A Resolution–Based Calculus For Temporal Logics

Dissertation

zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Technischen Fakultät
der Universität des Saarlandes
von

Andreas Nonnengart

Saarbrücken, 1995

Berichterstatter:	Priv. Doz. Dr. Hans Jürgen Ohlbach
	Prof. Dr. Dov Gabbay
Dekan:	Prof. Dr.-Ing. Helmut Bley

Preface

The increasing interest in applying temporal logics in various areas of computer science requires the development of efficient means that allow to reason within such logics. Usually this is realized by an implementable calculus and indeed remarkable progress has been made in the last two decades. The approaches developed so far can be roughly divided into two main categories: Either known techniques are extended to cope with the temporal logic syntax, or translation techniques into predicate logic are defined which allow to exploit already existing calculi. The former approach has the advantage that derivations remain within the temporal logic syntax, whereas the latter approach benefits from many years (in fact decades) of experience gained in classical logic theorem proving. The approach proposed in this work is based on a particular translation method into classical first-order predicate logic which utilizes certain interesting translational invariants. The reader is assumed to have detailed knowledge of automated theorem proving and formal logic, in particular classical first-order predicate logic. Although the itroduction of modal and temporal logics is fairly self-contained at least some knowledge of these logic areas would be quite helpful.

This work is a dissertation submitted under the same title to the "Universität des Saarlandes" in 1995. The research was partly carried out in the framework of the project LOGO under grant ITS 9102 at the Max Planck Institute for Computer Science, Saarbrücken.

Contents

Acknowledgments

In the first place I want to thank my supervisor Hans Jürgen Ohlbach who brought me into the field of modal and temporal logic theorem proving. Without his constant support, his strong encouragement and patience this work would not have been possible. I had the pleasure to follow him for many years now and he is in fact the person who guided me through the maze of science. The many fruitful discussions we had turned out to be a source of ideas that never seems to run dry.

I am also indebted to Dov Gabbay for his instant commitment to become one of the referees, for the fruitful discussions we had and for many helpful comments.

I also want to thank Harald Ganzinger for the opportunity to work in such a well-known and excellent research group at the Max Planck Institute and all the members of the group for providing such a wonderful research environment. Special thanks to Christoph Weidenbach and my room-mate Detlef Fehrer for so many interesting discussions.

Many thanks to Jörg Siekmann and his research group at Kaiserslautern University. He offered me my first job after I finished my Diploma thesis and during that time he was one of the main persons to turn to.

I benefited from the comments and help of many people and it seems impossible to list them without forgetting the one or the other. I thank all of them and want them to know that I highly appreciate their accompany.

Last but not least I want to thank Judith for her encouragement and patience and in particular my parents who made it at all possible for me to study computer science. I know that this was not a matter of course.

Saarbrücken, September 1995 Andreas Nonnengart

Zusammenfassung

Das ständig wachsende Interesse an Temporallogiken in zahlreichen Gebieten der Informatik verlangt nach Methoden, mit deren Hilfe effizient und schnell Schlußfolgerungen in diesen Logiken gezogen werden können. Üblicherweise geschieht dies durch die Entwicklung eines implementierbaren Kalküls, und tatsächlich wurden in den vergangenen Jahren bemerkenswerte Fortschritte in diese Richtung erzielt. Die bis heute bekannten Verfahren können grob in zwei Hauptkategorien eingeteilt werden: Entweder werden schon bekannte Techniken für andere Logiken (üblicherweise klassische Prädikatenlogik erster Stufe) erweitert, um mit der neuen Syntax zurechtzukommen, oder Übersetzungstechniken in die Prädikatenlogik werden definiert, welche es erlauben schon bekannte Kalküle wiederzuverwenden. Die ersteren Ansätze haben den Vorteil in der Syntax der Temporallogiken zu verbleiben, was die Lesbarkeit von Beweisen erleichtert, wohingegen die letzteren Ansätze von vielen Jahren – sogar Jahrzehnten – Erfahrung auf dem Gebiet des klassischen Theorembeweisens profitieren.

Der Ansatz der in dieser Arbeit vorgestellt wird, basiert auf einer bestimmten Übersetzungsmethode in die klassische Prädikatenlogik erster Stufe und einem darauf aufbauenden Kalkül, der gewisse interessante Übersetzungsinvarianten ausnützt. Vom Leser wird erwartet, daß seine/ihre Kenntnisse in Bezug auf die Prinzipien des automatischen Theorembeweisens über einfache Grundkenntnisse hinausgehen. Obgleich die Einführung in die Modal- und Temporallogiken ausführliche Kenntnisse auf diesen Gebieten nicht ausdrücklich verlangt, wäre ein vorangehendes Einarbeiten in diese Gebiete sicherlich recht hilfreich.

Temporallogik: Syntax und Semantik

Eine geeignete Auswahl an Operatoren für die zu behandelnden Temporallogiken ist im wesentlichen linguistisch motiviert. So zum Beispiel interessiert man sich für die Möglichkeit *es wird so sein, daß* ausdrücken zu können, ebenso wie *von jetzt ab* oder *bisher galt* bzw. *es war so, daß*. Hinzu kommen einfachere Ausdrücke wie *immer*, aber auch wesentlich kompliziertere wie z.B. *von … bis* oder *seit*. Für jede dieser Möglichkeiten zeitliche Zusammenhänge zu beschreiben, wurden logische Operatoren festgelegt, und tatsächlich ist ein Großteil der in der zeitgenössischen Literatur zu findenden Temporallogiken durch eine Teilmenge dieser Operatoren bestimmt.

Die Temporallogiken, welche in dieser Arbeit vorgestellt und bezüglich eines geeigneten Kalküls untersucht werden, basieren ebenfalls auf den oben informell aufgeführten Operatoren und deren zugehörigen Dualoperatoren. Die komplizierteste Logik darunter, läßt jeden dieser Operatoren und sogar zahlreiche weitere Varianten zu.

Zunächst zur Symbolik: Die temporallogischen Operatoren werden syntaktisch folgendermaßen unterschieden:

$\boxed{\mathrm{P}}\,\Phi$	bisher galt Φ (Gegenwart ausgeschlossen)
$\boxed{\mathrm{P}}_r\,\Phi$	bisher galt Φ (Gegenwart eingeschlossen)
$\boxed{\mathrm{F}}\,\Phi$	von jetzt ab gilt Φ (Gegenwart ausgeschlossen)
$\boxed{\mathrm{F}}_r\,\Phi$	von jetzt ab gilt Φ (Gegenwart eingeschlossen)
$\boxed{\mathrm{A}}_r\,\Phi$	immer Φ
$\langle\mathrm{P}\rangle\,\Phi$	Φ galt (es war so, daß Φ)
$\langle\mathrm{P}\rangle_r\,\Phi$	Φ gilt oder galt (Φ, oder es war so, daß Φ)
$\langle\mathrm{F}\rangle\,\Phi$	es wird so sein, daß Φ
$\langle\mathrm{F}\rangle_r\,\Phi$	Φ ist, oder es wird so sein, daß Φ
$\langle\mathrm{A}\rangle_r\,\Phi$	irgendwann Φ
$\Phi\,\mathbf{U}\,\Psi$	Φ gilt bis Ψ wahr ist
$\Phi\,\mathbf{S}\,\Psi$	Φ ist wahr seit Ψ galt

Tatsächlich gibt es noch eine Vielzahl weiterer Operatoren die mit den in dieser Arbeit vorgestellten Methoden behandelt werden können. Insbesondere die Operatoren **U** und **S** erlauben je nach Interpretation eine große Zahl an Varianten. Einige der obigen Operatoren, wie z.B. $\langle\mathrm{P}\rangle_r$, scheinen auf den ersten Blick eher künstlich zu sein als linguistisch motiviert. Tatsächlich tragen diese nur eine geringe eigenständige Bedeutung. Sie entstehen vielmehr nur aufgrund von Dualitätsbetrachtungen ($\langle\mathrm{P}\rangle_r = \neg\,\boxed{\mathrm{P}}_r\neg$). Die eigentliche Bedeutung von $\langle\mathrm{P}\rangle_r$ liegt also in *es ist nicht der Fall, daß bisher nicht.*

Die Sprache der Temporallogiken bezieht sich auf klassische Ausdrücke in Verbindung mit den oben genannten Operatoren, so daß z.B. eine Formel der Art

$$\forall x\ \boxed{\mathrm{P}}_r P(x) \wedge (Q\,\mathbf{U}\,S(x))$$

eine temporallogische Formel darstellt. Bedeutung erhalten derartige Formeln unter sogenannten temporallogischen Interpretationen, welche Informationen über Zeitpunkte und deren Beziehung untereinander tragen und für jeden Zeitpunkt gewisse lokale klassische Interpretationen bereitstellen. So z.B. gilt eine Formel $\langle\mathrm{F}\rangle\,\Phi$ in einer gegebenen Interpretation $\Im_{\mathrm{TL}}$ zum Zeitpunkt ι als *wahr*, wenn es einen späteren Zeitpunkt ι' als ι gibt, zu dem Φ bezüglich $\Im_{\mathrm{TL}}$ wahr ist. Etwas komplizierter gestaltet sich dies für Until- und Since-Operatoren. $\Phi\,\mathbf{U}\,\Psi$ gilt als *wahr* zum Zeitpunkt ι (in der Interpretation $\Im_{\mathrm{TL}}$), falls Ψ unter $\Im_{\mathrm{TL}}$ zu einem späteren Zeitpunkt als ι wahr ist und in der Zwischenzeit (d.h. für alle dazwischenliegenden Zeitpunkte) Φ unter $\Im_{\mathrm{TL}}$ gilt.

Relationale Übersetzung

Die obige Semantik erlaubt nun eine direkte Übersetzung in die Prädikatenlogik erster Stufe und zwar wie auf folgende Art und Weise exemplarisch dargestellt

$$\lceil \langle\mathrm{P}\rangle_r \rceil_u = \exists v\ v \leq u \wedge \lceil \Phi \rceil_v$$

wobei die speziellen Eigenschaften der Relation $\leq$ noch zu spezifizieren sind. So zum Beispiel werden dieser Relation in der Logik *K_tT4.3'* die Eigenschaften der Reflexivität, der Transitivität und der Rechtslinearität zugesprochen.

Eine solche Übersetzung ist im allgemeinen durchaus anwendbar, allerdings stellt es sich schnell heraus, daß schon sehr einfache Theoreme aufgrund der Übersetzung aber auch durch die recht komplizierte Theorie der gegebenen Ordnungen von üblichen Theorembeweisern kaum mehr geeignet behandelt werden können. Dieser Umstand verlangt nach Alternativen, und eine solche Alternative findet man in der semi-funktionalen Übersetzung.

Semi-funktionale Übersetzung

Ihren Ursprung hat die semi-funktionale Übersetzung in der sogenannten funktionalen Übersetzung wie sie u.a. von Hans Jürgen Ohlbach vorgestellt wurde. Die grundsätzliche Idee dahinter ist, die Verantwortung der Relationen, die durch die Übersetzung eingeführt wurden – im obigen Beispiel die Relation $\leq$ – auf eine Menge von geeigneten Funktionen zu übertragen und zwar derart, daß eine Folge von Relationen durch eine Schachtelung derartiger Funktionen simuliert wird. Daß eine solche Idee verwirklicht werden kann, ist beweisbar, indem gezeigt wird, daß für jede beliebige (seriale) Relation R auf $\mathcal{T} \times \mathcal{T}$ – wobei $\mathcal{T}$ höchstens abzählbar viele Elemente enthält – eine höchstens abzählbar große Menge F_R von (totalen) Funktionen existiert, sodaß gilt

$$\forall u, v\ R(u,v) \Leftrightarrow \exists f \in F_R\ f(u) = v$$

Eigentlich ist es dabei nicht unbedingt erforderlich von einer Menge von Funktionen zu sprechen; stattdessen kann die rechte Seite der obigen Äquivalenz ebenso durch $\exists x\ u\!:\!x = v$ ausgedrückt werden, wobei das Symbol „:" eine Funktion in Infixnotation darstellt. Diese Äquivalenz läßt sich nun auf zweierlei Arten ausnützen: Man kann jedes Vorkommen von R-Literalen in der Übersetzung einer noch zu beweisenden modallogischen Formel durch eine Gleichung (positiv oder negativ) ersetzen, oder aber man tut dies nur für bestimmte Vorkommen von R-Literalen, z.B. für alle positiven Vorkommen. Der erstere Ansatz mündet in der funktionalen, der letztere in der semi-funktionalen Übersetzung. Beide Verfahren haben gegenüber der relationalen Übersetzung den großen Vorteil, daß das Ergebnis nach Klauselnormalformbildung nicht unnötig stark in der Anzahl der Klauseln wächst, wobei bei der rein funktionalen Übersetzung außerdem die Länge der Klauseln stark verringert wird. Das sehr kompakte Ergebnis, das die funktionale Übersetzung liefert, ist der wesentliche Vorteil dieses Verfahrens. Bezahlt werden muß dieser Gewinn allerdings dadurch, daß die Hintergrundtheorien für die verschiedenen Modallogiken mit Hilfe einer Gleichungstheorie beschrieben werden müssen. So z.B. erhält man als Hintergrundtheorie für die Modallogik *S4* nicht mehr die Reflexivität und die Transitivität der Erreichbarkeitsrelation, sondern statt dessen zwei Gleichungen der Form

$$\begin{array}{l} \forall u\ \exists x\ u\!:\!x = u \\ \forall u, x, y\ \exists z\ u\!:\!z = u\!:\!x\!:\!y \end{array}$$

Derartige Gleichungssysteme (die für verschiedene Modallogiken beliebig komplex werden können) sind im allgemeinen sehr schwierig zu handhaben. Aus diesem Grunde wird häufig versucht in Fällen, in denen diese Theorie nur aus Unit-Gleichungen besteht, diese in einen geeigneten Theorie-Unifikationsalgorithmus zu übertragen. Mit Erfolg angewendet wurde dieses Verfahren bei den meisten Modallogiken deren Hintergrundtheorie sich aus einer Teilmenge der Eigenschaften Reflexivität, Serialität, Transitivität, Euklidizität und Symmetrie beschreiben läßt.

Die so beschriebene Vorgehensweise ist nicht diejenige, welche in der vorliegenden Arbeit verfolgt wurde, da insbesondere in den Temporallogiken, aber auch in vielen interessanten Modallogiken, die Hintergrundtheorie nicht durch eine Menge von Unit-Gleichungen dargestellt werden kann und somit die Möglichkeit, diese Hintergrundtheorie in eine geeignete Theorie-Unifikation zu transponieren, nicht mehr gegeben ist. Ohne diese Möglichkeit ist es allerdings für jeden Theorembeweiser fast unmöglich auch nur sehr einfache modallogische Theoreme zu beweisen, und man ist somit gezwungen nach anderen Lösungsmöglichkeiten zu suchen.

Die semi-funktionale Übersetzung stellte sich als interessante Alternative zur funktionalen Übersetzung heraus und zwar nicht nur in Bezug auf diejenigen Logiken, für die die funktionale Übersetzung keine Theorie-Unifikation zuläßt, sondern auch in Fällen, in denen eine Theorie-Unifikation durchaus vorstellbar ist und vielleicht sogar schon entwickelt wurde. Die grundsätzliche Idee hinter der semi-funktionalen Übersetzung liegt darin, nur einen Operatortyp (die $\Box$-Operatoren) relational, den anderen Operatortyp hingegen (also die $\Diamond$-Operatoren) funktional zu übersetzen. Der scheinbare Nachteil dieses Ansatzes, nämlich, daß zwar die Anzahl der Klauseln gegenüber des funktionalen Ansatzes nicht erhöht wird, wohl aber die Länge der Klauseln, wird dadurch wieder wettgemacht, daß die Hintergrundtheorie für die verschiedenen Modallogiken nicht mehr durch Gleichungssysteme zu beschreiben sind, sondern sich im wesentlichen aus der rein relationalen Übersetzung übernehmen lassen. Ein weiterer wesentlicher Vorteil dieses semi-funktionalen Ansatzes liegt in einer einfachen syntaktischen Invariante begründet, welche sich aus dieser Übersetzung ergibt. Man kann nämlich zeigen, daß das Ergebnis der Übersetzung einer beliebigen modallogischen Formel keine Vorkommen von positiven R-Literalen mehr enthält. Diese Übersetzungsinvariante läßt sich auf interessante Art und Weise verschiedentlich ausnützen. Zum einen erlaubt sie es in vielen Fällen die Nicht-Axiomatisierbarkeit gewisser Eigenschaften der Erreichbarkeitsrelationen zu beweisen. Als ein einfaches Beispiel sei hier die Irreflexivität genannt. Dadurch, daß das Übersetzungsergebnis keine positiven R-Literale enthalten kann, müssen sich alle überhaupt vorkommenden positiven R-Literale in der Hintergrundtheorie der gegebenen Modallogik aufhalten. Die einfachste Hintergrundtheorieklausel, welche in jeder serialen Modallogik vorkommt, besteht aus der Unit-Klausel $R(u, u\!:\!x)$. Diese Klausel drückt nichts anderes aus, als daß alles, was von einer beliebigen Funktion x angewendet auf die Welt u aus zu erhalten ist, auch schon von R erreicht werden könnte und spiegelt somit eine der beiden Richtungen der anfänglich genannten Äquivalenz wider. Es ist also so, daß für die einfachste seriale Modallogik *KD* nur ein einziges positives R-Literal betrachtet werden muß – nämlich $R(u, u\!:\!x)$ – und zwar völlig unabhängig davon, welches Theorem zu beweisen ist. Somit wäre diese eine Unit-Klausel der einzig mögliche Resolutionspartner für die Irreflexivitätsklausel. Allerdings ist zwischen diesen beiden Literalen kein Resolutionsschritt möglich und somit kann die Irreflexivitätsklausel nicht zu einer möglichen Widerlegung der gegebene Klauselmenge beitragen. Wäre nun also die Irreflexivität axiomatisierbar, dann gäbe es auch Formeln, deren Beweisbarkeit von der Irreflexivität abhängt. Allerdings haben wir erkannt, daß die Beweisbarkeit keiner Formel von der Irreflexivität abhängen kann und somit ist die Irreflexivität nicht axiomatisierbar. Diese Argumentation läßt sich nun auf beliebige Eigenschaften von Erreichbarkeitsrelationen erweitern, sofern diese Eigenschaften mit der Unit-Klausel $R(u, u\!:\!x)$ konsistent sind und nur aus negativen R-Literalen bestehen. Keine solche Eigenschaft läßt sich also axiomatisieren. Dieses Ergebnis ist eine der Folgen der semi-funktionalen Übersetzungsidee. Bisher konnten ähnliche Teilergebnisse nur aufgrund von komplizierten modelltheoretischen Betrachtungen gewonnen werden.

Solche Aussagen bezüglich der Nicht-Axiomatisierbarkeit von Eigenschaften der Erreichbarkeitsrelationen treffen zu können, ist nur ein Vorteil des semi-funktionalen Ansatzes. In Bezug auf die praktische Verwendbarkeit beim automatischen Theorembeweisen für Modallogiken ergibt sich allerdings ein noch wesentlich interessanterer Vorteil, die Möglichkeit der partiellen Saturierung von Hintergrundtheorien.

Partielle Saturierung von Hintergrundtheorien

Unter der partiellen Saturierung einer Hintergrundtheorie ist das Berechnen aller herleitbaren Konsequenzen aus der gegebenen Theorie zu verstehen. Dabei wird angenommen, daß diese Hintergrundtheorie all das widerspiegelt, was über ein bestimmtes Prädikat bekannt ist. Damit wäre es also möglich, die Saturierung der Theorie anstelle ihrer selbst zum Beweis eines Theorems zu verwenden. Unumgänglich ist dabei allerdings, daß das zu beweisende Theorem keine weiteren Aussagen bezüglich des durch die Theorie beschriebenen Prädikates macht. Allerdings ist eine solche Saturierung im allgemeinen unendlich groß, sodaß sich die Frage des Ersetzens der Theorie durch ihre Saturierung eigentlich nicht stellt. Dennoch kann das Berechnen einer Saturierung (mit Hilfe von Formelschemata) sehr nützlich sein, nämlich dann, wenn es gelingt eine Formelmenge zu finden, die in gewisser Form „einfacher" ist als die ursprüngliche Hintergrundtheorie, aber dennoch die gleiche Saturierung besitzt. In diesem Falle ist es möglich, die Hintergrundtheorie durch die alternative Formelmenge zu ersetzen und zwar ohne daß sich etwas an der Erfüllbarkeit oder Unerfüllbarkeit der gegebenen Gesamtformelmenge ändert.

Diese grundsätzliche Idee läßt sich nun auf die Modal- und Temporallogiken übertragen. Mit dem Wissen, daß die einzigen Vorkommen von positiven R-Literalen in der modallogischen Hintergrundtheorie liegen und diese charakteristisch ist für die Logik in der man rechnen möchte (und nicht etwa für das zu beweisende Theorem), kann also die Hintergrundtheorie durch ihre eigene Saturierung ersetzt werden. Dies sei zunächst anhand einer einfachen Modallogik, nämlich *S4*, beschrieben.

Die Hintergrundtheorie für *S4* nach semi-funktionaler Übersetzung lautet

$$\begin{array}{l} R(u,u) \\ R(u,u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \end{array}$$

D.h. *S4* ist dadurch charakterisiert, daß die zugehörige Erreichbarkeitsrelation sowohl reflexiv als auch transitiv ist. Die zusätzliche Unit-Klausel $R(u,u\!:\!x)$ stammt aus der Übersetzung und ist bei jeder serialen Modal- oder Temporallogik hinzuzufügen. Wichtig ist, festzuhalten, daß diese Klauseln tatsächlich unser gesammtes Wissen in *S4* darstellen das wir über R haben, da die Übersetzung einer beliebigen modallogischen Formel keinerlei positive R-Literale erzeugen wird. Man kann sich nun recht leicht davon überzeugen, daß die Saturierung dieser *S4*-Hintergrundtheorie aus allen Unit-Klauseln der Form $R(u,u\!:\!x_1\!:\!x_2\!:\,\ldots\,:\!x_n)$ mit $n \geq 0$ besteht. Offensichtlich ist diese Saturierung unendlich groß, allerdings ist es durchaus möglich, eine alternative Klauselmenge zu finden, die in gewisser Weise einfacher ist als die *S4*-Hintergrundtheorie und dennoch die gleiche Saturierung erzeugt. Eine solche alternative Klauselmenge besteht aus

$$\begin{array}{l} R(u,u) \\ R(u,v) \Rightarrow R(u,v\!:\!x) \end{array}$$

Es ist somit möglich, statt der originalen *S4*-Hintergrundtheorie diese einfachere Theorie zu verwenden, ohne daß die Erfüllbarkeit oder Unerfüllbarkeit einer gegebenen Formel darunter leidet. Man beachte, daß die Möglichkeit der Saturierung einer modallogischen Hintergrundtheorie bei der relationalen Übersetzung nicht besteht, da in diesem Falle nicht davon ausgegangen werden kann, daß die einzig positiven R-Literale sich nur in der für die Modallogik charakteristischen Hintergrundtheorie befinden. Die relationale Übersetzung erzeugt durchaus weitere positive R-Literale, nämlich durch die Übersetzung von $\Diamond$-Formeln.

Eine weitere äußerst interessante Eigenschaft, die zwar nicht nur im Falle der semi-funktionalen Übersetzung gilt, sich dort allerdings besonders angenehm auswirkt, ist die sogenannte Konnektiertheit von Frames. Diese Eigenschaft drückt sich dadurch aus, daß es ausreicht, nur solche Welten zu betrachten, die von einer gegebenen Anfangswelt aus über die reflexive und transitive Hülle der Erreichbarkeitsrelation zu erreichen sind. Unter der semi-funktionalen Übersetzung bedeutet dies, daß beliebige Welten durch Terme der Art $\iota : x_1 : \ldots : x_k$ ersetzt werden können, wobei $k \geq 0$ ist und jedes x_i ein beliebiges Element aus F_R darstellt. So zum Beispiel erhält man nach Saturierung der Hintergrundtheorie für die Modallogik *S5* alle Unit-Klauseln der Art $R(u : x_1 : \ldots : x_n, u : y_1 : \ldots : y_m)$ mit $n, m \geq 0$ und somit unter der Konnektiertheitseigenschaft die wesentlich vereinfachte Form $R(u, v)$, d.h. die universelle Relation. Auf diese Weise lassen sich häufig recht komplizierte Hintergrundtheorien, wie z.B. die der Modallogik *KD45*, zu wenigen, oder wie im Falle von *KD45*, zu einer einzigen Unit-Klausel – nämlich $R(u, v : x)$ – reduzieren.

Als ein weiteres Beispiel, diesmal einer Temporallogik, betrachten wir $K_tT4.3 \oplus K_tD4.3'$, die kompliziertеste der in dieser Arbeit betrachteten Temporallogiken. Ihre Axiomatisierung besteht aus 18 temporallogischen Axiomen und die dadurch beschriebene Hintergrundtheorie (nach einigen offensichtlichen Vereinfachungen) aus 13 Klauseln mit insgesamt 34 Literalen. Alle diese Literale sind entweder R-Literale oder Gleichheitsliterale und es ist unmittelbar einsichtig, daß der Suchraum, der durch diese Hintergrundtheorie geöffnet wird, immens groß ist und tatsächlich von gängigen allgemeinen Theorembeweisern kaum mehr gehandhabt werden kann. Nach Saturierung und Anwendung der Konnektiertheitseigenschaft ergibt sich allerdings eine Hintergrundtheorie, die nur noch aus 8 Klauseln mit nurmehr 14 Literalen besteht.

Eine weitere interessante Anwendung der semi-funktionalen Übersetzung findet sich in der Behandlung von aufsteigenden, absteigenden oder auch beliebig variierenden Domäne bei der Behandlung von Modal- und Temporallogiken der ersten Stufe. Die Annahme der konstanten Domäne garantiert, daß Objekte, über die in einer Welt gesprochen werden kann, auch in allen anderen Welten zur Verfügung stehen. In aufsteigenden Domänen wird hingegen angenommen, daß zumindest keine Objekte „verloren“ gehen und in absteigenden Domänen, daß in erreichbaren Welten keine neuen Objekte hinzukommen. Derartige Aussagen werden auf Modellebene durch gewisse zusätzliche Annahmen über die semantischen Eigenschaften von Quantoren repräsentiert. Diese können daraufhin ähnlich wie Modaloperatoren gehandhabt werden, d.h. die Übersetzung eines Allquantors (Existenzquantors) entspricht im wesentlichen der Übersetzung eines $\Box$-Operators ($\Diamond$-Operators), wenn auch mit einer anderen „Erreichbarkeitsrelation“. Dies ist am Beispiel einer einfachen Modallogik ausgeführt. Die Erweiterung auf kompliziertere Modal- und Temporallogiken ist danach offensichtlich.

Obgleich man im allgemeinen durch die oben genannten Verfahren eine signifikante Verbesserung erhält, ist es dennoch durchaus möglich, noch weitere Verbesserungen zu finden. So z.B. wird gar nicht erst nach alternativen Hintergrundtheorien Ausschau gehalten, sondern direkt versucht, die berechneten Saturierungen in geeigneten Inferenzregeln widerzuspiegeln. Auf diese Weise kann mit logikspezifischen Inferenzregeln gearbeitet werden und es ist überhaupt nicht mehr nötig, mit komplexen Hintergrundtheorien umgehen zu können.

Von Saturierungen zu Inferenzregeln

Zur Erläuterung betrachten wir noch einmal die Modallogik *S4*, die sich ihrer Einfachheit wegen besonders gut als einleitendes Beispiel eignet. Wie schon vorher erwähnt, ist die *S4*-Hintergrundtheorie durch die Unit-Klauseln der Form $R(u, u : x_1 : \ldots : x_n)$ beschrieben. Offensichtlich kann also ein beliebiges negatives R-Literal genau dann mit einem Element dieser Saturierung resolviert werden, wenn dieses negative R-Literal von der Art ist, daß das erste Argument mit einem Präfix des zweiten Argumentes unifizierbar ist. Diese Beobachtung führt unmittelbar zu einer geeigneten Inferenzregel, nämlich

$$\frac{\neg R(\alpha, \beta : \gamma) \vee C}{\sigma C}$$

wobei σ den allgemeinsten Unifikator von α und β bezeichnet und $\beta : \gamma$ darstellen soll, daß der gegebene Term aufgespalten werden kann in einen Präfix β und einen Suffix γ. Die Korrektheit dieser Regel ergibt sich unmittelbar aus dem Wissen um die Saturierung der *S4*-Hintergrundtheorie. Ihre Vollständigkeit – im Zusammenspiel mit der klassischen Resolution und Faktorisierung – verlangt allerdings ein gewisses trickreiches Vorgehen. Dies liegt darin begründet, daß zwar jede Anwendung dieser Regel einer Folge von Resolutionsschritten mit der alternativen *S4*-Hintergrundtheorie entspricht, nicht so aber umgekehrt. In gewisser Weise realisiert diese Inferenzregel also eine bestimmte Strategie auf den möglichen Resolutionsschritten mit der Hintergrundtheorie. Der Vollständigkeitsbeweis für dieses Inferenzsystem ist somit eigentlich ein Vollständigkeitsbeweis für eine bestimmte Resolutionsstrategie.

Auf analoge, wenn auch wesentlich kompliziertere Weise, lassen sich auch komplexere Modal- und Temporallogiken behandeln. So zum Beispiel besteht das Inferenzsystem für die schon oben genannte Temporallogik $K_tT4.3 \oplus K_tD4.3'$ aus 7 zusätzlichen Regeln, die alle aus der Saturierung der $K_tT4.3 \oplus K_tD4.3'$ Hintergrundtheorie entstanden sind.

Zur funktionalen Übersetzung

Wie eingangs schon erwähnt hat die funktionale Übersetzung – insbesondere in Bezug auf die Temporallogiken – den Nachteil, daß die Hintergrundtheorie in Gleichungsform wiedergegeben ist und sich deswegen nur sehr schwer handhaben läßt. Eigentlich geschieht der funktionalen Übersetzung bei dieser Behauptung unrecht, da dieser Effekt gar nicht direkt an der Übersetzung liegt, sondern von der strikten Ersetzung der R-Literale auch innerhalb der Hintergrundtheorie herrührt. Tatsächlich ist es durchaus möglich, die funktionale Übersetzung für Formeln zu wählen und dennoch das oben erwähnte Inferenzsystem zu verwenden. Allerdings ist es dann auch nötig, die Unifikation und die Anwendung von Substitutionen etwas zu verändern. So z.B. sind dann zwei beliebige Terme der Art $\alpha : x$ und β unifizierbar unter der Substitution $x/\alpha \rightsquigarrow \beta$. Intuitiv soll ein solcher Ausdruck ausdrücken, daß x durch etwas ersetzt werden soll, das, angewendet auf α, zu β führt. Die Anwendung einer solchen Substitution erzeugt dabei ein Residuum welches den Fall abdeckt, daß β überhaupt nicht von α aus erreichbar ist. So z.B. entsteht aus einem Literal der Form $P(\alpha : x)$ nach Anwendung der obigen Substitution die (bedingte) Instanz $\neg R(\alpha, \beta) \vee P(\beta)$, d.h. falls β von α aus erreichbar ist, dann gilt $P(\beta)$ (da x dann geeignet instanziiert werden kann). Auf diese Weise arbeitet man eigentlich mit der funktionalen Übersetzung, erzeugt allerdings hin und wieder R-Literale als Residuen, die wiederum mit

Hilfe des für diese Logik bestimmten Inferenzsystems behandelt werden können (gegebenenfalls muß der Begriff des allgemeinsten Unifikators um die obige Instanziierungsmöglichkeit erweitert werden). Einige Inferenzschritte, die nach semi-funktionaler Übersetzung möglich waren, werden dadurch ausgeschlossen. Der Gewinn ist dabei dennoch nicht ganz so spektakulär wie bei den vorangegangenen Verbesserungen, und deshalb wurde dieses Verfahren nur für einfache Modallogiken untersucht. Die Anwendung auf kompliziertere Modal- und Temporallogiken ist danach allerdings offensichtlich.

Behandelte Logiken

Für jede der in dieser Arbeit behandelten Logiken gilt im wesentlichen die gleiche Vorgehensweise, d.h. zunächst wird aus der Axiomatisierung der jeweiligen Logik die Hintergrundtheorie ermittelt, diese wird dann saturiert und das Ergebnis dieser Saturierung wird danach in eine geeignete Inferenzregel gegossen. Die Korrektheit dieser Inferenzregeln ergibt sich dabei fast unmittelbar aus der Saturierung der Hintergrundtheorie. Die Vollständigkeit hingegen verlangt häufig ein tiefergreifendes Vorgehen. Die jeweiligen Beweise sind für jede der behandelten Logiken durchgeführt worden.

Das Hauptinteresse in dieser Arbeit galt den Temporallogiken. Dennoch ist das vorgestellte Verfahren auch (und im besonderen) für die Behandlung von Modallogiken geeignet. Diese besitzen häufig eine weniger aufwendige Axiomatisierung als die Temporallogiken und eignen sich schon aus diesem Grunde besonders als einleitende Beispiele zur Demonstration der vorgestellten Techniken. Die folgenden Modallogiken wurden in dieser Arbeit untersucht:

K	keine besonderen Eigenschaften
KD	Serialität
KB	Symmetrie
KT	Reflexivität
K4	Transitivität
K5	Euklidizität
KDB	Serialität und Symmetrie
KD4	Serialität und Transitivität
KD5	Serialität und Euklidizität
K45	Transitivität und Euklidizität
KD45	Serialität, Transitivität und Euklidizität
S4	Reflexivität und Transitivität
S5	Reflexivität, Symmetrie und Transitivität
S4.2	Reflexivität, Transitivität und Konfluenz
S4.3	Reflexivität, Transitivität und Rechtslinearität
KD4.3'	Serialität, Transitivität und schwache Rechtslinearität
S4F	Reflexivität, Transitivität und „Separiertheit“
S4⊕*KD4*	Kombination von *S4* und *KD4*
S4.3⊕*KD4.3'*	Kombination von *S4.3* und *KD4.3'*

Interessanterweise stellte sich bei einigen dieser Modallogiken heraus – z.B. *S5*, *KD5*, *KD45*, *KDB* – daß die Hintergrundtheorie nach semi-funktionaler Übersetzung und Saturierung unter der Konnektiertheitseigenschaft sich zu wenigen (manchmal auch nur einer einzigen) Unit-Klauseln reduzieren läßt. Für derartige Theorien ist es dann selbstverständlich nicht nötig eigens

Inferenzregeln zu bestimmen; sie können direkt in die semi-funktionale Übersetzung einbezogen werden.

Temporallogiken, die in dieser Arbeit behandelt wurden, sind die folgenden:

K_t	keine besonderen Eigenschaften
K_tD	Serialität
K_tD4	Serialität und Transitivität
K_tT4	Reflexivität und Transitivität
$K_tT4.3$	Reflexivität, Transitivität und Rechts- Linkslinearität
$K_tD4.3$'	Serialität, Transitivität und schwache Rechts- Linkslinearität
$K_tD4 \oplus K_tT4$	Kombination aus K_tD4 und K_tT4
$K_tT4.3 \oplus K_tD4.3$'	Kombination aus $K_tD4.3$' und $K_tT4.3$
K_i	Intervallogik mit Monotonie
K_p^s	Intervallogik mit Monotonie, Reflexivität, Transitivität (und Konvexität)

Zusätzlich zu diesen Temporallogiken wurden weitere Temporaloperatoren betrachtet, deren Hinzufügen nichts an der zugrundeliegenden Zeitstruktur änderten und dennoch von einigem Interesse sind. Unter diesen Operatoren finden sich z.B. *Immer* und verschiedene Varianten von *von ... bis* und *seit*.

Vergleich zu gängigen Verfahren

Die in der zeitgenössigen Literatur mit dem hier vorgestellten Verfahren am engsten verwandten Methoden können in der relationalen Übersetzung und der rein funktionalen Übersetzung wiedergefunden werden. Erstere hat den großen Nachteil, daß schon recht einfache Theoreme durch die Übersetzung derartig aufgebläht werden, daß kein gängiger Theorembeweiser mit ihnen fertig würde. Letztere vermeiden zwar diesen Nachteil, allerdings wird die Hintergrundtheorie mit Hilfe eines Gleichungssystems beschrieben, welches sich im allgemeinen nicht in einen geeigneten Theorie-Unifikationsalgorithmus einbinden läßt. So z.B. wird in der funktionalen Übersetzung aus der schwachen Rechtslinearität für die Logik *KD4.3'*

$$u\!:\!f(u,v) = v \vee u = v \vee v\!:\!g(u,v) = u$$

d.h. eine Klausel, die insbesondere in Kombination mit der aus der Transitivität stammenden Unit-Gleichung $u\!:\!h(u, u\!:\!x, u\!:\!x\!:\!y) = u\!:\!x\!:\!y$, jeden Beweiser in sehr große Schwierigkeiten bringt. Derartige Probleme mit der Gleichheitsbehandlung werden im semi-funktionalen Ansatz größtenteils vermieden. Der Fairness halber sollte allerdings erwähnt werden, daß dies auch nicht ganz umsonst geschieht; die stattdessen eingeführten Inferenzregeln, wie z.B. die *KD4.3'* Inferenzregel, können durchaus an sehr vielen Stellen einer gegebenen Klauselmenge angreifen. Dies geschieht allerdings gezielter als im rein funktionalen Ansatz.

Der Vergleich der hier vorgestellten Methode mit gänzlich anderen Inferenzmethoden für Temporallogiken ist nicht ganz einfach. Als Beispiel seinen hier typische Tableau- und Sequenzen-Kalküle erwähnt, die sich nur unwesentlich voneinander unterscheiden. Tableau-Systeme eignen sich insbesondere für Modallogiken wie z.B. *KT* oder *S4*, da sich die Axiomatisierung dieser Logiken fast unmittelbar in den Tableauregeln widerspiegelt. Eine typische Tableauregel für *S4* lautet daher

$$\frac{\Diamond A, \Diamond\Delta, \Box\Gamma, \Omega}{A, \Box\Gamma}$$

Diese Regel ist informell etwa so zu interpretieren: Durch $\Diamond A$ wird die Existenz einer Nachfolgewelt garantiert, in der A gilt. In allen Nachfolgewelten müssen die Formeln aus Γ gelten und aufgrund der Transitivität wird sogar $\Box\Gamma$ zu übernehmen sein. Der klassische Anteil Ω, sowie das Wissen um andere erreichbare Welten ($\Diamond\Delta$), kann dabei ignoriert werden, da in *S4* Strukturen keinerlei Zusammenhänge zwischen verschiedenen erreichbaren Welten existieren müssen. Diese obige Regel spiegelt also im wesentlichen die Transitivität der Erreichbarkeitsrelation in *S4* wider. Problematischer wird der Tableau-Ansatz für rechtslineare Erreichbarkeitsrelationen, wie sie typischerweise in Temporallogiken vorkommen. In diesem Fall reicht es nicht aus, eine geeignete $\Diamond$-Formel auszuwählen und die anderen einfach zu ignorieren. Die Rechtslinearität macht nämlich gewisse Aussagen über die Zusammenhänge verschiedener Welten; insbesondere garantiert sie, daß zwei beliebige erreichbare Welten vergleichbar sind. Dies hat zur Folge, daß für zwei gegebene $\Diamond$-Formeln zu unterscheiden ist, welche „zuerst" wahr wird. Beide Möglichkeiten müssen betrachtet werden, und somit werden zwei $\Diamond$-Formeln eine Verzweigung des Tableaus in zwei Folgeäste erzwingen. Dies ist für nur zwei $\Diamond$-Formeln noch nicht allzu aufwendig. Hat man aber mehrere solcher Formeln im aktuellen Tableau, ist jede mögliche Reihenfolge zu berücksichtigen. Dies hat zur Folge, daß bei n $\Diamond$-Formeln eine Verzweigung in $n!$ (in Worten: n Fakultät) Äste zu betrachten ist; ein ganz erheblicher Aufwand, vorausgesetzt, die zu beweisende Formeln ist nicht trivial. Im Falle der schwachen Rechtslinearität verschlimmert sich dieser Nachteil sogar noch, da dann für je zwei $\Diamond$-Formeln eine Verzweigung in jeweils drei Äste zu betrachten ist.

Auch diese Problematik ist im vorliegenden Ansatz wesentlich abgeschwächt. Als typisches Beispiel betrachten wir die folgende Formelmenge

$$\begin{array}{l} \Diamond(P_1 \wedge \Box\neg P_2) \\ \Diamond(P_2 \wedge \Box\neg P_3) \\ \ldots \\ \Diamond(P_{n-1} \wedge \Box\neg P_n) \\ \Diamond(P_n \wedge \Box\neg P_1) \end{array}$$

Für jedes $n \geq 1$ ist diese Formelmenge *S4.3*-unerfüllbar und nach den vorherigen Betrachtungen hat ein Tableaukalkül $n!$ Äste zu überprüfen (wobei jeder dieser Äste durch *S4* Regeln geschlossen werden kann). Dies bedeutet selbstverständlich einen exponentiell großen Aufwand betreiben zu müssen. Im semi-funktionalen Ansatz entstehen zunächst $2 \times n$ Klauseln und mit Hilfe von n Reolutionsschritten (nämlich gerade denen auf den P_i-Literalen) entsteht eine aus n Unit-Klauseln bestehende unerfüllbare Klauselmenge. Diese Klauselmenge beschreibt einen sogenannten „Zykel" und die *S4.3* Inferenzregeln sind gerade dazu da, einen Zykel der Länge n in einen Zykel der Länge $n-1$ zu überführen. Nach insgesamt $2 \times n$ Inferenzschritten wird also eine Widerlegung gefunden; eine wesentliche Beschleunigung gegenüber der Tableaumethode.

Intervallogiken

Obgleich der hier vorgestellte Ansatz eigentlich zur Anwendung auf punkt-orientierte Zeitlogiken gedacht ist, ist die angebotenene Vorgehensweise keineswegs nur auf derartige Logiken beschränkt. In Intervallogiken betrachtet man gewisse Zeiträume (eben Intervalle) anstelle von Zeitpunkten und interessiert sich für die Beschreibung von Ereignissen und nicht mehr unbedingt von augenblicklichen Situationen. Dabei unterscheidet man zwischen Logiken, in denen die Beschreibung solcher Zeiträume durch feste Grenzen (Zeitpunkte) gegeben ist und solchen

Logiken, in denen Intervalle als die eigentlichen Primitive angesehen werden (und somit der Begriff des Zeitpunktes keine Bedeutung mehr hat).

Beide Ansätze lassen sich mit der in dieser Arbeit vorgestellten Methode behandeln. Im ersteren Falle ist es problemlos möglich, eine geeignete (semi-funktionale) Übersetzung zu definieren, die den geforderten Ansprüchen genügt; im letzteren Falle kann aufgrund der Axiomatisierung eine Hintergrundtheorie bestimmt werden, die nach Saturierung und Transformierung in geeignete Inferenzregeln sich ebenfalls der semi-funktionalen Übersetzung nicht verschließt, ja sogar ein einfaches und kompaktes Inferenzsystem zuläßt. Dies ist exemplarisch anhand von Intervallogiken von Humberstone und von van Benthem aufgezeigt.

Einschränkungen

Prinzipiell anwendbar ist der hier vorgestellte Ansatz für alle Modal- und Temporallogiken deren Frame-Eigenschaften sich in der Prädikatenlogik erster Stufe beschreiben lassen. Damit ist auch schon eine der gesetzten Schranken offensichtlich: Eigenschaften der Erreichbarkeitsrelation die nicht prädikatenlogisch erfaßt werden können, sind mit diesem Ansatz auch nicht handhabbar. Beispielsweise betrachtet man in der Anwendung von Temporallogiken in der Programmverifikation gerne *diskrete* Zeitstrukturen, in denen gewährleistet ist, daß zwischen zwei beliebigen Zeitpunkten nur endlich viele andere Zeitpunkte liegen können. Diese Eigenschaft ist nicht prädikatenlogisch beschreibbar und entzieht sich somit auch dem hier vorgestellten Ansatz.

Allerdings können auch dann Probleme auftreten, wenn alle Eigenschaften der betrachteten Erreichbarkeitsrelationen durch Ausdrücke der klassischen Logik erster Stufe repräsentiert sind. Leider existiert kein automatisches Verfahren, welches die Saturierung einer Hintergrundtheorie vollständig berechnen kann. Die Bestimmung einer solchen Saturierung, das Entwickeln einer alternativen Hintergrundtheorie bzw. ihre Transformation in geeeignete Inferenzregeln ist immer noch zu einem nicht unwesentlichen Teil ein kreativer Akt und leider nicht vollständig automatisierbar.

Zukünftige Betrachtungen

Die in dieser Arbeit betrachteten Zeitstrukturen sind von möglichst allgemeiner Art. Die einzige Eigenschaft, die von dieser Allgemeinheit abweicht, ist die Linearität. Diese Eigenschaft wurde auch schon deshalb ausgewählt, weil sie sich der Behandlung durch die funktionale Übersetzung in Bezug auf die Umsetzung in geeignete Unifikationsalgorithmen entzieht. Es sind allerdings auch noch andere Eigenschaften denkbar, die man Zeitstrukturen zugestehen kann und die nicht in dieser Arbeit behandelt wurden. Dazu gehört z.B. die *Dichtheit*. Diese Eigenschaft kann prinzipiell auch mit der vorgeschlagenen Methode behandelt werden. Dies tatsächlich zu tun, sei späteren Untersuchungen überlassen.

Weitere Eigenschaften (und Operatoren) sind dadurch definiert, daß explizit verzweigende Strukturen betrachtet werden und zwar nicht nur im Sinne der Modallogik *S4* sondern sogar durch ein explizites Quantifizieren über *mögliche Zukünfte*. Auch derartige Logiken wurden in dieser Arbeit nicht untersucht; es steht allerdings auch für diese Sprachen einer Behandlung im Sinne der hier vorgestellten Methoden nichts im Wege, vorausgesetzt, es steht eine Modelltheorie zur Verfügung, die eine Übersetzung in die Prädikatenlogik erster Stufe erlaubt.

Zuguterletzt sei noch einmal auf die Diskretheit eingegangen. Wie oben schon erwähnt, ist diese Eigenschaft nicht prädikatenlogisch beschreibbar und kann somit auch nicht direkt mit den Saturierungsmethoden behandelt werden. Allerdings ist es möglich, derartige Eigenschaften (dazu gehört auch die sogenannte *Löb-Eigenschaft*) in einer Fixpunkt-Sprache zu beschreiben. Wie so etwas möglich ist, ist auch in der Arbeit beschrieben. Es ist sicherlich sinnvoll, solche Fixpunkte dahingehend zu untersuchen, ob und wie sie zur Saturierung beitragen können. Es wird Aufgabe späterer Untersuchungen sein, festzustellen, ob derartige Erweiterungen denkbar und wie sie gegebenenfalls auszuführen sind.

1

Temporal Logic – Why Bother?

Is there a difference between the two statements "for any two distinct instants, one is earlier and one is later" and "whatever will have been either was, or is, or will be"? Should we be able to conclude "I will have been writing" from "I am now writing"? And what is the relationship between the two phrases "Peter is sitting" and "all moments identical with the present are (timelessly) moments when Peter is seated"?

Questions like these have preoccupied philosophers, linguists, computer scientists, and AI researchers, and made them highly interested in a formal investigation of temporal relationships. Philosophers, for instance, tried to learn how to avoid confusing the tensed from the tenseless as exemplified by the first question above. The part "for any two distinct instants, one is earlier and one is later" is tenseless, i.e. the "is" does by no means refer to the present. Rather it is used temporally indefinite (or timeless), whereas "whatever will have been either was, or is, or will be" is definitely a tensed statement, nevertheless stating essentially the same thing as we shall see later.

Linguists often grapple with the problem that in the formal examination of natural language dialogues utterances frequently refer to some temporal order which is usually expressed, at least in part, through changes in verb-form, or tenses. Intuitively, we would not hesitate to confirm that "whatever is, will have been" just as we would not disagree that "whatever was going to happen will (sooner or later) have happened". In linguistics there is an attempt to provide an idealized model for aspect in natural language which may shed some light to many interesting features of its deep semantic structure.

Probably due to a renewed appreciation for the fact that issues involving tense touch on certain issues of philosophical importance (determinism, causality, and the nature of events, of time and of change) the semantics of tense has received a great deal of attention in the contemporary literature of linguistics and philosophy.

Researchers working in areas of computer science like database theory and concurrent networks found it more and more important to be able to distinguish between static information and processes which change the environment dynamically. This gave rise to new and promising research areas like temporal databases and formal program verification.

This is even more observable in the field of artificial intelligence where it is tried to build a bridge between philosophy, mathematics and linguistics. Here it often does not suffice to merely represent temporal relationships; there is also a need for efficient reasoning calculi which make it possible to draw inferences from the formally described situations.

Realizing what the problems are inevitably raises the question of how these problems can be overcome. There are three possible responses. As P. F. Strawson noted in his *Introduction to logical theory* (see (Strawson 1952)) standard (classical) logic, in the form of the predicate calculus, seems ill-equipped to cope with statements containing tensed verbs or explicit temporal reference. He sees phenomena like the one given in the second question on the very beginning of this section as an indication of the inherent limitation of formal logic, showing that it is incapable of representing statements of ordinary language.

(Quine 1960), on the other hand, proposes to view tensed statements as to be paraphrased into an atemporal form and represented in a many-sorted predicate calculus thus making it fit patterns of classical logic. This then leads to a reformulation of tenses as given in the third question from above which — for most people — seems at best awkward and at worst misleading.

In the last few decades a third – more positive – response has appeared which takes the form of developing autonomous formal logics of tense and temporality as first undertaken by Arthur Prior (see (Prior 1957) for a first attempt along these lines). Such is the approach of *Temporal (or Tense) Logic*[1]. The objective of temporal logic is both, to elucidate reasoning with statements which have some temporal aspect and to give something of the rigour of modern logical systems to a language whose sentences resemble those of natural language in being true at one time and false at another. Slipping between the alternatives posed by Quine and Strawson, such logics offer a neat compromise. On the one hand, temporality is preserved against Quine's atemporal paraphrases, and on the other hand, the scope of formal logic is extended so as to take steps against Strawson's misgivings about the limits of logic.

But how should tensed arguments be treated formally? The aim should be to come up to meet all parties involved, for, from a philosophical and logical point of view, it should be tried to break with the traditional view that reasoning can only involve timeless eternal propositions. From the linguistic point of view, a logical description of such a ubiquitous and important phenomenon as tense will obviously be quite welcome. In this sense, temporal logic forms a bridge between linguistics and mathematics.

Tensed arguments obviously have as unstated premises certain assumptions about the structure of time. Judging which assumptions are physically or meta-physically correct is the job of a cosmologist or physicist rather than of a logician. The logician's job is to formalize such assumptions in logical symbolism. In addition to the formalization of tensed statements, the systemization of inferences involving such statements has been a primary aim of tense logicians. But here, as in all non-standard areas of logical theory, a difficult question arises as to which

[1]Throughout this text the two notions *Temporal Logic* and *Tense Logic* will be used interchangeably. Although *Tense* and *Temporal* should not be used synonymously the corresponding logics are not that different.

assumptions about time should be taken axiomatically in the development of deductive systems of temporal logic. Without an exploration of various possibilities any proposed answer will inevitably seem arbitrary. Thus many systems of temporal logic have been created which incorporate differing tense logical principles. Such systems allow the investigation of these principles taken singly or in groups. And, in this manner, the covert assumptions behind intuitive appraisals of tensed arguments are brought to the surface.

But what should the basic entities of temporal logic be? Some people might think of time intervals, others of time points or instants. Some might think about absolute measures of clocks, others about the relative (temporal) occurrence of certain events taking place. A formal examination of time thus requires to fix the basics, i.e. the questions have to be answered whether we are interested in periods or instances and whether we prefer absolute measurements over the relative order of the primitives, be it points or intervals. One might object that such differentiations are not really necessary, for one could view intervals as convex sets of time instants and moments as indivisible intervals. And just as the geometry of Space can be axiomatized taking unextended points as basic entities, it can equally well be axiomatized by certain regular open solid regions such as spheres. Likewise, the order of Time can be described either in terms of instants or in terms of periods of nonzero duration (see (Humberstone 1979) and (van Benthem 1990)).

The difference might not be too crucial at the first glance but it actually *is* if we are about to interpret whatever we gain as a formal logic underneath these primitives. What all these logics have in common is that certain truths and falsities are associated with the temporal primitives. But what can possibly be true on intervals and what can possibly be true at instants? Usually intervals are associated with certain events or processes[2] and instants with certain momentary situations. Point structures thus provide with a static view of time, a snapshot of the environment, whereas interval structures force us to view the world as a dynamic sequence of events. This shows that it is really most important to distinguish between moments and intervals, for, just as an example, how could we possibly describe that "a car is driving" only with the help of time instants? It would certainly not make very much sense to say that the car is moving in every instant which occurs in a certain convex set of time instants, since *movement* doesn't mean anything in a static description (i.e. in a single moment). Similarly, we run into difficulties if we try to describe a static truth with the help of events.

The switch from instants to periods is often motivated by a desire to model certain features of natural language. One of these is aspect, the verbal feature which indicates whether we are thinking of an occurrence as an event whose temporal stages do not concern us or as a protracted process. In part, the switch is motivated by a philosophical belief that periods are somehow more basic than instants. This motivation would be more convincing were periods not assumed (as they are in many recent works) to have sharply-defined (i.e. instantaneous) beginnings and ends. It may also be remarked that at the level of experience some occurrences do appear to be instantaneous. Thus the philosophical belief that every occurrence takes some time (period) to occur is not obviously true on any level.

What weighs even more is that — up to now — there is no really satisfactory solution to the problem of finding a suitable sound and complete axiomatization for interval based temporal logics. Some results along these lines have been obtained but unfortunately none as definitive as those of instant based temporal logic.

[2]There are exceptions, however, and some of these exceptions are handled in this thesis as well.

The objective of this thesis is not to *invent* new temporal logics, neither is it intended to develop suitable axiomatizations for interval logics. It is rather meant to provide efficient calculi for temporal logics whose syntax, semantics, and, in particular, axiomatizations are well understood today. Therefore, and also because of their close relationship to the (in some sense simpler) modal logics, calculi for temporal logics based on point structures are examined.

The main idea behind the approach proposed in this thesis is briefly described as follows: Given a temporal language which is in a sense general enough to cover various temporal logics occurring in the contemporary literature, a translation is defined which allows us to describe temporal logic sentences in the language of the first-order predicate calculus[3]. In fact, there are several such possible translation methods known today. The approach developed here is particularly interesting for it combines many advantages of the translation methods known so far and that without incorporating too many of their respective disadvantages. One such advantage lies in the fact that the translation result is comparatively small in the number of clauses. Another advantage is that the translation output is in Horn form if the input happened to be in Horn form. Most striking is the fact that the result thus obtained can be strictly seperated from the background theory of the logic under consideration. This fact makes it often possible to show the non-axiomatizability of certain accessibility relation properties in a nice and elegant way.

This new translation method forms the first part of the technique proposed in this thesis. The second part concerns the development of calculi which allow us to reason efficiently within the logics we are interested in.

The separation of the theorem to be proved on the one hand from the background theory which is induced by the temporal or modal logic in question on the other hand makes it possible to consider this very theory independently from the rest of the clause set. It usually consists of a set of clauses which state certain particular properties of the "Earlier–Later" relation (or the accessibility relation in case of modal logics). These properties are characteristic for the logic and thus do not depend on the theorem to be proved. The idea is now to compute everything that might be deducible from this clause set beforehand, i.e. to saturate the theory (with respect to the inference system which is resolution and maybe also paramodulation). Evidently, this cannot be done explicitly since this would usually result in infinitly many new clauses. However, these derivations obey a certain syntactic form and the aim is therefore to make use of the knowledge about this form. This can be done in two ways: Either an alternative theory clause set is found which would produce exactly the same saturation and therefore may replace the original theory, or a suitable set of inference rules is defined which covers the resposibilities of the background theory in the sense that every application of such an inference rule corresponds to a resolution step with an element of the saturation. As a matter of fact, such an inference system realizes a certain strategy on the resolution process and showing the refutation completeness of this system thus corresponds to proving the completeness of a certain resolution strategy.

As the above description shows, this technique is not at all restricted to the application to modal and temporal logics. It is general enough to be applied to any problem in which a theory can be seperated from the theorem to be proved such that this theory can possibly be saturated. After having found a saturation – i.e. clause schemata that contain exactly the clauses that can be derived from the theory – it is usually fairly simple to obtain a suitable set of inference rules

[3]Note that such a translation does not necessarily lead to Quine's opinion that temporal relationships should be encoded in the predicate calculus anyway. The translation result is going to be used only for the reasoning process; the *user* of the temporal logic under consideration is not supposed to work directly with this translation.

which may replace the original theory.

Remark: The logics considered in this work are all first-order logics. Still, the choice of operators and the properties of the underlying modal or temporal structures are rather motivated by axioms known from the corresponding propositional fragments. In this sense interesting axiom schemata are introduced as if we had a propositional language in mind. Such axioms usually induce certain properties of the underlying structure and these properties are taken over to the first-order level. To state it differently, we are rather interested in model-theoretic properties than in Hilbert axiomatizations. Nevertheless such an axiomatization (for the propositional case) is considered to motivate accessibility relation properties we want to deal with.

1.1 Structure of the Work

Chapter 1: Temporal Logic – Why Bother?
The chapter you are currently looking at. Its purpose is to briefly describe the background of temporal logic and what made me interested in working on such an issue.

Chapter 2: Temporal Logic In this chapter the temporal logics as they are considered throughout this thesis are introduced. This covers an examination of possible temporal operators and properties of the underlying temporal structure as well as the formal definition of both the syntax and the semantics of temporal logics.

Chapter 3: A Digression to Modal Logics Many modal logics can be viewed as a basis of temporal logics and in fact main differences can merely be found in some special operators and certain accessibility relation properties which might not be very common in the modal logic area. Although this chapter is called and meant as a *digression* to modal logics it became the biggest part of this thesis. This is so because most of the inference techniques for temporal logics are first developed for modal logics and the results thus obtained are utilized in later chapters.

Chapter 4: Back to Temporal Logic The preceding chapter contains the formal basis for the development of a resolution-based calculus for the temporal logics introduced in the second chapter. I was mainly interested in combinations of temporal logics as they appear in the temporal logic literature and that without any special assumptions about the underlying structure of time.

Chapter 5: The Linearity Assumption From a computational point of view the linearity assumption is a fairly complex one. It is interesting in so far as many people think of time as a linear sequence of moments. The problem with linearity from a theorem proving perspective is that, in practice, it significantly increases the complexity of finding a proof. Linearity is rarely considered in the modal logic area and in fact almost no theorem proving system for modal logics is able to deal with this assumption in an efficient way.

Chapter 6: A Short Digression to Interval Logics Although the temporal logics considered in this work are point–based it is often possible to apply the very same techniques to logics which use intervals as their basic temporal entities. How this works is exemplified in this chapter with the help of some more or less simple interval logics.

Chapter 7: Summary and Further Work This final chapter contains a brief summary of the whole approach and an outlook of what should be investigated (in the light of this approach) in the near future. It concludes with a short comparison with other (related) approaches.

2

Temporal Logic

2.1 Introduction

The main idea behind instant-based Temporal (or Tense) Logics is the extension of classical propositional or first-order logic by some additional new primitive operators which represent certain temporal relationships.

Which additional primitives are to be chosen highly depends on the expressive power required. So, for instance, one could extend classical logic by the addition of a new operator, say $\Box$, such that $\Box\Phi$ is a formula if and only if Φ is one, and which has the intended meaning: Φ is *always true* or in other words: Φ is *true at any time.* This operator immediately induces another operator dual[1] to $\Box$, say $\Diamond$, which has the intended meaning: Φ is true *at some time.* Such a particular interpretation of the operators $\Box$ and $\Diamond$ has some immediate effect on the properties of the logic: For instance it will be required that if some formula Φ is *always* true then it is obviously true *now*, written $\Box\Phi \Rightarrow \Phi$. Also we might say that *at some time* is independent from *now*, or in other words: if Φ is true *at some time* then exactly this fact holds *at any time*, written $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$. These two implications should hold for any formula Φ and hence can be viewed as axiom schemata which partly define our logic. In fact everybody familiar with modal logics will immediately recognize these properties as the characteristic axiom schemata which have to be added to the basic modal logic K in order to get *S5*. This logic *S5* has been very well examined in the last decades and indeed there exist fairly efficient calculi for *S5* already[2]. Hence at first glance the choice of these particular operators seems to be a pretty good one, for the syntactical and semantical extensions to classical logic are ostensibly small and the development

[1]By duality we mean a relationship analogous to the correspondence between universal and existential quantifiers.

[2]See also Chapter 3.

of a calculus is fairly straightforward if at all necessary. And if the expressive power of this logic suffices for one's own purposes, there is nothing else to be done.

In general, however, one will sooner or later realize that this logic is only of very limited expressive power. For example this extension does not allow us to represent something like: *it will be the case that*. But of course we can easily change the interpretation of the $\Box$-operator to: *from now on* or in other words: *henceforth*. In this case the $\Diamond$-operator, since it is dual to the $\Box$, has to be interpreted as: *eventually it will be the case that*[3]. What kind of properties do we expect for this logic? First, we have again that anything which holds *henceforth* also holds *now*. Second, we might want to express that the future of the future is itself future and this could be done by adding the axiom $\Diamond\Diamond\Phi \Rightarrow \Diamond\Phi$. These two properties are in fact the characteristic axiom schemata for another well-known modal logic, namely *S4*, and if there are no more additional properties required this would be our desired logic.

The same would hold if we were only interested in past operators with an analogous interpretation, or in operators which exclude the present. The former case again results in *S4* the latter in *K4* or *KD4*.

But still one might not be satisfied with the means thus provided. There may be a need for past operators as well as future operators and this demand changes the situation considerably as there is now a multi-modal logic to be considered. Multiple modalities are not entirely new in the modal logic community, although they are often considered as being independent from each other. Nevertheless, in temporal logics these modalities do in some sense *interfere* and therefore this extension can be viewed as one of the simplest logics which separate modal logics from temporal logics[4].

As an example let us consider the so-called *Priorian*[5] tense operators F, P, G, H where $F\Phi$ means: Φ *will be true* and $P\Phi$ means: Φ *was true*. The operators G and H then are the duals to F and P respectively. This logic is characterized by two modal K-fragments, i.e. the classical propositional calculus together with

$$G(\Phi \Rightarrow \Psi) \Rightarrow (G\Phi \Rightarrow G\Psi) \qquad H(\Phi \Rightarrow \Psi) \Rightarrow (H\Phi \Rightarrow H\Psi)$$

$$\frac{\Phi}{G\Phi} \qquad\qquad \frac{\Phi}{H\Phi}$$

one for the future (left column) and one for the past (right column) and the axiom schemata $FF\Phi \Rightarrow F\Phi$ and $PP\Phi \Rightarrow P\Phi$ which express that the future of the future is again the future and the past of the past is again the past. In addition we need some "mixing" schemata which express something in the lines of: If *henceforth* Φ has been true then Φ is true *now* and also: If it will be the case that hitherto Φ then, in particular, Φ is true *now*. Formally: $PG\Phi \Rightarrow \Phi$ and $FH\Phi \Rightarrow \Phi$.

In (Kamp 1968) two very interesting independent operators had been introduced, namely "Until" and "Since" which, as Kamp showed, are expressively complete in the sense that they

[3]Note that this *eventually* includes the present, i.e. something which is true *now* is automatically true *eventually*.

[4]Standard text-books often refer to E. J. Lemmon's system K_t as the minimal tense logic. This logic is essentially the combination of two modal logics K (one for the past and one for the future fragment) together with schemata which define the correlation between the past and the future. Transitivity of the earlier-later relation is thereby not assumed. Our examination of temporal logics will begin with an analysis of this logic K_t.

[5]Due to Arthur Prior.

can be used to represent all the operators from above (and in fact any temporality one can think of) provided the underlying time structure is continuous like the reals. Informally Φ Until Ψ means: Ψ holds *eventually* and *in the meantime* Φ holds. Thus $\Diamond\Phi$ can be represented as: **True** Until Φ and $\Box\Phi$ as: not (**True** Until $\neg\Phi$). Such a minimal set of operators is interesting for theoretical considerations; in a language which is supposed to represent temporal relationships as "natural" as possible, however, the necessary translations are not appropriate and tend to produce too much of representational overhead. Nevertheless both "Until" and "Since" are interesting by themselves as they represent phrases which are very common in natural language usage.

So far we considered different types of temporal operators but sometimes not only a certain set of operators is required but also there is a need for special properties of the underlying time structure. Mostly, a single linear time axis is assumed. Nevertheless one might equally imagine a branching in the past as well as in the future. Branching in the past is rarely assumed, but branching in the future may help to represent different future alternatives[6]. On a linear time axis the $\Diamond$-operator is therefore to be interpreted as: *inevitably it will happen that* whereas on a tree structure it means: *there is a future alternative where possibly* or *it can happen that.* Hence, the choice between branching and linear structures is very closely related to the choice of interpreting the $\Diamond$ either as *inevitably* or as *possibly.*

Further questions which might also arise are for example whether time is dense in the sense that between any two instants there is a third one, or, alternatively, whether time is discrete, i.e. between any two instants there are only finitely many others. Other typical questions which arise are: *is there a beginning of time, an end, or both*? As a matter of fact all these properties are axiomatizable in the propositional case and except for the discreteness property they are expressible for first-order temporal logic as well. But such a lifting to the first-order level immediately gives rise to new questions like: *is the domain constant over time or does it vary and if so, how*? Obviously not each of these decisions inevitably leads to an entirely different logic. So for example, one might view first-order temporal logic as a generalization of propositional temporal logic, varying domains as a generalization of constant domains, the possibility of having a beginning and an end of time as a generalization of endless time, and last but not least past and future operators as a generalization of the modal logic *S4* or *KD4*[7].

Certainly it would be most convenient to have some sort of *most general* temporal logic, however, as it turns out, this is not possible. Nevertheless, we can get a fairly good approximation by considering a first-order linear temporal logic with varying domains which allows us to specify beginning and end of time and which incorporates at least the temporal operators mentioned above.

The aim of this chapter is to introduce such a logic (i.e. its syntax and its semantics). Later, efficient resolution-based calculi will be provided which allow us to reason within these logics and some of their variants.

[6]In this case the time structure is essentially a tree where the different branches contain the future possibilities that cannot yet be known. One might also consider a branching in the past for in cases in which a certain order of events has been forgotten. However, in my view, this should rather be described with additional epistemic operators of knowledge and belief for otherwise it would be impossible to describe that the order is remembered.

[7]One might as well think of branching future as an extension of linear future. And indeed if for the branching case there are newly introduced operators like *inevitably* and *possibly henceforth*, we could call it a generalization of linear time. For the operators mentioned above, however, these logics are uncomparable (Lamport 1980), (Emerson and Halpern 1983).

For the beginning, however, we emphasize on constant domains and endless time structures. Other generalizations will be investigated later.

2.2 Syntax and Semantics of Temporal Logic

We can immediately distinguish between three kinds of temporal operators, namely those with a universal character like *always, henceforth, hitherto*, those with an existential character like *at some time, eventually, previously*, and those with a mixed character like *until* and *since*. Each of these groups can be split into two subgroups, one which contains the corresponding operators that may refer to the present (as for instance *always, henceforth*) and one which contains the operators that exclude the present (as e.g. *eventually, previously*). Unfortunately, in the temporal logic literature it is often the case that different publications use the same symbols with different meanings. The first thing to do therefore is to provide a uniform syntactical description of the temporal operators known from the literature.

For instance, we would like to have a single basic symbol for operators with a universal character and also a single basic symbol for operators with an existential character. For the first class we like to take the symbol $\Box$ and for the second the symbol $\Diamond$ in order to avoid conflicts as much as possible. Now what kind of $\Box$-operators do we want? There are the past operators, the future operators and the operators which refer to the whole time axis. These operators have to be distinguished syntactically and we do so by an extra inner symbol which may be either "F", "P", or "A" according to whether the operator refers to the future, the past or anywhere on the time axis. Thus we have the symbols $\boxed{\mathrm{F}}$, $\boxed{\mathrm{P}}$ and $\boxed{\mathrm{A}}$ respectively. Analogously there are the corresponding $\Diamond$-operators: $\langle\mathrm{F}\rangle$, $\langle\mathrm{P}\rangle$ and $\langle\mathrm{A}\rangle$.

Next we have to indicate whether the present is included or not. The symbols from above are meant to represent the operators which exclude the present; if the present is to be included, a subscript "r" is added to the respective symbol[8]. So for example, *always* will be represented as $\boxed{\mathrm{A}}_r$, and *it was the case that* as $\langle\mathrm{P}\rangle$.

For the "Until" and "Since" operators things get a little more complicated. They combine both an existential phrase and a universal phrase. The existential part comes from the second argument of the operator which says: *there is a future moment such that* and the universal part comes from the first argument, namely *and for every moment inbetween it holds that*. For the first part we again have to distinguish between operators which may refer to the present and operators which may not. For the second part we have to indicate whether the respective interval borders are open or closed, and finally there is a need to differentiate between the respective strong or weak versions[9]. The basic symbol we are going to use for the "until" will be $\mathbf{U}$ and $\mathcal{U}$ and for the "Since" it will be $\mathbf{S}$ and $\mathcal{S}$ for the strong and weak versions respectively. For any of these four symbols there is actually a need for three arguments; two which express whether the interval borders are open or closed and one which excludes or includes the present for the existential phrase.. However, the addition of three more arguments to temporal operators does not increase the readability of logical formulae. We therefore decide to leave the Until- and Since-operator symbols as they are, and that with the intended meaning that the present is

[8]In modal logics the inclusion of the actual world is enforced by the requirement that the accessibility relation is reflexive, whence the subscript "r" comes from.

[9]The strong version of Φ Until Ψ guarantees that Ψ will inevitably hold, whereas the weak version does not. In the literature the phrase *unless* often occurs for a *weak until*

excluded for the "eventually"-part and the interval borders are both open. Unless otherwise stated this will be the informal semantics for the respective symbols.

Now we are in a position where we can describe the temporal logic syntax. As usual we have the logical symbols $\vee$, $\wedge$, $\neg$, $\Rightarrow$, and $\Leftrightarrow$ representing *or*, *and*, *not*, *implies*, and *equivalent* respectively. Also there are the symbols $\forall$ and $\exists$ representing the universal and the existential quantifiers. In addition we have the unary temporal operators $\boxed{F}_r$, $\boxed{F}$, $\boxed{P}_r$, $\boxed{P}$, and $\boxed{A}_r$[10] together with their dual counterparts $\Diamond_{F,r}$, $\Diamond_F$, $\Diamond_{P,r}$, $\Diamond_P$, and $\Diamond_{A,r}$ and the binary temporal operators $\mathbf{U}$, $\mathcal{U}$, $\mathbf{S}$ and $\mathcal{S}$. These symbols altogether are called the *logical symbols* of the temporal logic language. In addition there are the non-logical symbols which form the *signature* of the logic language in question. These are defined below.

DEFINITION 2.2.1 (THE SIGNATURE OF TL)
The alphabet of our temporal logic language consists of the operators and connectives from above as well as of the following sets of symbols:

- $\mathbf{V}$ *is a set of variable symbols*
- $\mathbf{F}$ *is a set of function symbols*
- $\mathbf{P}$ *is a set of predicate symbols*

The tuple $\Sigma_{\mathrm{TL}} := (\mathbf{V}, \mathbf{F}, \mathbf{P})$ *is called a* signature *for TL.*

Together with the preliminaries from above, the recursive definition of terms, atoms, and formulae is as usual:

DEFINITION 2.2.2 (TERMS, ATOMS, AND FORMULAE)
Let Σ_{TL} *be a signature for TL.*
Terms, atoms and formulae are defined as follows:

- *Each variable symbol* x *is a term*
- *If* f *is an n-ary function symbol and* $t_1 \ldots t_n$ *are terms then* $f(t_1, \ldots, t_n)$ *is a term*
- *If* P *is an n-ary predicate symbol and* $t_1 \ldots t_n$ *are terms then* $P(t_1, \ldots, t_n)$ *is an atom*
- *Each atom is a formula*
- *If* Φ *and* Ψ *are formulae and* x *is a variable symbol then* $\neg\Phi$, $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \Rightarrow \Psi$, $\Phi \Leftrightarrow \Psi$, $\forall x\, \Phi$ *and* $\exists x\, \Phi$ *are formulae*
- *If* Φ *is a formula and* $\triangle$ *is a unary temporal operators then* $\triangle\Phi$ *is a formula.*
- *If* Φ *and* Ψ *are formulae and* $\triangle$ *is a binary temporal operator then* $\Phi\triangle\Psi$ *is a formula*

We shall sometimes refer to $\mathbf{Form}_{\mathrm{TL}}$ *as the set of TL-formulae.*

[10] We might also think of an *always* which excludes the present. However, there seems no obvious need for an operator which states *at any time but now.*

The set $\mathbf{Form}_{\mathrm{TL}}$ represents the *language* of our logic. However there is still no *meaning* associated with the respective sentences. The predicate symbols as well as the function symbols are meant to denote predicates (relations) and functions in the "real" world. Such a correlation is usually expressed by a so-called *signature interpretation* which maps the respective symbols to real objects. The same will be done here, although there is some slight complication. What we need is a flexible interpretation, i.e. symbols may change their meaning over time. For instance, the constant *US–President* has a different value today than it had some 20 years ago. We therefore have to consider different signature interpretations for different time instants and we call these *local* to the given moment of time.

DEFINITION 2.2.3 (ALGEBRAS AND STRUCTURES)
An algebra *consists of a non-empty set (the algebra's domain) together with operations (functions) on this set. A* structure *is an algebra which additionally contains a set of relations over the algebra's domain.*

Algebras and structures are always associated with a given signature. The respective symbols of the given signature are interpreted with respect to a structure if each function symbol has its counterpart in the algebra and each predicate symbol has its counterpart in the structure's relation set.

A local signature interpretation in the sense from above is thus a structure which interprets function and predicate symbols with respect to some particular time instant, namely the one which is associated with this very structure. This evidently means that any symbol can be interpreted differently in different moments and they are called *flexible* then for that obvious reason. Symbols which do not change their meaning over time are said to be *rigid*.

A non-temporal formula (sentence) is said to hold in a particular instant if it holds under the associated local signature interpretation (structure). For an arbitrary temporal formula Φ we have to note that the reference point in time is shifted by the given scope of temporal operators. So for example $\boxed{\mathrm{A}}_r\Phi$ *holds now* if and only if *it is always true that (Φ holds now).* Hence, in order to interpret formulae we have to be able to refer to the *current now* and its respective local signature interpretation. An interpretation in our sense therefore includes a mapping from instants to local signature interpretations.

We shall distinguish between frames and interpretations where a frame is the basic underlying structure which includes the set of time instants and the corresponding earlier-later relation on those instants. From the introduction we know that there are certain agreed requirements on this relation. These will be fixed by the following definition:

DEFINITION 2.2.4 (FLOW RELATIONS)
A binary relation $\prec$ is called a flow relation *if it obeys the following properties:*

- *it is irreflexive, i.e.* $\forall x\; x \nprec x$
- *it is transitive, i.e.* $\forall x, y, z\; x \prec y \wedge y \prec z \Rightarrow x \prec z$
- *it is linear, i.e.* $\forall x, y\; x \neq y \Rightarrow x \prec y \vee y \prec x$
- *it is infinite, i.e.* $\forall x\; (\exists y\; x \prec y \wedge \exists z\; z \prec x)$

Such flow relations are what most people have in mind when they think about properties of *earlier-later* as a temporal relationship.

At this stage we have almost everything that is necessary to give a *meaning* to the sentences of the temporal logic language. What remains is to define what we understand by *interpretations* and how these interpretations can act as models for temporal logic formulae.

Definition 2.2.5 (Frames and Interpretations)
By a frame $\mathcal{F}_{\mathrm{TL}}$ for a signature Σ_{TL} we understand any pair $(\mathcal{T}, \prec)$ where $\mathcal{T}$ is a (non-empty) set of time instants and $\prec$ is a flow relation over $\mathcal{T} \times \mathcal{T}$.
By an interpretation $\Im_{\mathrm{TL}}$ for Σ_{TL} we understand any tuple $(\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \tau, \phi)$ where

- *$\mathcal{D}$ is the universe of discourse*
- *$\mathcal{F}_{\mathrm{TL}}$ is a frame for Σ_{TL}*
- *$\Im_{\mathrm{loc}}$ is a mapping from $\mathcal{T}$ to a set of local signature interpretations where all of the local signature interpretations have a common domain which is $\mathcal{D}$*
- *τ is the current instant, the* now
- *ϕ is a variable assignment, i.e. a function which maps variable symbols to elements of $\mathcal{D}$.*

Note that, by this definition, we assume a unique domain over the set of time instants; we are therefore dealing with a constant domain *time structure. The case of* varying domains *(i.e. the case where each time instant may have its own local domain) will be considered later.*

Frames and interpretations are the means to interpret formulae. The preliminaries from above already provide a hint on how this is actually going to be performed, namely by an inductive definition on the structure of the given temporal formula. The most basic case evidently is the interpretation of a single predicate. Such a predicate usually contains terms and therefore we also need a possibility to evaluate terms in a given interpretation. This is done by extending the notion of an interpretation to a homomorphism as follows: Each interpretation can easily evaluate variables by its variable assignment component. For more complex terms $f(t_1, \ldots, t_n)$ first the function symbol f has to be interpreted by the appropriate local signature interpretation. This on the other hand can be found by applying the mapping $\Im_{\mathrm{loc}}$ to the given current instant. Now after f has been interpreted the result has to be applied to the evaluation of the respective arguments and we are done.

Definition 2.2.6 (Interpretation of Terms)
Arbitrary terms t are evaluated as follows:

$$\Im_{\mathrm{TL}}(t) = \begin{cases} \phi(t) & \text{if } t \text{ is a variable symbol} \\ \hat{f}(\Im_{\mathrm{TL}}(t_1), \ldots, \Im_{\mathrm{TL}}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

where $\Im_{\mathrm{TL}} = (\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \tau, \phi)$ and $\hat{f}$ is the function associated with f under the local signature interpretation $\Im_{\mathrm{loc}}(\tau)$. For convenience we shall abbreviate this by $\hat{f} = (\Im_{\mathrm{loc}}(\tau))(f)$ in the sequel.

The following standard notation is introduced for convenience:

Definition 2.2.7
Let ϕ be a variable assignment. We define:

$$\phi(y)[x/a] = \begin{cases} \phi(y) & \text{if } y \neq x \\ a & \text{otherwise} \end{cases}$$

We usually abbreviate $(\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \tau, \phi[x/a])$ *by* $\Im_{\mathrm{TL}}[x/a]$ *and* $(\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \chi, \phi)$ *by* $\Im_{\mathrm{TL}}[\chi]$ *whenever* $\Im_{\mathrm{TL}}$ *is given by* $(\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \tau, \phi)$.

Formulae are interpreted with the help of a satisfiability relation $\models$ which takes two arguments, an interpretation and a formula, and returns true if and only if the formula holds in that particular interpretation.

DEFINITION 2.2.8 (SATISFIABILITY)
Let $\Im_{\mathrm{TL}} = (\mathcal{D}, \mathcal{F}_{\mathrm{TL}}, \Im_{\mathrm{loc}}, \tau, \phi)$ *be a TL-interpretation where* $\mathcal{F}_{\mathrm{TL}} = (\mathcal{T}, \prec)$ *is a frame for the signature* Σ_{TL}. *A formula* Φ *is said to hold in the interpretation* $\Im_{\mathrm{TL}}$ *if and only if* $\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi$ *holds, where* $\models_{\mathrm{TL}}$ *is recursively defined as follows:*
The first part is standard for first-order logics.

$$
\begin{array}{lll}
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} P(\ldots, t_i, \ldots) & \textit{iff} & \Im_{\mathrm{Loc}}(\tau)(P)(\ldots, \Im_{\mathrm{TL}}(t_i), \ldots) \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \neg\Phi & \textit{iff} & \textit{not } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \vee \Psi & \textit{iff} & \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \textit{ or } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Psi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \wedge \Psi & \textit{iff} & \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \textit{ and } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Psi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \Rightarrow \Psi & \textit{iff} & \textit{not } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \textit{ or } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Psi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \Leftrightarrow \Psi & \textit{iff} & \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi \Rightarrow \Psi \textit{ and } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Psi \Rightarrow \Phi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \forall x\, \Phi & \textit{iff} & \Im_{\mathrm{TL}}[x/a] \models_{\mathrm{TL}} \Phi \textit{ for every } a \in \mathcal{D} \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \exists x\, \Phi & \textit{iff} & \Im_{\mathrm{TL}}[x/a] \models_{\mathrm{TL}} \Phi \textit{ for some } a \in \mathcal{D}
\end{array}
$$

The second part covers the $\Box$ *and* $\Diamond$*-operators. Since the* $\Diamond$*-operator is dual to the* $\Box$ *a short form for the* $\Diamond$*-formulae is sufficient to give the idea.*

$$
\begin{array}{lll}
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{F}}_r\Phi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Phi \textit{ for any } \chi \in \mathcal{T} \textit{ with } \tau \preceq \chi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{F}}\Phi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Phi \textit{ for any } \chi \in \mathcal{T} \textit{ with } \tau \prec \chi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{P}}_r\Phi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Phi \textit{ for any } \chi \in \mathcal{T} \textit{ with } \chi \preceq \tau \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{P}}\Phi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Phi \textit{ for any } \chi \in \mathcal{T} \textit{ with } \chi \prec \tau \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{A}}_r\Phi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Phi \textit{ for any } \chi \in \mathcal{T} \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Diamond\Phi & \textit{iff} & \textit{not } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Box\neg\Phi
\end{array}
$$

where $\preceq$ *denotes the reflexive closure of* $\prec$

There are two basic symbols for the Until *and two basic symbols for the* Since *as we learned at the beginning of this chapter and actually each of them needs three arguments. It is enough here to provide the definition for the basic* Until *and* Since *formulae. The corresponding definitions for the others are obtained by replacing the one or the other* $\prec$ *with* $\preceq$.

$$
\begin{array}{lll}
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathbf{U}\,\Psi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Psi \textit{ for some } \chi \in \mathcal{T} \textit{ with } \tau \prec \chi \textit{ and} \\
 & & \Im_{\mathrm{TL}}[\xi] \models_{\mathrm{TL}} \Phi \textit{ for any } \xi \in \mathcal{T} \textit{ with } \tau \prec \xi \textit{ and } \xi \prec \chi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathcal{U}\,\Psi & \textit{iff} & \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathbf{U}\,\Psi \textit{ or } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{F}}\Phi \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathbf{S}\,\Psi & \textit{iff} & \Im_{\mathrm{TL}}[\chi] \models_{\mathrm{TL}} \Psi \textit{ for some } \chi \in \mathcal{T} \textit{ with } \chi \prec \tau \textit{ and} \\
 & & \Im_{\mathrm{TL}}[\xi] \models_{\mathrm{TL}} \Phi \textit{ for any } \xi \in \mathcal{T} \textit{ with } \chi \prec \xi \textit{ and } \xi \prec \tau \\
\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathcal{S}\,\Psi & \textit{iff} & \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi\,\mathbf{S}\,\Psi \textit{ or } \Im_{\mathrm{TL}} \models_{\mathrm{TL}} \boxed{\mathrm{P}}\Phi
\end{array}
$$

An interpretation $\Im_{\mathrm{TL}}$ *is said to* satisfy *a formula* Φ *if* $\Im_{\mathrm{TL}} \models_{\mathrm{TL}} \Phi$. Φ *is called* satisfiable *then and the corresponding interpretation is called a* model *for* Φ. *We call* Φ unsatisfiable *if no model for* Φ *exists.*

It is easy to see that most (linear) temporal logics known from the literature are instances (or subsystems) of this logic. For instance we get the first-order version of *S5* by eliminating all temporal operators but $\boxed{\mathrm{A}}_r$ and $\diamondsuit\!\!\!\!\;\mathrm{A}_r$. On the other hand first-order *S4.3* is realized by considering $\boxed{\mathrm{F}}_r$ and $\diamondsuit\!\!\!\!\;\mathrm{F}_r$ only whereas first-order linear Tense-Logic in the sense of Prior can be achieved by eliminating all operators but $\boxed{\mathrm{F}}$, $\boxed{\mathrm{P}}$, $\diamondsuit\!\!\!\!\;\mathrm{F}$, and $\diamondsuit\!\!\!\!\;\mathrm{P}$. Finally Kamp's tense logic essentially contains no other operators than $\mathbf{U}$ and $\mathbf{S}$ which are in some sense most general as any other temporal operator can be represented by these two. For instance $\boxed{\mathrm{A}}$ is equivalent to the conjunction of $\boxed{\mathrm{F}}$ and $\boxed{\mathrm{P}}$ where $\boxed{\mathrm{F}}\Phi$ can be expressed by $\neg(\mathbf{True}\ \mathbf{U}\ \neg\Phi)$ and $\boxed{\mathrm{P}}\Phi$ by $\neg(\mathbf{True}\ \mathbf{S}\ \neg\Phi)$. Proofs of these facts are easy but lengthy and will therefore be omitted here. The interested reader is referred to (Kamp 1968).

So far the temporal logic and some of its variants have been defined. There remains the question how it is possible to reason within these logics. Evidently, because of the rather complicated properties of flow relations one cannot seriously expect a – in any sense – simple calculus. And indeed the calculus itself and its preliminaries as they are developed for this thesis require a fairly gentle introduction. And what could be more *gentle* in this sense than the examination of logics which can be viewed as the forefathers of instant temporal logic, the modal logics. On the one hand their relation to temporal logics is close enough such that developments of calculi can quite easily be carried over to temporal logics. On the other hand they (or at least many of them) are simple enough to demonstrate main technical ideas.

3

A Digression to Modal Logics

A detailed overview on modal logics is out of the scope of this work. The reader not familiar with the basic modal logic principles is referred to (Chellas 1980) and (Hughes and Cresswell 1968). Nevertheless, it might be necessary to recall some technical preliminaries at least as long as they play a crucial role for the sections to follow.

In modal logics we talk essentially about the same non-logical and logical symbols as in temporal logics, however, we consider merely the two additional (modal) operators $\Box$ and $\Diamond$. This shows that the temporal logics we have in mind are actually (at least to some extent) multi-modal logics, i.e. modal logics with multiple modal operators. Temporal operators like *Until* and *Since* do not quite fall into this category, though.

The syntax and semantics for modal logic is by no means different to temporal logic's syntax and semantics, i.e. we also consider frames as a pair which consists of a set of worlds[1] and an accessibility relation[2]. A first-order modal logic interpretation then consists of a domain, a frame, a local signature interpretation which associates structures with each world, an inital world, and a variable assignment just as we had it for temporal logic. Also the interpretation of terms and the satisfiability of formulae does not change at all.

Since this work aims at providing a reasoning calculus for temporal logics and since modal logics are supposed to serve as a special case, we are now faced with the question how it is possible to reason within modal logics.

In fact, various calculi have been developed up to date, be it tableau, natural deduction or Gentzen systems (see (Hughes and Cresswell 1968) and (Fitting 1983) for instance). There

[1] These happened to be time instants in temporal interpretations.

[2] The earlier-later relation in temporal logics.

also had been attempts to extend the resolution idea in a way such that some kind of standard resolution can be applied to modal logics as well (see e.g. (Fariñas del Cerro 1985)).

Another approach – the one which is proposed in this thesis – is to find some suitable translation from modal logic into first-order predicate logic that allows us to exploit the reasoning techniques developed for classical logic in many years of examination and development[3].

The first idea one might have along these lines is to translate the semantics definition for modal logic directly into the predicate calculus and to utilize standard predicate logic theorem provers to perform reasoning within modal logics. This method – at least in its naive form – shows to be only little interesting because of the "practical incompleteness" it evokes. Nevertheless, since it will form a basis for the sequel, it is presented here[4].

3.1 Relational Translation for Modal Logics

For convenience recall the syntax and semantics definition for modal logics.

DEFINITION 3.1.1 (FRAMES AND INTERPRETATIONS)
By a frame $\mathcal{F}_{\mathrm{ML}}$ we understand any pair $(\mathcal{W}, \Re)$ where $\mathcal{W}$ is a non-empty set (of worlds) and $\Re$ is an arbitrary binary relation on $\mathcal{W}$ called the accessibility relation *between worlds.*
By a modal logic interpretation $\Im_{\mathrm{ML}}$ based on a frame $\mathcal{F}_{\mathrm{ML}} = (\mathcal{W}, \Re)$ we understand any tuple $(\mathcal{D}, \mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ where

- *$\mathcal{D}$ denotes a set of individuals; the universe of discourse*
- *$\mathcal{F}_{\mathrm{ML}}$ is a frame*
- *$\Im_{\mathrm{loc}}$ is a mapping from $\mathcal{W}$ to the set of local structures, where the respective domains all are identical to $\mathcal{D}$.*
- *τ denotes the actual world (the current situation)*
- *ϕ is a variable assignment, i.e. a function which maps domain variables to elements of the domain.*

We shall usually call $\Im_{\mathrm{ML}}$ a modal logic interpretation *over the modal logic signature.*

The similarities between temporal logic frames and interpretations are evident. For the moment, the only difference lies in the accessibility relation which is not yet supposed to obey any special properties. Also there is no particular difference in the interpretation of terms, nor in the definition of the satisfiability relation. For convenience, however, the satisfiability definition for modal logics is repeated here, since the following translation approach to be defined later directly depends on it.

[3]Note that this does not at all mean to agree with Quine's opinion that any special treatment, be it for temporalities or for modalities, should be encoded in a many-sorted classical setting. It just means that *reasoning* should be performed classically; the logic language remains as is (i.e. with modal or temporal operators).

[4]Actually, this kind of translation occurs in various books and papers (see e.g. (Moore 1980)). Unfortunately, the corresponding soundness and completeness proofs, i.e. the proofs that a given modal formula is (modal logic) satisfiable iff its translation is classically satisfiable, are usually omitted.

DEFINITION 3.1.2 (SATISFIABILITY)
Let $\Im_{\mathrm{ML}} = (\mathcal{D}, \mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ *be a modal logic interpretation where* $\mathcal{F}_{\mathrm{ML}} = (\mathcal{W}, \Re)$ *is a frame. A formula* Φ *is said to hold for the interpretation* $\Im_{\mathrm{ML}}$ *if and only if* $\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Phi$ *holds, where* $\models_{\mathrm{ML}}$ *is recursively defined as follows:*

$$\Im_{\mathrm{ML}} \models_{\mathrm{ML}} P(\ldots, t_i, \ldots) \quad \textit{iff} \quad \Im_{\mathrm{loc}}(\tau)(P)(\ldots, \Im_{\mathrm{ML}}(t_i), \ldots)$$

The cases for the classical logical connectives should be clear

$$\begin{array}{lll} \Im_{\mathrm{ML}} \models_{\mathrm{ML}} \forall x \Phi & \textit{iff} & \Im_{\mathrm{ML}}[x/a] \models_{\mathrm{ML}} \Phi \textit{ for every } a \in \mathcal{D} \\ \Im_{\mathrm{ML}} \models_{\mathrm{ML}} \exists x \Phi & \textit{iff} & \Im_{\mathrm{ML}}[x/a] \models_{\mathrm{ML}} \Phi \textit{ for some } a \in \mathcal{D} \\ \Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Box \Phi & \textit{iff} & \Im_{\mathrm{ML}}[\chi] \models_{\mathrm{ML}} \Phi \textit{ for every } \chi \in \mathcal{W} \\ & & \textit{such that } \Re(\tau, \chi) \\ \Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Diamond \Phi & \textit{iff} & \Im_{\mathrm{ML}}[\chi] \models_{\mathrm{ML}} \Phi \textit{ for some } \chi \in \mathcal{W} \\ & & \textit{with } \Re(\tau, \chi) \end{array}$$

An interpretation $\Im_{\mathrm{ML}}$ *is said to* satisfy *a formula* Φ *if* $\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Phi$. Φ *is called* satisfiable *then and the corresponding interpretation is called a* model *for* Φ. *We call* Φ unsatisfiable *if no model for* Φ *exists.*

The simplest *normal* modal logic (called K) is axiomatized by the standard axioms for classical propositional logic and the Modus Ponens inference rule together with the axiom schema $\Box(\Phi \Rightarrow \Psi) \Rightarrow (\Box\Phi \Rightarrow \Box\Psi)$, which is usually called the *K-axiom*, plus the additional *necessitation rule*[5]

$$\frac{\vdash \Phi}{\vdash \Box\Phi}$$

It can be shown that a modal logic formula is provable in this axiomatization if and only if it holds in every interpretation of the above kind (i.e. without any special $\Re$-properties). This kind of semantics (called possible-world-semantics, or Kripke-style semantics) had been developed by Saul Kripke in the early sixties (Kripke 1963). What made this kind of semantics particularly interesting was the realization that certain axiom schemata added to this axiomatization are mirrored by certain properties of the accessibility relation (see also (Hughes and Cresswell 1968), (Chellas 1980)), also called the background theory of the logic under consideration. It is not necessary here to provide a detailed list of all modal logics, schemata, and corresponding accessibility relation properties. Only some of the most important modal logics which will play a role in the sequel are summarized in Table 3.1 together with their respective background theories.

Such correspondences can be found in the standard modal logic literature. Nevertheless, it is often interesting to know how these are obtained. This is of particular importance if an axiom occurs which is not as common as the ones listed above. A method for computing such correspondences is presented in Section 3.1.2.

Since different modal logics are distinguished by their respective additional axiom schemata and since these axioms are mirrored on the semantics level by certain accessibility relation properties, the relational translation not only has to take into account the theorem to be proved; also the characterizing properties are to be added. Moreover, in order to be able to prove that the given translation preserves both satisfiability and unsatisfiability the given signature and interpretations have to be translated as well.

[5]These axioms say nothing about the $\Diamond$-operator and therefore a $\Diamond$ is in fact considered as a short form for $\neg\Box\neg$. If this is not desired, the schema $\Diamond\Phi \equiv \neg\Box\neg\Phi$ is to be added to the axiomatization.

LOGIC	SCHEMATA	PROPERTIES
KT	$\Box\Phi \Rightarrow \Phi$	$\forall u\ \Re(u,u)$
KB	$\Phi \Rightarrow \Box\Diamond\Phi$	$\forall u,v\ \Re(u,v) \Rightarrow \Re(v,u)$
KD4	$\Box\Phi \Rightarrow \Diamond\Phi$ $\Box\Phi \Rightarrow \Box\Box\Phi$	$\forall u \exists v\ \Re(u,v)$ $\forall u,v,w\ \Re(u,v) \wedge \Re(v,w) \Rightarrow \Re(u,w)$
S4	$\Box\Phi \Rightarrow \Phi$ $\Box\Phi \Rightarrow \Box\Box\Phi$	$\forall u\ \Re(u,u)$ $\forall u,v,w\ \Re(u,v) \wedge \Re(v,w) \Rightarrow \Re(u,w)$
KD45	$\Box\Phi \Rightarrow \Diamond\Phi$ $\Box\Phi \Rightarrow \Box\Box\Phi$ $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$	$\forall u \exists v\ \Re(u,v)$ $\forall u,v,w\ \Re(u,v) \wedge \Re(v,w) \Rightarrow \Re(u,w)$ $\forall u,v,w\ \Re(u,v) \wedge \Re(u,w) \Rightarrow \Re(v,w)$
S5	$\Box\Phi \Rightarrow \Phi$ $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$	$\forall u\ \Re(u,u)$ $\forall u,v,w\ \Re(u,v) \wedge \Re(u,w) \Rightarrow \Re(v,w)$

Table 3.1: Modal Logic Correspondences

From the definitions above we see that a modal logic essentially consists of the following parts; the signature, i.e. the alphabet of the language, a (usually recursive) definition of the elements of the logic (the sentences), the definition of appropriate interpretations, and a satisfiability relation between interpretations and sentences. Now, given two different logics, a certain translation is supposed to map these parts from the one logic to the corresponding parts of the other. Therefore such a translation has to be split into three main components: a signature translation, a formula translation and an interpretation translation.

3.1.1 The Respective Translation Components

In this section the respective components for the relational translation approach are presented and that together with the necessary soundness and completeness proofs. For convenience I will use the same "translation symbol" $\lceil\ \rceil$ for each of the components. It will always be clear from the context which translation component is meant.

DEFINITION 3.1.3 (THE SIGNATURE TRANSLATION)
Let $\Sigma_{\mathrm{ML}} := (\mathbf{V}, \mathbf{F}, \mathbf{P})$ *be a modal logic signature.*

For each n-place f in $\mathbf{F}$ let f' be a new function symbol and for each n-place P in $\mathbf{P}$ let P' be a new predicate symbol. Additionally we assume a sort symbol, W, which is supposed to represent the sort of worlds under consideration and D which denotes the sort of individuals[6].
Then let $\mathbf{F}' = \{f' \mid f \in \mathbf{F}\} \cup \{\iota\}$ *and* $\mathbf{P}' = \{P' \mid P \in \mathbf{P}\} \cup \{R\}$.

We then define: $\lceil\Sigma_{\mathrm{ML}}\rceil = (\mathbf{V}, \mathbf{F}', \mathbf{P}')$

$\lceil\Sigma_{\mathrm{ML}}\rceil$ *is then called the* predicate logic signature generated from Σ_{ML}.

[6]The reader should not be too bothered with this introduction of sorts which have not been mentioned before. In fact, the most we are dealing with is a multi-sorted rather than an order-sorted scenario in which the respective sorts cannot be mixed up since each argument position has a fixed sort and there are no sub-sort relations involved.

Obviously the symbols R and ι are supposed to refer to the accessibility relation $\Re$ and the "actual" world τ respectively.

Definition 3.1.4 (The Formula Translation)
There are actually two parts to be defined: a translation of terms and *a translation of formulae. For convenience both are given by the same symbol.*

Let t be an arbitrary term and let u denote a world, i.e. u is either a variable symbol of sort W or the world constant ι. We define $\lceil t \rceil_u$ by induction on the structure of t:

$$\begin{array}{lcl} \lceil x \rceil_u & = & x \\ \lceil f(t_1, \dots, t_n) \rceil_u & = & f'(u, \lceil t_1 \rceil_u, \dots, \lceil t_n \rceil_u) \end{array}$$

Now let Φ be an arbitrary modal logic formula. $\lceil \Phi \rceil_u$ is inductively defined as follows:

$$\begin{array}{lcl} \lceil P(t_1, \dots, t_n) \rceil_u & = & P'(u, \lceil t_1 \rceil_u, \dots, \lceil t_n \rceil_u) \\ \lceil \neg \Phi \rceil_u & = & \neg \lceil \Phi \rceil_u \\ \lceil \Phi \vee \Psi \rceil_u & = & \lceil \Phi \rceil_u \vee \lceil \Psi \rceil_u \\ \multicolumn{3}{l}{\textit{and analogously for the other classical connectives}} \\ \lceil \forall x\ \Phi \rceil_u & = & \forall x\ \lceil \Phi \rceil_u \\ \lceil \exists x\ \Phi \rceil_u & = & \exists x\ \lceil \Phi \rceil_u \\ \lceil \Box \Phi \rceil_u & = & \forall v\ R(u, v) \Rightarrow \lceil \Phi \rceil_v \\ \lceil \Diamond \Phi \rceil_u & = & \exists v\ R(u, v) \wedge \lceil \Phi \rceil_v \end{array}$$

The initial call for the translation of an arbitrary modal formula Φ is then $\lceil \Phi \rceil_\iota$, where ι denotes the initial (or actual) world.

Definition 3.1.5 (The Interpretation Translation)
Let $\Im_{\mathrm{ML}} = (\mathcal{D}, \mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ be a modal logic interpretation over the signature Σ_{ML}.
For any function symbol f in Σ_{ML} let $\Im_{\mathrm{loc}}(\tau)(f) = \hat{f}_\tau$ and define $\hat{f}$ as:

$$\hat{f}(\tau, t_1, \dots, t_n) = \hat{f}_\tau(t_1, \dots, t_n)$$

Analogously for any predicate symbol P let $\Im_{\mathrm{loc}}(\tau)(P) = \hat{P}_\tau$ and define

$$\hat{P}(\tau, t_1, \dots, t_n) = \hat{P}_\tau(t_1, \dots, t_n)$$

Now let $\mathcal{M}$ be a $\lceil \Sigma_{\mathrm{ML}} \rceil$-structure with:

- $\mathcal{M}(f') = \hat{f}$ *for each function symbol* f
- $\mathcal{M}(P') = \hat{P}$ *for each predicate symbol* P
- $\mathcal{M}(R) = \Re$
- $\mathcal{M}(\iota) = \tau$

We then define $\lceil \Im_{\mathrm{ML}} \rceil = (\mathcal{M}, \phi)$ and call $\lceil \Im_{\mathrm{ML}} \rceil$ the (classical) interpretation generated from $\Im_{\mathrm{ML}}$.

As we now have a translation from first-order modal logic into first-order predicate logic, we have to show that the translation indeed behaves as desired, i.e. we have to show that whenever a modal logic formula has a (Kripke-) model then its translation has a (classical) model as well and vice versa. The corresponding proofs are performed by induction over the structure of modal logic formulae. To this end the following auxiliary lemma turns out to be useful.

LEMMA 3.1.6
Let $\Im_{\mathrm{ML}} = (\mathcal{D}, (\mathcal{W}, \Re), \Im_{\mathrm{loc}}, \tau, \phi)$ be a modal logic interpretation over the signature Σ_{ML}, let Φ be a modal logic formula, χ be a world from $\mathcal{W}$, and let u be some world term. Then

$$\lceil \Im_{\mathrm{ML}}[\chi] \rceil \models_{\mathrm{PL}} \lceil \Phi \rceil_\iota \textit{ iff } \lceil \Im_{\mathrm{ML}} \rceil [u/\chi] \models_{\mathrm{PL}} \lceil \Phi \rceil_u$$

where $\models_{\mathrm{PL}}$ denotes the classical predicate logic satisfiability relation.

Proof: First it has to be show that the respective interpretations of terms are identical. This is done by an induction over the term structure:

$$\begin{array}{rcl}
\lceil \Im_{\mathrm{ML}}[\chi] \rceil (\lceil x \rceil_\iota) & = & \phi(x) \\
 & = & \lceil \Im_{\mathrm{ML}} \rceil [u/\chi](\lceil x \rceil_u) \\
\lceil \Im_{\mathrm{ML}}[\chi] \rceil (\lceil t \rceil_\iota) & = & \lceil \Im_{\mathrm{ML}}[\chi] \rceil (\lceil f(\ldots, t_i, \ldots) \rceil_\iota) \\
 & = & \lceil \Im_{\mathrm{ML}}[\chi] \rceil (f'(\iota, \ldots, \lceil t_i \rceil_\iota, \ldots) \\
 & = & \hat{f}(\chi, \ldots, \lceil \Im_{\mathrm{ML}}[\chi] \rceil (\lceil t_i \rceil_\iota), \ldots) \\
 & = & \hat{f}(\chi, \ldots, \lceil \Im_{\mathrm{ML}} \rceil [u/\chi](\lceil t_i \rceil_u), \ldots) \\
 & & \text{by the induction hypothesis} \\
 & = & \lceil \Im_{\mathrm{ML}} \rceil [u/\chi](f'(u, \ldots, \lceil t_i \rceil_u, \ldots) \\
 & = & \lceil \Im_{\mathrm{ML}} \rceil [u/\chi](\lceil f(\ldots, t_i, \ldots) \rceil_u) \\
 & = & \lceil \Im_{\mathrm{ML}} \rceil [u/\chi](\lceil t \rceil_u)
\end{array}$$

Now we can start to prove the lemma for arbitrary modal logic formulae. Note that if Φ is an atom then the proof works similar to the case of the complex terms from above. Also if $\Phi = \neg\Psi$ or $\Phi = \Psi_1 \circ \Psi_2$, where $\circ$ is any classical logical connective, there is no problem at all. Even if $\Phi = \forall x\ \Psi$ or if $\Phi = \exists x\ \Psi$ no difficulties do occur. Therefore consider the case where $\Phi = \Box\Psi$:

$$\begin{array}{ll}
\multicolumn{2}{l}{\lceil \Im_{\mathrm{ML}}[\chi] \rceil \models_{\mathrm{PL}} \lceil \Box\Psi \rceil_\iota} \\
\text{iff} & \lceil \Im_{\mathrm{ML}}[\chi] \rceil \models_{\mathrm{PL}} \forall v\ R(\iota, v) \Rightarrow \lceil \Psi \rceil_v \\
\text{iff} & \lceil \Im_{\mathrm{ML}}[\chi] \rceil [v/\xi] \models_{\mathrm{PL}} R(\iota, v) \Rightarrow \lceil \Psi \rceil_v \\
 & \text{for any world } \xi \\
\text{iff} & \Re(\chi, \xi) \text{ implies } \lceil \Im_{\mathrm{ML}}[\chi] \rceil [v/\xi] \models_{\mathrm{PL}} \lceil \Psi \rceil_v \\
 & \text{for any world } \xi \\
\text{iff} & \Re(\chi, \xi) \text{ implies } \lceil \Im_{\mathrm{ML}}[\chi][\xi] \rceil \models_{\mathrm{PL}} \lceil \Psi \rceil_\iota \\
 & \text{by induction hypothesis} \\
\text{iff} & \Re(\chi, \xi) \text{ implies } \lceil \Im_{\mathrm{ML}}[\xi] \rceil \models_{\mathrm{PL}} \lceil \Psi \rceil_\iota
\end{array}$$

$$
\begin{array}{ll}
\text{iff} & \Re(\chi,\xi) \text{ implies } \lceil\Im_{\mathrm{ML}}\rceil[v/\xi] \models_{\mathrm{PL}} \lceil\Psi\rceil_v \\
& \text{by induction hypothesis} \\
\text{iff} & \Re(\chi,\xi) \text{ implies } \lceil\Im_{\mathrm{ML}}\rceil[u/\chi][v/\xi] \models_{\mathrm{PL}} \lceil\Psi\rceil_v \\
& u \text{ is not free in } \lceil\Psi\rceil_v \\
\text{iff} & \lceil\Im_{\mathrm{ML}}\rceil[u/\chi][v/\xi] \models_{\mathrm{PL}} R(u,v) \Rightarrow \lceil\Psi\rceil_v \\
\text{iff} & \lceil\Im_{\mathrm{ML}}\rceil[u/\chi] \models_{\mathrm{PL}} \forall v\ R(u,v) \Rightarrow \lceil\Psi\rceil_v \\
\text{iff} & \lceil\Im_{\mathrm{ML}}\rceil[u/\chi] \models_{\mathrm{PL}} \lceil\Box\Psi\rceil_u
\end{array}
$$

and similarly for $\Phi = \Diamond\Psi$

With the help of the above auxiliary lemma the soundness of the relational translation is proved as follows:

LEMMA 3.1.7
Let $\Im_{\mathrm{ML}} = (\mathcal{D}, (\mathcal{W}, \Re), \Im_{\mathrm{loc}}, \tau, \phi)$ be a modal logic interpretation over Σ_{M}, and let Ψ be a modal logic formula. Then

$$\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Psi \qquad \textit{iff} \qquad \lceil\Im_{\mathrm{ML}}\rceil \models_{\mathrm{PL}} \lceil\Psi\rceil_\iota$$

Proof: We first have to consider the evaluation of terms. Therefore we have to show that $\Im_{\mathrm{ML}}(t) = \lceil\Im_{\mathrm{ML}}\rceil(\lceil t\rceil_\iota)$ for an arbitrary term t.

$$
\begin{array}{lcl}
\Im_{\mathrm{ML}}(x) & = & \phi(x) \\
& = & \lceil\Im_{\mathrm{ML}}\rceil(\lceil x\rceil_\iota) \\
\Im_{\mathrm{ML}}(f(\ldots,t_i,\ldots)) & = & \hat{f}_\tau(\ldots,\Im_{\mathrm{ML}}(t_i),\ldots) \\
& = & \hat{f}(\tau,\ldots,\lceil\Im_{\mathrm{ML}}\rceil(\lceil t_i\rceil_\iota),\ldots) \\
& & \text{by the induction hypothesis} \\
& = & \lceil\Im_{\mathrm{ML}}\rceil(f'(\iota,\ldots,\lceil t_i\rceil_\iota,\ldots)) \\
& = & \lceil\Im_{\mathrm{ML}}\rceil(\lceil f(\ldots,t_i,\ldots)\rceil_\iota)
\end{array}
$$

Thus the relational translation behaves as desired on the evaluation of terms. With that we can now prove the lemma by induction on the structure of Ψ. Note that the base case (where Ψ is a literal) and the cases where Ψ is composed by two fomulae and a classical logical connective are again obvious. Also there are no problems if Ψ is a quantified formula. Therefore consider the case where $\Psi = \Box\Phi$:

$$
\begin{array}{ll}
\multicolumn{2}{l}{\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Box\Phi} \\
\text{iff} & \Re(\tau,\chi) \text{ implies } \Im_{\mathrm{ML}}[\chi] \models_{\mathrm{ML}} \Phi \\
& \text{for any world } \chi \\
\text{iff} & \Re(\tau,\chi) \text{ implies } \lceil\Im_{\mathrm{ML}}[\chi]\rceil \models_{\mathrm{PL}} \lceil\Phi\rceil_\iota \\
& \text{by the induction hypothesis} \\
\text{iff} & \Re(\tau,\chi) \text{ implies } \lceil\Im_{\mathrm{ML}}\rceil[w/\chi] \models_{\mathrm{PL}} \lceil\Phi\rceil_w \\
& \text{by Lemma 3.1.6} \\
\text{iff} & \lceil\Im_{\mathrm{ML}}\rceil[w/\chi] \models_{\mathrm{PL}} R(\iota,w) \Rightarrow \lceil\Phi\rceil_w \\
\text{iff} & \lceil\Im_{\mathrm{ML}}\rceil \models_{\mathrm{PL}} \lceil\Box\Phi\rceil_\iota
\end{array}
$$

and analogous for $\Psi = \Diamond\Phi$

Hence relational translation is sound in the sense that for any interpretation satisfying a modal logic formula Φ there is a predicate logic interpretation satisfying the translated version of Φ. In other words, the translation preserves satisfiability. In particular, if the accessibility relation $\Re$ for the given interpretation obeys certain properties the corresponding generated predicate logic interpretation satisfies the respective translations.

Now it is necessary to show that unsatisfiability is preserved as well, i.e. that the translation is complete. To this end a "reverse-translation" is defined which generates a modal logic interpretation from a given predicate logic interpretation. Note that in proving the soundness of the reverse-translation the completeness of the actual translation is shown.

DEFINITION 3.1.8 (THE REVERSE TRANSLATION $\lceil \Im_{\mathrm{PL}} \rceil^{-1}$)
Let $\Im_{\mathrm{PL}} = (\mathcal{M}, \phi)$ be a classical interpretation over the signature $\lceil \Sigma_{\mathrm{ML}} \rceil$ and let $\Im_{\mathrm{ML}} = (\mathcal{D}, (\mathcal{W}, \Re), \Im_{\mathrm{loc}}, \tau, \phi)$ be a modal logic interpretation over Σ_{ML} where:

- $\Re = \mathcal{M}(R)$
- $\mathcal{W} = \mathcal{M}(W)$
- $\mathcal{D} = \mathcal{M}(D)$
- $\tau = \mathcal{M}(\iota)$
- *for any τ' in $\mathcal{W}$: $\Im_{\mathrm{loc}}(\tau')(f) = \hat{f}_{\tau'}$ and $\Im_{\mathrm{loc}}(\tau')(P) = \hat{P}_{\tau'}$*

We define $\lceil \Im_{\mathrm{PL}} \rceil^{-1} = \Im_{\mathrm{ML}}$ and call $\Im_{\mathrm{ML}}$ the (modal logic) interpretation generated from $\Im_{\mathrm{PL}}$.

LEMMA 3.1.9
Let $\Im_{\mathrm{PL}}$ be a classical interpretation over the signature $\lceil \Sigma_{\mathrm{ML}} \rceil$ and let $\chi \in \mathcal{M}(W)$. Then for any modal logic formula Φ:

$$\Im_{\mathrm{PL}}[u/\chi] \models_{\mathrm{PL}} \lceil \Phi \rceil_u \qquad \textit{iff} \qquad \lceil \Im_{\mathrm{PL}} \rceil^{-1}[\chi] \models_{\mathrm{ML}} \Phi$$

Proof: Works similar to the soundness proof in Lemma 3.1.7. It is thus sufficient to consider only the critical case where $\Phi = \Box\Psi$.

$$\begin{array}{ll}
\multicolumn{2}{l}{\Im_{\mathrm{PL}}[u/\chi] \models_{\mathrm{PL}} \lceil \Box\Psi \rceil_u} \\
\text{iff} & \Im_{\mathrm{PL}}[u/\chi] \models_{\mathrm{PL}} \forall v\ R(u,v) \Rightarrow \lceil \Psi \rceil_v \\
\text{iff} & \Re(\chi,\xi) \text{ implies } \Im_{\mathrm{PL}}[u/\chi][v/\xi] \models_{\mathrm{PL}} \lceil \Psi \rceil_v \\
\text{iff} & \Re(\chi,\xi) \text{ implies } \Im_{\mathrm{PL}}[v/\xi] \models_{\mathrm{PL}} \lceil \Psi \rceil_v \\
 & \text{since } u \text{ is not free in } \lceil \Psi \rceil_v \\
\text{iff} & \Re(\chi,\xi) \text{ implies } \lceil \Im_{\mathrm{PL}} \rceil^{-1}[\xi] \models_{\mathrm{ML}} \Psi \\
 & \text{by the induction hypothesis} \\
\text{iff} & \Re(\chi,\xi) \text{ implies } \lceil \Im_{\mathrm{PL}} \rceil^{-1}[\chi][\xi] \models_{\mathrm{ML}} \Psi \\
\text{iff} & \lceil \Im_{\mathrm{PL}} \rceil^{-1}[\chi] \models_{\mathrm{ML}} \Box\Psi
\end{array}$$

COROLLARY 3.1.10
Let $\Im_{\mathrm{PL}}$ be a classical model for $\lceil \Phi \rceil_\iota$. Then $\lceil \Im_{\mathrm{PL}} \rceil^{-1}$ is a Kripke-model for Φ.
Proof: Follows easily from Lemma 3.1.9 if u is set to ι and χ is set to $\mathcal{M}(\iota) = \tau$.

Thus both soundness and completeness of the relational translation have been shown. Now assume that there is a modal logic formula Φ to be proved valid in some particular modal logic, say *KD45*. This can be performed by proving the unsatisfiability of $\neg\Phi$. In order to do this the relational translation technique allows us to translate $\neg\Phi$ into $\Psi = \lceil\neg\Phi\rceil_\iota$ and to prove the unsatisfiability of Ψ instead, provided the necessary additional axioms of the modal logic under consideration are added. In this example the respective axioms *D*, *4*, and *5* have to be included, i.e. $\forall x \exists y\ R(x,y)$, $\forall x,y,z\ R(x,y) \wedge R(y,z) \Rightarrow R(x,z)$ and $\forall x,y,z\ R(x,y) \wedge R(x,z) \Rightarrow R(y,z)$. The following theorem fixes this in a more general setting. It is hereby assumed that ML is an arbitrary modal logic (as e.g. *KD45* and that ML* represents the set of corresponding additional axioms (which are seriality, transitivity, and euclideaness in the case of *KD45*).

Theorem 3.1.11
Φ is a theorem of the modal logic ML if and only if any predicate logic interpretation satisfying the ML-Axioms (ML*) is a model for $\lceil\Phi\rceil_\iota$. Formally:

$$\models_{\mathrm{ML}} \Phi \qquad \textit{iff} \qquad \mathrm{ML}^* \models_{\mathrm{PL}} \lceil\Phi\rceil_\iota$$

Proof: Assume $\models_{\mathrm{ML}} \Phi$ and assume further that there exists a predicate logic interpretation which satisfies the axioms of ML* but not the formula $\lceil\Phi\rceil_\iota$. Then this PL-interpretation satisfies $\lceil\neg\Phi\rceil_\iota$. However, Corollary 3.1.10 then guarantees the existence of a modal logic interpretation which satisfies $\neg\Phi$ and this contradicts the assumption that $\models_{\mathrm{ML}} \Phi$.
On the other hand assume that every predicate logic interpretation which is a model for the ML*-Axioms also satisfies $\lceil\Phi\rceil_\iota$ and that there is a modal logic interpretation which is an ML-model for $\neg\Phi$. But then there exists a predicate logic interpretation satisfying the ML*-Axioms which is model for $\lceil\neg\Phi\rceil_\iota$ which again contradicts our assumption.

Thus we are able to reason within modal logics just by translating the theorem to be proved into classical first-order logic and to use standard predicate logic theorem provers (e.g. resolution-based) to do the job. However, as it turns out, almost every example – provided it is not too trivial – results in a clause set which requires to rummage through a huge search space simply because all the special knowledge we originally had about worlds and accessibility relations is not anymore separated from the actual theorem and thus can hardly be treated in a special manner. As an example consider the following:

Example 3.1.12
Suppose it is our aim to prove the *S4*-validity of the formula

$$\Diamond\Box P \Leftrightarrow \Diamond\Box\Diamond\Box P$$

After negation, translation and clause form generation we end up with a clause sets which consists of 14 fairly complicated clauses with 53 literals (see Table 3.2). What is particularly discouraging about these clauses is that almost every literal is an R-literal and that the search space is almost intractable for any standard theorem prover. And in spite of the fact that this is a rather simple theorem even sophisticated predicate logic theorem provers have great difficulties in actually finding a proof.

The aim should therefore be to keep the separation between the knowledge we have about the frame from the knowledge induced by the theorem. This can hardly be done on the basis of the relational translation as presented in this section. Rather we need another kind of translation – and thus modal logic semantics – which help us in this respect. Such a new translation and semantics (called *semi-functional*) is presented in later sections.

$$
\begin{array}{l}
R(\iota,a), R(\iota,b) \\
\neg R(b,x), R(\iota,a), R(x,f_1(x)) \\
\neg R(b,x), \neg R(f_1(x),y), R(\iota,a), P(y) \\
\neg R(a,x), P(x), R(\iota,b) \\
\neg R(a,x), \neg R(b,y), P(x), R(y,f_1(y)) \\
\neg R(a,x), \neg R(b,y), \neg R(f_1(y),z), P(x), P(z) \\
\neg R(\iota,x), \neg R(\iota,y), R(x,f_2(x)), R(y,f_3(y)) \\
\neg R(\iota,x), \neg R(\iota,y), \neg R(f_3(y),z), R(x,f_2(x)), R(z,f_4(z,y)) \\
\neg R(\iota,x), \neg R(\iota,y), \neg R(f_3(y),z), \neg P(f_4(z,y)), R(x,f_2(x)) \\
\neg R(\iota,x), \neg P(f_2(x)), \neg R(\iota,y), R(y,f_3(y)) \\
\neg R(\iota,x), \neg P(f_2(x)), \neg R(\iota,y), \neg R(f_3(y),z), R(z,f_4(z,y)) \\
\neg R(\iota,x), \neg P(f_2(x)), \neg R(\iota,y), \neg R(f_3(y),z), \neg P(f_4(z,y)) \\
R(x,x) \\
\neg R(x,y), \neg R(y,z), R(x,z)
\end{array}
$$

Table 3.2: Clauses for $\Diamond\Box P \not\Leftarrow \Diamond\Box\Diamond\Box P$

However, before we take a look at the semi-functioal translation approach let us first examine how modal logic correspondences can be obtained algorithmically for this will also be needed later on.

3.1.2 On Automating Correspondence Theory

Most of the accessibility relation properties that are induced by certain axiomatizations of modal and temporal logics considered in this work can actually be found in the standard literature. However, it is often not obvious at the first glance how these properties can be obtained from a given axiomatization. Until recently, such examinations of frame properties were mainly based either on pure model theoretic and algebraic considerations ((Segerberg 1971) and (Sahlqvist 1975)) or a rather non-algorithmic way of finding first-order equivalents for second-order (in fact Π^1_1) formulae (van Benthem 1984a). How such results can be obtained algorithmically was an open problem for quite some time.

In 1992 Dov Gabbay and Hans Jürgen Ohlbach (see (Gabbay and Ohlbach 1992a) and (Gabbay and Ohlbach 1992b)) came up with the SCAN algorithm which allows us to transform Σ^1_1-formulae, i.e. formulae of the form $\exists P\ \Psi$ where Ψ is a first-order formula, into equivalent first-order formulae provided they exist at all. Evidently, it is of no real importance whether such an algorithm transforms Σ^1_1-formulae or Π^1_1-formulae for a formula of the one kind can be viewed as the negation of a formula of the other kind. The main idea behind this approach is to transform the given Σ^1_1-formula $\exists P\ \Psi$ into clause normal form (thus skolemization gets involved) and to perform all possible *constraint* resolution steps between P-literals[7]. If this terminates all the clauses which contain P-literals can be deleted (by the so called *purity deletion*) and the remaining clause set provides with the desired result. However, since the original formula had been skolemized, it is necessary to "deskolemize" the result thus obtained, i.e. all terms which

[7]A constraint resolution step is a resolution step in which unification does not directly take place which means that inequalities get produced which act as certain resolution residues. Therefore two arbitrary terms a and b can be "unified" by introducing the residue $a \neq b$.

had been introduced by skolemization have to be back-translated into existential quantifiers. If this also succeeds, a first-order equivalent to the original second-order formula has been found.

There are two main possibilities which may prevent a successful termination of the whole approach. First, it may happen that the resolution process does not terminate because new clauses which contain P-literals get produced again and again and, second, even if this resolution process terminates, it might be impossible to deskolemize the resulting clause set. As a simple example for a successful application of the SCAN algorithm consider the modal axiom schema $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$. In the relational translation approach this schema gets translated into

$$\forall u \forall \Phi \ [(\exists v \ R(u,v) \wedge \Phi(v)) \Rightarrow (\forall v \ R(u,v) \Rightarrow \exists w \ R(v,w) \wedge \Phi(w))]$$

Here the formula variable Φ is universally quantified; in order to be able to apply the SCAN algorithm it is therefore necessary to negate this translation result (bearing in mind that the final outcome has to be negated once again). Negating and transforming into clause form then results in

$$\begin{array}{l} R(a,b) \\ \Phi(b) \\ R(a,c) \\ \neg R(c,x), \neg\Phi(x) \end{array}$$

Evidently, there is only one (constraint) resolution step possible between Φ-literals and after performing this step we get

$$\begin{array}{l} R(a,b) \\ R(a,c) \\ \neg R(c,b) \end{array}$$

The constants a, b and c had been introduced by skolemization, hence they have to be back-translated into existential quantifiers. We thus get:

$$\exists u,v,w \ R(u,v) \wedge R(u,w) \wedge \neg R(v,w)$$

Finally, this result has to be negated again and we end up with

$$\forall u,v,w \ R(u,v) \wedge R(u,w) \Rightarrow R(v,w)$$

i.e. the axiom schema $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$ characterizes the euclidean frames.

The SCAN algorithm works fine for most of the axiom schemata that usually occur in the modal and temporal logic literature[8]. Nevertheless, some examples are known for which first-order equivalents exist but SCAN does not terminate when fed with these examples. Usually, the resolvents SCAN produces while in progress do obey certain syntactic patterns and it can be observed that SCAN's only problem is that it is not able to deal with such patterns. This observation lead to another method of eliminating second-order quantifiers in (Nonnengart and Szałas 1995). In this approach such patterns are represented by fixpoint formulae of the kind $\mu P.\Phi(P)$ ($\nu P.\Phi(P)$ respectively), where P is positive in Φ (i.e. the negation normal form of Φ contains only positive Ps). It is not necessary to provide a very detailed description of the technical background behind this approach. Only the most important points are presented here. The interested reader is referred to (Nonnengart and Szałas 1995).

[8]In cases where a given axiom schema has no first-order equivalent, however, the algorithm evidently cannot terminate successfully.

DEFINITION 3.1.13 (FIXPOINT CHARACTERIZATION)

$$\mu P.\Phi(P) \Leftrightarrow \bigvee_{\beta\in\alpha} \Phi^{\beta}(\bot)$$

$$\nu P.\Phi(P) \Leftrightarrow \bigwedge_{\beta\in\alpha} \Phi^{\beta}(\top)$$

for some ordinal α. The least such ordinal is called the closure ordinal for $\Phi(P)$. *Note that $\mu P(\bar{x}).\Phi(P)$ is the least formula $\Psi(\bar{x})$ such that*

$$\Psi(\bar{x}) \Leftrightarrow \Phi([P/\Psi(\bar{x})])$$

THEOREM 3.1.14 (THE ELIMINATION THEOREM)
If Φ and Ψ are positive w.r.t. P then the closure ordinal for $\Phi(P)$ is less than or equal to ω, and

$$\begin{gathered}\exists P[(\forall \bar{y}\ P(\bar{y}) \Rightarrow \Phi(P)) \wedge \Psi(P)] \\ \Leftrightarrow \\ \Psi[P/\nu P(\bar{y}).\Phi(P)]\end{gathered}$$

where the above substitutions exchange the variables bound by fixpoint operators by the corresponding actual variables of the substituted predicate.

Proof: (Outline; the reader interested in more details is referred to (Nonnengart and Szałas 1995)).

That the closure ordinal for $\Phi(P)$ is $\leq \omega$ can be seen by showing that Φ is continuous w.r.t. P, i.e. Φ distributes over infinite disjunctions for P.

If $\Psi[P/\nu P(\bar{y}).\Phi(P)]$ holds then it is easy to check that $\exists P[(\forall \bar{y}\ P(\bar{y}) \Rightarrow \Phi(P)) \wedge \Psi(P)]$ by letting $P(\bar{y}) = \mu P(\bar{y}).\Phi(P)$.

Finally, let P such that $\forall \bar{y}\ (P(\bar{y}) \Rightarrow \Phi(P)) \wedge \Psi(P)$. First we show by induction over i that $P(\bar{y}) \Rightarrow \Phi^{i}(\top)$.

Base case $i = 0$: holds trivially.

Induction step: suppose $P(\bar{y}) \Rightarrow \Phi^{i}(\top)$. Φ is positive w.r.t. P and thus also monotone w.r.t. P. Hence $\Phi(P(\bar{y})) \Rightarrow \Phi(\Phi^{i}(\top)) \Leftrightarrow \Phi^{i+1}(\top)$. Since (by assumption) $P(\bar{y}) \Rightarrow \Phi(P)$ we have that $P(\bar{y}) \Rightarrow \Phi^{i+1}(\top)$ and we are done with the induction proof.

Thus $P(\bar{y}) \Rightarrow \bigwedge_{i\in\omega} \Phi^{i}(\top)$ and since the closure ordinal has been shown to be $\leq \omega$ we also have $P \Rightarrow \nu P.\Phi(P)$. Now recall that Ψ is positive (and thus monotone) w.r.t. P and that by assumption $\Psi(P)$ holds. From these two facts it follows that $\Psi[P/\nu P(\bar{y}).\Phi(P)]$ and we are done.

Analogously, it is possible to show that

$$\begin{gathered}\exists P[(\forall \bar{y}\ P(\bar{y}) \vee \Phi(\overline{P})) \wedge \Psi(\overline{P})] \\ \Leftrightarrow \\ \Psi[\overline{P}/\nu \overline{P}(\bar{y}).\Phi(\overline{P})]\end{gathered}$$

i.e. the sign of the P is of no importance.

EXAMPLE 3.1.15
Let us apply this kind of second-order quantifier elimination to the example schema $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$ again. Evidently, just as for the application of the SCAN algorithm the (relational) translation result of this schema has to be negated first in order to be able to apply the Elimination Theorem. This then results in:

$$\exists u \exists \Phi \begin{bmatrix} \exists v\ R(u,v) \wedge \Phi(v) \\ \wedge \\ \exists w\ R(u,w) \wedge \forall y\ R(w,y) \Rightarrow \neg\Phi(y) \end{bmatrix}$$

The Elimination Theorem requires a certain form to be applied. A transformation into this form then leads to

$$\exists u, v \exists \Phi \begin{bmatrix} \forall z\ \Phi(z) \vee z \neq v \\ \wedge \\ R(u,v) \\ \wedge \\ \exists w\ R(u,w) \wedge \forall y\ R(w,y) \Rightarrow \neg\Phi(y) \end{bmatrix}$$

From the Elimination Theorem we know that this is equivalent to

$$\exists u, v \begin{bmatrix} R(u,v) \\ \wedge \\ \exists w\ R(u,w) \wedge \forall y\ R(w,y) \Rightarrow \nu\Phi.y \neq v \end{bmatrix}$$

which can easily be simplified (using e.g. $\nu\Phi.y \neq v \Leftrightarrow y \neq v$) to

$$\exists u, v\ R(u,v) \wedge \exists w\ R(u,w) \wedge \neg R(w,v)$$

A final negation then provides us with

$$\forall u, v, w\ R(u,v) \wedge R(u,w) \Rightarrow R(w,v)$$

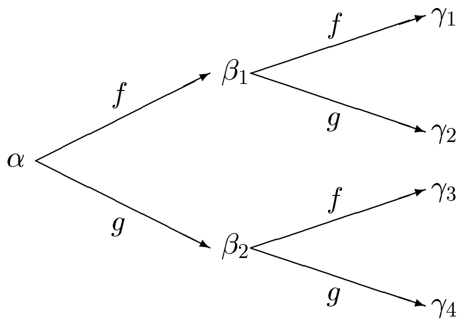
The advantage of this approach on second-order quantifier elimination over SCAN is that it always terminates provided the input formula can be transformed into the form required for this Elimination Theorem. However, this cannot always be guaranteed unless we apply a second-order skolemization in order to get this very form. This fact is actually not too surprising for there evidently exist second-order formulae which have no first-order equivalent (and may even not be representable by such fixpoint formulae).

Note that such a second–order quantifier elimination actually applies only to formulae which can be brought into the form required for the Elimination Theorem. Modal schemata are not of this form. Finding characteristic properties from modal schemata is therefore also a matter of the translation (in fact the modal logic semantics). However, it is known that not all modal logics are complete w.r.t the Kripke style semantics. It is therefore usually necessary to show the completeness of the axiomatization independently (see also (Sahlqvist 1975)).

3.2 Semi-Functional Translation

3.2.1 Background

In 1988 Hans Jürgen Ohlbach came up with the so called *functional translation* for modal logics which was supposed to eliminate most of the representational overhead that gets produced in the relational translation approach[9]. The main idea behind this functional translation method was based on the observation that the responsibility of an arbitrary binary relation $R \subseteq W \times W$ can be taken over by some suitable set of (partial) functions as it is illustrated in the following example



where each arrow represents an R-connection. In this example R consists of six pairs, however, as the example shows, we might equally assume two functions f and g (so-called *access functions*) together with the element α and would still be able to access any element from α by some nested function application. So for instance, $\gamma_2 = g(f(\alpha))$ and $\beta_2 = g(\alpha)$. Evidently, this requires at least as many functions as there are R-successors for an arbitrary element. Therefore, if there is an element which has infinitly many R-successors then also infinitly many such functions are needed. Intuitively, we might think of these functions as follows: Assume an arbitrary element a and consider all elements x such that $R(a, x)$. In general, R is not a function, i.e. a might have several R-successors. Now assume a function which maps a to the set of its R-successors and furthermore consider the set of functions which select elements out of such a set. These selector functions in comination with the mapping to the R-successors essentially form the set of access functions F_R that are examined in the functional translation approach.

As a matter of fact, it can easily be shown that under this construction (and provided the relation R is serial) the formula

$$\forall u, v\ R(u, v) \Leftrightarrow \exists f \in F_R\ f(u) = v$$

holds and therefore any occurrence of an R-literal can be substituted by an equation. Now consider the relational translation of a $\Box$-formula

$$\lceil \Box \Phi \rceil_u = \forall v\ R(u, v) \Rightarrow \lceil \Phi \rceil_v$$

If we replace the R-literal in this translation according to the above equivalence we get as an intermediate result[10].

$$\lceil \Box \Phi \rceil_u = \forall v\ (\exists f \in F_R\ f(u) = v \Rightarrow \lceil \Phi \rceil_v)$$

which is equivalent to

$$\forall v \forall f \in F_R\ (f(u) = v \Rightarrow \lceil \Phi \rceil_v)$$

[9]See (Ohlbach 1989), (Ohlbach 1988), (Ohlbach 1991), (Auffray and Enjalbert 1992), (Fariñas del Cerro and Herzig 1988).

[10]The elements of F_R do not necessarily have to be thought of as all the functions which map worlds to R-accessible worlds. We shall later see that a denumerable set suffices for our purposes. It is thus possible to introduce an "apply"-functions which accepts an element f of F_R and a world α and returns $f(\alpha)$. In Hans Jürgen Ohlbach's notation the apply is represented by a $\downarrow$ thus $f(\alpha)$ becomes $\downarrow (f, \alpha)$.

This can finally be simplified to

$$\forall f \in F_R \ \lceil \Phi \rceil_{f(u)}$$

Similarly, the translation of a $\Diamond$-formula then results in

$$\lceil \Diamond \Phi \rceil_u \Leftrightarrow \exists f \in F_R \ \lceil \Phi \rceil_{f(u)}$$

and it is obvious that no R-literals can occur in the translation result of an arbitrary modal formula.

However, R-literals were not only introduced by the translation of modal formulae; also the background theory of the logic under consideration usually says something about R (e.g. reflexivity and transitivity of R in case of *S4*). Unfortunately, substituting the R-literals in such a background theory does not lead to something that nice. For instance, reflexivity gets replaced by

$$\forall u \exists f \ f(u) = u$$

and transitivity becomes

$$\forall u, f, g \ \exists h \ f(g(u)) = h(u)$$

It is fairly evident that any pure R-formula gets substituted by an equational formula which cannot be simplified any further and therefore the price to be paid for the much nicer functional translation is in the equational clauses to be added. Nevertheless, often some simplifications are possible. For example, it can be shown that the equation which represents the reflexivity of the R-relation can be rewritten as

$$\exists f \ \forall u \ f(u) = u$$

Also, if the background theory happens to be a Horn theory (as it is in case of *S4*) then the background theory after functional translation consists of unit equations. This then allows us to exchange the equational theory by some suitable theory unification algorithm. Up to date theory unification algorithms for some of the most common modal logics have been developed.

Summarizing, the functional translation approach reduces the number and size of the translation result significantly when compared with the relational translation. The price to be paid for this reduction lies in the equational theory which has to be added and unless this theory can be transformed into a suitable theory unification algorithm it turns out that such an equational theory is hard to deal with for standard predicate logic theorem provers.

Thus both approaches, the relational and the functional translation method have advantages and disadvantages. The relational translation results in huge clause sets and fairly simple – at least easy to understand – background theories and the functional translation results in much smaller clauses but rather complicated equational background theories.

Now, the idea of the semi-functional translation is to combine the advantages and to avoid the disadvantages of the two other approaches if possible. As it will turn out the size of the clause set after semi-functional translation will be just as big as in case of the functional approach (the clause might get bigger though) and the resulting background theory does not contain any equations (provided the corresponding theory in the relational translation does not). As we shall see later it is even possible to find certain restrictions on the background theory after semi-functional translation which allow us to simplify this theory so considerably that it sometimes even can be described by a single unit clause.

In what follows the semi-functional translation approach is described. It is called *semi-functional* because the operator $\Diamond$ is translated functionally whereas the $\Box$ is translated as in the relational case. Just as in the functional translation case its definition depends on the question whether we consider serial or non-serial accessibility relations. Since we are mainly interested in serial relations for the application to temporal logics, this case is considered first. The non-serial case is briefly examined in Section 3.4 (for some additional information see (Nonnengart 1993) and (Nonnengart 1992)).

3.3 Serial Modal Logics

There are two possible cases to be distinguished. Either the modal logic under consideration is serial or it is not. Evidently, if the logic we are interested in is *not* serial then there exists at least one world, say χ, which has no access to any other world. Since the functions we are looking for are supposed to map worlds to accessible worlds none of these functions can be defined on χ and thus cannot be total. In serial modal logics we may restrict our attention to total functions as will be seen shortly.

DEFINITION 3.3.1 (FUNCTIONAL DECOMPOSITION)
Let $\Re$ be an arbitrary serial binary relation over a denumerable set $\mathcal{W}$. A denumerable set $F_\Re$ of total functions from $\mathcal{W}$ to $\mathcal{W}$ is called a functional decomposition *for $\Re$ if $\Re(\alpha, \beta)$ is tantamount to the existence of an element f in the set $F_\Re$ with $f(\alpha) = \beta$.*

As mentioned earlier, such a functional decomposition is supposed to take over some of the responsibilities of the accessibility relation. We therefore extend the notion of frames and interpretations accordingly.

DEFINITION 3.3.2 (EXTENDED FRAMES)
Let $\mathcal{F}_{\mathrm{ML}} = (\mathcal{W}, \Re)$ be an arbitrary modal logic frame and let $F_\Re$ be a functional decomposition of $\Re$. Then we call the triple $(\mathcal{W}, \Re, F_\Re)$ a functional extension *of $\mathcal{F}_{\mathrm{ML}}$.*

DEFINITION 3.3.3 (SEMI-FUNCTIONAL INTERPRETATIONS)
Let $\Im_{\mathrm{ML}} = (\mathcal{D}, \mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ be a modal logic interpretation and let $(\mathcal{W}, \Re, F_\Re)$ be a functional extension of $\mathcal{F}_{\mathrm{ML}}$. Then we call $(\mathcal{D}, (\mathcal{W}, \Re, F_\Re), \Im_{\mathrm{loc}}, \tau, \phi)$ a semi-functional modal logic interpretation generated from *$\Im_{\mathrm{ML}}$.*

Since interpretations have been changed slightly, the definition of the satisfiablity relation has to be changed as well.

DEFINITION 3.3.4 (SATISFIABILITY RELATION)
Compared to Definition 3.1.2 the satisfiability changes only with respect to $\Diamond$-formula, namely

$$\Im_{\mathrm{EML}} \models \Diamond\Phi \text{ iff } \Im_{\mathrm{EML}}[f(\tau)] \models \Phi$$

where $\Im_{\mathrm{EML}} = (\mathcal{D}, (\mathcal{W}, \Re, F_\Re), \Im_{\mathrm{loc}}, \tau, \phi)$ and f is some element of $F_\Re$. All other cases remain as before.

It has to be shown that there is actually no real difference between a relational Kripke-interpretation and a corresponding semi-functional interpretation as it is defined above. To this end it is necessary to guarantee the existence of a functional decomposition for arbitrary serial accessibility relations and also to show that Kripke-models behave just like extended models do. The latter can be shown as follows:

LEMMA 3.3.5

Let $\Im_{\mathrm{ML}}$ be an arbitrary serial modal logic (possible world) interpretation and let Φ be an arbitrary modal logic formula. Moreover, let $\Im_{\mathrm{EML}}$ be an extended interpretation generated from $\Im_{\mathrm{ML}}$. Then

$$\Im_{\mathrm{ML}} \models \Phi \quad \textit{iff} \quad \Im_{\mathrm{EML}} \models \Phi$$

Proof: By induction on the recursive structure of Φ. Since the two kinds of models differ only in the interpretation of $\Diamond$-formulae this is the only case to be considered here.

Consider $\Phi = \Diamond\Psi$: Since $\Im_{\mathrm{ML}}$ satisfies Φ we have that – assuming the initial world is τ – $\Im_{\mathrm{ML}}[\chi]$ satisfies Ψ for some world χ with $\Re(\tau, \chi)$. Since $\Im_{\mathrm{EML}}$ is an extended interpretation generated from $\Im_{\mathrm{ML}}$ we have that the set $F_{\Re}$ is a functional decomposition of $\Re$, thus $\Re(\alpha, \beta)$ holds iff there is an f in $F_{\Re}$ such that $f(\alpha) = \beta$ for any α and β. Therefore – and by the induction hypothesis – it follows that $\Im_{\mathrm{EML}}[f(\tau)]$ satisfies Ψ for some f and thus $\Im_{\mathrm{EML}}$ is a model for $\Diamond\Psi$.
For the other direction assume that $\Im_{\mathrm{EML}} \models \Diamond\Psi$. Acording to Definition 3.3.4 this is equivalent to $\Im_{\mathrm{EML}}[f(\tau)] \models \Psi$ for some $f \in F_{\Re}$. By the definition of a functional decomposition we have that $\Re(\tau, f(\tau))$ and thus we know that for some world χ, namely $f(\tau)$, it holds that $\Re(\tau, \chi)$ and $\Im_{\mathrm{EML}}[\chi] \models \Psi$. By the induction hypothesis this is equivalent to $\Im_{\mathrm{ML}}[\chi] \models \Psi$ for some χ with $\Re(\tau, \chi)$ which is tantamount to $\Im_{\mathrm{ML}} \models \Diamond\Psi$.

There are several ways how to show the existence of a functional decomposition for a given accessibility relation. A non-constructive, but nevertheless fairly elegant way works as follows: Consider the set of all functions f from $\mathcal{W}$ to $\mathcal{W}$ such that $f(\alpha) = \beta$ implies $\Re(\alpha, \beta)$. This set of functions obviously is a functional decomposition of $\Re$ and, according to the famous Löwenheim-Skolem Theorem, there even exists a countable subset of such functions which obey the desired properties.

However, there is also a constructive proof:

LEMMA 3.3.6

Let $\Re$ be an arbitrary serial relation over $\mathcal{W} \times \mathcal{W}$. If $\mathcal{W}$ is a countable set then there exists a countable functional decomposition $F_{\Re}$ of $\Re$.

Proof: It will be shown that $\Re$ can be split into countably many disjoint total and functional subrelations. To this end we arrange the pairs in $\Re$ in a two-dimensional field in such a way that the pairs in a single column all have an identical first element and there are no two columns which contain pairs with identical first element. Pairs may occur arbitrary often in one column, though. Then the subrelations we were looking for can be found by simply collecting the elements of the respective rows. Formally:

Define for any pair (x, y) in $\mathcal{W} \times \mathcal{W}$: $(x, y) \approx (u, v)$ iff $x = u$.
Obviously $\approx$ denotes an equivalence relation. It is thus possible to introduce equivalence classes $[\,]/_{\approx}$ by:

$$[(x, y)]/_{\approx} = \{(u, v) \in \Re \mid (x, y) \approx (u, v)\}$$

and

$$\Re/_{\approx} = \{[(x, y)]/_{\approx} \mid (x, y) \in \Re\}$$

Both $[(x, y)]/_{\approx}$ and $\Re/_{\approx}$ are denumerable, therefore there exist surjective mappings θ : $\mathrm{Nat} \to \Re/_{\approx}$ and $\delta_i : \mathrm{Nat} \to \theta(i)$.
Then define $f_j = \{\delta_k(j) \mid k \in \mathrm{Nat}\}$ and the functional decomposition $F_{\Re}$ of $\Re$ is then simply given by the set $\{f_j \mid j \in \mathrm{Nat}\}$.

The relational translation method was based on the relational (Kripke-style) semantics for modal logics. Analogously, the semi-functional translation approach is based on the extended (semi-functional) semantics as it is described above. We therefore have to define a suitable signature-, interpretation-, and formula translation just as it had been done for the relational translation approach.

DEFINITION 3.3.7 (THE SIGNATURE TRANSLATION $\lfloor\Sigma_{\mathrm{ML}}\rfloor$)
$\lfloor\Sigma_{\mathrm{ML}}\rfloor$ differs from the relational signature translation $\lceil\Sigma_{\mathrm{ML}}\rceil$ only in the additional binary function symbol ":" (written in infix notation).

Note that ":" corresponds to the "apply"-function in the functional translation approach.

DEFINITION 3.3.8 (THE INTERPRETATION TRANSLATION $\lfloor\Im_{\mathrm{ML}}\rfloor$)
The only difference to the relational case lies in the interpretation of the additional binary function symbol : which accepts a world and an element of the functional decomposition of the accessibility relation and returns a new world such that $\lfloor\Im_{\mathrm{ML}}\rfloor(:)(\alpha,\beta) = \gamma$ if and only if $\beta(\alpha) = \gamma$.

DEFINITION 3.3.9 (THE FORMULA TRANSLATION $\lfloor\Phi\rfloor_u$)
The only difference to the relational translation lies in the translation of the $\Diamond$-operator.

$$\lfloor\Diamond\Phi\rfloor_u = \exists x \in F_R\ \lfloor\Phi\rfloor_{u:x}$$

In the sequel we shall sometimes omit the sort informations if they are clear from the context.

At this stage all preliminaries are provided to prove the main theorem of this section. It has to be shown that this translation preserves both satisfiablity and unsatisfiablity.

THEOREM 3.3.10
Let Φ be a modal logic formula in negation normal form[11]. Then Φ is unsatisfiable (in the serial modal logic ML) iff $\lfloor\Phi\rfloor_\iota$ cannot be satisfied by any classical model satisfying both `Axioms` *and $\forall u, x\ R(u, u\!:\!x)$[12] where* `Axioms` *represent the properties induced by the additional axiom schemata for the logic ML.*

Proof: From Theorem 3.1.11 we know that the relational translation is both sound and complete, i.e. a formula Φ is ML-satisfiable iff there exists a classical interpretation $(\mathcal{M}, \phi)$ which is a model for $\forall x\ \lceil\Phi\rceil_x$ and which interprets the accessibility predicate R just as the accessibility relation $\Re$. Moreover, we know from Lemma 3.3.6 that for every serial relation R there exists a functional decomposition F_R. Thus we do not lose or gain anything if we simply add the formula[13]

$$\forall u, v\ R(u, v) \Leftrightarrow \exists x\ u\!:\!x = v$$

[11] A formula is in negation normal form if it contains no implication or equivalence and all negations are moved inwards as far as possible (i.e. directly in front of the atoms). Evidently any formula can easily be transformed into an equivalent formula in negation normal form.

[12] This unit clause describes the important connection between functional decompositions and accessibility relations. Essentially it guarantees that whatever we can get by the application of an element of the functional decomposition is accessible.

[13] Note that this formula is just the same as the one which had to be introduced in the functional translation approach.

Now we can show — in analogy to the proof for Theorem 3.1.11 — that Φ is ML-unsatisfiable if and only if $\lfloor\Phi\rfloor_\iota$ has classical model satisfying both `Axioms` and $\forall u, v\ R(u,v) \Leftrightarrow \exists x\ u\!:\!x = v$. To this end it suffices to show that the claim holds for $\Diamond$-formulae, other formulae are treated identically as in Theorem 3.1.11.

Hence consider $\Phi = \Diamond\Psi$ and let $\Im_{\mathrm{ML}}$ be a model for Φ, i.e. $\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Phi$. What has to be shown is that — under the additional equivalence $\forall u, v\ R(u,v) \Leftrightarrow \exists x\ u\!:\!x = v$ — $\forall u\ (\exists v\ R(u,v) \wedge \Phi[v] \Leftrightarrow \exists x\ \Phi[u\!:\!x])$. This, however, is in fact trivial.

Thus what we have shown up to now is that Φ is ML-unsatisfiable iff $\lfloor\Phi\rfloor_\iota$ cannot be satisfied by any classical model satisfying both `Axioms` and $\forall u, v\ R(u,v) \Leftrightarrow \exists x\ u\!:\!x = v$. Remains to be shown that the left-to-right implication of the latter equivalence is superfluous.

First consider the two implications one by one. The right-to-left implication states that for any u and v: if there exists an x such that $u\!:\!x = v$ then $R(u,v)$ holds, which is equivalent to $\forall u, x\ R(u, u\!:\!x)$. The other implication can be viewed as a conditioned equation and if we would add its clause form to an arbitrary clause set this would result in:

$$\neg R(u,v) \vee u\!:\!f(u,v) = v$$

Note that the variables on the left-hand-side of this equation form a proper superset of the variables on its right-hand-side. Thus it is possible to view this equation as directed from left to right. Now the negation normal comes into play. The idea is to apply this equation[14] to every non-variable subterm which occurs somewhere in the clause set. Recall that the only thing we know about the other clauses is that they have been generated by the semi-functional translation method. What is remarkable about this translation is that it produces clauses which do not contain any functional decomposition variables provided the formula to be translated is in negation normal form[15]. Thus the conditioned equation cannot be applied to any of these clauses, neither can it be applied to any of the `Axioms`. The only possibility at all is to apply it to the unit $R(u, u\!:\!x)$ but this results in a tautology. Hence this conditioned equation needs not be applied to any clause in the clause set and therefore it is superfluous.

Thus the semi-functional translation behaves as desired. A first easily observable advantage of it is that it produces much less clauses than the relational translation.

Definition 3.3.11 (Flattened Form of Modal Formulae)
The flattened form *of a modal logic formula can be obtained by simply ignoring the modal operators, i.e.*

$$\begin{array}{lcl} \mathit{flat}(P(\ldots)) & = & P(\ldots) \\ \mathit{flat}(\Box\Phi) & = & \mathit{flat}(\Phi) \\ \mathit{flat}(\Diamond\Phi) & = & \mathit{flat}(\Phi) \\ \mathit{flat}(\Phi \vee \Psi) & = & \mathit{flat}(\Phi) \vee \mathit{flat}(\Psi) \\ \multicolumn{3}{l}{\textit{and similarly for the other connectives}} \end{array}$$

The flattened form is sometimes also called the skeleton *of the formula.*

[14] Evidently, by *applying an equation* a usual paramodulation step is meant.
[15] The trivial induction on the formula structure is omitted here.

LEMMA 3.3.12
Let Φ be a modal logic formula in negation normal form and let Ψ be the skeleton (flattened form) of Φ. Then the clause normal form of $\lfloor\Phi\rfloor_w$ consists of exactly as many clauses as the clause normal form of Ψ.

Proof: By induction on the structure of Φ.

The case were Φ is a simple atom is trivial. Also if Φ is a negative literal or a conjunction there are no problems. Thus the only cases to be considered are those were Φ is a disjunction or a formula with a modal operator as top symbol.

If $\Phi = \Diamond\Phi'$ then $\lfloor\Phi\rfloor_w = \exists x\ \lfloor\Phi'\rfloor_{w:x}$. By induction hypothesis we have that the clause normal form (cnf for short) of $\lfloor\Phi'\rfloor_{w:x}$ contains exactly as many clauses as the cnf of the skeleton of Φ' and we are done.

If $\Phi = \Box\Phi'$ then $\lfloor\Phi\rfloor_w = \forall v\ \neg R(w,v) \vee \lfloor\Phi'\rfloor_v$ whose cnf contains as many clauses as the cnf of $\lfloor\Phi'\rfloor_v$ and we are again done by the induction hypothesis.

Finally, if $\Phi = \Phi' \vee \Phi''$ and the cnf of $\lfloor\Phi'\rfloor_w$ contains n clauses and the cnf of $\lfloor\Phi''\rfloor_w$ contains m clauses then $\lfloor\Phi\rfloor_w$ contains $n \times m$ clauses which is just as many as (by induction hypothesis) the number of clauses as in the skeleton of Φ.

What has been gained so far? In fact, two of the main goals mentioned at the begining of this chapter are already fulfilled. The first one was to get a more compact translation result by reducing the number of clauses that are generated and the second one was the separation of the background theory from the translated theorem. However, there is a third invariant of this translation approach which turns out to be useful:

LEMMA 3.3.13
Let Φ be a modal logic formula in negation normal form. Then the clause normal form of $\lfloor\Phi\rfloor u$ does not contain any positive R-literal.

Proof: Simple induction on the structure of Φ.

Thus positive R-literals can occur only in the background theory of the modal logic under consideration. We shall take advantage of this observation later.

3.3.1 The Semi-Functional Translation From Another Perspective

The semi-functional translation as it has been defined above has its origin in the functional translation approach for modal logics. In fact, the term "functional" might be a bit misleading. It is not really necessary to think of functions as the elements of the newly introduced sort since typical function properties as for instance the fact that two functions are equal if they agree in all their arguments are not needed. Rather we should think of a more or less arbitrary new sort and a new symbol ":" (which is to be interpreted as a function) and examine such a new translation in a more abstracted manner.

To this end consider an arbitrary translation of the modal operators. For simplicity let us assume that either this translation is already dual in the two modal operators or, otherwise, one of the operators, say the $\Box$, is translated as in the relational translation approach[16]. Let us then

[16]We assume here that the modal logic axiomatization is complete w.r.t. the relational semantics.

consider the *general translation schema*:

$$\begin{array}{rcl} \lfloor \Box\Phi \rfloor_u & = & \forall w\ R(u,w) \Rightarrow \lfloor \Phi \rfloor_w \\ \lfloor \Diamond\Phi \rfloor_u & = & \Omega(\Phi, u, \lfloor\ \rfloor) \end{array}$$

i.e. the translation result on a formula $\Diamond\Phi$ is a function, say Ω, which depends on Φ, u and also on the translation function $\lfloor\ \rfloor$. This Ω may introduce some new functions, predicates and sorts which we call the Ω-symbols[17]. At this stage nothing very strange has happened yet for we could still prove the same theorems as before provided we treat every $\Diamond$ as a $\neg\Box\neg$. However, if we do not want to translate the $\Diamond$ in terms of the $\Box$ we have to realize that these two operators are not anymore duals to each other as it is generally required for modal logics. In order to overcome this problem we have to consider only interpretations in which the schema $\Box\Phi \Leftrightarrow \neg\Diamond\neg\Phi$ is valid. Evidently, this schema evaluates to $\top$ in the relational translation approach and therefore requires no special treatment there. Similarly, the functional translation approach treats the two operators in a dual manner and this schema is again not necessary. This is not true anymore for the semi-functional translation method and since we do not want to deal with such schemata directly we have to find out whether $\Box\Phi \Leftrightarrow \neg\Diamond\neg\Phi$ corresponds to a first-order property of the function, sort and relation symbols involved, i.e. we try to eliminate the Φ from the translation of this axiom schema. Now let us assume that such a second-order elimination is successful and results in a formula Ψ. Evidently, Ψ is a first-order formula on R, u, and the Ω-symbols then and we are forced to add Ψ to the formula we get after applying the new translation to any alledged modal logic theorem for otherwise we could not even guarantee that $\neg\Diamond P \Rightarrow \Box\neg P$ is provable. However, we are not finished yet. Ψ tells us about a correspondence between R and the Ω-symbols. and such a correspondence in fact might induce certain properties on R itself.

Example 3.3.14
As a simple example consider

$$\Omega(\Phi, u, \lceil\ \rceil) = \forall w\ S(u,w) \Rightarrow \lceil \Phi \rceil_w$$

In this case the only Ω-symbol is S and the second-order quantifier elimination of $\Box\Phi \Leftrightarrow \neg\Diamond\neg\Phi$ results in

$$\begin{array}{l} \forall u \exists v\ S(u,v) \wedge R(u,v) \\ \forall u,v,w\ R(u,v) \wedge S(u,w) \Rightarrow v = w \end{array}$$

These two clauses, our Ψ, will have to be added for they guarantee the duality of $\Box$ and $\Diamond$. Now, if it were the case that for any R there is an S with such properties, in other words the formula $\exists S\ \Psi$ is equivalent to $\top$, everything would be fine and we had an alternative translation for arbitrary modal logics. Unfortunately, but not too surprising, $\exists S\ \Psi$ is *not* equivalent to $\top$. Once again with the help of the second-order quantifier elimination we can find out that

$$\exists S\ \Psi \Leftrightarrow \forall u \exists v\ R(u,v) \wedge \forall w\ R(u,w) \Rightarrow v = w$$

i.e. such an S exists if and only if R denotes a total function. Hence this translation is sound and complete for every modal logic where R is functional, i.e. the axiom schema $\Box\Phi \Leftrightarrow \Diamond\Phi$ holds[18].

[17] In case of the relational translation we have that $\Omega(\Phi, u, \lceil\ \rceil) = \exists w\ R(u,w) \wedge \lceil \Phi \rceil_w$ and no new symbols are introduced and in case of the (semi-)functional translation $\Omega(\Phi, u, \lfloor\ \rfloor) = \exists x \in F_R\ \lfloor \Phi \rfloor_{u:x}$ and the Ω-symbols consist of the new sort symbol F_R and the function symbol ":".

[18] In fact, S is then even identical to R.

The general idea can thus be summarized as follows: Given an alternative translation (for $\Diamond$-formulae in our case) try to find a first-order equivalent for the duality schema $\Box\Phi \Leftrightarrow \neg\Diamond\neg\Phi$. If this succeeds we get a formula Ψ on R, "=", and the Ω-symbols and what remains is to find a first-order equivalent for

$$\exists\Omega\text{-symbols } \Psi$$

This way we end up with a first-order formula on R which states the exact condition on the accessibility relation under which the alternative translation is both sound and complete (provided Ψ is added to the formula to be proved).

Now let us apply this idea to the (semi-)functional translation approach. Here we have

$$\Omega(\Phi, u, \lfloor\ \rfloor) = \exists x \in F_R\ \lfloor\Phi\rfloor_{u:x}$$

and the first step according to the above observation is to find a first-order equivalent for the duality schema. As a matter of fact this results in the two clauses (our Ψ)

$$\begin{array}{l} \forall u, x \in F_R\ R(u, u:x) \\ \forall u, v\ R(u,v) \Rightarrow \exists x \in F_R\ u:x = v \end{array}$$

Remains to be shown under which conditions on R there exist such a sort F_R and a function ":" such that Ψ holds. This is actually another second-order quantification elimination problem but we can avoid the application of the elimination procedure here for we already know the result by Lemma 3.3.6, i.e. such a new sort and such a new function exist if and only if R is serial.

Note that this result shows that in case of non-serial modal logics a different translation has to be chosen. We shall defer this issue until Section 3.4.

3.3.2 (Partial) Saturation of Background Theories

According to Lemma 3.3.13 no positive R-literals do occur in the translation result whatsoever the input modal formula looks like. This means that the only positive R-literals that might at all occur in the clause set appear in the background theory that is induced by the modal logic under consideration. This fact can be utilized by computing everything beforehand that can possibly be derived from this background theory, i.e. this theory gets *saturated.* Such a saturation characterizes the modal logic and is thus independent from the theorem to be proved.

NOTATION 3.3.15
We call a clause C P-positive (P-negative) if there is a positive (negative) occurrence of a P-literal in C. If, in addition, C is not P-negative (not P-positive) then we call C pure-P-positive (pure-P-negative).

DEFINITION 3.3.16 ((PARTIAL) SATURATION)
Let P be a designated predicate symbol and let $\mathcal{C}$ be a set of P-positive clauses. Then we call the set of clauses we get by resolution within $\mathcal{C}$ and whose elements are pure-P-positive, i.e. the set

$$\{C \mid \mathcal{C} \vdash_{\text{res}} C \textit{ and } C \textit{ is pure-}P\textit{-positive}\}$$

the saturation *of $\mathcal{C}$ with respect to P.*

As a little example consider the clause set $\{P(a), \neg P(x) \vee P(f(x))\}$. Its only clause which is pure-positive w.r.t. P is $P(a)$. However, it is possible to derive more pure-P-positive clauses by resolution, namely all unit clauses in the set $\{P(f^n(a)) \mid n \geq 0\}$.

Knowing about the saturation of a given clause set is often quite helpful as the following lemma states.

Lemma 3.3.17
Let $\mathcal{C}$ be a clause set and let P be a designated predicate symbol. Moreover let $\mathcal{C}' \subseteq \mathcal{C}$ be exactly the subset of $\mathcal{C}$ whose elements are positive w.r.t. P. If $\mathcal{C}''$ is the saturation of $\mathcal{C}'$ with respect to P then $\mathcal{C}$ is unsatisfiable iff $\mathcal{C} \setminus \mathcal{C}' \cup \mathcal{C}''$ is unsatisfiable.

Proof: Easy by Definition 3.3.16.

The problem with the above lemma is that saturations are usually infinite. However, if we are able to find a finite alternative clause set with exactly the same saturation we can use this one instead.

Theorem 3.3.18
Let $\mathcal{C}$ be a finite clause set, let P be a designated predicate symbol, and let $\mathcal{D} \subseteq \mathcal{C}$ contain exactly the P-positive clauses of $\mathcal{C}$. Moreover, let $\mathcal{B}$ be a finite set of P-positive clauses whose saturation w.r.t. P is identical to $\mathcal{D}$'s saturation w.r.t. P. Then $\mathcal{C}$ is unsatisfiable iff $(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{B}$ is unsatisfiable.

Proof: Follows immediately from the lemma above.

Thus the idea is to extract $\mathcal{D}$ and to find a hopefully simpler clause set $\mathcal{B}$ with the same saturation.

Example 3.3.19
Consider the simple background theory given by the clauses:

$$\begin{array}{l} P(a) \\ P(f(a)) \\ \neg P(x), \neg P(f(x)), P(f(f(x))) \end{array}$$

Its saturation is $\{P(f^n(a)) \mid n \geq 0\}$ and a fairly easy way to prove this is as follows. First it has to be shown that each of these elements can indeed be derived. A simple induction over n will do. For the base case we have to check whether $P(a)$ is derivable from the original clause set. Since $P(a)$ is even contained in this clause set we are already done. For the induction step assume that it has been shown that $P(f^k(a))$ is derivable for all $k < n$. We thus know that both $P(f^{n-2}(a))$ and $P(f^{n-1}(a))$ are derivable and therefore $P(f^n(a))$ can be obtained by two resolution steps with the third clause from the original clause set. Thus $\{P(f^n(a)) \mid n \geq 0\}$ is at least contained in the saturation we are looking for. Remains to be shown that the converse also holds, i.e. that the saturation is contained in the derived clause set. To this end it suffices to show that any pure-P-positive clause which is derivable from $\{P(f^n(a)) \mid n \geq 0\}$ and the P-positive clauses of the clause set under consideration is already of the form $P(f^n(a))$. Evidently, resolution steps between $P(f^k(a))$, $P(f^l(a))$ and $\neg P(x), \neg P(f(x)), P(f(f(x)))$ are possible only if $k = l+1$ (or $l = k+1$) and they result in $P(f^{l+2}(a))$ ($P(f^{k+2}(a))$ respectively). This derived unit belongs to $\{P(f^n(a)) \mid n \geq 0\}$ and we are done.

Now consider the somewhat "simpler" clause set

$$\begin{array}{l} P(a) \\ \neg P(x), P(f(x)) \end{array}$$

Similarly to the above we can show that the saturation of this clause set is also $\{P(f^n(a)) \mid n \geq 0\}$ and therefore (according to the above Theorem 3.3.18) this new clause set may be used to replace the original background theory.

Recall that these two clause sets are not at all equivalent. It is the mere fact that they do form a background theory in the sense that they contain the only P-positive literals occurring anywhere in the whole clause set under consideration which allows us to perform such a "simplification". Hence what we utilize here is not only that the background theory is *something* we know about P but is indeed *all* we know about P.

Some Simple Modal Logic Examples For Partial Saturation

There are two of the best known serial modal logics where this saturation approach does not lead to anything new, namely the modal logics *KD* and *KT* (also often simply called *T*). However, this does not bother us too much for these modal logics background theories are represented by one or two unit clauses anyway and thus the amount of search necessary for a theorem prover to prove a theorem in such a logic is certainly not too heavily influenced by these theories[19].

Now let us consider another fairly simple, although not at all trivial, modal logic, namely *KDB* which is axiomatized by the additional axiom schemata $\Box\Phi \Rightarrow \Diamond\Phi$ and $\Phi \Rightarrow \Box\Diamond\Phi$. As we know from the beginning of this chapter (and of course also from the standard modal logic literature) these two axiom schemata characterize the seriality and the symmetry of the underlying accessibility relation. The background theory for *KDB* therefore is:

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u, v) \Rightarrow R(v, u) \end{array}$$

According to the principle described in the previous section it is our aim to saturate this background theory[20]. This saturation can be found very easily and we end up with

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u\!:\!x, u) \end{array}$$

Hence these two unit clauses are sufficient as the background theory for *KDB* and, although this seems to be of minor effect, such a replacement at least avoids unwanted cycles in the search space.

As another, slightly more complicated, example consider the modal logic *S4* which is characterized by the accessibility relation properties reflexivity and transitivity[21]. The whole back-

[19]Note that for the logic *KD* we could incorporate this very unit clause directly in the translation description. Interestingly this would result in exactly the same clause set we would get if we applied the functional translation approach.

[20]Recall that under *KDB* these two clauses indeed describe *all* we know about the background theory and not just *something* we know of it.

[21]The corresponding axiom schemata are $\Box\Phi \Rightarrow \Phi$ and $\Box\Phi \Rightarrow \Box\Box\Phi$.

ground theory is therefore given by

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u, u) \\ R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \end{array}$$

Again we have to saturate this clause set bearing in mind that this is indeed all we know about R, for any formula to be proved unsatisfiable (in *S4*) won't contain R-positive clauses. Let us show now that the saturation consists of the (infinite) set of unit clauses of the form $\{R(u, u\!:\!x_1\!:\cdots:\!x_n) \mid n \geq 0\}$. To this end we first have to show that the purely positive R-clauses in the background theory (which are $R(u, u\!:\!x)$ and $R(u, u)$) are contained in this set, however, this turns out to be trivial. Then we have to show that the application[22] of two arbitrary elements of the alleged saturation to the transitivity clause does not produce anything new, and indeed, resolving $R(u, u\!:\!x_1\!:\cdots:\!x_n)$ and $R(u, u\!:\!y_1\!:\cdots:\!y_m)$ with the first two literals in $R(u, v) \wedge R(v, w) \Rightarrow R(u, w)$ results in $R(u, u\!:\!x_1\!:\cdots:\!x_n\!:\!y_1\!:\cdots:\!y_m)$ which is already contained in $\{R(u, u\!:\!x_1\!:\cdots:\!x_n) \mid n \geq 0\}$. Finally, we have to show that the supposed saturation is not too big, i.e. that each of its elements can in fact be derived and this follows by a simple induction over n.

So far we have found the saturation of the *S4* background theory. Now we have to find an alternative clause set with exactly the same saturation but which is in some sense simpler than the original one. Finding such an alternative clause set is still to be performed by a *good guess*; it is not yet known how this could be automatized in general and in fact there are some reasonable doubts that a "complete" automatization can be defined. For this example, however, it is not very hard to find a suitable clause set, namely

$$\begin{array}{l} R(u, u) \\ R(u, v) \Rightarrow R(u, v\!:\!x) \end{array}$$

or, equally simple,

$$\begin{array}{l} R(u, u) \\ R(u\!:\!x, v) \Rightarrow R(u, v) \end{array}$$

It is easy to show – similarly to the above – that the saturation of this clause set is identical to the saturation of the *S4* background theory and what we have gained is that we may replace the background theory for *S4* by the two simpler clauses

$$\begin{array}{l} R(u, u) \\ R(u, v) \Rightarrow R(u, v\!:\!x) \end{array}$$

In particular the vanishing of the transitivity clause turns out to be of major importance here. At this stage it might be helpful to illustrate the effect of the semi-functional translation together with saturation with the help of a little example.

EXAMPLE 3.3.20
Recall Example 3.1.12 on page 25 where the *S4*-validity of $\Diamond\Box Q \Leftrightarrow \Diamond\Box\Diamond\Box Q$ was to be shown.

[22] Sometimes I will use the term *applying a clause* as short for *performing a resolution step with this clause as one of the parent clauses.*

Semi-functional translation then results in the following clause set:

$$\begin{array}{l} \neg R(\iota : a, x) \vee \neg R(\iota : b, y) \vee \neg R(y : f(y), z) \vee Q(x) \vee Q(z) \\ \neg R(\iota, x) \vee \neg Q(x : g(x)) \vee \neg R(\iota, y) \vee \neg R(y : h(y), z) \vee \neg Q(z : k(z, y)) \\ R(u, u) \\ \neg R(u, v) \vee R(u, v : x) \end{array}$$

This set of clauses is not only much smaller than what we got from the relational translation approach; also the search space has been reduced so considerably that no standard theorem prover for classical predicate logic will have particular difficulties with it.

It should now be obvious how a logic like *KD4* has to be treated. Its background theory is described by

$$\begin{array}{l} R(u, u : x) \\ R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \end{array}$$

and it can very easily be shown that the saturation of this theory consists of the set

$$\{R(u, u : x_1 : \cdots : x_n) \mid n \geq 1, x_i \in F_R\}$$

i.e. it differs from the saturation of the *S4*-theory only in one respect: it lacks the reflexivity clause $R(u, u)$. However, it is just as easy to find an alternative clause set for this background theory, as there is for example

$$\begin{array}{l} R(u, u : x) \\ R(u, v) \Rightarrow R(u, v : x) \end{array}$$

To show that this clause set indeed has the same saturation can be proved in exactly the same way as it had been done in case of *S4*.

3.3.3 Exploiting Connectedness

So far only a few of the best known modal logics have been examined with respect to semi-functional translation and partial saturation (of background theories). Nevertheless, these techniques can equally well be applied to other well known modal logics, as e.g. *S5*, and they help to simplify the background theory for actually every modal logic commonly known from the literature (provided the accessibility relation properties are first-order).

But we can do even better, and that with the help of the so-called connectedness property of modal frames. Before coming to this let us first have a look at the technique applied to the logic *S5*.

As is well known, *S5* can be axiomatized by the schemata $\Box\Phi \Rightarrow \Phi$, $\Box\Phi \Rightarrow \Box\Box\Phi$, and $\Phi \Rightarrow \Box\Diamond\Phi$ or, equivalently, by $\Box\Phi \Rightarrow \Phi$ and $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$. The correspondence axioms for the accessibility relation are therefore reflexivity, transitivity and symmetry in the former and reflexivity and euclideanness in the latter axiomatization. Both are evidently equivalent and hence it does not make any difference whether we choose the one or the other. Now, the saturation of either background theory consists of all the unit clauses of the form $\{R(u : x_1 : \cdots x_n, u : y_1 : \cdots y_m) \mid n, m \geq 0\}$ as can easily be found out by applying the procedure described in the last section. Also, it is not very hard to find an alternative clause set which is in some sense simpler than the original one but gnerates the very same saturation, for instance

$$\begin{array}{l} R(u,u) \\ R(u,v) \Rightarrow R(u,v\!:\!x) \\ R(u,v) \Rightarrow R(u\!:\!x,v) \end{array}$$

which is still rather complicated.

Nevertheless, this background theory can be much more simplified bearing in mind that we may consider *connected* frames as defined in (Segerberg 1971):

Definition 3.3.21 (Connected Frames)
A frame $\mathcal{F}_{\mathrm{ML}} = (\mathcal{W}, \Re)$ is said to be connected *if there exists a world τ in $\mathcal{W}$ such that for every world χ in $\mathcal{W}$ we have that $\Re^\star(\tau,\chi)$ where $\Re^\star$ denotes the reflexive and transitive closure of $\Re$.*

Hence, in a connected frame any world can be reached from an initial world by zero or more R-steps and it is thus impossible to have two or more unconnected "islands" of worlds.

Definition 3.3.22 (Generated Frames (Segerberg))
Given an arbitrary frame $\mathcal{F}_{\mathrm{ML}} = (\mathcal{W}, \Re)$ and an arbitrary world τ in $\mathcal{W}$ we define $\mathcal{W}' = \{\chi \in \mathcal{W} \mid \Re^\star(\tau,\chi)\}$ and $\Re' = \{(\chi,\xi) \in \Re \mid \chi,\xi \in \mathcal{W}'\}$. The frame $(\mathcal{W}', \Re')$ is then called the frame generated from $\mathcal{F}_{\mathrm{ML}}$ *(with initial world τ).*

Evidently, every generated frame is connected. What is more remarkable, however, is that modal logics are not able to distinguish between connected and unconnected frames and this is shown by the following

Lemma 3.3.23 (Segerberg)
Let $\mathcal{F}_{\mathrm{ML}}$ be a modal logic frame and let $\Im_{\mathrm{ML}} = (\mathcal{D}, \mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ be an arbitrary interpretation based on $\mathcal{F}_{\mathrm{ML}}$. Moreover, let Φ be an arbitrary modal logic formula and let $\Im'_{\mathrm{ML}} = (\mathcal{D}, \mathcal{F}'_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ where $\mathcal{F}'_{\mathrm{ML}}$ is the frame generated from $\mathcal{F}_{\mathrm{ML}}$ (with initial world τ). Then

$$\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \Phi \text{ iff } \Im'_{\mathrm{ML}} \models_{\mathrm{ML}} \Phi$$

Proof: Can be found in (Segerberg 1971) and (Chellas 1980).

Hence it is possible to restrict our attention to connected frames and we have to find out what connectedness actually means in extended (semi-functional) frames.

Lemma 3.3.24
An extended frame (interpretation) is connected (with initial world τ) iff for every world χ there exist some $\gamma_1, \ldots, \gamma_n \in F_R$ ($n \geq 0$) such that

$$\chi = \tau\!:\!\gamma_1\!:\!\gamma_2\!:\!\gamma_3\!:\cdots:\!\gamma_n$$

Proof: In connected frames each world χ can be reached from the initial world τ by a finite sequence of R-transitions, i.e. there exist worlds $w_1, \ldots, w_{n-1}$ such that

$$R(\tau, w_1) \wedge R(w_1, w_2) \wedge \ldots \wedge R(w_{n-1}, \chi)$$

which – in the extended functional frames – is just $\tau\!:\!\gamma_1\!:\!\gamma_2\!:\!\gamma_3\!:\cdots:\!\gamma_n$ for appropriate γ_i.

Thus we may assume that (in the object level syntax)

$$\forall x \exists y_1, \ldots, y_n \; x = \iota : y_1 : \cdots : y_n$$

holds and this can be utilized as follows.

Recall that the saturation of *S5* resulted in an infinite clause set which consists of all unit clauses of the form $R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m)$ with $n, m \geq 0$ (and universally quantified variables u, x_i, y_j). Now consider the subset we get after instantiating the variable u with ι, the initial world. Then both arguments of the R-literals are of the form $\iota : z_1 : \cdots : z_n$ where every z_i is universally quantified and therefore this term can represent any world. Thus we have that (under the connectedness assumption) $R(\iota : x_1 : \cdots : x_n, \iota : y_1 : \cdots : y_m)$ which can be simplified to $R(v, w)$, i.e. the universal relation, which obviously subsumes any of the unit clauses described by $R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m)$. More formally:

$$\begin{gathered} \forall x \exists y_1, \ldots, y_n \; x = \iota : y_1 : \cdots : y_n \\ \Rightarrow \\ \forall u, x_i, y_j \; R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m) \Leftrightarrow \forall v, w \; R(v, w) \end{gathered}$$

Thus, instead of considering the still rather complicated background theory for *S5* as it is described above, we can simplify (in fact generalize) this theory to the universal relation.

Now, let us have a look at two slightly different modal logics, *KD45* and *KD5*. These are axiomatized by the clause set

$$\begin{array}{ll} R(u, u : x) & \\ R(u, v) \wedge R(v, w) \Rightarrow R(u, w) & \quad (KD45) \\ R(u, v) \wedge R(u, w) \Rightarrow R(v, w) & \end{array}$$

and

$$\begin{array}{ll} R(u, u : x) & \\ R(u, v) \wedge R(u, w) \Rightarrow R(v, w) & \quad (KD5) \end{array}$$

respectively, i.e. seriality and euclideanness (*KD5*) and, additionally, transitivity (*KD45*).

A few simple inductions show that the respective saturations then consist of the unit clause sets with all elements of the form $R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m)$ where $m \geq 1$ and $n \geq 0$ for *KD45* and $n \geq 1$ for *KD5* (in addition, the saturation of *KD5* also contains $R(u, u : x)$). Both are therefore quite closely related to *S5*. Unfortunately, since $m \geq 1$ (and $n \geq 1$ for *KD5*) these two arguments are not yet in the form we would like to have them in order to be able to apply our connectedness assumption. However, since $m \geq 1$ we know that $m - 1 \geq 0$ and we therefore get for all $m \geq 1$ and $n \geq 0$ — thus for *KD45* —

$$\begin{gathered} \forall x \exists y_1, \ldots, y_k \; x = \iota : y_1 : \cdots : y_k \\ \Rightarrow \\ \forall u, x_i, y_j \; R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m) \Leftrightarrow \forall v, w, x \; R(v, w : x) \end{gathered}$$

and for all $m \geq 1$ and $n \geq 1$ — thus for *KD5* —

$$\begin{gathered} \forall x \exists y_1, \ldots, y_k \; x = \iota : y_1 : \cdots : y_k \\ \Rightarrow \\ \forall u, x_i, y_j \; R(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m) \Leftrightarrow \forall v, w, x, y \; R(v : x, w : y) \end{gathered}$$

Logic	Background Theory
KD	$R(u, u:x)$
KT	$R(u, u:x)$ $R(u, u)$
KDB	$R(u, u:x)$ $R(u:x, u)$
KD4	$R(u, u:x)$ $R(u, v) \Rightarrow R(u, v:x)$
S4	$R(u, u)$ $R(u, v) \Rightarrow R(u, v:x)$
KD5	$R(\iota, \iota:x)$ $R(u:x, v:y)$
KD45	$R(u, v:x)$
S5	$R(u, v)$

Table 3.3: Simplified background theories

The unit clause $R(v, w:x)$ we got for *KD45* in fact subsumes the whole saturation and can therefore be used as the background theory for the logic *KD45* whereas the unit clause $R(v:x, w:y)$ subsumes almost the whole saturation for *KD5*; the only clause which is not subsumed is $R(u, u:x)$ and therefore the background theory for *KD5* can be described by the two unit clauses $R(u, u:x)$ and $R(v:x, w:y)$[23].

Table 3.3 summarizes the results obtained so far concerning the background theories for some of the best known serial modal logics provided the semi-functional translation is applied (note that all variables are assumed to be universally quantified) It should be remarked at this stage how these results, in particular for the logics *S5*, *KD45*, and *KD5* are to be interpreted since, at least at the first glance, their simplicity and generality might be surprising. As Krister Segerberg found out already in the early '70s by examining the model theory for various modal logics (Segerberg 1971), the characteristic frames for *S5* are in fact so-called *clusters* by which he meant sets of worlds such that each world has access to any other world including itself, hence it is allowed to assume that the accessibility relation for *S5* is the universal relation over $\mathcal{W} \times \mathcal{W}$. The models for *KD45* are (according to his investigations) characterized by either a single cluster as for *S5* or a single world together with a cluster such that this particular world has access to each element of the cluster. The characteristic frames for *KD5* are almost exactly as those for *KD45*; the only difference is that the single world does not necessarily have access to *all* elements in the cluster.

If we compare these Segerberg results with the saturated background theory we got for these

[23] Actually this can be even further simplified, for this first unit clause is subsumed for every instance but one, namely $R(\iota, \iota:x)$. Hence the two unit clauses $R(\iota, \iota:x)$ and $R(v:x, w:y)$ would already suffice in case of *KD5*.

logics then we can see that for *S5* we indeed got the universal accessibility relation and that for *KD45* indeed every world has access to every other world which itself is somehow accessed and that for *KD5* indeed any two worlds which have predecessors have access to each other.

It seems remarkable how the Segerberg results which have been found by model theoretic examinations are mirrored in the saturation approach and that with mere syntactic means.

3.4 Non-Serial Modal Logics

The semi-functional translation as described in the previous section works not only for the serial case but also for non-serial accessibility relations, however it has to be changed slightly then. And although non-serial modal logics are not of main interest in the examination of temporal logics, it might nevertheless be interesting here to see how the approach works in non-serial frames, be it just for completeness reasons.

The first problem we are faced with is that in the non-serial case there is no functional decomposition for the accessibility relation as defined in Definition 3.3.1 on page 32. At least we cannot guarantee such a suitable set of *total* functions and in classical predicate logic all functions are usually assumed to be total. We can overcome this problem by arbitrarily extending these functions so that they really become total, but also taking care for which worlds this has happened. Recall that for each world which has $\Re$-successors the generated functions in the corresponding functional decomposition are indeed defined. These worlds are called *normal worlds* then and it can be shown that functional decompositions defined in this slightly different way obey the property that a world α can access a world β if and only if α is *normal* and there exists an element in the functional decomposition of the accessibility relation which maps the α to the β.

Thus the semi-functional translation then has to be changed to ($N(w)$ denotes the *normality*-predicate):

$$\lfloor \Diamond \Phi \rfloor_w = N(w) \land \exists x \; \lfloor \Phi \rfloor_{w:x}$$

i.e. it differs from the semi-functional translation for serial modal logics only in the additional *normality*-literal $N(w)$.

However, the basic background theory also changes. In the serial case we always have at least the unit clause $R(u, u\!:\!x)$ in the background theory. Now, in the non-serial case, this gets slightly more complicated, namely the conjunction of $N(u) \Rightarrow R(u, u\!:\!x)$ and $R(u, v) \Rightarrow N(u)$.

Now, what effect does this have in the application of the saturation method? The method itself evidently does not have to be changed, it has been described for general first-order theorem proving. But obviously, since the background theory has changed for the non-serial case, the saturation result will also change. Let us have a look at an example, say the modal logic *KB* whose background theory (after semi-functional translation) gets

$$\begin{array}{l} \neg N(u) \lor R(u, u\!:\!x) \\ \neg R(u, v) \lor R(v, u) \\ \neg R(u, v) \lor N(u) \end{array}$$

where the final clause needs not be considered for the moment since it does not contain any positive R-literals.

Now, saturating the first two clauses is very easy and results in

$$\neg N(u) \vee R(u, u\!:\!x)$$
$$\neg N(u) \vee R(u\!:\!x, u)$$

and together with $\neg R(u, v) \vee N(u)$ we would actually be done. Nevertheless, after realizing that the translation of any formula produces clauses without any negative N-literals, we see a chance for some further simplification. First note that the N-literal in the additional clause $\neg R(u, v) \vee N(u)$ can only be resolved within the background theory. We therefore do perform these possible resolution steps beforehand and end up with the theory

$$\neg N(u) \vee R(u, u\!:\!x)$$
$$\neg N(u) \vee R(u\!:\!x, u)$$
$$\neg N(u) \vee N(u\!:\!x)$$

This third clause is quite interesting, because of self-resolution possibilities which result in $\neg N(u) \vee N(u\!:\!x_1\!:\cdots:\!x_n)$ and the special case with u instantiated to ι this (after applying the connectedness assumption again) leads to $\neg N(\iota) \vee N(v\!:\!x)$. This clause states that if the initial world is normal then every world which is somehow accessible is normal as well. On the other hand, if the initial world is not normal then no world can be normal (again by the connectedness assumption). Altogether this means that either *each* world is normal or *no* world is normal and this can be represented by $\neg N(u) \vee N(v)$.

Thus reasoning in *KB* can be performed as follows: First try to prove the theorem within *KDB* and if this succeeds prove the theorem under the assumption that no world is normal, i.e. ignore all those clauses which contain N-literals. Both proofs can evidently be performed much more easily than a single proof with the full *KB* background theory.

Again, after a little thought, it is not very surprising that under *KB* either each world or no world is normal, since the symmetry axiom already ensures that every world which has a predecessor also must have a successor and since it can be assumed by the connectedness assumption that the initial world is the only world which possibly has no predecessor then either every world has a successor or every world in $\mathcal{W}$ (which then consists only of the initial world) has no successor.

This very technique can equally be applied to other non-serial modal logics as well, as for instance *K45* or *K5* with a similar result. As an example consider *K5* with its background theory

$$N(u) \Rightarrow R(u, u\!:\!x)$$
$$R(u, v) \Rightarrow N(u)$$
$$R(u, v) \wedge R(u, w) \Rightarrow R(v, w)$$

During the saturation process one derives $N(u) \Rightarrow R(u\!:\!x, u\!:\!y)$ and this, together with $R(u, v) \Rightarrow N(u)$ leads again to $N(u) \Rightarrow N(u\!:\!x)$ which, as we recall, guarantees that either every or no world is normal. Therefore reasoning within *K5* and *K45* can be performed by first proving the theorem within *KD5* (*KD45* respectively) and then, under the assumption that the initial world is the only world at all, prove it again with a significantly reduced clause set. All in all this shows that reasoning within these non-serial modal logics is scarcely harder than reasoning within their serial version.

One slight exception to this rule can be found in the logic *K4*. Here it cannot be derived that either every world or no world is normal. We are therefore forced to perform the whole saturation for *K4* and can hardly hope for similar simplifications, although the simplifications we are about to obtain are still considerable. Recall the *K4* background theory:

$$\begin{array}{l} N(u) \Rightarrow R(u, u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R(u,v) \Rightarrow N(u) \end{array}$$

First of all note that the third of these clauses is actually superfluous and that because of the following reason. In the construction of a functional decomposition for a non-serial accessibility relation we artificially augmented the domain of the respective functions such that they became total. Those worlds for which such an augmentation occurred were considered as *not normal* and the effect of this operation had to be taken into account in the modal logic background theory by adding

$$\forall u, v\ R(u,v) \Leftrightarrow N(u) \wedge \exists x\ u\!:\!x = v$$

instead of the simpler

$$\forall u, v\ R(u,v) \Leftrightarrow \exists x\ u\!:\!x = v$$

which we had for serial modal logics. However, this equivalence for non-serial accessibility relations is actually a bit overloaded. As a matter of fact it suffices if the normality predicate occurs only in one of the two directions. Hence

$$\begin{array}{l} R(u,v) \Rightarrow \exists x\ u\!:\!x = v \\ N(u) \wedge \exists x\ u\!:\!x = v \Rightarrow R(u,v) \end{array}$$

does the same job bearing in mind that the only critical case in the construction of a suitable functional decomposition is that the mere existence of a function which maps α to β does not yet guarantee that $\Re(\alpha, \beta)$ holds. It also has to be ensured that the α is normal and thus the functions have not been augmented for α. For the other direction it is not really necessary to guarantee the normality of α if we know already that α has a successor in the $\Re$-relation. Nevertheless, adding the normality of α is by no means wrong; it is just not necessary but still might be helpful as we saw in the earlier examples. For the background theory for *K4* it turns out to be of no help and hence is ignored.

Performing the saturation process for the other two clauses results in all clauses of the form

$$N(u) \wedge N(u\!:\!x_1) \wedge \ldots \wedge N(u\!:\!x_1\!:\cdots:\!x_{n-1}) \Rightarrow R(u, u\!:\!x_1\!:\cdots:\!x_n)$$

with $n \geq 1$. Finding a suitable alternative clause set which generates exactly the same saturation can now easily be found and we end up with

$$\begin{array}{l} N(u) \Rightarrow R(u, u\!:\!x) \\ N(v) \wedge R(u,v) \Rightarrow R(u, v\!:\!x) \end{array}$$

which still is significantly simpler than the original background theory from above.

3.5 Other Miscellaneous Modal Logics

The examples presented so far had in common that they were represented by sets of Horn clauses. However, this approach is by no means restricted to Horn clauses, it works equally well for non-Horn sets, although finding the saturation and a suitable alternative background theory usually turns out to be more difficult then.

As a little example consider the modal logic *S4F* which is axiomatized by the usual axioms for *S4* together with the axiom schema

$$\Phi \wedge \Diamond\Box\Psi \Rightarrow \Box(\Diamond\Phi \vee \Psi)$$

In order to determine the first-order property induced by this extra axiom we have to apply our second-order quantifier elimination theorem, i.e. we have to solve the problem:

$$\exists u \exists \Phi, \Psi \ \lceil \Phi \wedge \Diamond\Box\Psi \wedge \Diamond(\Box\neg\Phi \wedge \neg\Psi)\rceil_u$$

After translation and some simple transformations we get

$$\exists u, v \ \exists \Phi, \Psi \left[\begin{array}{c} \forall w \ \Psi(w) \vee \neg R(v, w) \\ \wedge \\ \Phi(u) \wedge R(u, v) \\ \wedge \\ \exists v' \ R(u, v') \wedge \neg\Psi(v') \wedge \forall w \ R(v', w) \Rightarrow \neg\Phi(w) \end{array} \right]$$

Eliminating Ψ thus results in

$$\exists u, v \ \exists \Phi \left[\begin{array}{c} \Phi(u) \wedge R(u, v) \\ \wedge \\ \exists v' \ R(u, v') \wedge \neg R(v, v') \wedge \forall w \ R(v', w) \Rightarrow \neg\Phi(w) \end{array} \right]$$

A further transformation into the required for the Elimination Theorem leads to

$$\exists u, v, x \ \exists \Phi \left[\begin{array}{c} \forall w \ \neg\Phi(w) \vee \neg R(x, w) \\ \wedge \\ \Phi(u) \wedge R(u, v) \wedge R(u, x) \wedge \neg R(v, x) \end{array} \right]$$

and after eliminating Φ we and up with

$$\exists u, v, x \ R(u, x) \wedge R(u, v) \wedge \neg R(v, x) \wedge \neg R(x, u)$$

Hence the original axiom schema is equivalent to

$$\forall u, v, w \ R(u, v) \wedge R(u, w) \Rightarrow R(w, u) \vee R(v, w)$$

which is a non-Horn formula. As a full background theory for *S4F* we thus get

$$\begin{array}{l} R(u, u) \\ R(u, u\!:\!x) \\ \neg R(u, v), \neg R(v, w), R(u, w) \\ \neg R(u, v), \neg R(u, w), R(w, u), R(v, w) \end{array}$$

which indeed looks fairly complicated.

Now, an easy induction shows that the saturation for this clause set is given by

$$R(u, v\!:\!x_1\!: \cdots :\!x_n), R(v, w)$$

with $n \geq 0$[24].

This saturation looks suffciently simple to find a suitable alternative clause set with exactly the same result. And indeed such a possibility is e.g. the theory given by

$$\begin{array}{l} R(u,v) \Rightarrow R(u,v\!:\!x) \\ R(u,v) \vee R(v,w) \end{array}$$

Although this clause set is undoubtly much more convenient than the original background theory it is still possible to compute a lot of redundant clauses, as for example any clause of the form

$$R(u,v) \vee R(v,w\!:\!x_1\!:\cdots:\!x_m)$$

which is subsumed by $R(u,v) \vee R(v,w)$. A simple possibility how such redundancies can be avoided is to introduce new auxiliary predicates as follows

$$\begin{array}{l} S(u,u) \\ S(u,v) \Rightarrow S(u,v\!:\!x) \\ S(u,v) \Rightarrow R(x,v) \vee R(u,y) \end{array}$$

where the predicate symbol S acts as an auxiliary predicate which is new to the whole clause set, i.e. it must not occur in the theorem to be proved. Evidently, the saturation w.r.t S results in $S(u,u\!:\!x_1\!:\cdots:\!x_n)$ with $n \geq 0$ and therefore the only derivable pure-R-positive clauses are exactly those from the saturation of the original clause set.

The usage of such auxiliary predicates often turns out to be very useful. As a final example consider the modal logic *S4.2* which is axiomatized by the *S4*-Axioms $\Box\Phi \Rightarrow \Phi$ and $\Box\Phi \Rightarrow \Box\Box\Phi$ together with the so called *directedness* or *confluence* property given by $\Diamond\Box\Phi \Rightarrow \Box\Diamond\Phi$. Whereas the first two axioms characterize the reflexive and transitive frames, the third axiom defines the following property:

$$\forall x,y,z\; R(x,y) \wedge R(x,z) \Rightarrow \exists w\; R(y,w) \wedge R(z,w)$$

These three formulae are again pretty complicated as a background theory and we therefore try some simplification with the help of the saturation technique (together with the connectedness assumption and possibly auxiliary predicates).

First observe that from this axiomatization it can be proved that the axiom schema

$$\Diamond\Box(\Phi \Rightarrow \Psi) \Rightarrow (\Diamond\Box\Phi \Rightarrow \Diamond\Box\Psi)$$

is valid. Also it can easily be seen that the inference rule

$$\frac{\Phi}{\Diamond\Box\Phi}$$

is valid as well.

Now define a new modal operator, say $\blacksquare$, by

$$\blacksquare\Phi \Leftrightarrow \Diamond\Box\Phi$$

[24]The reader might wonder why the saturation of such a non-Horn clause set does not produce arbitrary long clauses. And indeed, usually this would happen unless the length of clauses is reduced by simple replacement factorization steps. This example is just of that kind.

There is nothing unusual with this new operator, however, we want to deal with it in just the same way as we treat the other operators, i.e. we want to be able to translate formulae containing this $\blacksquare$ into first-order logic and that with the help of a suitable accessibility relation. To this end we have to guarantee that this new operator obeys both the K-Axiom $\blacksquare(\Phi \Rightarrow \Psi) \Rightarrow (\blacksquare\Phi \Rightarrow \blacksquare\Psi)$ and the necessitation rule

$$\frac{\Phi}{\blacksquare\Phi}$$

and indeed by the observations from above these two preliminaries are fulfilled. Now consider again the directedness property for *S4.2* which is given by $\Diamond\Box\Phi \Rightarrow \Box\Diamond\Phi$. Under the definition of $\blacksquare$ this is equivalent to $\blacksquare\Phi \Rightarrow \neg\blacksquare\neg\Phi$ which in fact characterizes the seriality of the new accessibility relation. Thus instead of the directedness property we simply include the seriality of the new accessibility relation, say S. The price to be paid for this is that we have to find (and to add) the first-order property induced by the definition of the $\blacksquare$-operator. Fortunately, this is not too difficult. As a matter of fact it results in the two clauses[25] (considering semi-functional translation of course):

$$\begin{array}{l} \forall u \exists x \forall w\; R(u\!:\!x, w) \Rightarrow S(u,w) \\ \forall u,v,w\; R(u,v) \wedge S(u,w) \Rightarrow R(v,w) \end{array}$$

We therefore end up with the following first-order axiomatization of the background theory for *S4.2* under semi-functional translation:

$$\begin{array}{l} R(u, u\!:\!x_R) \\ R(u,u) \\ \neg R(u,v), \neg R(v,w), R(u,w) \\ S(u, u\!:\!x_S) \\ \neg R(u\!:\!f_R(u), w), S(u,w) \\ \neg R(u,v), \neg S(u,w), R(v,w) \end{array}$$

where the respective index denotes the functional decomposition the respective symbol is supposed to belong to. This clause set seems to be even more complicated than the one we originally had. Nevertheless, it has a big advantage over the former: It can more easily be saturated. And in performing this saturation we finally end up with (again under the connectedness assumption):

$$\begin{array}{ll} R(u, u\!:\!x_1\!:\cdots:\!x_n) & n \geq 0 \\ R(u, v\!:\!f(v)\!:\!y_1\!:\cdots:\!y_m) & m \geq 0 \end{array}$$

This looks much better[26], for we can very easily imagine a clause set which produces exactly the same saturation, namely

$$\begin{array}{l} R(u,u) \\ \neg R(u,v), R(u, v\!:\!x) \\ R(u, v\!:\!f(v)) \end{array}$$

Note that the first two of these clauses in fact form the background theory for *S4*. Thus in order to get the theory for *S4.2* it is sufficient to add the simple unit clause $R(u, v\!:\!f(v))$ to

[25] This result can be obtained by another application of the Elimination Theorem. The proof is simple and is therefore omitted here.

[26] The watchful reader might have noticed that the saturation should actually contain the clause schema $R(u, v\!:\!x_S\!:\!z_1\!:\cdots:\!z_k)$ as well, however, this one is not really necessary for every formula to be translated will not contain any $\blacksquare$-subformula.

the theory for *S4*. At the first glance, this result is quite surprising for the skolem function f depends on v only and not on both u and v. Frames with such properties thus consist of an *S4* structure together with a non-degenerated cluster (Segerberg 1971) such that each world of the *S4* structure has access to this very cluster. However, as Segerberg already claimed, *S4.2* frames are characterized by non-degenerated clusters with a final (last) cluster and therefore our result from above is in fact identical to what Segerberg had found out by applying certain model-theoretic filtration techniques.

3.6 On Definability and Axiomatizability

Several modal logics have been examined in the earlier sections of this chapter. Originally these modal logics were given by Hilbert axiomatizations (see e.g. (Lewis 1912a), (Lewis 1913), (Lewis 1912b), (Lewis 1914)). After Saul Kripke defined the so-called possible world semantics (cf. (Kripke 1963)) the area of modal logic correspondence theory was born (see (van Benthem 1990) and (van Benthem 1984a)). Here it was the aim to examine how certain axiom schemata correspond to properties of the underlying accessibility relation, i.e. to find out which properties can be axiomatized with the help of axiom schemata and which axioms do have first-order correspondences. Some axioms and their corresponding properties have already been mentioned in previous chapters of this work. So, for instance, we considered properties like reflexivity, transitivity, and euclideanness which descend from the axiom schemata $\Box\Phi \Rightarrow \Phi$, $\Box\Phi \Rightarrow \Box\Box\Phi$, and $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$ respectively.

For many applications, however, one might have certain accessibility properties in mind and wonder whether these can be at all axiomatized. So, for instance, one might be interested in *irreflexivity* rather than reflexivity and is desperately searching for a suitable axiom. Unfortunately, the poor fellow can't be very successful in his attempt and it is quite instructive to see – with the help of the semi-functional translation approach – why this is so.

Recall that under semi-functional translation the only positive R-literals do occur in the background theory. Hence, if the translated formula contains R-literals at all (and it usually will) then these are negative. Now, the irreflexivity property is described by a single unit clause, namely $\neg R(u,u)$ and the only way how this unit clause can act as a resolution partner is by unifying it with a positive R-literal which has to be contained in the background theory. However, the simplest background theory – the one for *KD* (or *K* if we also consider non-serial modal logics) – consists of only one such clause which is $R(u, u\!:\!x)$ ($N(u) \Rightarrow R(u, u\!:\!x)$ respectively) and it is obvious that no resolution step is possible between this theory clause and the irreflexivity axiom which is thus shown to be superfluous. Since the irreflexivity axiom cannot provide anything new, it cannot be axiomatized, for otherwise, assume there were a schema Φ which axiomatizes irreflexivity. Then it would be possible to refute the negation of Φ after adding the unit clause $\neg R(u,u)$. But since this unit clause has no resolution partner this refutation could have been performed already without this very unit clause and therefore irreflexivity would follow from *KD* or *K* alone which it obviously does not.

Now let us have a look at *asymmetry*, i.e. the accessibility relation property $\forall x, y\ R(x,y) \Rightarrow \neg R(y,x)$. This property, written in clause normal form, consists entirely of negative literals and, in spite of the fact that certain resolution steps are possible, it is evident that the corresponding resolvents cannot be further processed, they are dead-ends. It thus makes no difference whether we add asymmetry or not; modal logics are not able to distinguish between arbitrary frames and asymmetric frames and hence there is no axiom schema for this property.

As another example for an accesssibility relation property which cannot be axiomatized in the modal logic language consider *weak discreteness.* We call a frame *weak discrete* if every world has a direct successor according to the accessibility relation, formally

$$\forall x \exists y\ R(x,y) \wedge \neg\exists z\ R(x,z) \wedge R(z,y)$$

In the semi-functional approach this can slightly be simplified to

$$\forall x \exists y \forall z\ \neg R(x,z) \vee \neg R(z,x\!:\!y)$$

and the transformation into clause normal form then results in $\neg R(x,z), \neg R(z,x\!:f(x))$. This time there are two possible resolution steps with the background theory clause $R(u,u\!:\!x)$ but, evidently, the two resolvents thus obtained are both dead-ends and therefore weak discreteness is also not axiomatizable.

What these examples have in common is that the accessibility relation properties which fail to be axiomatizable are pure-R-negative (i.e. their respective negation normal form consists only of some negative but no positive R-literals). This is by no means a coincidence as the lemma below demonstrates.

Lemma 3.6.1
A pure-R-negative accessibility relation property[27] *which is at all consistent with* $R(u,u\!:\!x)$ *cannot be axiomatized.*

Proof: Along the lines of the examples from above. There is no resolution step possible between the property clauses and the clauses stemming from the theorem to be proved. However, any resolvent which can be derived with the help of the background theory is again a negative clause and can never lead to a refutation unless the background theory itself together with the negative property are contradictory but this has been ruled out.

The above preliminary that the accessibility relation property must be consistent with the background theory is absolutely necessary, for there are in fact pure-R-negative properties which can be axiomatized, witness $\forall x,y\ \neg R(x,y)$ which is axiomatized by $\Box(\Phi \wedge \neg\Phi)$.

Intuitively, one may view the above result as follows: Axiom schemata say something about the *existence* rather than the *non-existence* of accessibilities. So, for instance, the axiom $\Box\Phi \Rightarrow \Phi$ in some sense "generates" arrows from worlds to themselves and the schema $\Phi \Rightarrow \Box\Diamond\Phi$ "generates" the additional reversal for each arrow. Negative properties, however, exclude certain accessibilities and are therefore – up to some minor pathological exceptions – not axiomatizable.

Such pure-R-negative properties are by no means the only ones which are not axiomatizable. Recall that by Segerberg's connectedness result we know that modal logics cannot distinguish between connected and unconnected frames. So, for instance, it is impossible to axiomatize the universal relation. Equivalence relations, however, are axiomatizable, and – as we found out with the saturation approach – this together with connectedness results in the universal relation.

Another property for which it is easy to see that it is not axiomatizable is connectedness itself, for, otherwise, there would exist formulae which are valid in connected frames but not in unconnected frames and Segerberg's result would not hold. Let us summarize this in the following lemma:

[27] Recall that a formula is pure-R-negative if its negation normal form contains no positive R-literal.

LEMMA 3.6.2
Let Ψ be an accessibility relation property for which another strictly weaker property Ψ' exists such that Ψ' together with the assumption of connectedness implies Ψ but Ψ' without the connectedness assumption does not imply Ψ. Then Ψ is not axiomatizable.

Proof: Suppose there were an axiom which characterizes the class of frames with property Ψ and consider an arbitrary instance of this schema. Since modal logics cannot distinguish between connected and unconnected frames it follows that this instance is provable from Ψ' already. Therefore (since the instance was arbitrary) the whole axiom schema is valid in every frame with property Ψ' and thus Ψ follows from Ψ'. This, however, contradicts the assumption that Ψ' alone does not imply Ψ and therefore Ψ cannot be axiomatizable.

This result can be even further generalized to:

LEMMA 3.6.3
Let $\mathcal{C}$ be a class of frames with property Φ, let Ψ be a property consistent with Φ and let Ψ' be strictly weaker than Ψ such that Ψ' together with connectedness implies Ψ (but Ψ' alone does not imply Ψ). Then Ψ is not axiomatizable.

As an example consider *S4*-frames and the additional property of *strong directedness*:

$$\forall x, y \exists z\ R(x, z) \wedge R(y, z)$$

It is sufficient – according to the above lemma – to find a strictly weaker property which implies strong directedness on *S4* frames under the connectedness assumption. Such a property is not very hard to find, namely *weak directedness* (see Section 3.5)

$$\forall u, x, y\ R(u, x) \wedge R(u, y) \Rightarrow \exists z\ R(x, z) \wedge R(y, z)$$

sometimes also called *diamond-property* or *confluence-property.* Since in connected frames all worlds are accessible from some initial world τ by the reflexive and transitive closure of the accessibility relation and since the accessibility relation for *S4* is already reflexive and transitive we have that $R(\tau, w)$ for all worlds w. This – together with weak directedness – immediately leads to strong directedness.

3.7 Varying Domains

In varying domains we do no more assume that there is a single domain common to all worlds. Rather we assume that each world might have its own universe of discourse. This slightly changes some of the former definitions. For instance, interpretations do not refer to a unique domain, but the respective domains are all given by the mapping $\Im_{\mathrm{loc}}$ which maps worlds to structures. This means that mentioning the domain is not anymore necessary in the interpretations; the respective local domains can be accessed via $\Im_{\mathrm{loc}}$. Also the satisfiability relation changes accordingly:

DEFINITION 3.7.1 (SATISFIABILITY IN VARYING DOMAINS)
Let $\Im_{\mathrm{ML}} = (\mathcal{F}_{\mathrm{ML}}, \Im_{\mathrm{loc}}, \tau, \phi)$ be a (varying domain) modal logic interpretation.

$$\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \forall x\ \Phi \quad \textit{iff} \quad \Im_{\mathrm{ML}}[x/a] \models_{\mathrm{ML}} \Phi \textit{ for any } a \in \textit{domain}(\Im_{\mathrm{loc}}(\tau))$$
$$\Im_{\mathrm{ML}} \models_{\mathrm{ML}} \exists x\ \Phi \quad \textit{iff} \quad \Im_{\mathrm{ML}}[x/a] \models_{\mathrm{ML}} \Phi \textit{ for some } a \in \textit{domain}(\Im_{\mathrm{loc}}(\tau))$$

The other cases remain as before.

This new definition obviously has an immediate effect on the relational translation of modal logic formulae. In addition to the new predicate symbol R we also have to add a symbol E which is supposed to represent the "existence" relation.

DEFINITION 3.7.2 (THE FORMULA TRANSLATION)

$$\begin{aligned} \lceil \forall x\ \Phi \rceil_u &= \forall x\ E(u,x) \Rightarrow \lceil \Phi \rceil_u \\ \lceil \exists x\ \Phi \rceil_u &= \exists x\ E(u,x) \wedge \lceil \Phi \rceil_u \end{aligned}$$

All other cases remain as before.

Note that by this definition it is easy to see that constant domains are a special case of varying domains since in constant domains the additional unit clause $E(u,x)$ which just states that any element occurs in every world simplifies the above to Definition 3.1.4.

We omit the soundness and completeness proofs of this formula translation since they do not provide anything particularly new[28]. Much more interesting is a closer look at the E-predicate and its occurrences inside translated formulae. As a matter of fact, any such occurrence is very similar to the occurrences of R-predicates inside translated formulae. Thus, the introduction of a functional decomposition for this "existence" predicate makes it possible to consider a significant simplification:

LEMMA 3.7.3
Let F_E be a functional decomposition of the binary relation E. Then each element of F_E denotes a total function and the relational translation can be modified to the (semi-) functional version:

$$\begin{aligned} \lfloor \forall x\ \Phi \rfloor_u &= \forall y\ \lfloor \Phi \rfloor_u[x/u\!:\!y] \\ \lfloor \exists x\ \Phi \rfloor_u &= \exists y\ \lfloor \Phi \rfloor_u[x/u\!:\!y] \end{aligned}$$

where y ranges over F_E and $\lfloor \Phi \rfloor_u[x/u\!:\!y]$ means that any occurrence of x inside $\lfloor \Phi \rfloor_u$ is to be replaced by $u\!:\!y$.

Proof: Totality of the functions is guaranteed by the fact that no domain is empty. Soundness and completeness of the translation is proved along the lines of the proofs in the constant domains case. Note that if the domains may vary arbitrarily then the background theory for the *existence*-predicate consists only of the unit clause $E(u, u\!:\!x)$. The translation given above just reflects that this unit clause can already be incorporated into the translation procedure.

This translation already takes into account that there is no further theory behind the "existence"-predicate. Very often, however, one is interested in certain extra properties as e.g. increasing domains or decreasing domains. These are examined in the following two sections.

[28]The only thing that should be mentioned here is that an appropriate interpretation translation which maps varying domain modal logic interpretations to classical interpretations evidently has to combine all local domains into a single domain and distinguishes the respective domains only with the help of the "existence" relation. This means that the interpretation of a certain predicate might hold for an element even though this element does not belong to the current domain. However, this is but a harmless property; it just means that predicates are in fact interpreted globally rather than locally.

3.7.1 Increasing Domains

The axiom schema which describes increasing domains is $\Box\forall x\Phi(x) \Rightarrow \forall x\Box\Phi(x)$[29] and its relational reading gets

$$\forall u, v, x\ E(u, x) \wedge R(u, v) \Rightarrow E(v, x)$$

i.e. if the domain element x exists in world u und v is accessible from u then x exists in v as well. W.l.o.g. we can assume that the accessibility relation is not symmetric for otherwise we would immediately be faced with a constant domain structure. Now recall that in arbitrary varying domains the only theory clause we had to include was the simple unit clause $\forall u, x\ E(u, u\!:\!x)$ and in fact, we were able to incorporate this simple clause by modifying the formula translation similar to the way we treated the modal logic *KD*.

For more complicated modal logics the additional theory axioms could not be simplified to a single unit clause and therefore a translation had to be chosen which also produces negative R-literals, but nevertheless can be used to simplify the theory axioms. Similar things happen here in case of increasing domains. First of all, the formula translation from above has to be changed to:

$$\begin{aligned} \lfloor \forall x\ \Phi \rfloor_u &= \forall x\ E(u, x) \Rightarrow \lfloor \Phi \rfloor_u \\ \lfloor \exists x\ \Phi \rfloor_u &= \exists y\ \lfloor \Phi \rfloor_u [x/u\!:\!y] \end{aligned}$$

i.e. $\lfloor\ \rfloor$ treats the quantifiers $\forall$ and $\exists$ similar to the modal operators. It is easy to see that the clause normal form of the translation of any modal logic formula which is in negation normal form contains no positive E-literal.

Thus we are in a position where we can try to simplify the theory axiom for increasing domains, again analogously to the simplification of the accessibility relation properties given earlier.

A first simplification is:

$$\forall u, x, y\ E(u, y) \Rightarrow E(u\!:\!x, y)$$

if R is serial or

$$\forall u, x, y\ N(u) \wedge E(u, y) \Rightarrow E(u\!:\!x, y)$$

if seriality of R is not assumed.

It can easily be checked that the additional assumption of R's transitivity does not lead to anything further. Also reflexivity has no effect that is not yet stated by the above simplification. Therefore this simplification is indeed characteristic for increasing domains if we consider the modal logics *K*, *KD*, *KT*, *K4*, *KD4*, and *S4*[30].

Euclideanness of the accessibility relation also doesn't lead to any particular problems. Note again, that euclidean frames consist either of a single non-degenerated cluster or a single world followed by such a non-degenerated cluster. Obviously – under the assumption that the domains are increasing – there are constant domains within clusters thus the only additional axioms to be added are those which state that any domain element occurs in any world which is different from the initial world ι, and this can simply be done by adding the unit clause $\forall u, x, y\ E(u\!:\!x, v\!:\!y)$ where x ranges over the functional decomposition of R and y over the functional decomposition

[29] Applying the Elimination Theorem to this schema is an easy exercise and is omitted here.

[30] Note that the simplified axiom also guarantees that domain elements exist in worlds which can be accessed by more than one R-step, regardless of whether R is transitive or not.

of E[31]. This way we may simplify the property $E(u, u\!:\!x)$ to $E(\iota, \iota\!:\!x)$ and indeed these two unit clauses are sufficient for both *KD5* and *KD45*.

Adding reflexivity or symmetry immediately leads to constant domains and the non-serial cases are captured by the slightly modified clause

$$\forall u, v, x, y\ N(u) \Rightarrow E(u\!:\!x, v\!:\!y)$$

i.e. the normality of u acts as a constraint.

3.7.2 Decreasing Domains

The characteristic axiom schema for decreasing domains is $\forall x \Box \Phi(x) \Rightarrow \Box \forall x \Phi(x)$ and the corresponding relational property is thus

$$\forall u, v\ \forall x\ E(u, x) \wedge R(v, u) \Rightarrow R(v, x)$$

Its similarity to the axiom schema for increasing domains suggests an analogous way of dealing with it. As can easily be found out the theory clauses to be added then are $E(u, u\!:\!y)$ and $E(u\!:\!x, z) \Rightarrow E(u, z)$ for *KD*, *KT*, *KD4*, and *S4* and $E(\iota, \iota\!:\!x)$ and $E(u, v\!:\!x\!:\!y)$ for *KD5* and *KD45* (where x ranges over F_E and y ranges over F_R respectively). It is evident that for the non-serial modal logics the respective N-literals also have to be taken into account.

3.8 Multi-Modal Logics

Most temporal logics can be viewed as multi-modal logics, i.e. as modal logics with several modal operators. It therefore makes sense to introduce the basic techniques that we are going to apply to temporal logics already in this chapter.

The principle behind multi-modalities is in fact very simple if the corresponding basics for single modalities are already understood. Evidently, since there are several modal operators under consideration there are also several accessibility relations to be introduced in the semantics definition[32]. Thus multi-modal frames consist of a set of worlds together with a set of accessibility relations, one for each modal $\Box$-operator.

Actually, in some sense we already became acquainted with such a multi-modal logic. Recall the examination of the modal logic *S4.2* for which we introduced a new modality $\blacksquare$ just for the purpose of being able to derive a simple saturation. We were not really interested in formulae which include this new operator, nevertheless its corresponding accessibility relation had to be taken into account during the saturation process.

Now, all the techniques developed for single modalities apply for multi-modalities as well, i.e. relational and semi-functional translations are possible, saturations can be performed and alternative background theories can be searched for.

As a simple "practical" example consider the epistemic modalities of belief such that for each of the agents involved in a certain setting there is a "belief"-operator. For instance, $\Box_x \Phi$

[31] This explanation is rather informal. The fact can be formally proved by saturating the background theory and applying the connectedness assumption. Since this does not significantly differ from the earlier examples this formal proof is omitted here.

[32] Obviously, since we are talking about a Kripke-style possible world semantics here, all modalities involved are implicitly supposed to obey the K-axiom and the necessitation rule.

then represents: "agent x believes Φ". Typically for such a scenario is the wish to be able to talk about a *mutual belief*, which not only expresses what all the agents believe but also what they belief that the others belief etc. A simple axiomatization of such a mutual belief operator then might look as follows:

$$\begin{array}{lcl} \Box_{MB}\Phi & \Rightarrow & \Phi \\ \Box_{MB}\Phi & \Rightarrow & \forall x\, \Box_x\Phi \\ \Box_{MB}\Phi & \Rightarrow & \forall x\, \Box_x\Box_{MB}\Phi \\ \Box_{MB}\Phi & \Rightarrow & \Box_{MB}\forall x\, \Box_x\Phi \\ \Box_{MB}\Phi & \Rightarrow & \Box_{MB}\Box_{MB}\Phi \end{array}$$

i.e. if something is mutually believed it is true and also everybody believes it and everybody believes that it is mutually believed and it is mutually believed that everybody believes it and, finally, every mutual belief is itself mutually believed.

What has to be done next after finding such a suitable axiomatization is to look for first-order accessibility relation properties which correspond to these very axioms. And if we do so for the axiomatization above we end up with

$$\begin{array}{lcl} R_{MB}(u,u) & & \\ R_x(u,v) & \Rightarrow & R_{MB}(u,v) \\ R_x(u,v) \wedge R_{MB}(v,w) & \Rightarrow & R_{MB}(u,w) \\ R_{MB}(u,v) \wedge R_x(v,w) & \Rightarrow & R_{MB}(u,w) \\ R_{MB}(u,v) \wedge R_{MB}(v,w) & \Rightarrow & R_{MB}(u,w) \end{array}$$

where an R_x represents the accessibility relation for the belief operator for agent x and R_{MB} evidently describes the mutual belief accessibility relation.

Thus we are at a stage where we can apply the saturation technique to R_{MB}[33] and we end up with all units of the form

$$R_{MB}(u, u\!:\!x_1\!:\!x_2\cdots\!:\!x_n)$$

where $n \geq 0$ and each of the x_i may belong to any of the respective functional decompositions.

Now, finding an alternative clause set which generates exactly the same saturation is a very simple task indeed. One such possiblity is for instance given by

$$\begin{array}{lcl} R_{MB}(u,u) & & \\ R_{MB}(u,v) & \Rightarrow & \forall x_\alpha\, R_{MB}(u, v\!:\!x_\alpha) \end{array}$$

where α is either an agent's name or just MB.

If we compare these two clauses with the original background theory for mutual belief we immediately see how considerable simplifications by saturation can be, and that not only for single modalities but also for multi-modal logics.

3.9 From Saturations to Inference Rules

For quite a lot of modal logics the semi-functional translation approach allowed us to simplify the background theory after saturation so considerably that is was often even reduced to a few

[33]For simplicity we assume here that no further properties for any of the accessibility relations involved is assumed.

unit clauses. There were exceptions, however, witness *S4*, for which the saturation was not that successful, although the simplifications are still significant.

The idea to follow in this section is – instead of trying to find an alternative clause set for the saturation found – to cast the whole saturation set into a suitable inference rule.

The method how this can be done will be illustrated by some examples.

3.9.1 An Inference System for *S4*

Recall the saturation for the modal logic *S4*:

$$\begin{array}{l} R(u,u) \\ R(u,u:x) \\ \dots \\ R(u,u:x_1:\dots:x_n) \\ \dots \end{array}$$

What can immediately be seen from this set is that each first argument of the respective R's is a variable and that each second argument "starts" with the same variable. This observation guarantees that – given an arbitrary unsatisfiable formula to be refuted – a corresponding finite and unsatisfiable set of ground instances of clauses taken from this formula contains negative R-literals only of the form $R(\alpha,\alpha:\beta_1:\dots:\beta_k)$ where α and the β_i are arbitrary ground world terms. This, on the other hand, gives rise to the assumption that it might be sufficient to always unify the first argument of such a negative R-literal with a *prefix* of its second argument and thus to forget about the whole background theory or its saturation.
This idea is formalized in the sequel.

DEFINITION 3.9.1 (WORLD-TERM PREFIXES)
Let $t = \alpha:\beta_1:\dots:\beta_n$ be a world term.
Then we call any $t' = \alpha:\beta_1:\dots:\beta_k$ with $0 \le k \le n$ a prefix *of t.*

DEFINITION 3.9.2 ((MOST GENERAL) S4 PREFIX UNIFIER)
Let t and t' be two world terms and let σ be a most general unifier of t and some prefix of t'.
Then we call σ a most general *S4* prefix unifier *of the (ordered) pair (t,t').*

DEFINITION 3.9.3 (THE *S4* INFERENCE RULE)
Let α and β be two arbitrary world terms and let C be an arbitrary clause. Then

$$\frac{\neg R(\alpha,\beta),C}{\sigma C}$$

where σ is an S4 prefix unifier of (α,β) is called the S4 Inference Rule.

DEFINITION 3.9.4 (THE *S4* INFERENCE SYSTEM)
The S4 Inference System consists of the standard resolution and factorization rules together with the S4 Inference Rule.

Such prefix unifiers are supposed to eliminate negative R-literals in the clause set. How this is done, and why this is indeed correct is described in the following lemma.

LEMMA 3.9.5
Let $\Im$ be an S4-model for the clause $\neg R(\alpha, \beta), C$ and let σ be a prefix-unifier for α and β. Then $\Im$ is also an S4-model for σC.

Proof: Obviously $\Im$ is a model for $\neg R(\sigma\alpha, \sigma\beta), \sigma C$. Remains to be shown that $\neg R(\sigma\alpha, \sigma\beta)$ is *S4*-unsatisfiable. This, however, follows simply from the fact that $\sigma\alpha$ is a prefix of $\sigma\beta$ and this contradicts one of the clauses $R(u, u\!: x_1\!: \ldots : x_n)$.

Remains to be shown that this *S4*-inference rule can really act as a replacement for the *S4* background theory, i.e. it has to be shown that the resulting inference system is both sound and complete.

LEMMA 3.9.6
The S4 Inference System is sound.

Proof: Resolution and factorization have not been changed at all compared to the classical case and are thus sound. Soundness for the *S4* Inference Rule follows from the lemma above.

The usual way to prove the completeness of a resolution based inference system is by first showing the completeness for the ground case and to prove that any such ground refutation can be lifted to the non-ground level. Essentially this will be done here as well, however, as it turns out, this is not quite as simple as in the classical predicate logic case.

LEMMA 3.9.7
Let $\mathcal{C}$ be an S4-unsatisfiable set of ground clauses (without positive R-literals). Then $\mathcal{C}$ can be refuted with the help of the S4 Inference System.

Proof: This works essentially as in the classical case. The only thing to be considered is the special treatment of the (negative) R-literals. According to the Herbrand Theorem, finitly many ground instances of the background theory saturation are sufficient. Hence, every negative ground R-literal is of the form $\neg R(\alpha, \beta)$ where α is a prefix of β and therefore each resolutions step on R-literals can be replaced by an application of the *S4* Inference Rule.

In spite of the similarities between this ground completeness proof and the corresponding proof in the classical case we run into some troubles when we try to follow a similar analogy for the lifting to the non-ground level as illustrated by the following simple example.

EXAMPLE 3.9.8
Consider the simple two-literal clause $\neg R(\iota, u), \neg R(\iota, u\!: f(u))$. A possible unsatisfiable ground instance of this clause is $\neg R(\iota, \iota\!: a), \neg R(\iota, \iota\!: a\!: f(\iota\!: a))$.
Now, on this ground level the *S4* Inference Rule can be applied to either of the two ground literals and doing so results in $\neg R(\iota, \iota\!: a)$ (or $\neg R(\iota, \iota\!: a\!: f(\iota\!: a))$ respectively). Unfortunately, neither of the corresponding applications on the non-ground level produces a result which is more general than the non-ground resolvent and thus the lifting fails.

Nevertheless, the above example gives us a hint how this problem can be overcome. For, if we had chosen another ground instance, namely

$$\neg R(\iota, \iota), \neg R(\iota, \iota\!: f(\iota))$$

we would be able to lift at least one of the two possible ground refutations. The problem is thus that an arbitrary given unsatisfiable ground instance is not yet enough; it ought to be in some sense *minimal*.

DEFINITION 3.9.9 (MINIMAL GROUND SUBSTITUTIONS)
Let $\mathcal{C}$ be a clause set and let σ and λ be two ground substitutions (on the variables occurring in $\mathcal{C}$). σ is called smaller *than λ w.r.t. the given clause set $\mathcal{C}$, denoted by $\sigma \preceq_{\mathcal{C}} \lambda$, if for every world variable u in $\mathcal{C}$ we have that $\sigma(u)$ is a prefix of $\lambda(u)$. σ is called* strictly smaller *than λ on $\mathcal{C}$ (written $\sigma \prec_{\mathcal{C}} \lambda$) if $\sigma \preceq_{\mathcal{C}} \lambda$ but not $\lambda \preceq_{\mathcal{C}} \sigma$. σ is called* minimal *w.r.t. $\mathcal{C}$ if $\sigma\mathcal{C}$ is unsatisfiable and there is no $\lambda \prec_{\mathcal{C}} \sigma$ such that $\lambda\mathcal{C}$ is unsatisfiable.*

LEMMA 3.9.10
For each unsatisfiable clause set $\mathcal{C}$ there exists a minimal ground substitution σ such that $\sigma\mathcal{C}$ is unsatisfiable.

Proof: Follows simply from the fact that each world-term has only finitely many prefixes.

DEFINITION 3.9.11 (OVERESTIMATED VARIABLES)
Let $\neg R(\alpha, u : \beta_1 \cdots : \beta_n)$ be a literal with world variable u and let σ be a ground substitution such that $\sigma(\alpha)$ is a proper prefix of $\sigma(u)$[34]. This occurrence of u is then called overestimated *by σ and the literal $\neg R(\alpha, u : \beta_1 \cdots : \beta_n)$ is called* unprocessable *under σ.*

DEFINITION 3.9.12 (RELEVANT PREFIXES)
Consider the literal $L = \neg R(\alpha, \beta)$ and let σ be a ground substitution such that $\sigma\alpha$ is a prefix of $\sigma\beta$. Then we call the smallest prefix of β, say β_1, such that $\sigma\alpha$ is a prefix of $\sigma\beta_1$ the relevant prefix *of β with respect to L and σ.*

Intuitively, unprocessable literals are those literals on which a potential step with the *S4* Inference Rule cannot be lifted. In such a case the right hand side of this literal starts with a world variable which is the relevant prefix w.r.t. the given literal and the given ground substitution. We are now going to show that inference steps on such unprocessable literals are not necessary.

LEMMA 3.9.13
Minimality of ground substitutions is preserved under the S4 Inference Rule provided it is applied to a single S4-unsatisfiable clause.

Proof: Let $C = \neg R(\alpha, \beta), C'$ be a single *S4*-unsatisfiable clause and let $\sigma C = \neg R(\sigma\alpha, \sigma\alpha : \gamma), \sigma C'$ be an *S4*-unsatisfiable ground instance of C such that the application of the *S4* Inference Rule to $\neg R(\sigma\alpha, \sigma\alpha : \gamma)$ can be lifted. Let $\mu C'$ be the corresponding (lifted) result on C. μ is more general than σ therefore there exists a λ such that $\sigma = \lambda\mu$. Pictorially, we have the following situation:

$$\begin{array}{ccc} \neg R(\alpha, \beta), C' & \stackrel{\sigma}{\longmapsto} & \neg R(\sigma\alpha, \sigma\alpha : \gamma), \sigma C' \\ \hline \mu C' & \stackrel{\lambda}{\longmapsto} & \sigma C' \end{array}$$

Now assume that λ is not minimal w.r.t. $\mu C'$. Then there exists a $\lambda' \prec \lambda$ such that $\lambda'\mu C'$ is unsatisfiable. Define σ' as $\sigma' = \lambda'\mu$. Then $\sigma' = \lambda'\mu \prec \lambda\mu = \sigma$. Moreover we know that $\sigma' C$ is unsatisfiable since $\lambda'\mu C'$ is unsatisfiable and $\lambda'\mu C$ (and thus $\sigma' C$) is equivalent to $\lambda'\mu C'$. Hence $\sigma' C$ is unsatisfiable and $\sigma' \prec \sigma$ and therefore σ is not minimal w.r.t. C which contradicts our assumption and we are done.

LEMMA 3.9.14
Let C be a single S4-unsatisfiable clause and let σC be a minimal unsatisfiable ground instance of C. Then there exists an application of the S4 Inference Rule which can be lifted to C.

[34] By a *proper* prefix of a world-term γ we mean a prefix γ' of γ such that γ is not a prefix of γ'

Proof: Suppose every literal in C is unprocessable. Then each literal is of the form $\neg R(\alpha_i, u_i : \beta_i)$ and any such mention of u_i is overestimated. However, since σ is minimal, there must be an occurrence of u_i where u_i is not overestimated. This cannot be on a right-hand side of one of the R-literals because otherwise this R-literal would not be unprocessable or the occurrence would not belong to the corresponding relevant prefix. Now take any of these overestimated variables, say u, and find a literal where u is not overestimated. The right-hand-side of this literal must contain an overestimated variable (since again this literal would not be unprocessable otherwise) and this variable (let us call it v) must be different from u; in fact we know that $\sigma(u)$ is a proper prefix of $\sigma(v)$. Now we start the whole search again with v and this leads us to a variable w such that $\sigma(v)$ is a proper prefix of $\sigma(w)$ and so on. Thus if every literal is unprocessable then there are infinitly many different world variables and this is impossible. Therefore there exists at least one literal which is not unprocessable and thus the corresponding Inference Rule step on σC can be lifted.

LEMMA 3.9.15
Let C be a single S4-unsatisfiable clause. C can be refuted with the help of the S4 Inference Rule.

Proof: Let σ be minimal w.r.t. C (σ exists because of Lemma 3.9.10). The proof is performed by induction over the length of σC.

Base case: σC is a unit clause. By the previous lemma we know that there exists a σ-processable literal and since the clause under consideration consists of only one literal this one must be σ-processable. Hence an *S4* inference step can be performed and we immediately get the empty clause.

Induction step: By the previous lemma we know that there is a liftable *S4* Inference Rule step and by Lemma 3.9.13 the ground result is a minimal instance of the corresponding resolvent on the non-ground level. Since the size of the resolvent is smaller than the one for σC the induction hypothesis applies and we are done.

LEMMA 3.9.16
Resolution and factorization suffice to derive a single S4-unsatisfiable clause from an arbitrary S4-unsatisfiable clause set.

Proof: Let $\mathcal{C}$ be the *S4*-unsatisfiable clause set and let σ be an arbitrary ground substitution such that $\sigma\mathcal{C}$ is *S4*-unsatisfiable but no proper subset of $\sigma\mathcal{C}$ is. Moreover let $\sigma\mathcal{C}'$ be obtained from $\sigma\mathcal{C}$ by ignoring all R-literals. Evidently, for each R-literal in $\sigma\mathcal{C}$ we have that its left-hand argument is a prefix of its right-hand argument. Furthermore we know that $\sigma\mathcal{C}'$ is classically unsatisfiable. Hence $\sigma\mathcal{C}'$ can be refuted (by resolution alone) and performing exactly the same derivation steps on $\sigma\mathcal{C}$ results in a pure-R-negative clause (with literals of the form $\neg R(\alpha, \alpha : \cdots)$). This derived clause is clearly *S4*-unsatisfiable and therefore the lemma holds for the ground case. However, all inference steps performed up to now were only resolutions steps and therefore the derivation of the *S4*-unsatisfiable clause can be lifted to $\mathcal{C}$ (resolution and factorization steps). Hence we are able to derive a *S4*-unsatisfiable clause from an arbitrary *S4*-unsatisfiable clause set and we are done.

LEMMA 3.9.17
The S4 Inference System is refutation complete.

Proof: Let $\mathcal{C}$ be an unsatisfiable clause set. By Lemma 3.9.16 we have that a single *S4*-unsatisfiable clause can be derived by resolution and factorization and from Lemma 3.9.15 such a clause can be refuted with the help of the *S4* Inference Rule.

Thus we have shown that this inference system is both sound and complete. For convenience this is fixed by the following theorem.

THEOREM 3.9.18
Let Φ be a modal logic formula in negation normal form. Then

$$\Phi \textit{ is S4-unsatisfiable} \quad \textit{iff} \quad \lfloor \Phi \rfloor_\iota \textit{ can be refuted with the S4 Inference System}$$

Proof: Follows from Lemmas 3.9.6 and 3.9.17.

The completeness proof might suggest that it is necessary to first apply resolution and factorization steps until an R-clause is derived which then hopefully can be refuted by the *S4* Inference Rule. However, this order has been chosen only to simplify the completeness proof. It should be evident that applications of the *S4* Inference Rule do not necessarily have to be delayed until pure-R-negative clauses are derived.

3.9.2 An Inference System for *KD4*

At the first glance it seems obvious how an inference system for the modal logic *KD4* might look like. The only difference between the saturations of the background theories for *S4* and *KD4* lies in the absence of the reflexivity unit clause and this suggests that the notion of a *KD4* prefix unifier has to be changed accordingly.

So let us fix this with the help of some useful definitions.

DEFINITION 3.9.19 (*KD4* PREFIX UNIFIER)
Let (s, t) be an ordered pair of world terms and let σ be the most general unifier of s and a proper *prefix of t. Then σ is called a KD4* prefix unifier *of the pair (s, t).*

DEFINITION 3.9.20 (THE *KD4* INFERENCE RULE)
The rule

$$\frac{\neg R(\alpha, \beta), C}{\sigma C}$$

$$\textit{where } \sigma \textit{ is a KD4 prefix unifier for } (\alpha, \beta)$$

is called the KD4 Inference Rule.

For *S4* the corresponding inference rule was sufficient in order to get a refutation complete inference system together with standard resolution and factorization. However, the *KD4* Inference Rule is not enough for this purpose. As a trivial counter-example consider the unit clause $\neg R(\iota, u)$ which is obviously *KD4*-unsatisfiable but cannot be refuted with the help of the *KD4* Inference Rule. It is thus necessary to add a further inference rule which helps in such cases.

DEFINITION 3.9.21 (THE *KD4* ELIMINATION RULE)
The rule

$$\frac{\neg R(\alpha, u), C}{C^u_{\alpha:x}}$$

where u does not occur in α
and x is a new variable of sort F_R

is called the KD4 Elimination Rule*.*

DEFINITION 3.9.22 (THE *KD4* INFERENCE SYSTEM)
The KD4 Inference System *consists of the classical resolution and factorization rule together with the KD4 Inference Rule and the KD4 Elimination Rule.*

The soundness of the *S4* Inference System was pretty evident because of the background theory saturation we knew about. Analogously, neither the *KD4* Inference Rule nor the *KD4* Elimination rule poses any problems.

LEMMA 3.9.23
The KD4 Inference System is sound.

Proof: Soundness of the *KD4* Inference Rule follows immediately from our knowledge about the saturation of the *KD4* background theory.

For the soundness of the *KD4* Elimination Rule it suffices to bear in mind that arbitrary instantiation is sound and therefore – since u does not occur in α – it is certainly true that any *KD4* model $\Im$ for the clause $\neg R(\alpha, u), C$ is also a model for $\neg R(\alpha, \alpha : x), C^u_{\alpha:x}$. Now, the literal $\neg R(\alpha, \alpha : x)$ is *KD4*-unsatisfiable for any x. Thus $\Im$ is a model for $C^u_{\alpha:x}$ and we are done.

LEMMA 3.9.24
An application of the KD4 Inference Rule on a clause C with literal L corresponds to a resolution step between this literal L and an element of the saturation of the KD4 background theory.

Proof: Consider the clause

$$\neg R(\alpha, \beta), C$$

where σ is a *KD4* prefix unifier for (α, β) and let β' be the proper prefix of β such that σ is the most general unifier of α and β'. Then $\beta = \beta' : \beta_1 : \cdots : \beta_n$ with $n \geq 1$ and we thus have the situation

$$\neg R(\alpha, \beta' : \beta_1 : \cdots : \beta_n), C$$

with $\sigma\alpha = \sigma\beta'$. Now consider the unit clause

$$R(u, u : x_1 : \cdots : x_n)$$

from the saturation of the background theory for *KD4*. Evidently, a resolution step between the two latter clauses results in σC and this is just what has been claimed.

Recall that in the refutation completeness proof for the *S4* Inference System we could take into account the fact that a given *minimally* unsatisfiable ground instance guarantees that each world variable occurs somewhere *not* overestimated. This is not necessarily true anymore in case of

KD4 as the example with the unsatisfiable unit clause $\neg R(\iota, u)$ shows. A possible unsatisfiable ground instance of this clause might be $\neg R(\iota, \iota : a)$ and indeed this ground instance is minimal since there exists no other ground substitution μ such that $\neg R(\mu\iota, \mu(u))$ is unsatisfiable and $\mu(u)$ is a proper prefix of $\sigma(u)$. Nevertheless, u is evidently overestimated and therefore it cannot be guaranteed that every variable occurs somewhere *not* overestimated. It can be observed, however, that if a variable u occurs only overestimated in a single *KD4*-unsatisfiable clause then it occurs only on right-hand arguments of the respective R-literals. This fact allows us to perform suitable *KD4* Elimination Rule steps.

LEMMA 3.9.25
The KD4 Inference System is refutation complete.

Proof: In analogy to the completeness proof for *S4* it suffices to show that any single *KD4*-unsatisfiable clause can be refuted with the help of the *KD4* Inference System.

Now consider an arbitrary *KD4*-unsatisfiable clause C with a minimally unsatisfiable ground instance σC. If each variable which occurs overestimated also occurs somewhere on a left-hand argument of a R-literal then the proof that there is a liftable step with the *KD4* Inference Rule follows exactly the lines of the corresponding proof for the *S4* Inference Rule. Therefore assume that there is a variable, say u, which occurs only overestimated. By the observation from above we know that this u occurs only on right-hand arguments of the R-literals and thus C is of the form

$$\neg R(\alpha_1, u : \beta_1), \ldots, \neg R(\alpha_n, u : \beta_n), C'$$

and each $\sigma(\alpha_i)$ is a proper prefix of $\sigma(u)$. Hence all the $\sigma(\alpha_i)$ can be ordered under the prefix relation and there is a maximal one, say $\sigma(\alpha_k)$, such that for all $j \neq k$ we have that $\sigma(\alpha_j)$ is a prefix of $\sigma(\alpha_k)$, and even that $\sigma\beta_k = \emptyset$. This is so, because if each of the literals with maximal $\sigma\alpha_k$ had a non-empty $\sigma\beta_k$ then the minimality of σ would guarantee that u is not overestimated. Thus we have that $\beta_k = \emptyset$ and, moreover, that $\sigma(\alpha_k)$ is the direct prefix of $\sigma(u)$. Therefore the *KD4* inference step on $\neg R(\sigma\alpha_k, \sigma(u))$ can be simulated on the clause C by a *KD4* Elimination step in the sense that $\sigma C \setminus \{\neg R(\sigma\alpha_k, \sigma(u))\}$ – i.e. the result of a *KD4* Inference Rule application on σC – is an instance of $C^{u}_{\alpha_k : x}$ – the result of a *KD4* Elimination Rule application to C. Hence any *KD4*-unsatisfiable clause C can be reduced with the help of either the *KD4* Inference Rule or the *KD4* Elimination Rule to a smaller *KD4*-unsatisfiable clause and the lemma follows by induction over the length of C.

3.9.3 An Inference System for Mutual Belief

Let us again have a look at the Mutual Belief example in Section 3.8. After saturating the background theory we ended up with all units of the form

$$R_{MB}(u, u : x_1 : x_2 \cdots : x_n)$$

where $n \geq 0$ and each of the x_i belongs to any of the functional decompositions. The strong relation between this set of unit clauses and the saturation of the *S4* background theory is pretty obvious. They differ mainly in the fact that variables may belong to different functional decompositions. Nevertheless, the observations which lead us to the inference system for *S4* do apply here as well, i.e. whenever there is an occurrence of a negative R_{MB}-literal we try to unify its left-hand-side with a prefix of its right-hand-side.

Thus the inference system for Mutual Belief does not have to be changed at all compared with the *S4* Inference System, even the soundness and completeness proofs remain unchanged.

3.9.4 "Mixing" *S4* and *KD4*

As another example for multi-modalities consider a logic which is based on a *S4* and a *KD4* fragment such that the accessibility relation for the former is just the reflexive closure of the accessibility relation of the latter. Or, in other words, let $\Box$ be the modality of the *KD4* fragment and let $\Box^*$ be the modality of the *S4* part (with associated accessibility relation symbol R for $\Box$) then any formula of the form $\Box^*\Phi$ is true in a world w if and only if Φ is true in w and, in addition, in every world accessible from w by R as well. The basic principles behind this multi-modal logic will play a crucial role in the development of calculi for the temporal logics to be considered later.

As a first attempt to reason within this combination we might consider a translation approach which helps us with this problem, namely one which translates the *S4* modalities in terms of the *KD4* modalities. This would result in something like:

$$\begin{aligned} [\Box^*\Phi]^1_u &= [\Phi]^1_u \wedge \forall v\ R(u,v) \Rightarrow [\Phi]^1_v \\ [\Box\Phi]^1_u &= \forall v\ R(u,v) \Rightarrow [\Phi]^1_v \end{aligned}$$

The advantage of this kind of translation is that there is only one accessibility relation to be considered. Its disadvantage lies in the fact that $\Box^*$-formulae get translated into conjunctions and $\Diamond^*$-formulae get translated into disjunctions and therefore the clauses we obtain after clause normal form transformation become fairly big (and also we get lots of them).

As an alternative we might consider another translation, namely

$$\begin{aligned} [\Box^*\Phi]^2_u &= \forall v\ R^*(u,v) \Rightarrow [\Phi]^2_v \\ [\Box\Phi]^2_u &= \forall v\ R(u,v) \Rightarrow [\Phi]^2_v \end{aligned}$$

which keeps off a lot of this representational overhead.

These two translation approaches are indeed equivalent provided we guarantee that R^* represents the reflexive closure of R as is shown in the following lemma.

LEMMA 3.9.26
Let R^* represent the reflexive closure of R. Then $[\Phi]^1_u$ is equivalent to $[\Phi]^2_u$.

Proof: Follows by a simple induction over the structure of Φ. For the only interesting case (where $\Phi = \Box^*\Psi$) it suffices to realize that the translation $[\,]^1$ can be reformulated into:

$$[\Box^*\Psi]^1_w = \forall v\ (v = u \vee R(u,v)) \Rightarrow [\Psi]^1_v$$

Since R^* is the reflexive closure of R we have that

$$R^*(u,v) \Leftrightarrow u = v \vee R(u,v)$$

and thus

$$[\Box^*\Psi]^1_w \Leftrightarrow [\Box^*\Psi]^2_w$$

The translation $[\ ]^2$ is to be preferred because it results in fewer and smaller clauses. The price to be paid, however, is that we have to deal with a more complicated background theory whereas describing $\Box^*$ in terms of $\Box$ would mean to merely consider the *KD4* background theory. Under translation $[\ ]^2$ we have a *KD4* and a *S4* background theory together with the additional information that R^* denotes the reflexive closure of R. Whether or not this really simplifies reasoning within the combined logic will have to be investigated in the following.

Let us first have a look at the axiomatization of this very multi-modal logic (which we shall call *S4⊕KD4* in the sequel). It consists of the two modal logic fragments for *S4* and *KD4* and a combination axiom which guarantees that the one accessibility relation is indeed the reflexive closure of the other, hence the full axiomatization is given by

$$\begin{array}{l} \Box\Phi \Rightarrow \Diamond\Phi \\ \Box\Phi \Rightarrow \Box\Box\Phi \\ \Box^*\Phi \Rightarrow \Phi \\ \Box^*\Phi \Rightarrow \Box^*\Box^*\Phi \\ \Box^*\Phi \Leftrightarrow \Box\Phi \wedge \Phi \end{array}$$

where the usual K-axioms, the necessitation rules and Modus Ponens have to be added. Evidently, the first two axioms describe the *KD4* part and the third and fourth axiom the *S4* fragment. The final axiom then provides the relationship between the two $\Box$-operators such that we get the following background theory for *S4⊕KD4* (after semi-functional translation):

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R^*(u,u) \\ R^*(u, u\!:\!x^*) \\ R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) \\ R(u,v) \Rightarrow R^*(u,v) \\ R^*(u,v) \Rightarrow R(u,v) \vee u = v \end{array}$$

where x (x^*) ranges over the functional decomposition of R (R^*). Actually, the transitivity of R^* is redundant here for it follows already from the other properties (the reflexive closure of a transitive relation is transitive itself).

Our aim is now to saturate these properties and to find suitable inference rules which finally may replace the background theory (or its saturation). To this end we have to determine the set of pure-$R^{(*)}$-positive clauses that are derivable from the *S4⊕KD4* background theory by resolution on $R^{(*)}$-literals.

Notation 3.9.27
Given the functional decompositions $F_{R_1}, \ldots, F_{R_n}$ we denote by $\alpha \in (F_{R_1} \cup \ldots \cup F_{R_n})^*$ that α is a *sequence* $\alpha_1\!:\!\alpha_2\!:\cdots:\!\alpha_m$ such that each α_i ($1 \leq i \leq m$) is an element of $F_{R_1} \cup \ldots \cup F_{R_n}$. To indicate that at least one of the members of α belongs to F_R we write: $\exists\alpha_i \in \alpha\ \alpha_i \in F_R$. A sequence which consists only of functional decomposition variables is usually represented by an overlined variable symbol, e.g. by $\overline{x}$. Moreover, we sometimes write $\alpha\!:\!\beta$ to indicate that the world term in question can be split into a prefix α and a suffix sequence β.

LEMMA 3.9.28
The set of clause schemata of the form

$$\begin{array}{lll} R^*(u,u:\overline{x}) & \textit{with} & \overline{x} \in (F_R \cup F_{R^*})^* \\ R(u,u:\overline{x}) & \textit{with} & \exists x_j \in \overline{x}\ x_j \in F_R \\ R(u,u:x:\overline{y}) \vee u:x = u & \textit{with} & x:\overline{y} \in (F_{R^*})^* \end{array}$$

is derivable from the S4⊕KD4 background theory (where x^* means that x is a symbol representing an element of F_{R^*}.

Proof: First we show that $R^*(u,v) \Rightarrow R^*(u,v:x)$.

$$\begin{array}{ll} R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) & \text{Transitivity of } R^* \\ R(u,v) \Rightarrow R^*(u,v) & R \text{ is a subrelation of } R^* \\ R(u,u:x) & \text{Seriality of } R \\ \hline R^*(u,v) \wedge R(v,w) \Rightarrow R^*(u,w) & \text{by the first two} \\ R^*(u,v) \Rightarrow R^*(u,v:x) & \text{plus the third} \end{array}$$

Also we have that $R^*(u,v) \Rightarrow R^*(u,v:x^*)$ by

$$\begin{array}{ll} R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) & \text{Transitivity of } R^* \\ R^*(u,u:x^*) & \text{Seriality of } R^* \\ \hline R^*(u,v) \Rightarrow R^*(u,v:x^*) & \text{by resolution} \end{array}$$

Then it follows by induction that

$$\left.\begin{array}{l} R^*(u,u) \\ R^*(u,v) \Rightarrow \left\{ \begin{array}{l} R^*(u,v:x^*) \\ R^*(u,v:x) \end{array} \right. \end{array}\right\} \Rightarrow R^*(u,u:\overline{x})$$

This covers the first clause schema.

Similarly we proceed for the second schema. Note that we can show from the background theory that $R^*(u,v) \Rightarrow R(u,v:x)$, $R(u,v) \Rightarrow R(u,v:x^*)$ and $R(u,v) \Rightarrow R(u,v:x)$. Proofs of these facts are a bit tedious; they can easily be checked by a standard theorem prover, however. Then we get by induction that

$$\left.\begin{array}{l} R^*(u,u) \\ R^*(u,v) \Rightarrow R(u,v:x) \\ R(u,v) \Rightarrow \left\{ \begin{array}{l} R(u,v:x^*) \\ R(u,v:x) \end{array} \right. \end{array}\right\} \Rightarrow R(u,u:\overline{x})$$

where $\overline{x} \neq \emptyset$ and $\exists x_i \in F_R$ and this covers the second clause schema.

For the third schema recall that $R(u,v) \Rightarrow R(u,v:x^*)$ and that $R^*(u,u:\overline{x})$ (in particular in the case where $\forall x_i \in F_{R^*}$). Thus we can show (again by induction) that

$$\left.\begin{array}{l} R^*(u,u:\overline{x}) \text{ with } \forall x_i \in F_{R^*} \\ R(u,v) \Rightarrow R(u,v:x^*) \\ R^*(u,v) \Rightarrow R(u,v) \vee u = v \end{array}\right\} \Rightarrow R(u,u:\overline{x}:\overline{y}) \vee u = u:\overline{x}$$

where $\forall x_i, y_j \in F_{R^*}$. Thus, in particular, $R(u,u:x^*:\overline{y}) \vee u = u:x^*$ and we are done.

The clause schemata from the previous lemma do not cover all positive clauses which are somehow derivable from the background theory. They are sufficient, however, and this is shown next.

Lemma 3.9.29
Let Φ be a S4⊕KD4-unsatisfiable clause set. Then there exists a finite set of instances of the derived clause schemata, say $\mathcal{C}$, such that $\mathcal{C} \cup \Phi$ is (classically) unsatisfiable.

Proof: First recall that the transitivity of R^* is already derivable from the fact that R^* is the reflexive closure of the transitive relation R and thus is redundant. We now show two main facts:

1. $R^*(u,v) \Rightarrow R(u,v) \vee u = v$ can be simplified to $R(u, u\!:\!x^*) \vee u\!:\!x^* = u$
2. $R(u,v) \wedge R(v,w) \Rightarrow R(u,w)$ and $R(u,v) \Rightarrow R^*(u,v)$ become redundant under the derived clause schemata

It then follows immediately that the derived clause schemata cover all positive clauses which are derivable from the given background theory and therefore the lemma holds.

The first of the two facts can easily be proved as follows: Let Φ be the clause set which results from the semi-functional translation and clause form transformation of some *S4⊕KD4*-unsatisfiable formula. Then there exists a finite set of ground clauses from clauses of Φ and a finite set of ground clauses from the *S4⊕KD4* background theory such that the union of these finite clause sets is classically unsatisfiable and can be refuted (for example by some standard resolution theorem prover). In particular there is a finite set of instances of the clause $R^*(u,v) \Rightarrow R(u,v) \vee u = v$ involved and we are going to prove what has been claimed by an induction over the number of instances of this very clause. In the base case this clause is not required at all and we are already done. Therefore assume that there are $n > 0$ ground instances needed. Now consider any path[35] through the clause set which contains all the literals $\neg R^*(\alpha,\beta)$ which stem from $R^*(u,v) \Rightarrow R(u,v) \vee u = v$ and whose unsatisfiability requires at least one of these R^*-literals. Such a clause set must exist for otherwise all these instances were not necessary for this very Φ and we would be finished anyway. Now the only possibility to resolve these literals $\neg R^*(\alpha,\beta)$ is by a resolution step (paramodulation steps are not possible because there are no further equational clauses involved) with instances of $R^*(u,u)$, $R^*(u, u\!:\!x^*)$, or $R(u,v) \Rightarrow R^*(u,v)$. However, the first and the third of these possibilities can be ignored for they would result in tautologies. Remains the second possibility and this results in just the simplification that has been claimed.

The second of the facts from above can also very easily be shown. It suffices to observe that any pair (α,β) in the R relation is also in the R^* relation (which is obvious) and that the clauses containing R-literals are closed under transitivity. This requires four very simple cases to be checked. As an example consider the case where two instances of the clause schema $R(u, u\!:\!x^*\!:\!\overline{y}) \vee u\!:\!x^* = u$ are resolved with the transitivity clause. This results in $R(u, u\!:\!x^*\!:\!\overline{y}\!:\!v^*\!:\!\overline{w}) \vee u\!:\!x^* = u \vee u\!:\!x^*\!:\!\overline{y}\!:\!v^* = u\!:\!x^*\!:\!\overline{y}$ which is subsumed by $R(u, u\!:\!x^*\!:\!\overline{y}) \vee u\!:\!x^* = u$.

[35]By a path through a set of k ground clauses we understand any set of k literals which contains a literal from each of the k ground clauses.

Thus we are now at a stage where we can try to find an appropriate set of inference rules which replaces the saturation of the background theory. Such a set of inference rules is supposed to cover the responsibilities of each of the derived clause schemata and the close relation to both *S4* and *KD4* provides already with a hint how a suitable inference system for *S4*⊕*KD4* might look like. Essentially we need an *S4* rule for the R^*, the *KD4* rules for R and a further rule for the equational clauses (which say something about the relation between R and R^*).

DEFINITION 3.9.30 (THE *S4*⊕*KD4* INFERENCE RULES AND SYSTEM)
The S4⊕KD4 Inference System *consists of the classical resolution, factorization and paramodulation rules together with the following S4⊕KD4* Inference Rules

$$\frac{\neg R^*(\alpha, \beta : \gamma), C}{\sigma C} \qquad \frac{\neg R(\alpha, \beta : \gamma), C}{\sigma C}$$

$$\begin{array}{cc} \gamma \in (F_R \cup F_{R^*})^* & \exists x_i \in \gamma \; x_i \in F_R \\ \sigma = mgu(\alpha, \beta) & \sigma = mgu(\alpha, \beta) \end{array}$$

$$\frac{\neg R(\alpha, u), C}{C^u_{\alpha : x}} \qquad \frac{\neg R(\alpha, \beta : \gamma : \delta), C}{\sigma C, \sigma(\beta : \gamma) = \sigma\alpha}$$

$$\begin{array}{cc} \textit{provided } u \textit{ does} & \gamma : \delta \in (F_{R^*})^* \\ \textit{not occur in } \alpha & \sigma = mgu(\alpha, \beta) \end{array}$$

Note that it suffices to assume that the γ in the last of the above inference rules consists only of a single F_{R^*}-term.

Before the soundness and refutation completeness of this inference system is proved let us have a look at an example which shows the main difference between the two translation approaches.

EXAMPLE 3.9.31
Consider the modal logic formula

$$\Box^*((P \wedge \Box Q) \Rightarrow \Box^*(P \vee \Box^* Q))$$

After negation and clause normal form transformation we end up with

$$\begin{array}{l} P(\iota : a^*) \\ \neg R(\iota : a^*, u), Q(u) \\ \neg P(\iota : a^* : b^*) \\ \neg Q(\iota : a^* : b^* : c^*) \end{array}$$

In fact, there are only two inference steps possible, both with the second clause involved, namely a resolution step between the two Q-literals and a *KD4* Elimination Step on the R-literal. The latter immediately leads to a dead-end, the former, however, results in

$$\neg R(\iota : a^*, \iota : a^* : b^* : c^*)$$

Again there is only one possible further step, and that with the last one of the above rules. This step then yields

$$\iota : a^* : b^* = \iota : a^*$$

and this equation together with the two P-literals immediately leads to the empty clause.

But what would happen if we tried the other approach, i.e. if we translated the "starred" operators in terms of the other modalities. This would mean that we have to show the *KD4*-unsatisfiablity of the formula we get after translating each $\Box^*\Phi$ into $\Phi \wedge \Box\Phi$ and each $\Diamond^*\Phi$ into $\Phi \vee \Diamond\Phi$. As a matter of fact such a translation and a following clause form transformation results in a clause set which consists of about 40 clauses and approximately 180 literals. Presenting this clause form here would be a boring task indeed and is therefore omitted. What this shows, however, is how significant the savings are if we treat the $\Box^*$ as an operator on its own and not simply as an abbreviation.

We now proceed with the soundness and refutation completeness proofs for the *S4⊕KD4* Inference System.

LEMMA 3.9.32
The S4⊕KD4 Inference System is sound.

Proof: The proof is in full analogy to the corresponding proof for the inference systems for *S4* and for *KD4*. Only the last of the new *S4⊕KD4* Inference Rules – the one which introduces an equation – has not yet been considered. Its soundness, however, follows immediately from the fact that any of its applications corresponds to a resolution step with an instance of the clause schema

$$R(u, u\!:\!x^*\!:\!\overline{y}), u\!:\!x^* = u$$

which belongs to the saturation of *S4⊕KD4*'s background theory.

For the completeness proof of the Inference System we have to take into account that equations can be derived and therefore some equation handling is necessary. For convenience, we consider the so called "simultaneous paramodulation" as it had been introduced in (Benanav 1990). This approach is particularly interesting for it obeys a lifting lemma, something which is not possible for paramodulation in general. Informally, simultaneous paramodulation (s-paramodulation for short) differs from standard paramodulation in that an equation $\alpha = \beta$ is paramodulated into *every* subterm α whereas in standard paramodulation terms are replaced one at a time. Evidently, any s-paramodulation step can be simulated by a series of standard paramodulation steps.

LEMMA 3.9.33 (BENANAV'S S-PARAMODULATION LIFTING LEMMA)
Suppose that C_1, C_2, C_1', C_2' and C' are clauses such that C' is an s-paramodulant of C_2' into C_1' and that C_1 subsumes C_1' (i.e. some instance of C_1 is a subset of C_1') and C_2 subsumes C_2'. Then either C_2 subsumes C' or there exists an s-paramodulant C of C_2 into C_1 such that C subsumes C'.

LEMMA 3.9.34
The S4⊕KD4 Inference System is refutation complete on ground clauses.

Proof: This follows immediately from the resolution and s-paramodulation completeness and the fact that (on ground clauses) there is a one-to-one correspondence between resolution steps with an instance of the derived clause schemata and an application of one of the new inference rules.

What the general refutation completeness of the *S4⊕KD4* Inference System is concerned we are again faced with the problem that a general lifting lemma does not hold for the new inference rules. Therefore we are not able to lift an arbitrary refutation of an arbitrary unsatisfiable ground instance to the general level. What we can show, however, is that there exists an unsatisfiable ground instance and a refutation for this clause set which can be lifted. To this end some auxiliary lemmas turn out to be useful.

LEMMA 3.9.35
Let C be a single S4⊕KD4-unsatisfiable clause. C can be refuted with the help of the S4⊕KD4 Inference System.

Proof: Since C is *S4⊕KD4*-unsatisfiable there exists a minimal ground substitution σ such that σC is *S4⊕KD4*-unsatisfiable. σC can be refuted according to Lemma 3.9.34 but suppose that none of the steps in this refutation can be lifted. Then each of the literals in C is of the form $\neg R^{(*)}(\alpha_i, u_i : \beta_i)$ with u_i overestimated w.r.t. σ and at least one of these u_i occurs only on right-hand arguments (for otherwise we would be able to find infinitly many different variables – see the corresponding proofs for *S4* and *KD4*). However, not each of these literals can be a R^*-literal for otherwise this ground substitution were not minimal. Therefore – in analogy to the corresponding proof in case of *KD4* – we are able to find a maximal $\sigma\alpha_k$ (with $\beta_k = \emptyset$) such that the *KD4* Elimination Rule can be applied on $\neg R(\alpha_k, u_k)$ and this step lifts the corresponding step on $\neg R(\sigma\alpha_k, \sigma(u_k))$. Note that this very literal must indeed be a R-literal rather than a R^*-literal for otherwise u were not overestimated in this literal and the corresponding step could be lifted. Hence in any case there is a liftable step which reduces the size of σC and the proof is completed by a simple induction over the length of σC.

LEMMA 3.9.36
Any clause C of the form

$$\neg R^{(*)}(\alpha_1, u_1 : \beta_1), \ldots, \neg R^{(*)}(\alpha_n, u_n : \beta_n)$$

for which a ground substitution σ exists such that $\sigma\alpha_i$ is a proper prefix of $\sigma(u_i)$ is S4⊕KD4-unsatisfiable.

Proof: First we show that at least one of the u_i occurs only on right-hand side arguments of the literals in C. To this end consider any of the u_i, say u_1. If u_1 occurs in a left-hand argument, say in α_2, we know that $\sigma(u_1)$ is a proper prefix of $\sigma(u_2)$ and thus u_1 and u_2 are different. We start our search again, this time with u_2, i.e. if u_2 occurs on a left-hand-side, say in α_3, then u_1 and u_2 are both different from u_3 and this progression can only be stopped by finding a variable which does not occur on left-hand-sides.

Now consider C without the literals $\neg R^{(*)}(\ldots, u_i : \cdots)$. The very same situation holds for this subclause, i.e. there is a variable, say u_j, which occurs only on right-hand sides on this subclause (it might occur on a left-hand side in the literals $\neg R^{(*)}(\ldots, u_i : \cdots)$ though). Doing so for all world variables finally leads us to the possibility of permuting C such that for any literal $\neg R^{(*)}(\alpha_i, u_i : \beta_i)$ only the variables $u_1, \ldots, u_{i-1}$ can occur in α_i and, in particular, there is a variable u_j such that for every $\neg R^{(*)}(\alpha_j, u_j : \beta_j)$ no world variable at all occurs in α_j. Evidently, for any evaluation of the α_j we are able to instantiate u_j such that each $\neg R^{(*)}(\alpha_j, u_j : \beta_j)$ gets *S4⊕KD4*-unsatisfiable and that without instantiating any of the other world variables (this instantiation certainly depends on the domain variables in the α_j). Due to our construction there is now another world variable without any world variable on the left-hand side of its occurrences and again there is a suitable instantiation (of domain variables) which yields the unsatisfiablity of these literals and so on. The lemma thus follows immediately by a simple induction over the number of world variables u which occur in the form $\neg R^{(*)}(\ldots, u\colon \cdots)$.

LEMMA 3.9.37
Let C be a clause of the form

$$\neg R^{(*)}(\alpha_1, \beta_1), \ldots, \neg R^{(*)}(\alpha_n, \beta_n)$$

and let σ be a ground substitution such that $\sigma\alpha_i$ is a (proper) prefix of $\sigma\beta_i$ for every $1 \leq i \leq n$. Then either C is S4⊕KD4-unsatisfiable or a conditioned equation can be derived from σC with the help of the S4⊕KD4 Inference System and this derivation can be lifted to C.

Proof: W.l.o.g. we can assume that C and σC are of equal length. Since each $\sigma\alpha_i$ is a (proper) prefix of $\sigma\beta_i$ we have that either there is a liftable step or each of the β_i is of the form $u_i : \gamma_i$. In the former case we are either able to reduce the size of C by applying this liftable step or we already have derived a conditioned equation by this liftable derivation. In the latter case we are again guaranteed that at least one of these variables u_i occurs only on right-hand side arguments and the previous lemma then guarantees that C is *S4⊕KD4*-unsatisfiable.

NOTATION 3.9.38
A ground literal of the form $\neg R(\alpha, \alpha\colon\beta)$ is called *equality introducing* (an E-literal for short) if $\beta \in (F_{R^*})^*$. The *unconstraint part* of a clause set is obtained by ignoring the $R^{(*)}$-literals occurring in this clause set.

LEMMA 3.9.39
Let $C = C', \neg R(\alpha, \alpha\colon \beta_1^*\colon \cdots \colon \beta_n^*)$ be a clause in an S4⊕KD4-unsatisfiable set $\mathcal{C}$ of ground clauses. $\mathcal{C}$ remains unsatisfiable if we replace C by the set

$$\{C', \alpha\colon \beta_i^* = \alpha \mid 1 \leq i \leq n\}$$

Proof: Recall from Lemma 3.9.29 that a finite number of ground instances from the derived clause schemata suffices to refute a given *S4⊕KD4*-unsatisfiable clause set. Hence, bearing the above clause C in mind, the clause

$$R(\alpha, \alpha\colon \beta_1^*\colon \cdots \colon \beta_n^*), \alpha\colon \beta_1^* = \alpha$$

must be part of the background theory and the literal $R(\alpha, \alpha\colon \beta_1^*\colon \cdots \colon \beta_n^*)$ does not occur anywhere else in the clause set. Therefore the only possible resolution step between this theory clause and the given clause set results in replacing each occurrence of the literal $\neg R(\alpha, \alpha\colon \beta_1^*\colon \cdots \colon \beta_n^*)$ by $\alpha\colon \beta_1^* = \alpha$. Applied to C this results in

$$C', \alpha\colon \beta_1^* = \alpha.$$

Now there is a paramodulation step possible between this clause and the original clause C which leads to

$$C', \neg R(\alpha, \alpha\colon \beta_2^*\colon \cdots \colon \beta_n^*).$$

For this new negative R-literal the same observations can be made as for its ancestor $\neg R(\alpha, \alpha\colon \beta_1^*\colon \cdots \colon \beta_n^*)$, i.e. there exists a clause

$$R(\alpha, \alpha\colon \beta_2^*\colon \cdots \colon \beta_n^*), \alpha\colon \beta_2^* = \alpha$$

in the background theory and therefore the clauses

$$C', \alpha\!:\!\beta_2 = \alpha$$

and

$$C', \neg R(\alpha, \alpha\!:\!\beta_3^*\!:\cdots:\!\beta_n^*)$$

are derivable. This proceeds until we reach $C', \neg R(\alpha, \alpha\!:\!\beta_n^*)$ from which only the clause $C', \alpha\!:\!\beta_n^* = \alpha$ can be derived. Hence, everything which is derivable from the E-literal in original clause C is contained in the clause $C', \alpha\!:\!\beta_i^* = \alpha$ and we are done.

LEMMA 3.9.40
Let $\mathcal{C}$ be a S4⊕KD4-unsatisfiable set of clauses. Then $\mathcal{C}$ can be refuted with the help of the S4⊕KD4 Inference System.

Proof: Let $\sigma\mathcal{C}$ be an *S4⊕KD4*-unsatisfiable ground instance of $\mathcal{C}$. Evidently, according to Lemma 3.9.35 it suffices to show that there exists a liftable derivation of a single *S4⊕KD4*-unsatisfiable ground clause. The proof is performed by induction over the number of E-literals in $\sigma\mathcal{C}$.

Base case: there are no E-literals in $\sigma\mathcal{C}$. Then we are able to derive the empty clause from the unconstraint part of $\sigma\mathcal{C}$ and that with the help of resolution, factorization and s-paramodulation. Recall that any such inference can be lifted to the unconstraint part of $\mathcal{C}$. Performing exactly the same sequence of inference steps to $\sigma\mathcal{C}$ and $\mathcal{C}$ thus results in a single unsatisfiable clause and we are done.

Induction step: let σE be an E-literal in a clause $\sigma C \in \sigma\mathcal{C}$ and let $\sigma C' = \sigma C \setminus \{\sigma E\}$ and $\sigma\mathcal{C}' = (\sigma\mathcal{C} \setminus \{\sigma C\}) \cup \{\sigma C'\}$. By the induction hypothesis we are able to derive a single *S4⊕KD4*-unsatisfiable clause σD from $\sigma\mathcal{C}'$. Performing exactly the same derivation steps on $\sigma\mathcal{C}$ therefore results in a clause $\sigma D \vee \sigma E$ and, according to the induction hypothesis, this derivation can be lifted. Now it is easy to see that we may replace the original clause σC by $\sigma D \vee \sigma E$ and remain *S4⊕KD4*-unsatisfiable since by the induction hypothesis each literal in σD is *S4⊕KD4*-unsatisfiable. At this stage we can apply Lemma 3.9.39 which guarantees that $\sigma D \vee \sigma E$ can be replaced by the clauses $\sigma D \vee \alpha\!:\!\beta_i^* = \alpha_i$ without loosing unsatisfiability and we thus end up with a clause set with fewer E-literals. The Lemma therefore follows by the induction hypothesis.

3.9.5 An Inference System for *S4F*

Recall the saturation result for the background theory of *S4F*

$$R(u, v\!:\!x_1\!:\cdots:\!x_n), R(v, w)$$

An alternative clause set with exactly the same saturation is (see Section 3.5):

$$\begin{array}{l} S(u,u) \\ S(u,v) \Rightarrow S(u, v\!:\!x) \\ S(u,v) \Rightarrow R(x,v) \vee R(u,y) \end{array}$$

How could an appropriate inference rule for *S4F* look like which covers the responsibilities of this background theory? There obviously are some similarities between the *S4F* saturation and the *S4* saturation and indeed a suitable inference rule can be found along these lines, namely:

Definition 3.9.41 (The *S4F* Inference System)
Let σ is a S4 prefix unifier of (γ, β). Then we call

$$\frac{\begin{array}{c}\neg R(\alpha,\beta),\Phi\\ \neg R(\gamma,\delta),\Psi\end{array}}{\sigma\Phi,\sigma\Psi}$$

the S4F Inference Rule. *The S4F Inference System consists of the standard resolution and factorization rules together with the S4 and the S4F Inference Rules.*

Showing the soundness of this inference system is in fact a trivial task since we know about the soundness of the *S4* inference rule (and *S4F* is stronger than *S4*) and the soundness of the *S4F* inference rule follows immediately from the correctness of the *S4F* saturation.
Remains to show that this inference system is complete.

As a matter of fact it is fortunate that we already investigated the corresponding completeness of the *S4* Inference System for we are running into similar troubles here. Again, there is no problem to show the ground completeness and again, the actual problem lies with the lifting lemma. We therefore try to incorporate the knowledge we have gained from *S4* case into the corresponding proof for *S4F*.
To this end consider two other inference rules which – for the moment – are supposed to replace the *S4F* Inference Rule, namely

$$\frac{\begin{array}{c}\neg R(\alpha,\beta),\Phi\\ \neg R(\gamma,\delta),\Psi\end{array}}{\neg S(\gamma,\beta),\Phi,\Psi} \quad \text{and} \quad \frac{\neg S(\gamma,\beta),\Psi}{\sigma\Phi}$$

where σ is an *S4* prefix unifier of (α, β).
Evidently, these two alternative inference rules are taken directly from the saturation of the alternative background theory for *S4F* and thus their correctness follows trivially. Moreover, it is also very easy now to show the completeness of these two rules (together with resolution, factorization and the *S4* Inference Rule of course). As a matter of fact, completeness follows almost immediately from the completeness of the *S4* Inference System, for, assume we have an unsatisfiable set of (negative) R-clauses[36]. Then there exists an unsatisfiable set of ground instances from these clauses which can be refuted according to the ground completeness of these inference rules. However, steps performed with the first of these two rules can trivially be lifted since they do not require any variable instantiation. This way we can perform all these liftable steps one after the other and end up with a single unsatisfiable clause (which may contain both negative R-literals and negative S-literals). This very clause can now be refuted according to Lemma 3.9.15 and we are done.

Thus we have taken a major step towards the refutation completeness of the *S4F* Inference System. The rest is fixed by the following:

Theorem 3.9.42
The S4F Inference System is sound and complete.

[36]Standard resolution and factorization steps can be lifted anyway.

Proof: Soundness is no problem at all as mentioned earlier. Remains to show completeness and that with the help of the above observations concerning the completeness of the alternative inference system. To this end it suffices to show that there is a refutation with the alternative inference system such that an application of the first of the two new rules is always immediately followed by an application of the second new inference rule. This, however, is fairly easy, for consider the first liftable application of the second new inference rule (recall that sooner or later there must be such a liftable inference step). If this application immediately follows the first application of the first inference rule both together form a liftable step of the *S4F* inference rule. Otherwise we can at least restructure the (ground) refutation such that the step immediately preceding this liftable application becomes the first rule application at all. The theorem thus follows immediately by induction over the length of the refutation.

3.10 Functional Translation

Although the search for a suitable inference rule application has already been simplified significantly one still might object that quite a lot of negative R-literals remain which may act as parent clauses for the new inference rules. We therefore try to reduce the number of possible inference steps by a further restriction (on the *S4* Inference System for convenience) and that by applying the *S4* Inference Rule only in certain cases. To this end a new inference mechanism is developed which is again based on the R-literals and which (syntactically) forbids undesired inference steps.

Evidently, it would be quite welcome if a clause set contains only a few of these R-literals, for the new inference rules can only be applied to such literals. We therefore consider again the *functional translation* as proposed in (Ohlbach 1989), (Ohlbach 1988), (Ohlbach 1991), (Auffray and Enjalbert 1992), and (Fariñas del Cerro and Herzig 1988). Please note that only the formula translation method from the above approaches is considered here; the reasoning process to be defined later differs significantly from the above methods.

DEFINITION 3.10.1 (FUNCTIONAL TRANSLATION)
Let Φ be a modal logic formula.

$$\begin{aligned} \lceil \Box\Phi \rceil_u &= \forall x\ \lceil \Phi \rceil_{u:x} \\ \lceil \Diamond\Phi \rceil_u &= \exists x\ \lceil \Phi \rceil_{u:x} \end{aligned}$$

All the other cases are treated by the usual homomorphic extension of the above. The initial call is again: $\lceil \Phi \rceil_\iota$.

This functional translation can be viewed from the semi-functional perspective as a semi-functional translation followed by resolution steps with the single unit clause $R(u, u\!:\!x)$ wherever possible and finally throwing away all the old clauses from the translated formula which did contain negative R-literals. And in fact there is no difference at all to the semi-functional translation if we just kept the conditioned equation

$$R(u,v) \Rightarrow \exists x\ u\!:\!x = v$$

in the clause set for then the original clauses can easily be reconstructed[37].

The idea is now to change the unification in a way such that the effect of the conditioned equation is taken directly into account. This will result in resolvents which carry certain residues and these residues are to be manipulated by the *S4* Inference Rule. Note that after this functional translation is performed there are no R-literals left in the given clause set.

DEFINITION 3.10.2 (DIRECT PREFIX)
The direct prefix pre(α) *of a world term* α *is the unique prefix of* α *such that every other prefix of* α *is either* α *itself or a prefix of* pre(α)*. In the sequel, whenever the* the prefix *of some world term is mentioned the direct prefix is meant.*

DEFINITION 3.10.3 (THEORY UNIFICATION)
Let $\alpha : x$ *and* β *be two world terms. The substitution*

$$\sigma = \{x/\alpha \rightsquigarrow \beta\}$$

is called the theory unifier *for* $\alpha : x$ *and* β *with residue* $\neg R(\alpha, \beta)$ *(read as:* x *leads from* α *to* β*). Applying this substitution to a clause* C *results in the clause* $C^{\alpha:x}_{\beta} \vee \neg R(\alpha, \beta)$ *which is called the conditioned instance of* C*.*

DEFINITION 3.10.4 (THEORY UNIFICATION ALGORITHM)
A unification problem *consists of a pair* (E, σ) *where* E *is a set of equations and* σ *is a substitution. Such a unification problem is said to be solved if* E *is empty and* σ *is of the form* $\{[x_i/t_i] \mid 1 \leq i \leq n\}$ *where the* x_i *are variables and the* t_i *are terms that do not contain* x_i *and are of the same sort as* x_i*. Solving a unification problem means to apply certain transformation rules of the following kind:*

decomposition
$$(\{\alpha : a = \beta : b\} \cup E, \sigma) \longrightarrow (\{a = b, \alpha = \beta\} \cup E, \sigma)$$

orientation
$$(\{\alpha = \beta : x\} \cup E, \sigma) \longrightarrow (\{\beta : x = \alpha\} \cup E, \sigma)$$

application($x \notin \beta$)
$$(\{\alpha : x = \beta\} \cup E, \sigma) \longrightarrow (E^{\alpha:x}_{\beta}, [x/\alpha \rightsquigarrow \beta] \circ \sigma)$$

occurs check($x \in \beta$)
$$(\{\alpha : x = \beta\} \cup E, \sigma) \longrightarrow \textit{failure}$$

where $\circ$ *denotes the usual composition of substitutions. The above transformation system contains only those rules that are necessary to unify world terms. Other (classical) terms are unified as usual.*

The latter definitions would not make much sense if we had not the following property of translated formulae:

LEMMA 3.10.5
The prefix of functional decomposition variables is unique.

[37] As a matter of fact, this formula translation differs slightly from the other functional translation approaches. The differences are not crucial, however.

Proof: This holds obviously after translation and is not violated by any inference step (unification) (see also (Ohlbach 1989)).

DEFINITION 3.10.6 (FUNCTIONAL INFERENCE SYSTEM)
The functional *S4* Inference System *consists of resolution, factorization, and the S4 Inference Rule, however, R-substitution and thus unification of world-terms works according to the definition 3.10.3 and 3.10.4.*

Again it has to be shown that this inference system is sound and complete. To this end the following definition is quite helpful.

DEFINITION 3.10.7 (FULL RELATIVIZATION)
Let C be a clause generated after functionally translating the modal logic formula Φ. Let $\{x_1, \ldots, x_n\}$ be the set of universally quantified functional decomposition variables occurring in C and let

$$\sigma = \{\ldots, x_i/\mathrm{pre}(x_i) \leadsto u_i, \ldots\}$$

where the u_i are new world variables. Then σC is called the full relativization *of C.*

As an example consider the modal expression $\Box\Diamond\Box P$. Its functional translation (with initial world ι) results in

$$\forall x \exists y \forall z\; P(\iota : x : y : z)$$

The full relativization of this formula is then

$$\forall u \exists y \forall v\; \neg R(\iota, u) \vee \neg R(u : y, v) \vee P(v)$$

Evidently, the full relativization is a means to obtain the semi-functional translation result from a functionally translated modal formula.

LEMMA 3.10.8
The Functional Inference System is sound.

Proof: In fact, nothing has changed but unification. It is thus enough to show the soundness of the new substitution here. To this end it suffices to show that the full relativization is an equivalence transformation, since any substitution can be viewed as a classical instantiation of the full relativization.

Recall that a modal formula is *S4*-unsatisfiable if its functional translation together with the background theory

$$\begin{array}{l} R(u,v) \Rightarrow \exists x\; u : x = v \\ R(u,u) \\ R(u, u : x) \\ R(u, u : x : y) \\ \ldots \end{array}$$

is unsatisfiable. The first and the third of these clauses in fact describe the equivalence

$$R(u,v) \Leftrightarrow \exists x\; u : x = v$$

and indeed, according to this equivalence any clause and its full relativization are equivalent.

LEMMA 3.10.9
Let C be a single S4-unsatisfiable clause. Then C can be refuted with the help of the S4 Inference Rule (under the new unification).

Proof: By induction on the number of functional decomposition variables in C.

Base Case: no variables. Done by Lemma 3.9.17.

Induction Step: By Lemma 3.9.17 we know that there is at least a step in the full relativization of C which leads nearer towards the empty clause. Now suppose that in one of the residues, say $\neg R(\alpha, u)$, u is overestimated. Since $C_u^{\alpha:x}, \neg R(\alpha, u)$ contains less variables than C it can be refuted and that with a first step not involving the literal $\neg R(\alpha, u)$. This first step can be performed in C as well (modulo the different instantiation) and we end up either with a clause with less variables (in which case the induction hypothesis can be applied) or with a clause with the same variables but less literals. In this latter case we start the whole procedure with this new clause again and this ultimately leads to the empty clause.

For the other case assume that none of those world variables is overestimated. Then the clause $C_{\mathrm{pre}(x_i)}^{\mathrm{pre}(x_i):x_i}$ is unsatisfiable and contains less (in fact none at all) variables and can thus be refuted. The first step of this refutation can analogously be applied to C as well and we thus get a new clause which is unsatisfiable and smaller than C. Therefore we finally end up with the empty clause here, too.

With this almost all necessary steps towards the completeness proof for the Functional Inference System are done.

THEOREM 3.10.10
The Functional S4 Inference System is sound and complete.

Proof: Soundness follows from Lemma 3.10.8. Completeness is shown along the lines of the completeness proof for *S4* Inference System in Lemma 3.9.17. It suffices to show that it is always possible to derive a single unsatisfiable clause which then can be refuted according to Lemma 3.10.9.

The advantage of the functional approach is that the R-literals are hidden as long as possible. This means that inference steps which were possible in the semi-functional approach are not necessarily possible anymore. Thus the search space gets reduced even further.

At this stage a comparison between this Functional *S4* Inference System and the approaches proposed by e.g. Hans Jürgen Ohlbach seems adequate. What the formula translation is concerned the difference is very small indeed. The major difference lies within the reasoning process. In Ohlbach's approach every occurrence of an R-literal in the relational translation gets replaced by a (positive or negative) equation according to the definition

$$\forall u, v\ R(u, v) \Leftrightarrow \exists x\ u : x = v$$

in particular this means that the background theories which consist solely of R-literals in the relational and in the semi-functional translation are represented by sets of equational clauses. Unfortunately, reasoning with such equations is fairly complicated and therefore an attempt was made to cast these equations into suitable unification algorithms. Note that whenever the background theory in the relational translation is Horn then the resulting theory after functional translation consists of unit equations and indeed, the background theories for most of the well known modal logics consists of Horn formulae. Up to date unification algorithms for almost any combination of the schemata D, T, B, *4* and *5* have been developed. The advantage of this approach after such unification algorithms have been found is obviously that no further special reasoning within the background theory is necessary. However, it also has some disadvantages. First, the development of suitable unification algorithms is not a trivial task at all and this may be a major reason why such algorithms have been defined for a rather limited number of modal logics (accessibility relation properties in fact). Second, even if a unification algorithm has been defined it might produce exponentially many unifiers for a single unification problem (as the *S4* case shows for instance) and thus a single resolution step might produce exponentially many resolvents.

The Functional Inference System as proposed in this work avoids such difficulties at least to some extent. Recall that the theory unification as defined in Definition 3.10.3 is unitary. Application of such a unifier, however, results in a clause which is extended by some residue, namely an R-literal. In a sense, this R-literal contains the information about further possible unifiers as they are directly computed in Ohlbach's approach. In fact, if we tried to find all possible instantiations which make the resulting residue unsatisfiable we would obtain the set of unifiers computed by Ohlbach's unification algorithm. However, we are not forced to compute all such possible instantiations. In fact, we often even postpone any possible inference step with this residue until a clause is derived which contains nothing but R-literals. Until we reach this point variables get more and more instantiated and thus reduce the number of possible unification results. It thus makes sense to think of the Functional Inference System as a functional translation approach in Ohlbach's sense, however, combined with some kind of *lazy* unification. Nevertheless, although this sounds superior to Ohlbach's approach, it also has some disadvantages for it may unify terms which are not unifiable in Ohlbach's sense. This is possible in cases as the following: consider the two literals $P(\iota : a : x)$ and $\neg P(\iota : b : y)$. According to the Functional Inference System these two literals can be made complementary, e.g. by applying the substitution $x/\iota : a \rightsquigarrow \iota : b : y$ with residue $\neg R(\iota : a, \iota : b : y)$. In Ohlbach's approach these two literals could not be made complementary for any common instance of both argument terms would have to have both $\iota : a$ and $\iota : b$ as its prefix which is impossible. This impossibility is also hidden somewhere in the Functional Inference System, namely in the residue $\neg R(\iota : a, \iota : b : y)$. Note that this residue is equivalent to the disjunction of the two literal $\neg R(\iota : a, u)$ and $\neg R(\iota : b, u)$ (according to the full relativization) and it is impossible to "solve" both literals together. Thus the Functional Inference System may produce *impossible* constraints, something which cannot happen in Ohlbach's approach. A possibility how this problem could be overcome would be to test the constraint part of a resolvent for satisfiability immediately after it has been generated. Whether or not this turns out to be useful is a matter of future examinations. For the moment, the Functional Inference System at least seems to be an interesting alternative to Ohlbach's approach.

3.11 A Word on Constraint Resolution

The way we treat negative R-literals in translated formulae suggests a constraint resolution approach in the sense of (Bürckert 1990) and (Bürckert 1991). In fact, there already exists an approach along these lines which is proposed in (Scherl 1993). The main idea behind Bürckert's constraint resolution approach is to distinguish between the actual theorem to be proved and a certain background theory for which a special inference mechanism exists. Any clause then consists of two parts: A clause body and a certain constraint. Resolution steps can only be performed between literals in the clause bodies and the aim is to derive empty clause bodies instead of empty clauses. Such empty clause bodies do not yet guarantee the unsatisfiablity of the whole clause set, however. What remains to be shown is that the collection of constraints which are associated with empty clause bodies can be instantiated such that the result of this instantiation is valid in the background theory.

This very idea forms the basis of Richard Scherl's approach on modal logic theorem proving. He considers the relational translation for modal formulae and treats negative R-literals in translated formulae as constraints which have to be solved in a suitable background theory. However, the pure relational translation turned out to be not very convenient for, as we know from Section 3.1, it also produces positive R-literals in the translation result. Based on the following observation he therefore developed a means to avoid such positive R-literals in translated formulae.

Example 3.11.1
Consider the modal (sub-)formula $P \vee \Box\Box\Diamond P$. After relational translation we get

$$P(\iota) \vee \forall u\ R(\iota, u) \Rightarrow \forall v\ R(u, v) \Rightarrow \exists w\ R(v, w) \wedge P(w)$$

and a clause form transformation then results in

$$\begin{array}{l} P(\iota) \vee \neg R(\iota, u) \vee \neg R(u, v) \vee P(f(u, v)) \\ P(\iota) \vee \neg R(\iota, u) \vee \neg R(u, v) \vee R(v, f(u, v)) \end{array}$$

Now it can be shown[38] that this can be simplied – provided R is serial – to

$$P(\iota) \vee \neg R(\iota, u) \vee \neg R(u, v) \vee P(f(u, v))$$

where the clause

$$\neg R(\iota, u) \vee \neg R(u, v) \vee R(v, f(u, v))$$

has finally to be added to the whole clause set.

This can be done for any such positive occurrence of an R-literal inside formulae if it is only guaranteed that R is serial. It is thus possible to translate modal formulae in a way such that no positive R-literals occur in the translation result, although the background theory gets extended then. Now, after the background theory can be distinguished from the translation result it is possible to perform constraint resolution in Bürckert's sense. However, the constraint theory which now consists of the accessibility relation properties of the modal logic in question and the additional theory clauses from the translation often turn out to be pretty complicated. Scherl therefore proposes to perform the satisfiablity test for constraints by using Ohlbach's functional translation and unification approach.

At this stage is might be useful to compare Scherl's method with the (semi-)functional approach proposed in this work.

[38]This result has been further developed and generalized in (Ohlbach and Weidenbach 1995).

A first observation is that Scherl's way of extracting theory clauses from the translation result of a modal formula results in unneccessarily complicated additional clauses. For instance, in the example above the extra clause could be simplified to the unit clause

$$\forall u, v\; R(v, f(u, v))$$

i.e. the constraint part for such new theory clauses can always be ignored (see also (Ohlbach and Weidenbach 1995)). But now we can observe something very interesting: the extra clauses that are generated by translating arbitrary modal formulae must be unit clauses of the form

$$\forall \overline{x}\; R(\alpha_i, f_i(\overline{x}))$$

and the function symbols f_i all have to be different because each of them has been generated by some skolemization step. This allows us to provide a characterization of the background theory: it consists of the accessibility relation properties together with a finite set of unit clauses of the form $R(\alpha_i, f_i(\overline{x}))$ where all f_i are different. Since we know now, at least to some extent, how the background theory looks like we can try to saturate this background theory.

EXAMPLE 3.11.2
Consider the background theory for *S5*, i.e. reflexivity and euclideaness of the accessibility relation together with the unit clause $R(u, f(u))$. Note that this means that the translated formula contained only one $\Diamond$-operator which was in the scope of exactly one $\Box$. Saturating this simple clause set results in all clauses of the form

$$R(f^n(u), f^m(u)) \qquad \text{for } n, m \geq 0$$

as can easily be checked.

Now, how can we exploit the connectedness assumption here? As a matter of fact, connectedness is mirrored by the assumption that any world can be represented by a suitable application sequence of the skolem functions involved. In the example above this means that for every world u there exists an i such that $u = f^i(\iota)$ since f is the only skolem function occurring in the background theory. Now, the clause schema $R(f^n(u), f^m(u))$ holds for every u, in particular for ι and since $f^n(\iota)$ and $f^m(\iota)$ both denote arbitrary worlds we end up with the universal relation for R just as expected. Note that in general we have to assume arbitrarily many such skolem functions and this makes it hard to saturate the resulting unit clauses with the *S5* accessibility relation properties. These difficulties are avoided in the semi-functional approach for there the only additional clause is $R(u, u\!:\!x)$ which covers the responsibilities for all the units $\forall \overline{x}\; R(\alpha_i, f_i(\overline{x}))$.

Summarizing, Scherl's translation approach can be viewed as a first step towards the semi-functional translation method. Unfortunately he did not realize that his extra clauses can be simplified to units. Also, he did not think of saturating his background theory which would be possible, although not very easy, even if the simplification to unit clauses would not be performed. Finally, his idea to apply Ohlbach's theory unification in order to solve remaining constraints can only work if such a theory unification algorithm exists. However, as mentioned earlier in this chapter, such algorithms do only exist for a limited number of modal logics and in particular only for modal logics whose accessibility relation properties can be represented by Horn formulae. Nevertheless, for such modal logics it definitly makes sense to apply a constraint resolution approach.

In cases where the background theory does not consist of Horn formulae it might still be useful to think about constraint resolution and that in particular in the light of the semi-functional translation approach. However, one has to be aware of the problems that might arise then.

EXAMPLE 3.11.3
Consider again the multi-modal logic *S4⊕KD4* and show the validity of the formula

$$\Box Q \Rightarrow (P \vee \Box^*(P \Rightarrow Q))$$

Transforming the translation of its negation then results in the clause set

$$\begin{array}{c} \neg R(\iota, u) \vee Q(u) \\ \neg P(\iota) \\ P(\iota : a^*) \\ \neg Q(\iota : a^*) \end{array}$$

In constraint notation we therefore have something like

$$\begin{array}{lll} Q(u) & || & R(\iota, u) \\ \neg P(\iota) & || & \emptyset \\ P(\iota : a^*) & || & \emptyset \\ \neg Q(\iota : a^*) & || & \emptyset \end{array}$$

A resolution step between the Q-literals results in

$$\emptyset \quad || \quad R(\iota, \iota : a^*)$$

This constraint is not yet valid in the *S4⊕KD4* background theory. However, no further (standard) resolution step is possible as can easily be checked. Completeness of this approach is therefore only obtained if we also consider non-standard (constraint) resolution steps between the P-literals which would result in

$$\emptyset \quad || \quad \iota : a^* = \iota$$

Now the collection of the constraints associated with the empty clause bodies which is

$$R(\iota, \iota : a^*) \vee \iota : a^* = \iota$$

is valid in the *S4⊕KD4* background theory and we are done.

This example shows that it might be necessary to perform resolution steps between literals whose arguments are not unifiable. Unfortunately, there are usually many such possibilities in a given clause set and therefore a constraint resolution approach with background theories as the one for *S4⊕KD4* turns out to be rather disappointing.

4

Back To Temporal Logic

Instant temporal logics can be viewed as multi-modal logics with operators that refer to the future, the past, or segments of the time axis. In fact, various temporal logics occur in the literature and they differ mainly in the assumptions about the topology of time. Often even logics as *S4* and *S5* are sometimes called *temporal* simply because their modalities can have a temporal reading. Unfortunately, both *S4* and *S5* are too little expressive for most purposes since they refer to only one direction of time or even only to the entire time structure. Nevertheless, there are some fairly simple modal logics which – although having just one single modality – turned out to be useful for some theories with a temporal background. So, for instance, Goldblatt showed in (Goldblatt 1980) that the two-dimensional Minkowski Space-Time can be axiomatized by *S4.2*, a logic we examined in Section 3.5. In spite of such "pathological" cases temporal logics are usually considered as multi-modal and – up to some minor exceptions – these will be the logics we examine.

The techniques that have been developed for modal logics in the last chapter are going to be used for these temporal logics now. I.e. – assuming that we have an axiomatization for the temporal logic we are interested in – we determine the background theory of this logic with respect to the semi-functional semantics, saturate this background theory, derive a finite set of clause schemata which serves as an alternative background theory and finally either look for a set of clauses whose saturation contains exactly these clause schemata or try to cast the background theory represented by these schemata into suitable inference rules.

These techniques are applied to every temporal logic we are interested in. For convenience we shall start our examinations with the simplest *Tense Logics* that appear in the literature. By adding more and more new combinations and properties these logics are extended until we finally reach the "general" temporal logic defined in Chapter 2.

4.1 Lemmon's Minimal Tense Logic K_t

The simplest logic that occurs in the temporal logic literature is E. J. Lemmon's Minimal Tense Logic K_t (see (Prior 1967) and (Prior 1968)). It is motivated by the following minimal postulates regarding the truth-conditions of tense-logical formulae (in a more contemporary syntax):

$$\begin{array}{lll}
\Im \models_x \neg\Phi & \text{iff} & \Im \not\models_x \Phi \\
\Im \models_x \Phi \Rightarrow \Psi & \text{iff} & \Im \not\models_x \Phi \vee \Im \models_x \Psi \\
\Im \models_x \langle \mathrm{F} \rangle \Phi & \text{iff} & \exists y\ \Re(x,y) \wedge \Im \models_y \Phi \\
\Im \models_x \langle \mathrm{P} \rangle \Phi & \text{iff} & \exists y\ \Re(y,x) \wedge \Im \models_y \Phi
\end{array}$$

Lemmon showed that the tense logical formulae that can be derived from these postulates, propositional calculus and quantification theory are precisely those that can be derived by substitution, Modus Ponens, some standard axiomatization for propositional logic, the rules

$$\frac{\Phi}{\boxed{\mathrm{F}}\,\Phi} \qquad\qquad \frac{\Phi}{\boxed{\mathrm{P}}\,\Phi}$$

and the additional axioms

$$\begin{array}{l}
\boxed{\mathrm{F}}(\Phi \Rightarrow \Psi) \Rightarrow (\boxed{\mathrm{F}}\,\Phi \Rightarrow \boxed{\mathrm{F}}\,\Psi) \\
\boxed{\mathrm{P}}(\Phi \Rightarrow \Psi) \Rightarrow (\boxed{\mathrm{P}}\,\Phi \Rightarrow \boxed{\mathrm{P}}\,\Psi) \\
\langle \mathrm{F} \rangle\, \boxed{\mathrm{P}}\,\Phi \Rightarrow \Phi \\
\langle \mathrm{P} \rangle\, \boxed{\mathrm{F}}\,\Phi \Rightarrow \Phi
\end{array}$$

where $\boxed{\mathrm{F}}$ and $\boxed{\mathrm{P}}$ are short for $\neg \langle \mathrm{F} \rangle \neg$ and $\neg \langle \mathrm{P} \rangle \neg$ respectively.

Evidently, this Minimal Tense Logic is of very limited expressive power. Nevertheless it shows to be a good starting point for it serves as a first step towards the examination of more complex temporal logics.

Now, having a closer look at the K_t axiomatization we immediately notice that the only axioms we are not yet familiar with are $\langle \mathrm{F} \rangle\, \boxed{\mathrm{P}}\,\Phi \Rightarrow \Phi$ and $\langle \mathrm{P} \rangle\, \boxed{\mathrm{F}}\,\Phi \Rightarrow \Phi$. The other axioms and rules just guarantee that this logic is *normal*, which means that we may consider accessibility relations in the semantics. We therefore have to determine the first-order properties that are induced by the two new axioms and we do so by applying the Elimination Theorem 3.1.14 on the two relational translation results. Let us consider the axiom $\langle \mathrm{F} \rangle\, \boxed{\mathrm{P}}\,\Phi \Rightarrow \Phi$ first. After translation its negation we get

$$\exists u, v \exists \Phi \left[\begin{array}{c} R_F(u,v) \wedge \\ \forall w\ R_P(v,w) \Rightarrow \Phi(w) \wedge \\ \neg\Phi(u) \end{array} \right]$$

where R_F denotes the accessibility relation[1] for $\boxed{\mathrm{F}}$ and R_P denotes the accessibility relation for $\boxed{\mathrm{P}}$ respectively[2].

Applying the Elimination Theorem then results in

$$\exists u, v\ R_F(u,v) \wedge \neg R_P(v,u)$$

[1] The term accessibility relation is actually used for modal logics rather than temporal logics. In a temporal interpretation they should be called *earlier-later relation* for convenience.

[2] Thus R_F represents the *earlier*-relation and R_P denotes the *later*-relation.

and the original axiom schema is therefore equivalent to

$$\forall u, v\ R_F(u,v) \Rightarrow R_P(v,u)$$

Similarly, we obtain for the other axiom $\Diamond_P \boxed{F}\Phi \Rightarrow \Phi$ the first-order property

$$\forall u, v\ R_P(u,v) \Rightarrow R_F(v,u)$$

so that in fact the one relation turns out to be the converse of the other; something that could have been expected from Lemmon's original motivation.

Our aim is now to apply the semi-functional translation approach to this logic. Note, that seriality is not assumed for the respective accessibility relations and we therefore have to consider the semi-functional translation method for non-serial modal logics here. In doing so we end up with the following translation (see section 3.4)

$$\begin{array}{lcl}
\lfloor \boxed{F}\Phi \rfloor_u & = & \forall v\ R_F(u,v) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \boxed{P}\Phi \rfloor_u & = & \forall v\ R_P(u,v) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \Diamond_F \Phi \rfloor_u & = & N_F(u) \wedge \exists x_F\ \lfloor \Phi \rfloor_{u:x_F} \\
\lfloor \Diamond_P \Phi \rfloor_u & = & N_P(u) \wedge \exists x_P\ \lfloor \Phi \rfloor_{u:x_P} \\
\multicolumn{3}{l}{\text{where the other cases remain as before}}
\end{array}$$

together with the background theory

$$\begin{array}{l}
\forall u, x_F\ N_F(u) \Rightarrow R_F(u, u\!:\!x_F) \\
\forall u, x_P\ N_P(u) \Rightarrow R_P(u, u\!:\!x_P) \\
\forall u, v\ R_F(u,v) \Rightarrow R_P(v,u) \\
\forall u, v\ R_P(u,v) \Rightarrow R_F(v,u)
\end{array}$$

Evidently, N_F (N_P) denotes the normality predicate for R_F (R_P respectively) and the variables indices are supposed to indicate to which functional decomposition the variables belong to.

Because of the strong correlation between R_F and R_P it is worthwhile to replace occurrences of the one by the other in the translation definition. We thus get the following simplification

$$\begin{array}{lcl}
\lfloor \boxed{F}\Phi \rfloor_u & = & \forall v\ R(u,v) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \boxed{P}\Phi \rfloor_u & = & \forall v\ R(v,u) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \Diamond_F \Phi \rfloor_u & = & N_F(u) \wedge \exists x_F\ \lfloor \Phi \rfloor_{u:x_F} \\
\lfloor \Diamond_P \Phi \rfloor_u & = & N_P(u) \wedge \exists x_P\ \lfloor \Phi \rfloor_{u:x_P}
\end{array}$$

for the semi-functional translation and

$$\begin{array}{l}
\forall u, x_F\ N_F(u) \Rightarrow R(u, u\!:\!x_F) \\
\forall u, x_P\ N_P(u) \Rightarrow R(u\!:\!x_P, u)
\end{array}$$

for the background theory (where R is short for R_F and R_P has been rewritten in terms of R).

This background theory is already saturated and because of its simplicity it certainly makes not very much sense to think about inference rules which cover this theory's responsibilities. We therefore stop the examination on K_t here and consider some simple but nevertheless interesting extensions to K_t.

4.2 Simple Extensions to K_t

K_t is the weakest of all the temporal logics that have modalities for the past as well as for the future since the only assumption made for K_t is that the earlier- and the later-relation are converse. In analogy to the simplest extension to the modal logic K we assume that time has neither a beginning nor an end [3].

4.2.1 Adding Seriality to K_t

Interestingly, such an additional assumption simplifies the semi-functional translation and the corresponding background theory even further, because it it makes it possible to ignore the *normality*-predicate. The semi-functional translation for the logic K_tD is thus given by[4]

$$\begin{array}{rcl}
\lfloor \boxed{F}\, \Phi \rfloor_u & = & \forall v\ R(u,v) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \boxed{P}\, \Phi \rfloor_u & = & \forall v\ R(v,u) \Rightarrow \lfloor \Phi \rfloor_v \\
\lfloor \Diamond_F\, \Phi \rfloor_u & = & \exists x_F\ \lfloor \Phi \rfloor_{u:x_F} \\
\lfloor \Diamond_P\, \Phi \rfloor_u & = & \exists x_P\ \lfloor \Phi \rfloor_{u:x_P}
\end{array}$$

and the background theory simplifies to

$$\begin{array}{l}
\forall u, x_F\ R(u, u\!:\!x_F) \\
\forall u, x_P\ R(u\!:\!x_P, u)
\end{array}$$

4.2.2 Adding Transitivity to K_tD

As a further extension to the K_t we consider now the additional axiom schemata

$$\boxed{F}\,\Phi \Rightarrow \boxed{F}\,\boxed{F}\,\Phi \qquad \text{and} \qquad \boxed{P}\,\Phi \Rightarrow \boxed{P}\,\boxed{P}\,\Phi$$

which express transitivity of the Earlier- and of the Later-relation respectively. Actually, only one of these two axioms is necessary since – under the assumption that the Earlier-relation is converse to the Later-relation – transitivity of the one implies transitivity of the other.

We thus get as the background theory for K_t augmented by seriality and transitivity (called K_tD4 in the sequel[5])

$$\begin{array}{l}
R(u, u\!:\!x_F) \\
R(u\!:\!x_P, u) \\
R(u,v) \wedge R(v,w) \Rightarrow R(u,w)
\end{array}$$

The formula translation does not change at all compared to the definition above and we therefore know – provided the theorem to be proved is in negation normal form – that there are no positive occurrences of an R-literal in the translation result. Hence these three clauses describe *all* we know about the positive R-occurrences and we apply the saturation technique to this background theory.

[3] Such an additional assumption might cause problems if there is a wish for representing something like a "Big Bang". For most applications, however, these assumptions make sense.

[4] This name has been chosen for uniformity. As usual, serial versions of a modal logic are named by an additional D.

[5] As commonly used in the modal logic literature D stands for seriality and *4* stands for transitivity.

LEMMA 4.2.1
The saturation of the background theory for K_tD4 consists of all unit clauses of the form

$$R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m) \qquad \textit{with } n+m \geq 1$$

where the x_i range over the functional decomposition of R_P and the y_j range over the functional decomposition of R_F

Proof: We first have to show that each of these units indeed belongs to the saturation and we do so by induction over $n+m$.

Base Case: $n+m=1$.

Then the units under consideration are just $R(u, u\!:\!x_F)$ and $R(u\!:\!x_P, u)$ which obviously are contained in the saturation since they already belong to the background theory.

Induction Step: suppose we have already shown that $R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m)$ belongs to the saturation for all n, m with $n+m<k$. We now have to show that all of these units with $n+m=k$ are also contained in the saturation. We consider three different cases:

Case 1: $n=0$ and $m=k>1$. By induction hypothesis we know that $R(u, u\!:\!x_F)$ and $R(u, u\!:\!y_1\!:\cdots:\!y_{k-1})$ both belong to the saturation. Two simple resolution steps with the transitivity clause then result in the derivation of $R(u, u\!:\!y_1\!:\cdots:\!y_k)$.

Case 2: $n=k>1$ and $m=0$. Analogous.

Case 3: $n, m>0$ and $n+m=k$. By the induction hypothesis we know that both unit clause schemata $R(u, u\!:\!y_1\!:\cdots:\!y_m)$ and $R(u\!:\!x_1\!:\cdots:\!x_n, u)$ belong to the saturation. The desired result can be derived by two resolution steps between these units and the transitivity clause.

Now we have to show that no element of the saturation has been forgotten, i.e. we have to show that this is *all* we gain by saturation. To this end we check whether the alledged saturation already contains all pure-R-positive clauses that can be derived by resolving elements of this set of unit clauses with the transitivity clause. This, however, is just as simple. Consider one of these units, say $R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m)$, and perform a resolution step with the transitivity clause (e.g. its first literal). This results in $\neg R(u\!:\!y_1\!:\cdots:\!y_m, w) \vee R(u\!:\!x_1\!:\cdots:\!x_n, w)$ and a further resolution step with, say $R(u\!:\!x_1\!:\cdots:\!x_i, u\!:\!y_1\!:\cdots:\!y_j)$ (which can only be applied if either $m=0$ or $i=0$) then leads to either $R(u\!:\!x_1\!:\cdots:\!x_{i+n}, u\!:\!y_1\!:\cdots:\!y_j)$ or $R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_{m+j})$ depending on whether $m=0$ or $i=0$. In any case the result is already contained in the alledged saturation and we are done.

Knowing about a saturation is not yet enough; we have to find out whether there is a suitable alternative clause set with the same saturation or even a new inference rule which does a similar job[6].

[6]It can also be imagined to incorporate this saturation set in a suitable constraint satisfaction algorithm. For this logic K_tD4 this would mean to allow resolution steps only between literals which are *not* R-literals and to check whether each of the R-literals of the respective resolvents could be simultaneously brought into the form $\neg R(\alpha\!:\!\beta_1\!:\cdots:\!\beta_n, \alpha\!:\!\gamma_1\!:\cdots:\!\gamma_m)$ (where the β_i belong to the functional decomposition of R_P and the γ_j range over the functional decomposition of R_F) by some suitable instantiation. This works in principle but is not further considered here, for it does not quite fit into the treatment of some of the logics examined later.

LEMMA 4.2.2
The saturation of the clause set

$$\begin{array}{l} R(u, u\!:\!x_F) \\ R(u\!:\!x_P, u) \\ R(u,v) \Rightarrow R(u, v\!:\!x_F) \\ R(u,v) \Rightarrow R(u\!:\!x_P, u) \end{array}$$

is identical to the saturation of the K_tD4 background theory.

Proof: Obviously, the first and the third clause suffice to derive any unit clause schema of the form $R(u, u\!:\!y_1\!:\cdots:\!y_m)$ where $m \geq 1$ and the y_j range over the functional decomposition of R_F. In order to show that any unit clause of the form $R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m)$ can be derived we only have to perform n successive resolution steps with the fourth clause and we are done.

Showing that the saturation of the background theory for K_tD4 is all we can get is just as simple. It suffices to realize that any resolvent between a unit of the background theory and either clause from above results in a clause which is already contained in the saturation.

Whether or not the original background theory for K_tD4, i.e.

$$\begin{array}{l} R(u, u\!:\!x_F) \\ R(u\!:\!x_P, u) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \end{array}$$

should really be replaced by

$$\begin{array}{l} R(u, u\!:\!x_F) \\ R(u\!:\!x_P, u) \\ R(u,v) \Rightarrow R(u, v\!:\!x_F) \\ R(u,v) \Rightarrow R(u\!:\!x_P, u) \end{array}$$

is more or less a matter of taste. In theorem provers in which the transitivity clause does not receive any special treatment[7] the alternative theory usually behaves slightly better.

Now let us, instead of working with such an alternative background theory, try to find a suitable K_tD4 Inference System which can replace the background theory. This will be done in lines of section 3.9 where appropriate inference rules had been found for *S4*, *KD4*, Mutual Belief and *S4F*. The techniques developed there will find their counterpart in the application to K_tD4, although some slight changes are necessary. The newly to be defined inference rule for K_tD4 looks very similar to the *S4* Inference Rule; it is only the unification that changed.

DEFINITION 4.2.3 (THE K_tD4 INFERENCE RULE)
The rule

$$\frac{\neg R(\alpha\!:\!\gamma, \beta\!:\!\delta), C}{\sigma C}$$

$$\begin{array}{c} \textit{where } \sigma = mgu(\alpha, \beta), \\ \gamma \in (F_{R_P})^*,\ \delta \in (F_{R_F})^*, \\ \textit{and either } \gamma \neq \emptyset \textit{ or } \delta \neq \emptyset \end{array}$$

[7] see e.g. (Bachmair and Ganzinger 1994b)

is called the K_tD4 Inference Rule.

DEFINITION 4.2.4 (THE K_tD4 ELIMINATION RULES)
The K_tD4 Elimination Rules are defined in accordance with the definition 3.9.21, the KD4 Elimination Rule. Note that we need two such elimination rules for K_tD4, one for its future and one for its past fragment.

$$\frac{\neg R(\alpha, u), C}{C^u_{\alpha:x_F}} \quad \text{with } u \notin \alpha \qquad\qquad \frac{\neg R(u, \alpha), C}{C^u_{\alpha:x_P}} \quad \text{with } u \notin \alpha$$

where $u \notin \alpha$ is short for: there is no occurrence of u in α.

DEFINITION 4.2.5 (THE K_tD4 INFERENCE SYSTEM)
The K_tD4 Inference System *consists of the (classical) resolution and factorization rules together with the K_tD4 Inference Rule and the K_tD4 Elimination Rules.*

We now have to show that this inference system is both sound and refutation complete.

LEMMA 4.2.6
The K_tD4 Inference System is sound.

Proof: Soundness of the K_tD4 Inference Rule follows easily from the fact that the application of a K_tD4 prefix unifier for (α, β) to the literal $\neg R(\alpha, \beta)$ results in a K_tD4-unsatisfiable instance. The resolution and the factorization rule remain as in classical first-order predicate logic and thus both are evidently sound. What the K_tD4 Elimination Rules is concerned first note that instantiating u with $\alpha : x_F$ in a clause $\neg R(\alpha, u), C$ is sound and results in

$$\neg R(\alpha, \alpha : x_F), C^u_{\alpha:x_F}$$

Obviously, the literal $\neg R(\alpha, \alpha : x_F)$ is K_tD4-unsatisfiable and therefore this instance is equivalent to $C^u_{\alpha:x_F}$.

Proving the refutation completeness of the K_tD4 Inference System is again in the lines of Section 3.9.2, i.e. we show that a single unsatisfiable clause can be derived from an arbitrary unsatisfiable clause set and also that any unsatisfiable clause can be refuted.

Again we are faced with the problem that it is not always possible to lift a ground proof to the general level as exemplified by the following example.
Consider the clause $C = \neg R(u : f_P(u), \iota : a_F : b_F), C'$ and its ground instance (with $\sigma(u) = \iota : a_F : c_P$) $\sigma C = \neg R(\iota : a_F : c_P : f_P(\iota : a_F : c_P), \iota : a_F : b_F), \sigma C'$. Evidently, the K_tD4 Inference Rule can easily be applied to σC and this results in $\sigma C'$ then. However, the only corresponding steps that are possible on the non-ground level result in either of the clauses $C'[u/\iota : a_F : b_F]$ or $C'[u/\iota : a_F]$ or $C'[u/\iota]$, i.e. in resolvents which are *not* more general than $\sigma C'$. However, just as in the modal logic case this problem can be overcome by assuming *minimal* substitutions.

LEMMA 4.2.7
Let C be a single K_tD4-unsatisfiable clause. Then C can be refuted with the help of the K_tD4 Inference System.

Proof: Let σ be a minimal ground substitution for C. Evidently, each literal in σC can be deleted with the help of the *K_tD4* Inference Rule. Now suppose that none of these possible inference steps can be lifted to C. Then each literal in C is either of the form $\neg R(\alpha_i, u_i : \beta_i)$ or of the form $\neg R(u_i : \alpha_i, \beta_i)$ where u_i is overestimated w.r.t. σ. If each of these u_i on a right-hand (left-hand) argument would also occur in the relevant prefix of a left-hand (right-hand) argument then there were infinitly many instant variables[8] and this is impossible. Hence, there exists at least one u which occurs only on right-hand (left-hand) arguments. We thus have that (the case where u occurs only on left-hand sides is symmetric)

$$C = \neg R(\alpha_1, u : \beta_1), \ldots, \neg R(\alpha_n, u : \beta_n), C'$$

where u does not occur in the relevant prefixes of any α_i or any term in C'. Now consider σC. Since the σ-instance of each of these R-literals is *K_tD4*-unsatisfiable we have that

$$\sigma(\neg R(\alpha_i, u : \beta_i)) = \neg R(a_i : \gamma_i, a_i : \delta_i : \sigma(\beta_i))$$

where the a_i represents the common prefix of the respective left and right-hand sides. Now, each of these a_i is a proper prefix of $\sigma(u)$ and therefore the a_i can be ordered. This means that one of them, say a_1, is maximal in the sense that every a_i is a prefix of a_1. Since σ is minimal by assumption we can conclude that δ_1 consists of only one F_{R_F}-term and also that $\sigma(\beta_1)$ (and thus β_1) and γ_1 are of zero length for otherwise instantiating u by a_1 instead of $a_1 : \delta_1$ would be sufficient for the unsatisfiability of C (note that a_1 is maximal). Hence we have that the literal $\neg R(\alpha_1, u : \beta_1)$ is actually $\neg R(\alpha_1, u)$ and that $\sigma\alpha_1 = a_1$ and $\sigma(u) = a_1 : \gamma$ where γ is a one-element sequence of F_{R_F}-terms. We can therefore apply the first of the *K_tD4* Elimination Rules on the literal $\neg R(\alpha_1, u)$ and this results in a resolvent $\hat{C}$ such that $\sigma\hat{C} = \sigma(C \setminus \{\neg R(\alpha_1, u)\})$. As it turns out, although the application of the *K_tD4* Inference Rule to $\neg R(\sigma(\alpha_1), \sigma(u)$ cannot be lifted to $\neg R(\alpha_1, u)$, there is an application of a *K_tD4* Elimination Rule on the non-ground level with exactly the same effect.

LEMMA 4.2.8

The K_tD4 Inference System is refutation complete.

Proof: According to Lemma 4.2.7 it suffices to show that it is always possible to derive a single *K_tD4*-unsatisfiable clause from the translation of an arbitrary *K_tD4*-unsatisfiable formula Φ. This, however, is almost trivial, for consider an arbitary *K_tD4*-unsatisfiable ground instance $\sigma\mathcal{C}$ of $\mathcal{C}$, the clause set obtained after translating Φ (for convenience we assume that no proper subset of $\sigma\mathcal{C}$ is unsatisfiable). Evidently, the unconstraint part of $\sigma\mathcal{C}$ does not contain the empty clause[9] but nevertheless is *K_tD4*-unsatisfiable and can be refuted by resolution and factorization. These steps can be lifted and therefore the very same sequence of inference steps can be applied to $\mathcal{C}$ as well. This results in a pure-R-negative clause which must be *K_tD4*-unsatisfiable and we are done by Lemma 4.2.7.

[8]See the corresponding proof for the *S4* Inference System. Note that the term *instant variable* is used in temporal logics just as the term *world variable* has been used in modal logics.

[9]It can easily be checked that the translation definition guarantees that no pure-R-negative clause is generated.

4.2.2.1 Summarizing the calculus for K_tD4

The temporal logic K_tD4 is axiomatized by a usual axiomatization of classical propositional logic plus the K-axioms for the two operators $\boxed{\mathrm{F}}$ and $\boxed{\mathrm{P}}$, i.e.

$$\begin{array}{lcl} \boxed{\mathrm{F}}(\Phi \Rightarrow \Psi) & \Rightarrow & (\boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\Psi) \\ \boxed{\mathrm{P}}(\Phi \Rightarrow \Psi) & \Rightarrow & (\boxed{\mathrm{P}}\Phi \Rightarrow \boxed{\mathrm{P}}\Psi), \end{array}$$

the necessitation rules for $\boxed{\mathrm{F}}$ and $\boxed{\mathrm{P}}$, i.e.

$$\frac{\Phi}{\boxed{\mathrm{F}}\Phi} \qquad \frac{\Phi}{\boxed{\mathrm{P}}\Phi}$$

and, in additon, the axiom schemata

$$\begin{array}{l} \Phi \Rightarrow \boxed{\mathrm{F}}\langle\mathrm{P}\rangle\Phi \\ \Phi \Rightarrow \boxed{\mathrm{P}}\langle\mathrm{F}\rangle\Phi \\ \boxed{\mathrm{F}}\Phi \Rightarrow \langle\mathrm{F}\rangle\Phi \\ \boxed{\mathrm{P}}\Phi \Rightarrow \langle\mathrm{P}\rangle\Phi \\ \boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\boxed{\mathrm{F}}\Phi \\ \boxed{\mathrm{P}}\Phi \Rightarrow \boxed{\mathrm{P}}\boxed{\mathrm{P}}\Phi \end{array}$$

The semi-functional translation for a first-order constant domain temporal logic formula in negation normal form is given by

$$\begin{array}{lcl} \lfloor P(\ldots, t_i, \ldots)\rfloor_u & = & P(u, \ldots, \lfloor t_i\rfloor_u, \ldots) \\ \lfloor \neg\Phi\rfloor_u & = & \neg\lfloor\Phi\rfloor_u \\ \lfloor \Phi \wedge \Psi\rfloor_u & = & \lfloor\Phi\rfloor_u \wedge \lfloor\Psi\rfloor_u \\ \lfloor \boxed{\mathrm{F}}\Phi\rfloor_u & = & \forall v\ R(u,v) \Rightarrow \lfloor\Phi\rfloor_v \\ \lfloor \boxed{\mathrm{P}}\Phi\rfloor_u & = & \forall v\ R(v,u) \Rightarrow \lfloor\Phi\rfloor_v \\ \lfloor \langle\mathrm{F}\rangle\Phi\rfloor_u & = & \exists x_F\ \lfloor\Phi\rfloor_{u:x_F} \\ \lfloor \langle\mathrm{P}\rangle\Phi\rfloor_u & = & \exists x_P\ \lfloor\Phi\rfloor_{u:x_P} \\ \lfloor \forall x\ \Phi\rfloor_u & = & \forall x\ \lfloor\Phi\rfloor_u \\ \lfloor \exists x\ \Phi\rfloor_u & = & \exists x\ \lfloor\Phi\rfloor_u \end{array}$$

Showing the (first-order constant domain) K_tD4-validity of a modal formula Φ then means to transform $\neg\Phi$ into its negation normal form, say Ψ, and to refute the clause normal form of $\lfloor\Psi\rfloor_\iota$ with the help of the K_tD4 Inference System.

4.2.3 Adding Reflexivity to K_tD4

In K_tD4 the $\boxed{\mathrm{F}}$-operator is to be interpreted as: *it will be the case in all future times*, i.e. the present is not necessarily included. Often, however, one would rather prefer the interpretation "henceforth" instead which indeed includes the present. As proposed in the introductory Chapter 2 on temporal logic syntax and semantics this means that the operators $\boxed{\mathrm{F}}_r$ and $\boxed{\mathrm{P}}_r$ are to be chosen where the little subscript indicates that the reflexive closure of the Earlier- and the Later-relation are to be considered. In doing so, we have to add the axiom schemata

$$\begin{array}{l} \boxed{\mathrm{F}}_r\Phi \Rightarrow \Phi \\ \boxed{\mathrm{P}}_r\Phi \Rightarrow \Phi \end{array}$$

and leave the rest of the axiomatization as for K_tD4[10]. Let us call the resulting logic K_tT4, where the T now indicates that the seriality assumption has been replaced by the stronger reflexivity property[11].

For K_tT4 we now proceed as we did for K_tD4, i.e. we determine the saturation for the K_tT4 background theory which is given by

$$\begin{array}{l} R(u,u) \\ R(u,u\!:\!x_F) \\ R(u\!:\!x_P,u) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \end{array}$$

and try both, to find an alternative (and hopefully simpler) background theory and to cast this background theory into a suitable inference rule.

LEMMA 4.2.9
The saturation of the K_tT4 background theory consists of all unit clauses of the form

$$R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m)$$

with $n,m \geq 0$, the x_i range over the functional decomposition of R_P and the y_j range over the functional decomposition of R_F.

Proof: Analogous to the proof for Lemma 4.2.1.

Because of the strong similarities between the saturations for the background theories of K_tD4 and K_tT4 there is no problem at all in finding an alternative background theory for K_tT4.

LEMMA 4.2.10
The saturation of the clauses

$$\begin{array}{l} R(u,u) \\ R(u,v) \Rightarrow R(u,v\!:\!x_F) \\ R(u,v) \Rightarrow R(u\!:\!x_P,u) \end{array}$$

is identical to the saturation of the K_tT4 background theory.

Even the search for a suitable K_tT4 Inference Rule leaves no particular diffculties.

DEFINITION 4.2.11 (K_tT4 PREFIX UNIFIER)
Let (s,t) be an arbitrary pair of world terms such that there exists a future prefix s' of s and a past prefix t' of t. Then any unifier σ for s' and t' is called a K_tT4 prefix unifier *for the ordered pair (s,t).*

Note that the only difference between K_tT4 prefix unifiers and K_tD4 prefix unifiers is that the former does not require that at least one of the two prefixes has to be a proper prefix.

[10]where each temporal operator gets the subscript "r". Note that these two additional axioms subsume the seriality axioms which are therefore superfluous. Also note that actually only one of these two axioms is needed because – due to the fact that the Earlier-relation and the Later-relation are convers – the reflexivity of the one immediately implies the reflexivity of the other.

[11]Evidently, K_tT4 thus consists essentially of two modal $S4$ fragments together with the mixing axioms that guarantee that the one accessibility relation is the converse of the other.

DEFINITION 4.2.12 (THE K_tT4 INFERENCE RULE AND SYSTEM)
The rule

$$\frac{\neg R(\alpha,\beta), C}{\sigma C}$$

where σ is a K_tT4 prefix unifier for (α,β)
is called the K_tT4 Inference Rule.
The inference system which consists of the classical resolution and factorization rules together with the K_tT4 Inference Rule is called the K_tT4 Inference System.

Soundness and refutation completeness is now proved in analogy to the corresponding proofs for *S4* and the K_tD4 Inference System.

THEOREM 4.2.13
The K_tT4 Inference System is sound and refutation complete.

Proof: Soundness of the K_tT4 Inference Rule follows immediately from the fact that an application of a K_tT4 prefix unifier for a pair (α,β) to the literal $\neg R(\alpha,\beta)$ results in a literal which is complementary to one of the literals in the saturation of the K_tT4 background theory.

The refutation completeness proof is again split into two main parts. It has to be shown that

- a single K_tT4-unsatisfiable clause can be derived from an arbitrary K_tT4-unsatisfiable clause set
- such an unsatisfiable clause can be refuted

For the first part consider an arbitrary unsatisfiable ground instance of the clause set. Evidently, the unconstraint part of these ground clauses are unsatisfiable and, since unconstraint clauses do not contain any R-literals, can be refuted by resolution and factorization. This refutation can be lifted and the very same sequence of inference steps applied to the original clause set results in a single unsatisfiable clause. Remains to be shown that this resulting clause, say C, can be refuted. To this end consider a minimal unsatisfiable ground instance σC of C. Recall that the reflexivity of R guarantees that every instant variable[12] must occur somewhere not overestimated for otherwise the ground instance could not be minimal. Evidently, σC can easily be refuted by applying the K_tT4 Inference Rule to each of its literals in arbitrary order. Unfortunately, such a refutation cannot necessarily be lifted. What can be shown, however, is that an order on the literals exists such that the refutation of σC in this particular order can be lifted. This order is obtained by finding at least one *processable* literal, i.e. a literal for which the application of the K_tT4 Inference Rule on its ground instance can be lifted. Such a processable literal must exist if the ground instantiation is minimal for otherwise we would be able to find for each variable u which occurs somewhere overestimated another variable v (also somewhere overestimated) such that $\sigma(u)$ is a proper subterm of $\sigma(v)$. This, however, is impossible for there are only finitly many variables at all. Hence, there exists at least one processable literal in C and we are done by induction over the length of C.

[12]Instant variables are variables that range over the set of time instants just like – in a modal setting – world variables range over the set of worlds.

4.3 Mixing K_tD4 and K_tT4

The linguistic motivation for K_tD4 lies in the $\langle F \rangle$ and the $\langle P \rangle$-operators which are to be interpreted as "it will be" and "it was" respectively. K_tT4 is motivated by the two operators $\boxed{F}_r$ and $\boxed{P}_r$ which are to be read as "henceforth" and "hitherto". The respective dual operators in these two logics are of no real importance at least from a linguistics point of view. For instance, the $\boxed{F}$-operator in K_tD4 is to be read as: "in all (proper) future times" which rarely occurs in everyday natural language usage since the present is usually not excluded in such constructs. Neither does the $\langle P \rangle_r$ in K_tT4 have an obvious equivalent in natural language usage, since a term like "it was the case that P" should not hold simply because P is true *now*, even if it never was before.

Therefore one is actually interested in the two operators $\langle F \rangle$ and $\boxed{F}_r$ (and also $\langle P \rangle$ and $\boxed{P}_r$ for the past mirror images[13]) which, unfortunately, are *not* duals.

There are two possible solutions to this problem: the first (and technically simpler one) is to "translate" each $\boxed{F}_r$ and each $\langle F \rangle_r$ (and similarly for the mirror operators) in terms of $\boxed{F}$ and each $\langle F \rangle$ by extending the formula translation $\lfloor\ \rfloor$ accordingly:

$$\begin{array}{rcl} \lfloor \boxed{F}_r \Phi \rfloor_u & = & \lfloor \boxed{F} \Phi \rfloor_u \wedge \lfloor \Phi \rfloor_u \\ \lfloor \langle F \rangle_r \Phi \rfloor_u & = & \lfloor \langle F \rangle \Phi \rfloor_u \vee \lfloor \Phi \rfloor_u \end{array}$$

The advantage of this extension is obvious: we remain inside K_tD4 and we do not have to worry about any further interrelations.

Nevertheless, this translation extension also has an obvious disadvantage: the clause set obtained after clause form transformation gets rather big and that in the number but also in the size of clauses. It therefore makes sense to think about alternatives and one such alternative can be found along the lines of Section 3.9.4, i.e. all temporal operators are kept as primitives which means that no operator is translated in terms of another. Evidently, this requires to axiomatize the interrelation between $\boxed{F}_r$ and $\boxed{F}$ (and between $\boxed{P}_r$ and $\boxed{P}$ respecively) and that by adding the axiom schemata $\boxed{F}_r \Phi \Leftrightarrow \boxed{F} \Phi \wedge \Phi$ and $\boxed{P}_r \Phi \Leftrightarrow \boxed{P} \Phi \wedge \Phi$.

The effect of this method is that the formula translation remains as simple as it was before (for both K_tD4 and K_tT4) but there is a price to be paid for that: first-order properties induced by the above two equivalences are to be added to the background theory of the resulting combined logic.

Let us have a look at the axiomatization of this combined logic (which will be called $K_tD4 \oplus K_tT4$ in the sequel). It consists of all classical propositional tautologies together with the axiom schemata

$$\begin{array}{l} \boxed{F}_r \Phi \Rightarrow \Phi \\ \boxed{F} \Phi \Rightarrow \langle F \rangle \Phi \\ \boxed{F} \Phi \Rightarrow \boxed{F}\,\boxed{F} \Phi \\ \boxed{F}_r \Phi \Rightarrow \boxed{F}_r \boxed{F}_r \Phi \\ \Phi \Rightarrow \boxed{F} \langle P \rangle \Phi \\ \Phi \Rightarrow \boxed{F}_r \langle P \rangle_r \Phi \\ \boxed{F}_r \Phi \Leftrightarrow \boxed{F} \Phi \wedge \Phi \end{array}$$

plus their respective mirror images.

[13]By a mirror image of an axiom schema we understand the result of replacing all future operators by the corresponding past operators and vice versa.

Moreover, there are Modus Ponens and the usual Necessitation rules

$$\frac{\Phi, \Phi \Rightarrow \Psi}{\Psi}, \quad \frac{\Phi}{\boxed{\mathrm{F}}_r\Phi}, \quad \frac{\Phi}{\boxed{\mathrm{F}}\Phi}, \quad \frac{\Phi}{\boxed{\mathrm{P}}_r\Phi}, \quad \frac{\Phi}{\boxed{\mathrm{P}}\Phi}$$

As a matter of fact we are not really interested in such Hilbert style axiomatizations. We rather examine the background theory that is induced by these schemata (after semi-functional translation).

$$\begin{array}{l}
R(u, u\!:\!x_F) \\
R(u\!:\!x_P, u) \\
R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\
R^*(u,u) \\
R^*(u, u\!:\!x_F^*) \\
R^*(u\!:\!x_P^*, u) \\
R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) \\
R^*(u,v) \Rightarrow R(u,v) \vee u = v \\
R(u,v) \Rightarrow R^*(u,v)
\end{array}$$

where it has already been taken into account that $R_F(u,v)$ is identical to $R_P(v,u)$ (and similarly for R_F^* and R_P^*). Evidently, the "starred" symbols are those which stem from the subscribed operators. Actually, the transitivity of R^* is not really necessary here because it follows already from the other clauses (the reflexive closure of a transitive relation is itself transitive).

Not too surprising, this background theory is even more complicated than the background theories we considered so far and working directly with it seems hopeless. We therefore determine its saturation, i.e. the set of pure-R-positive clauses that are derivable from the theory (modulo subsumption).

Lemma 4.3.1

The saturation of the background theory for $K_tD4 \oplus K_tT4$ (modulo subsumption) consists of all clauses of the form

$$\begin{array}{ll}
R^*(u\!:\!\overline{x_P}, u\!:\!\overline{y_F}) & \\
R(u\!:\!\overline{x_P}, u\!:\!\overline{y_F}) & \\
& \text{where either } \overline{x_P} \notin (F_{R_P^*})^* \text{ or } \overline{y_F} \notin (F_{R_F^*})^* \\
R(u\!:\!\overline{x_P}\!:\!\overline{z_P}, u\!:\!\overline{y_F}\!:\!\overline{w_F}) \vee u\!:\!\overline{x_P} = u\!:\!\overline{y_F} & \\
& \text{where } \overline{x_P}, \overline{z_P} \in (F_{R_P^*})^* \text{ and } \overline{y_F}, \overline{w_F} \in (F_{R_F^*})^*
\end{array}$$

where $\overline{x_P}, \overline{z_P} \in (F_{R_P} \cup F_{R_P^*})^*$ and $\overline{y_F}, \overline{w_F} \in (F_{R_F} \cup F_{R_F^*})^*$.

Proof: First we have to show that all instances of the above clause schemata can be derived from the background theory. To this end note that from $R^*(u,v)$ it follows that $R^*(u, v\!:\!x_F^*)$ as well as $R^*(u\!:\!x_P^*, v)$, $R^*(u, v\!:\!x_F)$, and $R^*(u\!:\!x_P, v)$ (this can easily be checked either by hand or with the help of a standard theorem prover). We thus have

$$\left.\begin{array}{l}
R^*(u,u) \\
R^*(u,v) \Rightarrow \begin{cases} R^*(u, v\!:\!x_F^*) \\ R^*(u\!:\!x_P^*, v) \\ R^*(u, v\!:\!x_F) \\ R^*(u\!:\!x_P, v) \end{cases}
\end{array}\right\} \quad \Rightarrow \quad R^*(u\!:\!\overline{x_P}, u\!:\!\overline{y_F})$$

which covers the first clause schema.

For the second schema note that

$$\left.\begin{array}{l} R^*(u:\overline{x_P}, u:\overline{y_F}) \\ R^*(u,v) \Rightarrow \left\{\begin{array}{l} R(u,v:x_F) \\ R(u:x_P,v) \end{array}\right. \end{array}\right\} \Rightarrow \left\{\begin{array}{l} R(u:\overline{x_P}, u:\overline{y_F}:z_F) \\ R(u:\overline{x_P}:z_P, u:\overline{y_F}) \end{array}\right.$$

hence

$$\left.\begin{array}{l} R(u:\overline{x_P}, u:\overline{y_F}:z_F) \\ R(u:\overline{x_P}:z_P, u:\overline{y_F}) \\ R(u,v) \Rightarrow \left\{\begin{array}{l} R(u,v:x_F^*) \\ R(u:x_P^*,v) \\ R(u,v:x_F) \\ R(u:x_P,v) \end{array}\right. \end{array}\right\} \Rightarrow \begin{array}{l} R(u:\overline{x_P}, u:\overline{y_F}) \\ \qquad \text{where } \overline{x_P} \notin (F_{R_P^*})^* \text{ or } \overline{y_F} \notin (F_{R_F^*})^* \end{array}$$

Remains the third schema. First note that $R^*(u:\overline{x_P}, u:\overline{y_F})$ holds and that in particular for the case $\overline{x_P} \in (F_{R_P^*})^*$ and $\overline{y_F} \in (F_{R_F^*})^*$. Together with $R^*(u,v) \Rightarrow R(u,v) \vee u = v$ we thus have

$$\left.\begin{array}{l} R^*(u:\overline{x_P}, u:\overline{y_F}) \\ \qquad \text{where } \overline{x_P} \in (F_{R_P^*})^* \text{ and } \overline{y_F} \in (F_{R_F^*})^* \\ R(u,v) \Rightarrow \left\{\begin{array}{l} R(u,v:x_F^*) \\ R(u:x_P^*,v) \end{array}\right. \end{array}\right\} \Rightarrow \begin{array}{l} R(u:\overline{x_P}:\overline{z_P}, u:\overline{y_F}:\overline{w_F}) \vee u:\overline{x_P} = u:\overline{y_F} \\ \qquad \text{where } \overline{x_P}, \overline{z_P} \in (F_{R_P^*})^* \\ \qquad \text{and } \overline{y_F}, \overline{w_F} \in (F_{R_F^*})^* \end{array}$$

and we are done with the third clause schema as well.

Next we have to show that this is all that can be derived (as pure-$R^{(*)}$-positive). To this end we have to check whether the pure-$R^{(*)}$-positive clauses in the background theory are instances of the clause schemata (which is trivial) and whether all pure-$R^{(*)}$-positive clauses that are derivable from the clause schemata and the clauses

$$\begin{array}{l} R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) \\ R^*(u,v) \Rightarrow R(u,v) \vee u = v \\ R(u,v) \Rightarrow R^*(u,v) \end{array}$$

are already members of the clause schemata as well. This is just as easy for the second, third and fourth clause above. The only non-trivial case consists of the transitivity of R together with the third clause schema. Therefore consider a resolution step between the first literal of the transitivity clause and the third clause schema. This results in

$$u:\overline{y_F} = u:\overline{x_P} \vee \neg R(u:\overline{y_F}:\overline{w_F}, w) \vee R(u:\overline{x_P}:\overline{z_P}, w)$$

A further resolution step is only possible if either the $\overline{y_F}$ and the $\overline{w_F}$ in the above clause are both empty or the $\overline{x_P}$ and the $\overline{z_P}$ in the clause schema are both empty. In the first case we get

$$u:\overline{x_P}:\overline{z_P} = u:\overline{x_P}:\overline{z_P}:\overline{x_P}' \vee u:\overline{x_P} = u:\overline{y_F} \vee R(u:\overline{x_P}:\overline{z_P}:\overline{x_P}':\overline{z_P}', u:\overline{y_F}:\overline{w_F})$$

and in the second case we end up with

$$u\!:\!\overline{x_P} = u\!:\!\overline{y_F} \vee u\!:\!\overline{y_F}\!:\!\overline{w_F}\!:\!\overline{y_F}' = u\!:\!\overline{y_F}\!:\!\overline{w_F} \vee R(u\!:\!\overline{x_P}\!:\!\overline{z_P}, u\!:\!\overline{y_F}\!:\!\overline{w_F}\!:\!\overline{y_F}'\!:\!\overline{w_F}')$$

Obviously, both results are subsumed by the third clause schema and we are done.

Again there are two possibilities how to proceed: we either try to find a nice alternative background theory or we try to replace this saturation (and thus the whole background theory) by some suitable inference rules. We do not consider the former possibility here since a "nice" alternative background theory is not in sight. We therefore search for interesting inference rules and that in the lines of sections 4.2.2 and 4.2.3.

To this end we again "read" the inference rules from the saturation set and essentially adopt the inference rules from those we know already from other – more or less similar – logics.

DEFINITION 4.3.2 (THE $K_tD4 \oplus K_tT4$ INFERENCE SYSTEM)
We consider the following inference rules

$$\frac{\neg R^*(\alpha\!:\!\gamma, \beta\!:\!\delta), C}{\sigma C} \quad \begin{array}{c} \sigma = mgu(\alpha,\beta) \\ \gamma \in (F_{R_P} \cup F_{R_P^*})^* \\ \delta \in (F_{R_F} \cup F_{R_F^*})^* \end{array} \qquad \frac{\neg R(\alpha\!:\!\gamma, \beta\!:\!\delta), C}{\sigma C} \quad \begin{array}{c} \sigma = mgu(\alpha,\beta) \textit{ and} \\ \textit{either } \exists x \in \gamma\ x \in F_{R_P} \\ \textit{or } \exists y \in \delta\ y \in F_{R_F} \end{array}$$

$$\frac{\neg R(\alpha\!:\!\gamma_1\!:\!\gamma_2, \beta\!:\!\delta_1\!:\!\delta_2), C}{\sigma C, \sigma(\alpha\!:\!\gamma_1) = \sigma(\beta\!:\!\delta_1)} \quad \begin{array}{c} \sigma = mgu(\alpha,\beta) \\ \gamma_1,\gamma_2 \in (F_{R_P^*})^* \\ \delta_1,\delta_2 \in (F_{R_F^*})^* \end{array}$$

$$\frac{\neg R(\alpha, u), C}{C^u_{\alpha:x_F}} \quad \textit{where } u \notin \alpha \qquad \frac{\neg R(u, \alpha), C}{C^u_{\alpha:x_P}} \quad \textit{where } u \notin \alpha$$

The $K_tD4 \oplus K_tT4$ Inference System *then consists of (classical) resolution and factorization together with the above rules.*

Evidently, the first three rules correspond to the three clause schemata that have been derived by saturating the $K_tD4 \oplus K_tT4$ background theory and the final two are meant to take care of the K_tD4 specialties.

The effect of the one new inference rule above which introduces an equality may not be easily comprehensible at the first glance. The following example might help to understand what this rule is about.

EXAMPLE 4.3.3
Show the $K_tD4 \oplus K_tT4$-validity of

$$\langle\mathrm{P}\rangle_r(Q \wedge \langle\mathrm{P}\rangle_r \boxed{\mathrm{F}} P) \Rightarrow \boxed{\mathrm{F}}_r(Q \vee \boxed{\mathrm{F}}_r P)$$

i.e. try to refute $\langle\mathrm{P}\rangle_r(Q \wedge \langle\mathrm{P}\rangle_r \boxed{\mathrm{F}} P) \wedge \langle\mathrm{F}\rangle_r(\neg Q \wedge \langle\mathrm{F}\rangle_r \neg P)$.
Semi-functonal translation results in:

$$\begin{array}{l} Q(\iota : a_P^*) \\ \neg R(\iota : a_P^* : b_P^*, u), P(u) \\ \neg Q(\iota : c_F^*) \\ \neg P(\iota : c_F^* : d_F^*) \end{array}$$

There is only one reasonable first step (and that a resolution step between the P-literals) which results in:

$$\neg R(\iota : a_P^* : b_P^*, \iota : c_F^* : d_F^*)$$

The equality introducing inference rule from above then yields $\iota : a_P^* = \iota : c_F^*$ which together with the two Q-literals leads to the empty clause.

After the inference system for $K_tD4 \oplus K_tT4$ has been defined we have to show its soundness and refutation completeness and again this is fairly easy with the means provided in the earlier sections.

LEMMA 4.3.4
The $K_tD4 \oplus K_tT4$ Inference System is sound.

Proof: The application of any of the first three new inference rules corresponds to a resolution step with an instance of one of the derived clause schemata. The soundness for the final two rules is established in exactly the same way as in case of K_tD4.

LEMMA 4.3.5
A single $K_tD4 \oplus K_tT4$-unsatisfiable clause C can be refuted.

Proof: Works almost exactly as in the proof of Lemma 4.2.7. Only one particularity might be worth mentioning: Suppose that some instant variable occurred only on right-hand arguments of C and that no inference step on a minimal ground instance of C is liftable. In case of K_tD4 we then knew that C is of the form

$$\ldots \neg R(\alpha_i, u : \beta_i), \ldots, C'$$

and that for one of these literals, say $\neg R(\alpha, u : \beta)$, we have that $\beta = \emptyset$ and $\sigma\alpha$ is a direct prefix of $\sigma(u)$.

The same holds for $K_tD4 \oplus K_tT4$ as well, however, what if this very literal is not an R-literal but an R^*-literal? Fortunately, this cannot happen for then it would already be sufficient to instantiate u by the direct prefix of $\sigma(u)$ without loosing unsatisfiability. But then σ would not be minimal and therefore the literal in question must be an R-literal and the proof proceeds exactly as in Lemma 4.2.7.

LEMMA 4.3.6
The $K_tD4 \oplus K_tT4$ Inference System is refutation complete.

Proof: Exactly as in case of Lemma 3.9.40, i.e. it is shown that a single $K_tD4 \oplus K_tT4$-unsatisfiable clause can be refuted and that it is always possible to derive either a single $K_tD4 \oplus K_tT4$-unsatisfiable clause or a conditioned instance which can be used to derive such a $K_tD4 \oplus K_tT4$-unsatisfiable clause. Since there are only finitly many possibilities to derive conditioned instances this will ultimately terminate.

4.4 The *Always*-Operator

Up to now we examined logics which know about the *henceforth*, the *hitherto*, the *eventually*, and the *previously* together with the respective duals. In addition there may be a need for an *always* in order to be able to represent some overall truth. In the introductory chapter on temporal logics we already learned about a temporal logic which captures just this "always"-operator, although it didn't know how to treat any other of the operators we are interested in. This logic is actually the modal logic *S5* and in the chapter on modal logics we have learned how the semi-functional translation approach deals with *S5* very efficiently.

The problem with *S5* is its very limited expressive power. One can hardly imagine any serious application of temporal logic where *S5* would really suffice as *the* temporal logic. Nevertheless there might be a need for such operators *in addition* to the ones we have already learned about.

One possibility of adding such an *Always*-operator, say $\boxed{A}_r$, is to consider it as *not* primitive, i.e. by translating it in terms of the other operators we already know. A first idea in these lines would be to extend Ω_F by:

$$\begin{aligned} \lfloor \boxed{A}_r \Phi \rfloor_u &= \lfloor \boxed{F}_r \Phi \rfloor_u \wedge \lfloor \boxed{P}_r \Phi \rfloor_u \\ \lfloor \langle A\rangle_r \Phi \rfloor_u &= \lfloor \langle F\rangle_r \Phi \rfloor_u \vee \lfloor \langle P\rangle_r \Phi \rfloor_u \end{aligned}$$

in the sense that Φ is *always* true if and only if Φ will be true *henceforth* and was true *hitherto*. Alternatively, one could imagine to translate this new operator in terms of *always in the future* and *always in the past* by

$$\begin{aligned} \lfloor \boxed{A}_r \Phi \rfloor_u &= \lfloor \boxed{F} \Phi \rfloor_u \wedge \lfloor \Phi \rfloor_u \wedge \lfloor \boxed{P} \Phi \rfloor_u \\ \lfloor \langle A\rangle_r \Phi \rfloor_u &= \lfloor \langle F\rangle \Phi \rfloor_u \vee \lfloor \Phi \rfloor_u \vee \lfloor \langle P\rangle \Phi \rfloor_u \end{aligned}$$

where the respective $\lfloor \Phi \rfloor_u$ has to be added to ensure that the present is not excluded.

Even some kind of a mixture of the two possibilities above could be contemplated, as for instance

$$\begin{aligned} \lfloor \boxed{A}_r \Phi \rfloor_u &= \lfloor \boxed{F}_r \Phi \rfloor_u \wedge \lfloor \boxed{P} \Phi \rfloor_u \\ \lfloor \langle A\rangle_r \Phi \rfloor_u &= \lfloor \langle F\rangle \Phi \rfloor_u \vee \lfloor \langle P\rangle_r \Phi \rfloor_u \end{aligned}$$

One of the disadvantages all these possibilities have in common is that the clause set which is finally generated suffers from an exponential increase in the number of $\boxed{A}_r$-operators. Another disadvantage is that an obvious property one has in mind as an essential property for the *Always*-relation, the transitivity, does not follow from this translation. Whereas in standard modal logics a problem like the latter one can quite easily be overcome by simply adding the axiom schema $\boxed{A}_r\Phi \Rightarrow \boxed{A}_r\boxed{A}_r\Phi$ we do not have this possibility here simply because the $\boxed{A}_r$-operator is not primitive. We are therefore forced to think about alternatives.

The alternative proposed here is along the lines of the examination of $K_tD4 \oplus K_tT4$: we consider $\boxed{A}_r$ as a primitive operator and add the suitable schemata to the axiomatization. Notwithstanding, for reasons that will become clear in the following chapter we do *not yet* include the full axiom $\boxed{A}_r\Phi \Leftrightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$. Rather we simplify it slightly to only one of its two directions, namely $\boxed{A}_r\Phi \Rightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$. The schema $\boxed{A}_r\Phi \Rightarrow \boxed{A}_r\boxed{A}_r\Phi$, however, is to be added anyway.

The first of these axioms describes our main intuition about the *always* and the latter fills the transitivity gap. The semi-functional formula translation thus looks like this[14]:

$$\begin{array}{rcl} \lfloor \boxed{A}_r\Phi \rfloor_u &=& \forall v\ R_A(u,v) \Rightarrow \lfloor \Phi \rfloor_u \\ \lfloor \Diamond^{A}_r\Phi \rfloor_u &=& \exists x_A\ \lfloor \Phi \rfloor_{u:x_A} \end{array}$$

where the other cases remain as before.

The background theory for this logic[15] then consists of the background theory of $K_tD4{\oplus}K_tT4$ plus

$$\begin{array}{l} R_F^*(u,v) \Rightarrow R_A(u,v) \\ R_F^*(u,v) \Rightarrow R_A(v,u) \\ R_A(u,v) \wedge R_A(v,w) \Rightarrow R_A(u,w) \end{array}$$

This background theory is quite easy to be handled for us now since the only positive literals are R_A-literals, thus the saturation to be derived for this theory does not change for any of the other temporal relations. Now what the saturation of the R_A-predicate is concerned we end up with a neat result.

LEMMA 4.4.1
The saturation of the R_A-predicate results in the clauses of the form

$$R_A(u:x_1:\cdots:x_n, u:y_1:\cdots:y_m)$$

where $n, m \geq 0$ and each x_i, y_j belongs to the functional decomposition for any of the relations R_P, R_F, R_P^*, R_F^* or R_A.

Proof: From the reflexivity of R_F^* we derive the reflexivity of R_A. Now we can easily prove that

$$R_A(u,v) \Rightarrow \left\{ \begin{array}{l} R_A(u, v:x_F) \\ R_A(u, v:x_F^*) \\ R_A(u, v:x_P) \\ R_A(u, v:x_P^*) \\ R_A(u, v:x_A) \\ R_A(u:x_F, v) \\ R_A(u:x_F^*, v) \\ R_A(u:x_P, v) \\ R_A(u:x_P^*, v) \\ R_A(u:x_A, v) \end{array} \right.$$

From this it follows that at least all elements of the alledged saturation are derivable.

Remains to be shown that there are no other derivable unit clauses. To this end we apply this clause schema to the transitivity clause for R_A and end up with a unit which is already contained in the clause schema. For the other two new axioms we also cannot find anything that has not yet been derived and so we are done.

[14]Note that under the given axiomatization we indeed have that the *always*-relation is serial. Thus the semi-functional translation for serial logics suffices.

[15]From the axiomatization it follows that the axiom schemata for reflexivity and symmetry hold as well. The canonical name for this logic would thus be $K_tD4{\oplus}K_tT4{\oplus}K_tT5$. This looks rather awkward and will therefore be omitted in the sequel.

Now, what is so neat about this saturation? It doesn't look much simpler than the saturations for the other temporal relations. The point is that we are now able to take the connectedness assumption into account which we successfully utilized already in the chapter on modal logics.

LEMMA 4.4.2
The connectedness assumption for $K_tD4 \oplus K_tT4$ (with or without the $\boxed{A}_r$-operator) leads to

$$\forall u \exists x_1, \ldots, x_n \; u = \iota : x_1 : \cdots : x_n$$

for some $n \geq 0$ and each x_i belongs to one of the functional decompositions involved.

Proof: The connectedness assumption states that any world can be accessed from the initial world by the reflexive and transitive closure of the union of the respective accessibility relations, i.e.

$$\forall u \; u = \iota \vee R_F(\iota, u) \vee R_F^*(\iota, u) \vee \ldots \vee \exists v \; R_F(\iota, v) \wedge R_A(v, u) \vee \ldots$$

In terms of the semi-functional translation this means that

$$\forall u \; u = \iota \vee \exists x_F \; u = \iota : x_F \vee \exists x_F^* \; u = \iota : x_F^* \vee \ldots \vee \exists x_F, y_A \; u = \iota : x_F : y_A \vee \ldots$$

and this is just what has been claimed.

Temporal logics as an – in this sense – instance of multi-modal logics do not at all differ from modal logics what this assumption is concerned, i.e. the Segerberg result which states that no modal logics can distinguish between connected and non-connected frames also works for temporal logics. Thus we may assume that the temporal structure under consideration is connected and in doing so we end up with the following:

LEMMA 4.4.3
The saturation for R_A together with the connectedness assumption results in the universal relation for R_A, i.e.

$$\forall u, v \; R_A(u, v)$$

Proof: The saturation for R_A consists of all units of the form

$$R_A(u : x_1 : \cdots : x_n, u : y_1 : \cdots : y_m)$$

with $n, m \geq 0$ and each x_i, y_j belongs to the functional decomposition of any of the relations R_P, R_F, R_P^*, R_F^* or R_A according to Lemma 4.4.1. Thus, in particular we have that

$$R_A(\iota : x_1 : \cdots : x_n, \iota : y_1 : \cdots : y_m)$$

and each of these two argument schemas can be simplified to arbitrary variables according to Lemma 4.4.2.

Hence, the effect of adding an *always*-operator to the temporal logic $K_tD4 \oplus K_tT4$ (with respect to the semi-functional translation) can be summarized as follows: simply extend the formula translation $\lfloor \; \rfloor$ by

$$\begin{array}{rcl} \lfloor \boxed{A}_r \Phi \rfloor_u & = & \forall v \; \lfloor \Phi \rfloor_v \\ \lfloor \Diamond_r \Phi \rfloor_u & = & \exists x_A \; \lfloor \Phi \rfloor_{u : x_A} \end{array}$$

Note that the simple unit clause $R_A(u, v)$ has already been incorporated into the translation here and therefore needs not to be included in the background theory. This on the other hand means that no extra inference rules are necessary when the *Always* is included; everything necessary is already included by the translation.

5

The Linearity Assumption

Having a look back to the introductory chapter on temporal logics we can see that we have got quite far already. Up to now we considered several temporal logics with both past and future operators and the earlier-later properties we examined are approaching nearer and nearer towards the properties of a *flow relation*. Two major flow relation properties are not yet examined though, the irreflexivity and the linearity.

What the irreflexivity is concerned we already have an answer, although – unluckily – a negative one. According to Section 3.6 on page 52 there is no axiom schema which characterizes the irreflexive frames. This does not mean that no axiom schema *implies* irreflexivity[1], although none of the axiom schemata we are interested in is of such a kind.

However, recall that the reason why the irreflexivity is not axiomatizable lies in the fact that modal logics cannot distinguish between irreflexive and non-reflexive frames and therefore any formula that is valid in irreflexive frames is also valid in all non-reflexive frames simply because adding irreflexivity or not makes no difference (the other direction is trivial anyway for every irreflexive frame is a non-reflexive frame in particular). Still, this may be not very satisfying for *not assuming reflexivity* does not mean *assuming non-reflexivity*. Nevertheless, it carries us pretty near towards irreflexivity and this is the best we can achieve.

Another property we were interested in was *linearity* and we are faced with the question whether there is an axiom schema that characterizes this property. Unfortunately, the answer is again negative: There is no axiom schema that characterizes the linear frames for modal or

[1]For instance the so called Löb-Axiom given by $\Box(\Box\Phi \Rightarrow \Phi) \Rightarrow \Box\Phi$ characterizes the transitive and backward well-founded frames and thus implies irreflexivity. Note that this property induced by the Löb-Axiom is not first-order expressible and therefore cannot be handled directly by the approach presented in this work.

temporal logics and it will be shown why this is so with the means provided in Section 3.6. Nevertheless, there is no reason to be too bothered by this negative result for there are indeed axiom schemata that characterize some slightly weaker form of linearity, the forward- and the backward-linearity, and in the light of the connectedness assumption these will lead us to general linearity.

As with the other special properties like reflexivity and transitivity, the linearity property comes into play via some suitable axiom schemata. Under reflexivity, i.e. the only operators we assume are $\boxed{\mathrm{F}}_r$ and $\boxed{\mathrm{P}}_r$, typical linearity axioms are

$$\begin{array}{l}
\boxed{\mathrm{F}}_r(\boxed{\mathrm{F}}_r\Phi \Rightarrow \Psi) \vee \boxed{\mathrm{F}}_r(\boxed{\mathrm{F}}_r\Psi \Rightarrow \Phi) \\
\langle\mathrm{F}\rangle_r\,\Phi \wedge \langle\mathrm{F}\rangle_r\,\Psi \Rightarrow \langle\mathrm{F}\rangle_r(\Phi \wedge \langle\mathrm{F}\rangle_r\,\Psi) \vee \langle\mathrm{F}\rangle_r(\langle\mathrm{F}\rangle_r\,\Phi \wedge \Psi) \\
\langle\mathrm{P}\rangle_r\,\langle\mathrm{F}\rangle_r\,\Phi \Rightarrow \langle\mathrm{P}\rangle_r\,\Phi \vee \langle\mathrm{F}\rangle_r\,\Phi
\end{array}$$

for the forward (future) linearity and the respective mirror images for the backward (past) linearity. It might be surprising at the first glance that these three axiom schemata, although they look entirely different, all denote forward-linearity. Anyway, it can easily be checked that applying the Elimination Theorem on either of these axioms results in the first-order property

$$\forall u, v, w\ R^*(u, v) \wedge R^*(u, w) \Rightarrow R^*(v, w) \vee R^*(w, v)$$

i.e. in *right-linearity* on reflexive frames.

Such properties will have to be examined in this chapter. However, before they are added to the rather complicated temporal logics of the last chapter we start our investigations on the corresponding extensions to the slightly more gentle modal logics.

5.1 The Modal Logic *S4.3*

S4.3 is the smallest extension of *S4* that obeys the linearity assumption. Hence it is axiomatized by the already well-known axiomatization of *S4* plus one of the first two axioms from above, for instance $\Box(\Box\Phi \Rightarrow \Psi) \vee \Box(\Box\Psi \Rightarrow \Phi)$. The background theory under semi-functional translation is therefore given by:

$$\begin{array}{l}
R(u, u) \\
R(u, u\!:\!x) \\
R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \\
R(u, v) \wedge R(u, w) \Rightarrow R(v, w) \vee R(w, v)
\end{array}$$

This theory is already fairly complicated and theorem provers will have quite some difficulties in proving even simple theorems. Let us therefore try to simplify the above theory with the means provided so far.

Evidently, the background theory for *S4.3* is a superset of the background theory for *S4*. Hence, we get as a first part of the saturation of the background theory for *S4.3* all unit clauses of the form

$$R(u, u\!:\!x_1\!:\!x_2\!:\cdots:\!x_n)$$

with $n \geq 0$ and $x_i \in F_R$ for all $0 \leq 1 \leq n$. From this and the connectedness assumption it is possible to derive general linearity for R, i.e. $\forall u, v\ R(u, v) \vee R(v, u)$ as is shown in the following lemma.

LEMMA 5.1.1
Under the connectedness assumption the background theory for S4.3 is equivalent to

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R(u,v) \vee R(v,u) \end{array}$$

Proof: By the observation from above we have that $R(u, u\!:\!x_1\!:\!x_2\!:\cdots:\!x_n)$ for any world variable u and any $x_i \in F_R$. Two resolution steps with the transitivity clause then yield

$$R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m) \vee R(u\!:\!y_1\!:\cdots:\!y_m, u\!:\!x_1\!:\cdots:\!x_n)$$

with $n, m \geq 0$ and $x_i, y_j \in F_R$. In particular we then have that

$$R(\iota\!:\!x_1\!:\cdots:\!x_n, \iota\!:\!y_1\!:\cdots:\!y_m) \vee R(\iota\!:\!y_1\!:\cdots:\!y_m, \iota\!:\!x_1\!:\cdots:\!x_n)$$

Now recall that the connectedness assumption is expressed by

$$\forall u \exists z_1, \ldots, z_k\ u = \iota\!:\!z_1\!:\cdots:\!z_k$$

The arguments in the above two-literal clause schemata therefore refer to any world whatsoever, or, more formally,

$$\begin{array}{c} \forall u \exists z_1, \ldots, z_k\ u = \iota\!:\!z_1\!:\cdots:\!z_k \\ \Rightarrow \\ R(u\!:\!x_1\!:\cdots:\!x_n, u\!:\!y_1\!:\cdots:\!y_m) \vee R(u\!:\!y_1\!:\cdots:\!y_m, u\!:\!x_1\!:\cdots:\!x_n) \Leftrightarrow R(u,v) \vee R(v,u) \end{array}$$

and thus right-linearity can be replaced by general linearity which even subsumes reflexivity.

Hence, the saturation idea together with the connectedness assumption frees us from some superfluous ballast, namely the conditional part of right-linearity. The connectedness assumption is in fact essential in this proof for otherwise it were possible to have several unconnected (parallel) linear time axis and these could by no means be simplified to a single one. As a little side-effect, we therefore get the following corollary.

COROLLARY 5.1.2
Linearity is not modal logic axiomatizable.

Proof: According to Section 3.6 an accessibility relation property Φ is not axiomatizable if there exists a strictly weaker property which implies Φ under the connectedness assumption. Now, right-linearity is strictly weaker than linearity but nevertheless implies linearity in connected frames. Therefore, linearity is not axiomatizable.

What should be done next is to find a representation of the *S4.3* background theory which suits better the idea of having inference rules instead of a background theory. To this end, we should try to get rid of the transitivity clause in its usual form and we do so as follows.

LEMMA 5.1.3
The S4.3 background theory can be described by

$$\begin{array}{l} R(u,u) \\ R(u,v) \Rightarrow R(u, v\!:\!x) \\ R(u,v) \Rightarrow R(w,v) \vee R(u,w) \end{array}$$

Proof: According to Theorem 3.3.18 on page 39 it suffices to show that the saturations of both clause sets

$$\begin{array}{lll} R(u,u\!:\!x) & & \\ R(u,v)\wedge R(v,w)\Rightarrow R(u,w) & \text{A)} & \\ R(u,v)\vee R(v,u) & & \end{array} \quad \text{and} \quad \begin{array}{ll} R(u,u) & \\ R(u,v)\Rightarrow R(u,v\!:\!x) & \text{B)} \\ R(u,v)\Rightarrow R(w,v)\vee R(u,w) & \end{array}$$

are identical.

Evidently, clause set B) is implied by A) and therefore its saturation is contained in the saturation of A). It is therefore sufficient to show the existence of a clause schema which

1. is contained in the saturation of B)
2. contains the saturation of A)

Such a schema obviously describes the saturation of both A) and B) then.

Now, consider the clause schema

$$R(u_1,u_2\!:\!\overline{x_2})\vee R(u_2,u_3\!:\!\overline{x_3})\vee\ldots\vee R(u_{n-1},u_n\!:\!\overline{x_n})\vee R(u_n,u_1\!:\!\overline{x_1})$$

with $n\geq 1$ and $\overline{x_i}\in F_R$. For the first part we show by induction over n that

$$R(u_1,u_2)\vee R(u_2,u_3)\vee\ldots\vee R(u_{n-1},u_n)\vee R(u_n,u_1)$$

is contained in the saturation of B) and then that the right-hand argument of each such disjunctive element can be arbitrarily inflated by functional decomposition variables.

Base Case: $n=1$: trivial, since $R(u,u)$ is contained in B).

Induction Step: suppose $R(u_1,u_2)\vee R(u_2,u_3)\vee\ldots\vee R(u_{n-1},u_n)\vee R(u_n,u_1)$ has already be shown to belong to the saturation of B). A single resolution step with $R(u,v)\Rightarrow R(w,v)\vee R(u,w)$ then results in $R(u_1,u_2)\vee R(u_2,u_3)\vee\ldots\vee R(u_n,u_{n+1})\vee R(u_{n+1},u_1)$ and we are done.

Now note that for an arbitrary clause $C=R(\alpha,\beta)\vee C'$ it is possible to derive $\hat{C}=R(\alpha,\beta\!:\!x)\vee C'$ by a single resolution step with the clause $R(u,v)\Rightarrow R(u,v\!:\!x)$. What has been claimed thus follows easily by induction over the respective lengths of the functional decomposition variable sequences.

For the second part of the proof we have to check whether every clause in the saturation of A) is an instance of the above clause schema. Evidently, this is no problem for both $R(u,u\!:\!x)$ and $R(u,v)\vee R(v,u)$. Remains to be shown that any pure-R-positive clause that can be derived from

$$R(u_1,u_2\!:\!\overline{x_2})\vee R(u_2,u_3\!:\!\overline{x_3})\vee\ldots\vee R(u_{n-1},u_n\!:\!\overline{x_n})\vee R(u_n,u_1\!:\!\overline{x_1})$$

and the transitivity clause is already an instance of this schema. Thus consider the clause set

$$\begin{array}{l} \neg R(u,v)\vee\neg R(v,w)\vee R(u,w) \\ R(u_1,u_2\!:\!\overline{x_2})\vee\ldots\vee R(u_n,u_1\!:\!\overline{x_1}) \\ R(v_1,v_2\!:\!\overline{y_2})\vee\ldots\vee R(v_m,v_1\!:\!\overline{y_1}) \end{array}$$

The first resolution step results in:

$$\neg R(u_2 : \overline{x_2}, w) \vee R(u, w) \vee R(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R(u_n, u : \overline{x_1})$$

and after a second resolution step we end up with

$$R(u, v_2 : \overline{y_2}) \vee R(v_2, v_3 : \overline{y_3}) \vee \ldots \vee R(v_m, u_2 : \overline{x_2}) \vee R(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R(u_n, u : \overline{x_1})$$

This clause schema is indeed an instance of the schema we already have; hence we are done.

Now, for *S4.3* we know both its saturation and a suitable clause set which generates exactly this saturation and which is "simpler" than the original background theory. Thus we can try to find an appropriate inference rule for *S4.3* just as we did for the other modal and temporal logics.

A first idea for such an inference rule would be to "read" it directly from the saturation we know. This would result in something like:

$$\frac{\begin{array}{c}\neg R(\alpha_1, \beta_1), C_1 \\ \neg R(\alpha_2, \beta_2), C_2 \\ \vdots \\ \neg R(\alpha_n, \beta_n), C_n\end{array}}{\sigma C_1, \sigma C_2, \ldots, \sigma C_n}$$

where σ is an *S4* prefix unifier for (α_{i+1}, β_i) and (α_1, β_n). Such an inference rule, however, regardless whether it would turn out to be complete or not, does not quite suit the resolution paradigm which states that inference rules should be applied locally to one or two clauses rather than possibly the whole clause set.

Another possibility for a suitable inference rule can be found by having a closer look at the newly derived background theory for *S4.3*. Evidently, the first two clauses are those of the background theory for *S4* and thus the *S4* Inference Rule could act at least as one of the inference rules for *S4.3* as well.

What the third clause – $R(u, v) \Rightarrow R(w, v) \vee R(u, w)$ – is concerned it could be imagined to result in an inference rule like

$$\frac{\begin{array}{c}\neg R(\alpha, \beta), C_1 \\ \neg R(\gamma, \delta), C_2\end{array}}{\neg R(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is the most general (classical unifier) for (γ, β). This rule would be sound but, unfortunately, not complete (even together with the *S4* Inference Rule). The problem is that the interaction between the *S4* part and the third theory clause has not been taken into account this way. But we can quite easily overcome this problem by observing that the third theory clause can be generalized to

$$R(u, v) \Rightarrow R(w, v) \vee R(u, w : x_1 : \cdots : x_n)$$

and this generalization finally leads to the following:

DEFINITION 5.1.4 (THE *S4.3* INFERENCE RULE)
The rule

$$\frac{\begin{array}{c}\neg R(\alpha, \beta), C_1 \\ \neg R(\gamma, \delta), C_2\end{array}}{\neg R(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2} \qquad \textit{where } \sigma \textit{ is an S4 prefix unifier for } (\gamma, \beta)$$

is called the S4.3 Inference Rule.

DEFINITION 5.1.5 (THE *S4.3* INFERENCE SYSTEM)
The S4.3 Inference System *consists of the S4 Inference System together with the S4.3 Inference Rule*[2]*.*

LEMMA 5.1.6
The S4.3 Inference System is sound.

Proof: The soundness of the *S4* Inference System is evident. Remains the *S4.3* Inference Rule. We have to show that any *S4.3* interpretation which satisfies the antecedent of the *S4.3* Inference Rules also satisfies its consequent. To this end assume that there exists an *S4.3* interpretation $\Im$ which is a model for $\neg R(\alpha, \beta) \vee C_1$ and for $\neg R(\gamma, \delta) \vee C_2$ but not for $\neg R(\sigma\alpha, \sigma\delta) \vee \sigma C_1 \vee \sigma C_2$ where σ is an *S4* prefix unifier of (γ, β). This means that there exists a variant $\hat{\Im}$ of $\Im$ such that

$$\hat{\Im} \models R(\sigma\alpha, \sigma\delta) \wedge \neg R(\sigma\alpha, \sigma\beta) \wedge \neg R(\sigma\gamma, \sigma\delta)$$

$\hat{\Im}$ is an *S4.3* interpretation and therefore R is both linear and transitive. It thus follows that

$$\hat{\Im} \models R(\sigma\alpha, \sigma\gamma) \wedge \neg R(\sigma\alpha, \sigma\beta)$$

Now recall that $R(u, v) \Rightarrow R(u, v\!:\!x)$ for any *S4.3* interpretation and that $\sigma\gamma$ is a prefix of $\sigma\beta$ (since σ is an *S4* prefix unifier of (γ, β)). Hence $\hat{\Im}$ is a model for both $R(\sigma\alpha, \sigma\beta)$ and $\neg R(\sigma\alpha, \sigma\beta)$ which is a contradiction. It is thus not possible for any *S4.3* interpretation to satisfy the antecedent of the *S4* Inference Rule but not its consequent and we are done.

LEMMA 5.1.7
The S4.3 Inference System is ground complete.

Proof: Any application of one of the inference rules is justified by the saturation of the *S4.3* background theory.

The idea behind the completeness proof for the *S4.3* Inference System is to reduce it to the completeness proof for the *S4* Inference System. To this end we consider an alternative Inference Rule, namely

$$\frac{\begin{array}{c}\neg R(\alpha_1, \beta_1), C_1 \\ \neg R(\alpha_2, \beta_2), C_2\end{array}}{\neg R(\alpha_1, \beta_2), \neg R(\alpha_2, \beta_1), C_1, C_2}$$

This alternative rule would be very inefficient indeed for it can be applied to any two R-literals. However, it is pretty obvious that an application of this rule followed immediately by an application of the *S4* Inference Rule to the application result generates a resolvent that could have been obtained from a single application of the *S4.3* Inference Rule. Intuitively, the alternative inference rule tries to cut potential cycles into two smaller ones (where by a cycle we mean a set of R-literals which is inconsistent with the *S4.3* background theory). The *S4.3* Inference Rule does something similar, however, it cuts not arbitrarily but into two portions such that one of them is trivial (i.e. it can be eliminated with the *S4* Inference Rule). This does not only reduce the number of possible cuts, it also takes into account that every cycle is finite and must contain neighbourhood elements.

[2]Note that the *S4* Inference Rule can be viewed as a special case of the *S4.3* Inference Rule if the two parent clauses are identical.

We now proceed as follows: First we show that we are able to derive a single *S4*-unsatisfiable clause from an arbitrary *S4.3*-unsatisfiable set of R-clauses with the help of the alternative inference rule. This clause can then be refuted by a finite number of applications of the *S4* Inference Rule. Finally we show how the refutation obtained this way can be transformed into a refutation within the *S4.3* Inference System.

Lemma 5.1.8
It is possible to derive (by resolution and factorization) a S4.3-unsatisfiable set of R-clauses from an arbitrary S4.3-unsatisfiable clause set.

Proof: Obvious, since this can easily be done on the ground level and any resolution and factorization step can be lifted.

Lemma 5.1.9
A single S4-unsatisfiable clause can be derived from an arbitrary S4.3-unsatisfiable set $\mathcal{C}$ *of R-clauses and that with the help of the alternative Inference Rule.*

Proof: Let $\sigma\mathcal{C}$ be an *S4.3*-unsatisfiable ground instance of $\mathcal{C}$. It suffices to show that the lemma holds for $\sigma\mathcal{C}$ since an application of the alternative inference rule can trivially be lifted to $\mathcal{C}$ (there are no instantiations involved).

Consider any refutation of $\sigma\mathcal{C}$ (which exists according to Lemma 5.1.7). Evidently, whenever the *S4.3* Inference Rule gets applied it is also possible to apply the alternative inference rule to the very same literals. Thus if we replace every application of the *S4.3* Inference Rule by an application of the alternative inference rule and ignore every application of the *S4* Inference Rule we end up with a single *S4*-unsatisfiable clause.

Lemma 5.1.10
The S4.3 Inference System is refutation complete.

Proof: From Lemma 5.1.8 we know that a *S4.3*-unsatisfiable set of R-clauses can be derived from an arbitrary *S4.3*-unsatisfiable set of clauses by resolution and factorization. It would now be possible to derive a single *S4*-unsatisfiable clause from this clause set if we had the alternative inference rule at hand and, evidently, this single clause could easily be refuted by the *S4* Inference Rule. Remains to be shown that such a refutation can be transformed into a refutation within the *S4.3* Inference System.

To this end consider the first literal in the single unsatisfiable clause to be eliminated by the *S4* Inference Rule and find the parent clauses in the original clause set from where it descents. These parent clauses (and the involved literals) can be found since in the derivation of this very literal no variable instantiation has been performed. The corresponding application of the alternative inference rule together with the final application of the *S4* Inference Rule can now be performed by a single application of the *S4.3* Inference Rule (or by the *S4* Inference Rule if the literal under consideration had not been generated by the alternative inference rule). This way we are able to replace each *S4* Inference Rule application in the refutation of the single *S4*-unsatisfiable clause by an application of a rule in the *S4.3* Inference System and that without any need for the alternative inference rule. Hence there exists an (alternative) refutation that can be simulated by the *S4.3* Inference System and we are done.

5.2 Linearity and K_tT4

The modal logic *S4.3* can hardly be called a temporal logic. Recall that even the simplest temporal logic we considered, Lemmon's minimal tense logic K_t, knows about both past and future, although in a fairly restricted way. Therefore a linear temporal logic closest to *S4.3* would be $K_tT4.3$, i.e. the temporal logic for which the Earlier and the Later relations are reflexive, transitive and right-linear[3]. A possible axiomatization would thus be:

$$\begin{array}{c} \Diamond\!\!\!\!\scriptstyle{F}_r\, \Phi \wedge \Diamond\!\!\!\!\scriptstyle{F}_r\, \Psi \Rightarrow \Diamond\!\!\!\!\scriptstyle{F}_r(\Phi \wedge \Diamond\!\!\!\!\scriptstyle{F}_r\, \Psi) \vee \Diamond\!\!\!\!\scriptstyle{F}_r(\Diamond\!\!\!\!\scriptstyle{F}_r\, \Phi \wedge \Psi) \\ \Diamond\!\!\!\!\scriptstyle{P}_r\, \Phi \wedge \Diamond\!\!\!\!\scriptstyle{P}_r\, \Psi \Rightarrow \Diamond\!\!\!\!\scriptstyle{P}_r(\Phi \wedge \Diamond\!\!\!\!\scriptstyle{P}_r\, \Psi) \vee \Diamond\!\!\!\!\scriptstyle{P}_r(\Diamond\!\!\!\!\scriptstyle{P}_r\, \Phi \wedge \Psi) \end{array}$$

together with the well-known axiomatization for K_tT4. Note that indeed both of these axioms are necessary because, unlike transitivity, right-linearity of one does not imply right-linearity of the other.

Thus we are able to describe the background theory for $K_tT4.3$ as:

$$\begin{array}{l} R(u,u) \\ R(u,u\!:\!x_F) \\ R(u\!:\!x_P,u) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R(u,v) \wedge R(u,w) \Rightarrow R(v,w) \vee R(w,v) \\ R(v,u) \wedge R(w,u) \Rightarrow R(v,w) \vee R(w,v) \end{array}$$

and again we have to determine the saturation of this background theory.

LEMMA 5.2.1
The saturation of the $K_tT4.3$ background theory contains

$$R(u,v) \vee R(v,u)$$

under the connectedness assumption.

Proof: It is easy to show (whether by hand or by using a standard theorem prover) that

$$R(u,v) \vee R(v,u) \Rightarrow \left\{ \begin{array}{l} R(u,v\!:\!x_F) \vee R(v\!:\!x_F,u) \\ R(v\!:\!x_P,u) \vee R(u,v\!:\!x_P) \end{array} \right.$$

Since R is reflexive we get by induction

$$R(u\!:\!x_1\!:\!\cdots\!:\!x_n, u\!:\!y_1\!:\!\cdots\!:\!y_m) \vee R(u\!:\!y_1\!:\!\cdots\!:\!y_m, u\!:\!x_1\!:\!\cdots\!:\!x_n)$$

with $n,m \geq 0$ and $x_i, y_j \in F_{R_P} \cup F_{R_F}$.

Now the connectivity assumption guarantees that for each u there exist some $z_1, \ldots, z_k$ such that $u = \iota\!:\!z_1\!:\!\cdots\!:\!z_k$ hence we have that

$$R(u,v) \vee R(v,u)$$

as an element of the saturation of the $K_tT4.3$ background theory.

[3]Note that because of the strong correspondence between Earlier and Later the right-linearity of Later is tantamount to the left-linearity of Earlier.

With this result we can simplify the background theory for $K_tT4.3$ to:

$$\begin{array}{l} R(u,u) \\ R(u,u\!:\!x_F) \\ R(u\!:\!x_P,u) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R(u,v) \vee R(v,u) \end{array}$$

in a way similar to the case of *S4.3*. The advantage of this representation of the background theory is that it simplifies the process of finding the corresponding saturation.

LEMMA 5.2.2
The saturation of the $K_tT4.3$ background theory consists of all clauses of the form:

$$R(u_1\!:\!\overline{x_1},u_2\!:\!\overline{y_1}) \vee R(u_2\!:\!\overline{x_2},u_3\!:\!\overline{y_2}) \vee \cdots \vee R(u_{n-1}\!:\!\overline{x_{n-1}},u_n\!:\!\overline{y_{n-1}}) \vee R(u_n\!:\!\overline{x_n},u_1\!:\!\overline{y_n})$$

where $\overline{x_i} \in (F_{R_P})^*$ and $\overline{y_i} \in (F_{R_F})^*$.

Proof: Evidently, each of the positive clauses in the background theory is indeed contained in the alledged saturation. Also, if we apply arbitrary resolution steps between the transitivity clause and any instance of the clause schemata we end up with a clause that is already contained and thus these clause schemata at least contain the saturation.

Remains to be shown that these clauses are also contained in the saturation. To this end we first show that the clauses

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \cdots \vee R(u_{n-1},u_n) \vee R(u_n,u_1)$$

are derivable from the background theory and that by induction over n.

Base Case: $n = 1$, i.e. we consider the unit clause $R(u,u)$ which trivially belongs to the saturation.

Induction Step: assume we have shown that

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \cdots \vee R(u_{n-1},u_n) \vee R(u_n,u_1)$$

belongs to the saturation. Then we perform a resolution step between the literal $R(u_n,u_1)$ and the first literal of the transitivity clause which results in

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \cdots \vee R(u_{n-1},u_n) \vee R(u_n,w) \vee \neg R(u_1,w)$$

After a final resolution step with the linearity clause $R(u,v) \vee R(v,u)$ and a renaming of w into u_{n+1} we end up with

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \cdots \vee R(u_{n-1},u_n) \vee R(u_n,u_{n+1}) \vee \neg R(u_{n+1},u_1)$$

and we are done.

Now it suffices to show that

$$R(u,v) \Rightarrow \left\{ \begin{array}{l} R(u,v\!:\!x_F) \\ R(u\!:\!x_P,v) \end{array} \right.$$

i.e. each literals in the derived clause schema can be augmented by arbitrarily many future variables in the right and arbitrarily many past variables in the left-hand argument. Evidently, all clauses in the alledged saturation are thus derivable. Now, both clauses $R(u,v) \Rightarrow R(u, v : x_F)$ and $R(u,v) \Rightarrow R(u : x_P, v)$ follow immediately from the transitivity clause and $R(u, u : x_F)$ ($R(u : x_P, u)$ respectively). A simple induction over the length of the respective functional decomposition variable sequences then completes the first part of the proof, namely that the elements of the alledged saturation can indeed be derived.

Remains to be shown that actually nothing more is derivable. To this end it suffices to show that the transitivity clause gets redundant under the derived clause schemata. Proving this fact is a bit tedious but is included here for completeness reasons. First, for convenience, let us use the abbreviation $C[u_2, u_1]$ for the clause schema $R(u_2 : \overline{x_2}, u_3 : \overline{y_2}, \ldots, R(u_n \overline{x_n}, u_1 : \overline{y_n}))$, i.e. $C[u_2, u_1]$ covers the whole clause schema except for the literal $R(u_1 : \overline{x_1}, u_2 : \overline{y_1})$. Then the derived clauses are of the form

$$R(u : \overline{x}, v\overline{y}) \vee C[v, u]$$

and we consider resolution steps between this clause schema and the transitivity clause. After the first such step we get

$$\neg R(v : \overline{y}, w) \vee R(u : \overline{x}, w) \vee C[v, u]$$

A further resolution step is only possible if either this $\overline{y} = \emptyset$ or the $\overline{x}$ in $R(u : \overline{x}, v\overline{y}) \vee C[v, u]$ is empty. In the first case we have to resolve between

$$\begin{array}{l} \neg R(v, w) \vee R(u : \overline{x}, w) \vee C[v, u] \\ R(\alpha : \overline{z}, \beta : \overline{y}) \vee C[\beta, \alpha] \end{array}$$

and this results in

$$R(u : \overline{x}, \beta : \overline{y}) \vee C[\alpha : \overline{z}, u] \vee C[\beta, \alpha]$$

which can easily be seen to belong to the clause schema already and for the second case we perform a resolution step between

$$\begin{array}{l} \neg R(v : \overline{y}, w) \vee R(u : \overline{x}, w) \vee C[v, u] \\ R(\alpha, \beta : \overline{z}) \vee C[\beta, \alpha] \end{array}$$

which leads to

$$R(u : \overline{x}, \beta : \overline{z}) \vee C[v, u] \vee C[\beta, v : \overline{y}]$$

which obviously belongs to the clause schema as well.

It is thus not possible to derive anything new from the transitivity clause which is thus redundant.

The similarities between this saturation and the saturation of the background theory for *S4.3* are evident. The main difference lies in the additional past variables that may occur on left-hand side arguments. It is thus not too surprising that the inference system to be defined for $K_tT4.3$ is also very similar to the one of *S4.3*.

DEFINITION 5.2.3 (THE $K_tT4.3$ INFERENCE RULE AND SYSTEM)
The $K_tT4.3$ Inference Rule *is defined as*

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg R(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ *is a* K_tT4 *prefix unifier of* (γ,β).
The $K_tT4.3$ Inference System *consists of the* K_tT4 *Inference System together with the* $K_tT4.3$ *Inference Rule.*

The preliminaries provided in the previous section now make soundness and completeness proofs for the $K_tT4.3$ Inference System fairly easy. In fact they don't have to be changed at all. A mere repetition of the proofs is therefore omitted here.

THEOREM 5.2.4
The $K_tT4.3$ *Inference System is sound and refutation complete.*

Proof: The only difference to the *S4.3* case lies in the additional past operators involved. Therefore the completeness proofs are to be reduced to the corresponding proof in K_tT4 rather than *S4*. Everything else remains the same.

5.3 The *Always*-Operator revisited

Recall from Section 4.4 how the *Always* had been introduced to the (non-linear) temporal logics. The main idea was to consider the *Always*-operator $\boxed{A}_r$ as a new primitive and to add the axiom schemata $\boxed{A}_r\Phi \Rightarrow \boxed{A}_r\boxed{A}_r\Phi$ and $\boxed{A}_r\Phi \Rightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$. Finding the corresponding properties of the accessibility relations and saturation of the thus obtained background theory resulted in the universal relation for R_A. Now, what would have happened if we choosed the full equivalence $\boxed{A}_r\Phi \Leftrightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$ as one of the axiom schemata instead? Then, in addition, we would get the property

$$R_A(u,v) \Rightarrow R_F(u,v) \vee R_F(v,u)$$

as an element of the background theory and since R_A turned out to become the universal relation we would automatically get the full linearity of the underlying time structure.

This is the reason why we had to delay the full equivalence in Section 4.4 up to now: we were simply not yet able at that stage to talk about linearity. After we have learned how to deal with linearity, however, we can simply add the full equivalence to the axiom set and nothing else has to be changed. Thus, augmenting $K_tT4.3$ by the $\boxed{A}_r$-operator by adding the implication $\boxed{A}_r\Phi \Rightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$ is tantamount to augmenting K_tT4 by the full equivalence $\boxed{A}_r\Phi \Leftrightarrow \boxed{F}_r\Phi \wedge \boxed{P}_r\Phi$ (transitivity has to be added in both cases, evidently). Hence, it is possible to get linearity by a suitable introduction of the *Always* operator.

It might be surprising at the first glance that the introduction of an operator can influence properties of the other operators. Indeed, if there were no other additional axioms than the equivalence which defines $\boxed{A}_r$ in terms of $\boxed{F}_r$ and $\boxed{P}_r$ then such an influence would be impossible. However, this equivalence was not the only new axiom; recall that also the transitivity for the relation R_A – which does not follow from the other axioms – had been assumed as well. But evidently, any extra property for some relation that is defined in terms of other relations will also impose extra properties on these other relations and this is what happened with the transitivity assumption for the *Always*-relation.

5.4 Linearity Without Reflexivity

The (right-)linearity axiom $\Diamond\Phi \wedge \Diamond\Psi \Rightarrow \Diamond(\Diamond\Phi \wedge \Psi) \vee \Diamond(\Phi \wedge \Diamond\Psi)$ results – after semi-functional translation and under the implicit assumption of connectedness – in the property $R(u,v) \vee R(v,u)$ for all instants u and v. It thus implies reflexivity of the accessibility relation. Even without explicitly adding the reflexivity axiom to the axiomatiztion it at least yields reflexivity for every world that is not identical to the initial world τ. If this is too strong a property for the intended axiomatization we have to think about something different and this can be found in the axiom

$$\Diamond\Phi \wedge \Diamond\Psi \Rightarrow \Diamond(\Diamond\Phi \wedge \Psi) \vee \Diamond(\Phi \wedge \Psi) \vee \Diamond(\Phi \wedge \Diamond\Psi)$$

which states that two worlds are either comparable or identical, or, more formally,

$$R(u,v) \wedge R(u,w) \Rightarrow R(v,w) \vee v = w \vee R(w,v)$$

Note that the R corresponds to a $<$ rather than a $\leq$ this way.

In this section we are going to examine how the techniques developed for the previous sections can be exploited to find a suitable inference system for modal and temporal logics that obey this very property.

5.4.1 Linearity and *KD4*

Recall the background theory for *KD4*

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \end{array}$$

and add the right-linearity (without the reflexivity assumption)

$$R(u,v) \wedge R(u,w) \Rightarrow R(v,w) \vee v = w \vee R(w,v)$$

Let us call the logic under consideration *KD4.3'* where by *3'* we mean the modified right-linearity from above. This background theory can be somewhat simplified as follows:

LEMMA 5.4.1
The background theory for KD4.3' can be simplified to:

$$\begin{array}{l} R(u, u\!:\!x) \\ R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\ R(u,v) \vee u = v \vee R(v,u) \end{array}$$

Proof: Again this is done by exploiting Segerberg's connectedness assumption. First recall that the saturation of the *KD4* background theory results in all unit clauses of the form $R(u, u\!:\!\overline{x})$ with $\overline{x} \neq \emptyset$. Resolving these units with the right-linearity axiom $3'$ produces all clauses of the form

$$R(u\!:\!\overline{x}, u\!:\!\overline{y}) \vee u\!:\!\overline{x} = u\!:\!\overline{y} \vee R(u\!:\!\overline{y}, u\!:\!\overline{x})$$

with $\overline{x}, \overline{y} \neq \emptyset$. However, if exactly one of the $\overline{x}, \overline{y}$ has zero length then the disjunction holds trivially because of the already derived unit clauses and if both have zero length the axiom is a tautology simply because of the middle equation. We thus have the above schematic disjunction even without restrictions. Therefore, and by the connectedness assumption, we have that both $u{:}\overline{x}$ and $u{:}\overline{y}$ represent arbitrary worlds and thus the disjunctive schema can be simplified to the third clause from above.

Such a simplified background theory facilitates the problem of finding the saturation of the background theory.

LEMMA 5.4.2
The saturation of the KD4.3' background theory consists of all positive clauses of the form

$$R(u_1, u_2 : \overline{x_2}) \vee R(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R(u_n, u_1 : \overline{x_1})$$
$$R(u_1, u_2) \vee R(u_2, u_3) \vee \ldots \vee R(u_n, u_1) \vee u_1 = u_2$$

where $n \geq 1$ and $\overline{x_k} \neq \emptyset$ for some $1 \leq k \leq n$.

Proof: From the *KD4.3'* background theory it evidently follows that

$$R(u, v) \Rightarrow R(u, w) \vee R(w, v)$$

This – together with $R(u, v) \vee u = v \vee R(v, u)$ – immediately leads to the second clause schema.

For the first clause schema we have that

$$\left.\begin{array}{l} R(u, u : x) \\ R(u, v) \Rightarrow R(u, w) \vee R(w, v) \end{array}\right\} \Rightarrow R(u_1, u_2) \vee R(u_2, u_3) \vee \ldots \vee R(u_n, u_1 : x)$$

and this – together with $R(u, v) \Rightarrow R(u, v : x)$ – yields

$$R(u_1, u_2 : \overline{x_2}) \vee R(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R(u_n, u_1 : \overline{x_1})$$

with $n \geq 1$ and $\overline{x_k} \neq \emptyset$ for some $1 \leq k \leq n$.

Remains to be shown that these clause schemata are closed under transitivity. This is obvious whenever the first schema is involved. Also it is easy to see that the result of applying two resolutions steps between the transitivity clause and the second schema results in a clause which is subsumed by this second clause schema. Hence nothing more than the above clause schemata can be derived and we are done.

The knowledge we have gained about the saturation of the background theory almost immediately determines a suitable set of inference rules.

DEFINITION 5.4.3 (THE *KD4.3'* INFERENCE SYSTEM)
The KD4.3' Inference System *consists of the KD4 Inference System, i.e. classical resolution and factorization plus the extra KD4 rules*

$$\frac{\neg R(\alpha, \beta), C}{\sigma C} \qquad \textit{where } \sigma \textit{ is a KD4 prefix unifier for } (\alpha, \beta)$$

$$\frac{\neg R(\alpha, u), C}{C^u_{\alpha : x}} \qquad \textit{where } u \textit{ does not occur in } \alpha \textit{ and } x \textit{ is a new variable of sort } F_R$$

together with the following rules

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg R(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}\qquad \textit{where } \sigma \textit{ is an S4 prefix unifier of } (\gamma,\beta)$$

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\sigma\alpha = \sigma\beta, \sigma C_1, \sigma C_2}\qquad \textit{where } \sigma \textit{ is the most general unifier of } (\gamma,\beta) \textit{ and } (\alpha,\delta)$$

The first of the two new rules from above could equally be described by:

$$\frac{\begin{array}{c}\neg R(\alpha,\beta_1\!:\!\beta_2), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg R(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$
$$\text{where } \sigma = mgu(\gamma,\beta_1)$$

and it might be surprising at the first glance that it is not required that $\beta_2 \neq \emptyset$, or in other words, that we are considering an *S4* prefix unifier rather than a *KD4* prefix unifier. Informally, the reason for this is that in the saturation of the *KD4.3'* background theory it is not required – at least what the first clause schema is concerned – that each of the functional decomposition variable sequences has to be non-empty. There is only a need for at least one of these sequences to be non-empty; for all the others we have to be able to unify γ and $\beta_1\!:\!\beta_2$ and this is realized in the first one of the two new inference rules.

Note that the linearity assumption gives rise to an interesting inference rule that might help reducing the search space considerably, namely

$$\frac{\neg R(\alpha_1, u\!:\!\beta_1), \ldots, \neg R(\alpha_n, u\!:\!\beta_n), C}{C}$$
$$\text{where } u \notin \alpha_i \text{ and } u \notin C$$

Its soundness follows from the fact that – in the antecedent – the interpretation of one of the α_i must be maximal (latest) and that u could be instantiated to an arbitrary time instant later than this maximum. This was not possible in case of *KD4* for there it could not be guaranteed that the α_i are linearly ordered.

LEMMA 5.4.4
The KD4.3' Inference System is sound.

Proof: There is certainly no problem with the rules of the *KD4* Inference System for nothing has changed compared with *KD4*.

Soundness of the second new inference rule follows immediately from the background theory clause

$$R(u,v) \vee u = v \vee R(v,u)$$

For the first new inference rule assume there is an *KD4.3'* interpretation $\Im$ which satisfies both $\neg R(\alpha,\beta_1\!:\!\beta_2) \vee C_1$ and $\neg R(\gamma,\delta) \vee C_2$ but is not a model for $\neg R(\sigma\alpha,\sigma\delta) \vee \sigma C_1 \vee \sigma C_2$ where σ is the most general unifier for γ and β_1. Then there exists a variant $\hat{\Im}$ of $\Im$

which satisfies $R(\sigma\alpha, \sigma\delta)$ as well as $\neg\sigma C_1$, $\neg\sigma C_2$, $\neg R(\alpha, \beta_1 : \beta_2) \vee C_1$ and $\neg R(\gamma, \delta) \vee C_2$. In particular

$$\hat{\Im} \models R(\sigma\alpha, \sigma\delta) \wedge \neg R(\sigma\alpha, \sigma\beta_1 : \sigma\beta_2) \wedge \neg R(\sigma\gamma, \sigma\delta)$$

$\hat{\Im}$ is an *KD4.3'* interpretation and we thus have

$$\hat{\Im} \models R(\sigma\alpha, \sigma\delta) \wedge \neg R(\sigma\alpha, \sigma\beta_1 : \sigma\beta_2) \wedge (R(\sigma\delta, \sigma\gamma) \vee \sigma\delta = \sigma\gamma)$$

The first literal and the final disjunction together yield $R(\sigma\alpha, \sigma\gamma)$ either directly or by transitivity of R and therefore

$$\hat{\Im} \models R(\sigma\alpha, \sigma\gamma) \wedge \neg R(\sigma\alpha, \sigma\beta_1 : \sigma\beta_2)$$

But now recall that σ is the most general unifier of γ and β_1. This means that we actually have

$$\hat{\Im} \models R(\sigma\alpha, \sigma\gamma) \wedge \neg R(\sigma\alpha, \sigma\gamma : \sigma\beta_2)$$

This now leads to a contradiction because in any *KD4.3'* interpretation it holds that $R(u, v) \Rightarrow R(u, v : x)$ and therefore we can derive $R(\sigma\alpha, \sigma\gamma : \overline{x})$ from $R(\sigma\alpha, \sigma\gamma)$ and thus in particular

$$\hat{\Im} \models R(\sigma\alpha, \sigma\gamma : \sigma\beta_2) \wedge \neg R(\sigma\alpha, \sigma\gamma : \sigma\beta_2)$$

Hence every *KD4.3'* interpretation that satisfies the antecedent of the second new inference rule also satisfies its consequent and we are done.

A common way to prove the ground completeness of a resolution–based calculus is by an induction over the number of excess literals in a given unsatisfiable set of clauses. For the modal and temporal logics considered so far this was not really necessary for there were always possibilities to reduce the (theory)-unsatisfiability to certain special cases or even to the classical unsatisfiability. This does not work anymore here, at least not that easily, and we have to come back to formerly known procedures. Unfortunately, the excess literal idea neither works for simultaneous paramodulation cannot be handled properly. As an example consider the ground clauses

$$P(a) \vee Q(a)$$
$$a = b$$
$$\neg P(a)$$
$$\neg Q(b)$$

Evidently, $Q(a)$, $a = b$, and $\neg Q(b)$ are inconsistent and can be refuted by performing a s-paramodulation step from $a = b$ into $Q(a)$ with result $Q(b)$ and deriving the empty clause from $Q(b)$ and $\neg Q(b)$. Unluckily, if this sequence of inference steps is applied to the original clause set from above then we not end up with $P(a)$ but with $P(b)$ and the induction hypothesis cannot be applied[4]. We therefore switch to an alternative – though closely related – ground completeness proof, namely one which consists of an induction over the number of *unsolved paths* in a given theory-unsatisfiable clause set. The main idea behind this proof technique can briefly be described as follows. A path through a set of n clauses consists of n literals, such that there is a literal from each given clause. Evidently, if each path is classically unsatisfiable then the whole set of clauses is classically unsatisfiable and can be refuted with standard means.

[4]In temporal logics to be considered later it might even happen that a non-unit clause can be derived.

We are therefore done if we are able to show that a theory–unsatisfiable set of paths can be transformed (by applying the given inference rules) into classically unsatisfiable paths. An induction over the number of paths would not help, since each inference step usually increases the number of paths through a given clause set. Nevertheless, it is possible to focus on certain special paths, the so-called *unsolved paths* instead. In showing that the application of inference rules sooner or later decreases the number of unsolved paths the induction will go through.

DEFINITION 5.4.5 (GROUND PATHS, COVERING PATHS)
By a ground path *(or just path) $\mathcal{L}$ (of length n) we understand a sequence of ground literals*

$$\langle L_1, \ldots, L_n \rangle$$

A path is called (classically) satisfiable *if the conjunction of the literals contained in this path is (classically) satisfiable*[5] *and* unsatisfiable *otherwise. A path $\mathcal{L}$* covers *a path $\mathcal{K}$ if all models of $\mathcal{K}$ satisfy $\mathcal{L}$.*

The following facts follow immediately from the above definition.

LEMMA 5.4.6

- *If the path $\mathcal{L} = \langle L_1, \ldots, L_n \rangle$ covers the path $\mathcal{K} = \langle K_1, \ldots, K_n \rangle$ and L_{n+1} is an arbitrary ground literal then $\langle L_1, \ldots, L_n, L_{n+1} \rangle$ covers $\langle K_1, \ldots, K_n, L_{n+1} \rangle$*
- *If the path $\mathcal{K}$ is covered by an unsatisfiable path $\mathcal{L}$ then $\mathcal{K}$ is unsatisfiable*
- *An unsatisfiable path is covered by any path*

A path which contains complementary literals needs no theory step. Also any path which is covered by some other path is not of any interest for, if the covering path gets extended such that it becomes classically unsatisfiable then the (extension) of the covered path must be unsatisfiable as well according to the above lemma. Covered paths therefore need no special treatment. There is an exception, however. If there are two (or more) paths that cover each other then one of these paths has to act as a representative for these paths unless there exists a further path which covers both but is not covered by them.

DEFINITION 5.4.7 (PATHS THROUGH $\mathcal{C}$, $\#UP(\mathcal{C})$, SOLVED PATHS)
Let $\mathcal{C} = \langle C_1, \ldots, C_n \rangle$ be a list of clauses. Then any path $\langle L_1, \ldots, L_n \rangle$ such that $L_i \in C_i$ for all $1 \le i \le n$ is called a path through $\mathcal{C}$.

A path through $\mathcal{C}$ is called solved *if it either contains complementary literals or it is covered by some other path through $\mathcal{C}$ which is not covered by the path under consideration. By $\#UP(\mathcal{C})$ we understand the number of unsolved paths through $\mathcal{C}$ under the restriction that any two paths that cover each other count as one.*

LEMMA 5.4.8
Let the clause C be obtained by a KD4.3' Inference System step on the (ground) clause set $\mathcal{C}$. Then $\#UP(\mathcal{C} \cup \{C\}) \le \#UP(\mathcal{C})$.

[5] In the sequel *satisfiablity* means classical satisfiability unless otherwise stated.

Proof: This is trivial in case of the *KD4* Inference Rule for consider the situation

$$\frac{\begin{array}{c}\neg R(\alpha, \alpha\!:\!\beta), C_1' \\ C_2 \\ \cdots \\ C_n\end{array}}{C_1'}$$

Evidently, each path $\langle L_1, L_2, \ldots, L_n, L_1' \rangle$ through $\mathcal{C} \cup \{C\}$ is covered by $\langle L_1', L_2, \ldots, L_n, L_1' \rangle$ which is also a path through $\mathcal{C} \cup \{C\}$. In particular, every path which contains the $\neg R(\alpha, \alpha\!:\!\beta)$ gets covered and therefore it is even the case that $\#UP(\mathcal{C} \cup \{C\}) < \#UP(\mathcal{C})$.

In case of an s-paramodulation step we are faced with the following situation

$$\frac{\begin{array}{c}\alpha = \beta, C_1' \\ C_2 \\ \cdots \\ C_n\end{array}}{C_2\,[\alpha/\beta]\,, C_1'}$$

where $C_2\,[\alpha/\beta]$ is meant to express that any occurrence of α in C_2 is replaced by β. Now let $\mathcal{L} = \langle L_1, \ldots, L_n, L_{n+1} \rangle$ be a path through $\mathcal{C} \cup \{C\}$. If $L_1 \in C_1'$ then $\langle L_1, \ldots, L_n, L_1 \rangle$ covers $\mathcal{L}$ unless $L_{n+1} = L_1$ and if $L_1 = \alpha = \beta$ then $\langle L_1, L_2, \ldots, L_n, L_2\,[\alpha/\beta] \rangle$ covers $\mathcal{L}$ unless $L_{n+1} = L_2\,[\alpha/\beta]$. In either case the number of unsolved paths is at least not increased.

Remain the cases of the resolution rule and the new *KD4.3'* Inference Rules. The situation we then have is as follows

$$\frac{\begin{array}{c}L_1, C_1' \\ L_2, C_2' \\ \cdots \\ C_n\end{array}}{R, C_1', C_2'}$$

where R denotes the residue of the rule application (in case of a resolution rule application R is just *false*). Let $\mathcal{L} = \langle L_1, \ldots, L_n, L_{n+1} \rangle$ be a path through $\mathcal{C} \cup \{C\}$. Evidently, if $L_1 \in C_1'$ then $\langle L_1, \ldots, L_n, L_1 \rangle$ covers $\mathcal{L}$ unless $L_{n+1} = L_1$. Similarly, if $L_2 \in C_2'$ then $\langle L_1, L_2, \ldots, L_n, L_2 \rangle$ covers $\mathcal{L}$ unless $L_{n+1} = L_2$. This also works the other way round, i.e. if L_{n+1} is not R then $\mathcal{L}$ is covered by either $\langle L_{n+1}, L_2, \ldots, L_n, L_{n+1} \rangle$ or $\langle L_1, L_{n+1}, \ldots, L_n, L_{n+1} \rangle$. Thus the number of unsolved paths is again not increased.

LEMMA 5.4.9
A KD4.3'-unsatisfiable path $\mathcal{L}$ can be refuted by the KD4.3' Inference System.

Proof: This should be obvious if $\mathcal{L}$ contains no E-cycle[6]. On the other hand, if $\mathcal{L}$ contains an E-cycle of length n then this E-cycle can be used to derive $n-1$ equations (this can be shown by induction over n). Moreover, this E-cycle won't be necessary anymore then and the number of literals in $\mathcal{L}$ is reduced by one. Hence, the lemma follows by induction on the number of E-cycles contained in $\mathcal{L}$.

[6]Recall that an E-cycle is a set of negative R-literals from which an equation can be derived. Also note that it is never necessary to apply an equation to an element of the background theory since such an equation can only be obtained from the background theory which is supposed to be saturated.

LEMMA 5.4.10
The KD4.3' Inference System is refutation complete on ground clauses.

Proof: This follows by induction on the number of unsolved paths $\#UP(\mathcal{C})$ through the given clause set $\mathcal{C}$. In case of $\#UP(\mathcal{C}) = 0$ we have that $\mathcal{C}$ is classically unsatisfiable and therefore can be refuted. In case of $\#UP(\mathcal{C}) > 0$ recall that any inference step on a unsolved path does not increase the total number of unsolved paths. However, according to Lemma 5.4.9, any unsolved (unsatisfiable) path can be refuted and therefore – sooner or later – this number will decrease.

LEMMA 5.4.11
The KD4.3' Inference System is refutation complete.

Proof: In analogy to the corresponding proof for the *S4.3* Inference System, i.e. we delay unliftable *KD4* inference steps and replace *KD4.3'* Inference Rule applications on the ground instances of two clauses

$$\begin{gathered}\neg R(\alpha, u\!:\!\beta), C_1\\ \neg R(\gamma, \delta), C_2\end{gathered}$$

which are not liftable by the following inference step

$$\frac{\begin{gathered}\neg R(\alpha, u\!:\!\beta), C_1\\ \neg R(\gamma, \delta), C_2\end{gathered}}{\neg R(\alpha, \delta), \neg R(\gamma, u\!:\!\beta), C_1, C_2}$$

where the corresponding ground instance of $\neg R(\gamma, u\!:\!\beta)$ is *KD4*-unsatisfiable. Therefore, the original *KD4.3'* Inference Rule application can be simulated by a step of the alternative inference rule immediately followed by a *KD4* Inference Rule application. Moreover, each application of the alternative inference rule can be lifted and therefore it is possible to derive (on the ground level as well as on the non-ground level) a single *KD4*-unsatisfiable clause. Now note that this *KD4*-unsatisfiable clause can be refuted and this means that there is at least one R-literal in this clause on which a *KD4* Inference Rule application is possible. This possible step could have been performed already after this literal had been generated by the alternative inference rule and therefore an induction on the length of the unsatisfiable clause (or induction over the number of alternative inference rule applications) the refutation obtained can ultimatly be transformed into a refutation within the *KD4.3'* Inference System.

Note that the first of the two new *KD4.3'* Inference Rules could be further optimized. Consider the following set of clauses

$$\begin{gathered}\neg R(\alpha, \beta\!:\!\gamma), C_1\\ \neg R(\beta, \delta), C_2\\ \neg R(\delta, \alpha), C_3\end{gathered}$$

Evidently, it is possible to derive the clause C_1, C_2, C_3 from this clause set. However, after applying the first new inference rule to the first two clauses we would end up with $\neg R(\alpha, \delta), C_1, C_2$ and this result together with the third clause from above would yield $\alpha = \delta, C_1, C_2, C_3$ instead of what we originally expected. The reason for this is pretty obvious: After we performed the first inference step we lost the information that we are actually dealing with a proper cycle rather than with an equation introducing cycle. There are two ways to avoid such a problem. One possibility would be to change the order of inference steps. In the above example this would mean to start with the second and third clause – obtaining $\neg R(\beta,\alpha), C_2, C_3$ – and then add the first clause (with result $\neg R(\beta,\beta\!:\!\gamma), C_1, C_2, C_3)$. This finally leads to derive C_1, C_2, C_3 just as desired.

Another, maybe more sensible way of dealing with such possibilities would be to change the first of the two new inference rules to

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg R(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is the most general unifier of γ and β. This rule alone evidently would not be sufficient. We therefore have to add further rules of the form

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg S(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is a *KD4* prefix unifier of the pair (γ,β)

$$\frac{\begin{array}{c}\neg S(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg S(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is a *S4* prefix unifier of the pair (γ,β)

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg S(\gamma,\delta), C_2\end{array}}{\neg S(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is a *S4* prefix unifier of the pair (γ,β)

$$\frac{\begin{array}{c}\neg S(\alpha,\beta), C_1\\ \neg S(\gamma,\delta), C_2\end{array}}{\neg S(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is a *S4* prefix unifier of the pair (γ,β)

$$\frac{\neg S(\alpha,\beta), C_1}{\sigma C_1, \sigma C_2}$$

where σ is a *S4* prefix unifier of the pair (α,β)

where the predicate symbol S is new to the whole clause set. The close relationship between the $\neg S$ and the $\neg R$ is pretty evident. The main difference between these two is that the S contains the information that is has been obtained by a step which promises a proper cycle. Therefore, although this alternative Inference System contains more rules than what we originally had, it forbits some steps which are unnecessary but nevertheless possible in the *KD4.3'* Inference System.

5.4.2 The Linear Version of K_tD4

As we now know how to deal with the linearity property for the modal logic *KD4* we can easily apply the very same ideas to the next complicated of the temporal logics we consider, namely K_tD4. As a matter of fact, a calculus for the linear version of K_tD4 (which we shall call $K_tD4.3'$ in the sequel) can be easily obtained from the calculus for *KD4.3'*.

Let us first have a look at the axiomatization of $K_tD4.3'$. In addition to the standard necessitation rules and Modus Ponens we consider the following axiom schemata:

$$
\begin{array}{l}
\boxed{F}\,\Phi \Rightarrow \Diamond_{F}\,\Phi \\
\boxed{F}\,\Phi \Rightarrow \boxed{F}\boxed{F}\,\Phi \\
\boxed{P}\,\Phi \Rightarrow \Diamond_{P}\,\Phi \\
\boxed{P}\,\Phi \Rightarrow \boxed{P}\boxed{P}\,\Phi \\
\Phi \Rightarrow \boxed{F}\Diamond_{P}\,\Phi \\
\Phi \Rightarrow \boxed{P}\Diamond_{F}\,\Phi \\
\Diamond_{F}\,\Phi \wedge \Diamond_{F}\,\Psi \Rightarrow \Diamond_{F}(\Phi \wedge \Diamond_{F}\,\Psi) \vee \Diamond_{F}(\Phi \wedge \Psi) \vee \Diamond_{F}(\Diamond_{F}\,\Phi \wedge \Psi) \\
\Diamond_{P}\,\Phi \wedge \Diamond_{P}\,\Psi \Rightarrow \Diamond_{P}(\Phi \wedge \Diamond_{P}\,\Psi) \vee \Diamond_{P}(\Phi \wedge \Psi) \vee \Diamond_{P}(\Diamond_{P}\,\Phi \wedge \Psi)
\end{array}
$$

Note that there are two linearity axioms to be included, one for the past and one for the future fragment of the temporal logic under consideration, however, not both of the transitivity axioms are really needed; in the presence of the other schemata either one follows from the respective mirror image.

The background theory we obtain from this axiomatization then is:

$$
\begin{array}{l}
R(u, u\!:\!x_F) \\
R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\
R(u\!:\!x_P, u) \\
R(u,v) \wedge R(u,w) \Rightarrow R(v,w) \vee v = w \vee R(w,v) \\
R(u,w) \wedge R(v,w) \Rightarrow R(u,v) \vee u = v \vee R(v,u)
\end{array}
$$

where R denotes the accessibility relation R_F and the relation R_P is described in terms of R_F. As in the case *KD4.3'* these linearity axioms can be simplified and this will be shown in the following lemma.

LEMMA 5.4.12
The two linearity axioms in the background theory for $K_tD4.3$' can be simplified to the single clause

$$R(u,v) \vee u = v \vee R(v,u)$$

Proof: It can fairly easy be shown from the background theory (e.g. with the help of a standard theorem prover) that

$$R(u,v) \vee u = v \vee R(v,u) \Rightarrow \begin{cases} R(u, v\!:\!x_F) \vee u = v\!:\!x_F \vee R(v\!:\!x_F, u) \\ R(u\!:\!x_P, v) \vee u\!:\!x_P = v \vee R(v, u\!:\!x_P) \end{cases}$$

The following clause is clearly valid as well

$$R(\iota,\iota) \vee \iota = \iota \vee R(\iota,\iota)$$

for it denotes a trivial tautology. Both together therefore yield (by a simple induction)

$$R(\iota\!:\!\overline{x}, \iota\!:\!\overline{y}) \vee \iota\!:\!\overline{x} = \iota\!:\!\overline{y} \vee R(\iota\!:\!\overline{y}, \iota\!:\!\overline{x})$$

where $\overline{x}, \overline{y} \in (F_{R_P} \cup F_{R_F})^*$.

Now recall that the connectedness assumption guarantees that any time instant u can be represented in the form $u = \iota\!:\!\overline{z}$ with $\overline{z} \in (F_{R_P} \cup F_{R_F})^{*}$[7]. Hence the derived clause schema may be simplified to

$$R(u,v) \vee u = v \vee R(v,u)$$

and we are done.

LEMMA 5.4.13
The saturation of the K$_t$D4.3' background theory can be described by the following clause schemata:

$$R(u_1\!:\!\overline{x_1}, u_2\!:\!\overline{y_2}) \vee R(u_2\!:\!\overline{x_2}, u_3\!:\!\overline{y_3}) \vee \ldots \vee R(u_n\!:\!\overline{x_n}, u_1\!:\!\overline{y_1})$$
with $n \geq 1$, $\overline{x_i} \in (F_{R_P})^*$, $\overline{y_i} \in (F_{R_F})^*$, and not all $\overline{x_i}, \overline{y_j}$ are empty

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \ldots \vee R(u_n,u_1) \vee u_2 = u_1$$
with $n \geq 2$

Proof: First note that from the background theory it follows that $R(u,v) \Rightarrow R(u,w) \vee R(w,v)$ (this can easily be checked by any standard theorem prover). From this it follows that

$$\left.\begin{array}{l} R(u\!:\!x_P, u) \\ R(u,v) \Rightarrow R(u,w) \vee R(w,v) \end{array}\right\} \Rightarrow \begin{array}{r} R(u_1\!:\!x_P, u_2) \vee \ldots \vee R(u_{n-1},u_n) \vee R(u_n,u_1) \\ \text{with } n \geq 1 \end{array}$$

and similarly

$$\left.\begin{array}{l} R(u, u\!:\!x_F) \\ R(u,v) \Rightarrow R(u,w) \vee R(w,v) \end{array}\right\} \Rightarrow \begin{array}{r} R(u_1, u_2\!:\!x_F) \vee \ldots \vee R(u_{n-1},u_n) \vee R(u_n,u_1) \\ \text{with } n \geq 1 \end{array}$$

Therefore we have that

$$\left.\begin{array}{l} R(u_1\!:\!x_P, u_2) \vee \ldots \vee R(u_n,u_1) \\ R(u_1, u_2\!:\!x_F) \vee \ldots \vee R(u_n,u_1) \\ R(u,v) \Rightarrow R(u, v\!:\!x_F) \\ R(u,v) \Rightarrow R(u\!:\!x_P, v) \end{array}\right\} \Rightarrow \begin{array}{l} R(u_1\!:\!\overline{x_1}, u_2\!:\!\overline{y_2}) \vee \ldots \vee R(u_n\!:\!\overline{x_n}, u_1\!:\!\overline{y_1}) \\ \quad \text{with } n \geq 1,\ \overline{x_i} \in (F_{R_P})^*,\ \overline{y_i} \in (F_{R_F})^* \\ \quad \text{and for some } \overline{x_i}, \overline{y_j} \neq \emptyset \end{array}$$

and this covers the first clause schema.

For the second clause schema consider both

$$\begin{array}{c} R(u,v), R(v,u), u = v \\ \neg R(u,v), R(u,w), R(w,v) \end{array}$$

In case of $n = 2$ we are already done. For greater n assume we have already shown that

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \ldots \vee R(u_{n-1},u_1) \vee u_2 = u_1$$

A resolution step between $R(u_{n-1},u_1)$ and $\neg R(u,v), R(u,w), R(w,v)$ then leads to

$$R(u_1,u_2) \vee R(u_2,u_3) \vee \ldots \vee R(u_n,u_1) \vee u_2 = u_1$$

[7]Note that the connectedness assumption requires that the variables in $\overline{z}$ may belong to *any* of the functional decompositions of the accessibility relations in question.

and we are done by induction on n.

In order to guarantee that these are indeed all clauses that can be obtained we have to show that the clause schemata are closed under transitivity. This is again obvious whenever the first clause schema is involved, i.e. anything that can be derived is either already contained or subsumed by the first clause schema. The second clause schema is identical to the second clause schema in case of the logic *KD4.3'* and therefore nothing changes in this case and we are done.

DEFINITION 5.4.14 (THE $K_tD4.3$' INFERENCE SYSTEM)
The $K_tD4.3$' Inference System *consists of the classical resolution, factorization and paramodulation rules together with the K_tD4 Inference System and the following new inference rules:*

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\neg R(\sigma\alpha,\sigma\delta), \sigma C_1, \sigma C_2}$$

where σ is a K_tT4 prefix unifier of (γ,β)

$$\frac{\begin{array}{c}\neg R(\alpha,\beta), C_1\\ \neg R(\gamma,\delta), C_2\end{array}}{\sigma\beta = \sigma\alpha, \sigma C_1, \sigma C_2}$$

where σ is the most general unifier of (α,δ) and (γ,β)

Note that the close relationship between *KD4.3'* and $K_tD4.3$' is mirrored by the similarities between the respective background theory saturations and the corresponding Inference Systems. It is therefore not very surprising that the soundness and the refutation completeness proofs for the $K_tD4.3$' Inference System are almost identical to the corresponding proofs in case of *KD4.3'*. We therefore omit such a repetition and just fix the final result in a theorem.

THEOREM 5.4.15
The $K_tD4.3$' Inference System is sound and refutation complete.

5.4.3 Linearity plus *S4⊕KD4*

In the earlier sections concerning the linearity assumption we examined only rather simple modal and temporal logics. Now before we start with the development of a resolution based calculus for more complicated linear temporal logics we should try to find out how this linearity influences the saturation of the background theory for "mixed" modalities in the sense of *S4⊕KD4*. To this end we proceed as we did for the other logics we considered, i.e. we translate the Hilbert style axiomatization into a set of accessibility relation properties, simplify this set if possible, determine its saturation (under the semi-functional translation), and finally cast this saturation into some suitable inference rules.

To this end let us first have a look at the full axiomatization of *S4⊕KD4* under linearity

(*S4.3⊕KD4.3'* in the sequel).

$$\begin{array}{l}
\Box^*\Phi \Rightarrow \Phi \\
\Box^*\Phi \Rightarrow \Box^*\Box^*\Phi \\
\Diamond^*\Phi \wedge \Diamond^*\Psi \Rightarrow \Diamond^*(\Phi \wedge \Diamond^*\Psi) \vee \Diamond^*(\Diamond^*\Phi \wedge \Psi) \\
\Box\Phi \Rightarrow \Diamond\Phi \\
\Box\Phi \Rightarrow \Box\Box\Phi \\
\Diamond\Phi \wedge \Diamond\Psi \Rightarrow \Diamond(\Phi \wedge \Diamond\Psi) \vee \Diamond(\Phi \wedge \Psi) \vee \Diamond(\Diamond\Phi \wedge \Psi) \\
\Box^*\Phi \Leftrightarrow \Box\Phi \wedge \Phi
\end{array}$$

The background theory induced by this axiomatization is then given by

$$\begin{array}{l}
R^*(u, u\!:\!x^*) \\
R^*(u, u) \\
R^*(u, v) \wedge R^*(v, w) \Rightarrow R^*(u, w) \\
R^*(u, v) \wedge R^*(u, w) \Rightarrow R^*(v, w) \vee R^*(w, v) \\
R(u, u\!:\!x) \\
R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \\
R(u, v) \wedge R(u, w) \Rightarrow R(v, w) \vee v = w \vee R(w, v) \\
R(u, v) \Rightarrow R^*(u, v) \\
R^*(u, v) \Rightarrow R(u, v) \vee u = v
\end{array}$$

Evidently, some redundancies are contained in this property set. So, for instance, the transitivity of R^* is not necessary since it follows already from the transitivity of R and the fact that R^* denotes the reflexive closure of R. Moreover, the right-linearity clauses can be simplified:

LEMMA 5.4.16
Under the connectedness assumption the right-linearity clause R^* can be simplified to

$$R^*(u, v) \vee R^*(v, u)$$

and the right-linearity clause for R is redundant.

Proof: It is easy to check that

$$R^*(u, v) \vee R^*(v, u) \Rightarrow \begin{cases} R^*(u\!:\!x, v) \vee R^*(v, u\!:\!x) \\ R^*(u, v\!:\!x) \vee R^*(v\!:\!x, u) \\ R^*(u\!:\!x^*, v) \vee R^*(v, u\!:\!x^*) \\ R^*(u, v\!:\!x^*) \vee R^*(v\!:\!x^*, u) \end{cases}$$

Moreover we have that $R^*(\iota, \iota) \vee R^*(\iota, \iota)$ by the reflexivity of R^* and we therefore get by a simple induction that

$$R^*(\iota\!:\!\overline{x}, \iota\!:\!\overline{y}) \vee R^*(\iota\!:\!\overline{y}, \iota\!:\!\overline{x})$$

where $\overline{x}, \overline{y} \in (F_R \cup F_{R^*})^*$. Hence the connectedness assumption can be applied and we end up with the linearity clause for R^*

$$R^*(u, v) \vee R^*(v, u)$$

Now, two resolution steps between this linearity clause and $R^*(u, v) \Rightarrow R(u, v) \vee u = v$ result in

$$R(u, v) \vee u = v \vee R(v, u)$$

and therefore the right-linearity clause for R is redundant.

LEMMA 5.4.17
The saturation of the background theory for S4.3⊕KD4.3' is determined by

$$\begin{array}{l} R^*(u_1, u_2 : \overline{x_2}) \vee R^{(*)}(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R^{(*)}(u_{n-1}, u_n : \overline{x_n}) \vee R^{(*)}(u_n, u_1 : \overline{x_1}) \\ \qquad \textit{with } n \geq 1,\ \overline{x_i} \in (F_R \cup F_{R^*})^* \textit{ for all } 1 \leq i \leq n \\ R(u_1, u_2 : \overline{x_2}) \vee R(u_2, u_3 : \overline{x_3}) \vee \ldots \vee R(u_{n-1}, u_n : \overline{x_n}) \vee R(u_n, u_1 : \overline{x_1}) \\ \qquad \textit{with } n \geq 1,\ \overline{x_i} \in (F_R \cup F_{R^*})^* \textit{ for all } 1 \leq i \leq n,\ \overline{x_j} \notin (F_{R^*})^* \textit{ for some } 1 \leq j \leq n \\ R(u_1, u_2 : \overline{x_2} : \overline{y_2}) \vee \ldots \vee R(u_{n-1}, u_n : \overline{x_n} : \overline{y_n}) \vee R(u_n, u_1 : \overline{x_1} : \overline{y_1}) \vee u_i : \overline{x_i} = u_j : \overline{x_j} \\ \qquad \textit{with } n \geq 1,\ 1 \leq i, j \leq n, \textit{ and } \overline{x_k} : \overline{y_k} \in (F_{R^*})^* \textit{ for all } 1 \leq k \leq n \end{array}$$

where an $R^{(*)}$-literal arbitrarily denotes either an R- or an R^*-literal.

Proof: It is easy to check (e.g. with the help of a standard theorem prover) that each clause of the form

$$R^{(*)}(u, v) \Rightarrow R^{(*)}(u, w) \vee R^{(*)}(w, v)$$

which is not just $R^*(u, v) \Rightarrow R(u, w) \vee R(w, v)$ follows from the background theory. Moreover, it holds that each of the clauses

$$R^{(*)}(u, v) \Rightarrow R^{(*)}(u, v : x^{(*)})$$

(unless it is $R^*(u, v) \Rightarrow R(u, v : x^*)$) also follows from the background theory. Both together – in addition with the reflexivity of R^* – therefore lead to any of the first two clause schemata from above by a simple induction argument.

Similarly we can prove that any instance of the third clause schema is derivable. To this end it suffices to show that

$$R^*(u, v) \Rightarrow R(u, w) \vee R(w, v) \vee u = w$$

holds within the background theory. This – together with $R(u, v) \Rightarrow R(u, w) \vee R(w, v)$ and $R^*(u, u)$ – then guarantees the derivability of

$$R(u_1, u_2) \vee \ldots R(u_n, u_1) \vee u_2 = u_1$$

and therefore also of

$$R(u_1, u_2) \vee \ldots R(u_n, u_1) \vee u_i = u_1$$

This results and the clause $R(u, v) \Rightarrow R(u, v : x^*)$ finally results in any instance of the third clause schema.

In order to show that these schemata cover everthing that is derivable from the background theory we have to check whether they are closed under

$$\begin{array}{l} R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \\ R^*(u, v) \Rightarrow R(u, v) \vee u = v \\ R(u, v) \Rightarrow R^*(u, v) \end{array}$$

This is trivial in case of $R^*(u, v) \Rightarrow R(u, v) \vee u = v$ and $R(u, v) \Rightarrow R^*(u, v)$ and also very easy whenever one of the first two schemata is involved. Remains to be shown that the third clause schema is closed under transitivity. To this end let us abbreviate

$$R(u_1, u_2 : \overline{x_2} : \overline{y_2}) \vee \ldots \vee R(u_n, u_1 : \overline{x_1} : \overline{y_1})$$

by Φ_1 and

$$R(v_1, v_2 : \overline{z_2} : \overline{z_2}') \vee \ldots \vee R(v_m, v_1 : \overline{z_1} : \overline{z_1}')$$

by Φ_2. Now let $\sigma = \{v_l / u_{k+1} : \overline{x_{k+1}} : \overline{y_{k+1}}\}$ and let Φ_3 denote[8]

$$\begin{array}{l} R(u_1, u_2 : \overline{x_2} : \overline{y_2}) \vee \ldots \vee R(u_{k-1}, u_k : \overline{x_k} : \overline{y_k}) \vee \\ \quad R(u_{k+1}, u_{k+2} : \overline{x_{k+2}} : \overline{y_{k+2}}) \vee \ldots \vee R(u_{n-1}, u_n : \overline{x_n} : \overline{y_n}) \vee R(u_n, u_1 : \overline{x_1} : \overline{y_1}) \vee \\ \quad R(u_k, v_{l+1} : \overline{z_{l+1}} : \overline{z_{l+1}}') \vee R(v_{l+1}, v_{l+2} : \overline{z_{l+2}} : \overline{z_{l+2}}') \vee \ldots \vee R(v_m, v_1 : \overline{z_1} : \overline{z_1}') \vee \\ \quad R(v_1, v_2 : \overline{z_2} : \overline{z_2}') \vee \ldots \vee R(v_{l-2}, v_{l-1} : \overline{z_{l-1}} : \overline{z_{l-1}}') \vee R(v_{l-1}, \sigma(v_l) : \overline{z_l} : \overline{z_l}') \end{array}$$

Φ_3 looks rather complicated but nevertheless it is pretty obvious that $\Phi_3 \vee u_{i_1} : \overline{x_{i_1}} = u_{j_1} : \overline{x_{j_1}}$ belongs to the third clause schema. Now we perform two resolution steps between the transitivity clause and both $\Phi_1 \vee u_{i_1} : \overline{x_{i_1}} = u_{j_1} : \overline{x_{j_1}}$ and $\Phi_2 \vee v_{i_2} : \overline{z_{i_2}} = v_{j_2} : \overline{z_{j_2}}$. This results in

$$\Phi_3 \vee u_{i_1} : \overline{x_{i_1}} = u_{j_1} : \overline{x_{j_1}} \vee \sigma(v_{i_2}) : \overline{z_{i_2}} = \sigma(v_{j_2}) : \overline{z_{j_2}}$$

which is subsumed by $\Phi_3 \vee u_{i_1} : \overline{x_{i_1}} = u_{j_1} : \overline{x_{j_1}}$. Hence, nothing new can be gained from the transitivity and we are done.

Note that the third clause schema can equally be replaced by the two schemata

$$R(u_1, u_2 : \overline{y_2}) \vee R(u_2, u_3 : \overline{y_3}) \vee \ldots \vee R(u_{n-1}, u_n : \overline{y_n}) \vee R(u_n, u_1 : x_1^* \overline{y_1}) \vee u_1 : x_1^* = u_1$$

$$R(u_1, u_2) \vee \ldots \vee R(u_n, u_1), u_2 = u_1$$

Both are instances of the third schema but also, both together imply every instance of the third schema.

At this stage a little example seems convenient which shows how powerful the equality introduction has to be in order to form a complete set of inference rules.

EXAMPLE 5.4.18
Consider the two unit clauses

$$\begin{array}{l} \neg R(\iota : a, \iota : b : c^* : d^*) \\ \neg R(\iota : b, \iota : a : e^*) \end{array}$$

Informally they claim that $\iota : a$ is later than or equal to $\iota : b : c^* : d^*$ ($\iota : a \geq \iota : b : c^* : d^*$ for short) and that $\iota : b \geq \iota : a : e^*$. We know that $\iota : a : e^* \geq \iota : a$ and that $\iota : b : c^* : d^* \geq \iota : b$. Therefore, if one of the c^*, d^*, e^* were not the identity on its argument then we would have a contradiction. Thus we have that all of the following terms are equal

$$\begin{array}{l} \iota : a \\ \iota : a : e^* \\ \iota : b \\ \iota : b : c^* \\ \iota : b : c^* : d^* \end{array}$$

and we have to be able to derive this fact, or actually, we should be able to derive the empty clause from, say, $P(\iota : a : e^*)$ and $\neg P(\iota : b : c^*)$. Obviously, the third schema from above covers all

[8]Note that a subscript of the form $k + i$ is implicitly assumed to be *modulo* n and, similarly, a subscript of the form $l + i$ is implicitly assumed to be *modulo* m.

these possibilities. This shows that this schema is very powerful, but also that its transformation into a suitable inference rule would open quite a big search space. However, if we used the alternative equality introductions instead then there would be just two possible outcomes for the first application, namely $\iota : a : e^* = \iota : a$ and $\iota : b : c^* = \iota : b$. These two equations reduce the original set of unit clauses to

$$\begin{array}{l} \neg R(\iota : a, \iota : b : d^*) \\ \neg R(\iota : b, \iota : a) \\ P(\iota : a) \\ \neg P(\iota : b) \end{array}$$

Now a further equality introduction step can be applied to the first two R-literals, this time with the result $\iota : b : d^* = \iota : b$ which further reduces the first unit clause from above, yielding $\neg R(\iota : a, \iota : b)$. Up to now only the first of the two new schemata had come into play and what we gained is a possibility to apply the second new equality introduction which results in $\iota : a = \iota : b$. Now this – together with the two P-literals – immediately leads to the empty clause and we are done.

What this example shows is that the alternative equality introductions make it possible to get rid of unnecessary symbols step–by–step, whereas the more complicated clause schema which we derived by saturating the background theory provides all possibilities at once (whether they are needed or not).

In the above example we talked about inference rules as if we had them already. The way how they ought to be defined should be obvious, however.

DEFINITION 5.4.19 (THE *S4.3⊕KD4.3'* INFERENCE SYSTEM)
The S4.3⊕KD4.3' Inference System *consists of the S4⊕KD4 Inference System together with the following inference rules*

$$\frac{\begin{array}{c}\neg R^{(*)}(\alpha, \beta_1 : \beta_2), C_1 \\ \neg R^{(*)}(\gamma, \delta), C_2\end{array}}{\neg R^*(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2}$$

where $\sigma = mgu(\gamma, \beta_1)$ and either $\beta_2 \notin (F_{R^*})^*$ or at least one of the parent literals is an R^*-literal

$$\frac{\begin{array}{c}\neg R(\alpha, \beta_1 : \beta_2), C_1 \\ \neg R(\gamma, \delta), C_2\end{array}}{\neg R(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2}$$

where $\sigma = mgu(\gamma, \beta_1)$ and $\beta_2 \in (F_{R^*})^*$

$$\frac{\begin{array}{c}\neg R(\alpha, \beta), C_1 \\ \neg R(\gamma, \delta), C_2\end{array}}{\sigma\alpha = \sigma\beta, \sigma C_1, \sigma C_2}$$

$\sigma = mgu((\gamma, \beta), (\alpha, \delta))$

Soundness and completeness is now proved along the lines of the corresponding proofs for *S4⊕KD4*, *S4.3* and *KD4.3'*.

LEMMA 5.4.20
The S4.3⊕KD4.3' Inference System is sound.

Proof: Soundness of the third new inference rule follows immediately from the background theory clause $R(u, v) \lor u = v \lor R(v, u)$.

For the second new inference rule recall that any *S4.3⊕KD4.3'* interpretation satisfies $R^*(u,v) \vee R(v,u)$ as well as $R(u,v) \wedge R^*(v,w) \Rightarrow R(u,w)$ and $R(u,v) \Rightarrow R(u,v\!:\!\overline{x})$. Now assume there were a *S4.3⊕KD4.3'* interpretation which is a model for $\neg R(\sigma\alpha, \sigma\gamma\!:\!\sigma\beta_2)$ and $\neg R(\sigma\gamma, \sigma\delta)$ and which also satisfies $R(\sigma\alpha, \sigma\delta)$. This model would satisfy $R^*(\sigma\delta, \sigma\gamma)$ and therefore also $R(\sigma\alpha, \sigma\gamma)$ and even $R(\sigma\alpha, \sigma\gamma\!:\!\overline{x})$. But this contradicts the assumption that the interpretation under consideration satisfies $\neg R(\sigma\alpha, \sigma\gamma\!:\!\sigma\beta_2)$ and therefore the second new inference rule is sound.

Soundness of the first new inference rule follows analogously; the proof is just as easy as for the other rules and is therefore omitted here.

The first of the three new inference rules might require a little bit of an explanation. The remarkable thing about it is that an R^*-literal gets derived even though the parent clauses do not necessarily contain R^*-literals. In this case, however, this inference rule can only be applied if one of the elements of β_2 belongs to the functional decomposition of R. Now, this rule is supposed to reduce a potential proper cycle by one, i.e. to some extent it has to simulate the first of the clause schemata that have been derived for *S4.3⊕KD4.3'*. Now consider the possible instances of this very schema in which no R^*-literal occurs and for all but one, say L, of the sequences it is the case that every decomposition term consists of elements from F_{R^*}. The rule might now be applied to L and another clause and if we did derive a R-literal instead of an R^*-literal we would not be able to detect that we are actually dealing with a proper cycle instead of one that induces some equality. Deriving the R^*-literal safes us from such a mistake.

LEMMA 5.4.21
The S4.3⊕KD4.3' Inference System is refutation complete.

Proof: Ground completeness can be shown along the lines of Section 5.4.1. I.e. it has to be shown that a *S4.3⊕KD4.3'*-unsatisfiable set of unit clauses can be refuted and that the application of any inference rule does not increase the number of unsolved paths through the unsatisfiable clause set under consideration. Therefore any not yet solved path can eventually be refutes and hence the number of unsolved paths is eventually decreased which means that a (classically) unsatisfiable set of clauses can be derived. Since every (classically) unsatisfiable set of clauses can be refuted the empty clause can be derived from an arbitrary *S4.3⊕KD4.3'*-unsatisifiable set of ground clauses then. Now, showing that no inference rule application does increase the number of unsolved paths is in fact very easy and can be exemplified on one of the new inference rules (the corresponding proofs for the other new inference rules are similar). Let us consider the two clauses in the given clause set

$$\neg R^{(*)}(\alpha, \beta\!:\!\gamma), C_1'$$
$$\neg R^{(*)}(\beta, \delta), C_2'$$

with further clauses $C_3, \ldots, C_n$ such that the first new inference rule can be applied (i.e. either there exists a $\gamma_i \in F_R$ or at least one of the two $R^{(*)}$-literals is in fact a R^*-literal). Let $\langle L_1, \ldots, L_n\rangle$ be an aritrary path through $\langle C_1, \ldots, C_n\rangle$. The result of applying the first new inference rule to C_1 and C_2 then is

$$C_{n+1} = \neg R^*(\alpha, \delta), C_1', C_2'$$

Evidently, if $L_1 \in C_1'$ or if $L_{n+1} \in C_1'$ then each path $\langle L_1, \ldots, L_n, L_{n+1}\rangle$ is covered by $\langle L_1, \ldots, L_n, L_1\rangle$ ($\langle L_{n+1}, \ldots, L_n, L_{n+1}\rangle$ respectively). Also, if $L_2 \in C_2'$ or if $L_{n+1} \in C_2'$ then each path $\langle L_1, L_2, \ldots, L_n, L_{n+1}\rangle$ is covered by $\langle L_1, L_2, \ldots, L_n, L_2\rangle$ ($\langle L_1, L_{n+1}, \ldots, L_n, L_{n+1}\rangle$ respectively). In all other cases $\langle L_1, \ldots, L_n\rangle$ is just extended by $\neg R^*(\alpha, \delta)$ and therefore the number of unsolved paths in not increased.

In order to show that a single *S4.3⊕KD4.3'*-unsatisfiable path can be refuted consider $\mathcal{L} = \langle L_1, \ldots, L_n\rangle$. If $\mathcal{L}$ is classically unsatisfiable then there is no problem. Therefore assume that $\mathcal{L}$ has a classical model but nevertheless is *S4.3⊕KD4.3'*-unsatisfiable. W.l.o.g. we can assume that any equations in $\mathcal{L}$ can be ignored (they can be used to reduce the literals and we can assume that all possible reductions have been performed). By assumption there exists a finite number of instances of the clause schemata such that $\mathcal{L}$ together with these clauses is classically unsatisfiable. If no instance of the third clause schema is needed then there is no problem at all. Hence assume that at least one instance of the third clause schema is required in order to (classically) refute $\mathcal{L}$. It suffices then to show that an equation can be derived since such an equation can be used to reduce the literals in $\mathcal{L}$ and this can happen only finitly often so that – eventually – no further instance of the third clause schema is required. Recall that an equation can be derived if there is an instance of the third clause schema, say $\mathcal{C}$, such that for every R-literal in $\mathcal{C}$ the respective complement is contained in $\mathcal{L}$. Now assume that no such $\mathcal{C}$ does exist. Then each instance (and there is at least one) of the third clause schema contains an R-literal whose complement is not contained in $\mathcal{L}$. Hence, if we extend $\mathcal{L}$ by each such element we obtain a (classically) satisfiable path through a (classically) unsatisfiable set of clauses which is impossible. Therefore at least one equation can be derived and we are done.

5.5 More Linear Temporal Logics

The construction of an inference system for *S4.3⊕KD4.3'* marks a main step towards the resolution-based inference system for the first-order linear temporal logic we are interested in. Recall that we have to be able to deal with past operators as well as future operators and that in both versions of either excluding or including the present. Also there might occur other operators like the *Always* but this operator does not bother us too much for we have found out already that the saturation of the *Always* results in the universal accessibility relation and hence there are no complications to be expected.

5.5.1 The Temporal Logic $K_tT4.3 \oplus K_tD4.3$'

We are now at a stage where it is possible to introduce the most complicated of the temporal logics we want to consider, namely the temporal logic with past and future operators (both proper and not proper) and linearity in all its possible occurrences. The corresponding background theory will be the most expressive and complicated one we consider. And although some of the operators that have been introduced in the introductory chapter are not yet contained in this logic the background theory will not change later on.

The axiomatization of (the propositional fragment of) this temporal logic is fairly complicated but should not be omitted at this point (see Table 5.1). This axiomatization can be viewed as a piece-by-piece axiomatization of the temporal logic we are interested in. Some of the axioms above are in fact redundant, nevertheless this axiomatization covers what one might have in mind when thinking of a temporal logic based on these operators.

$\boxed{\mathrm{F}}_r \Phi \Rightarrow \Phi$
$\boxed{\mathrm{F}}_r \Phi \Rightarrow \boxed{\mathrm{F}}_r \boxed{\mathrm{F}}_r \Phi$
$\langle\mathrm{F}\rangle_r \Phi \land \langle\mathrm{F}\rangle_r \Psi \Rightarrow \langle\mathrm{F}\rangle_r(\langle\mathrm{F}\rangle_r \Phi \land \Psi) \lor \langle\mathrm{F}\rangle_r(\Phi \land \langle\mathrm{F}\rangle_r \Psi)$
the *S4.3* future fragment

$\boxed{\mathrm{F}} \Phi \Rightarrow \langle\mathrm{F}\rangle \Phi$
$\boxed{\mathrm{F}} \Phi \Rightarrow \boxed{\mathrm{F}} \boxed{\mathrm{F}} \Phi$
$\langle\mathrm{F}\rangle \Phi \land \langle\mathrm{F}\rangle \Psi \Rightarrow \langle\mathrm{F}\rangle(\langle\mathrm{F}\rangle \Phi \land \Psi) \lor \langle\mathrm{F}\rangle(\Phi \land \Psi) \lor \langle\mathrm{F}\rangle(\Phi \land \langle\mathrm{F}\rangle \Psi)$
the *KD4.3'* future fragment

$\boxed{\mathrm{F}}_r \Phi \Leftrightarrow \boxed{\mathrm{F}} \Phi \land \Phi$
the combination of future operators

$\boxed{\mathrm{P}}_r \Phi \Rightarrow \Phi$
$\boxed{\mathrm{P}}_r \Phi \Rightarrow \boxed{\mathrm{P}}_r \boxed{\mathrm{P}}_r \Phi$
$\langle\mathrm{P}\rangle_r \Phi \land \langle\mathrm{P}\rangle_r \Psi \Rightarrow \langle\mathrm{P}\rangle_r(\langle\mathrm{P}\rangle_r \Phi \land \Psi) \lor \langle\mathrm{P}\rangle_r(\Phi \land \langle\mathrm{P}\rangle_r \Psi)$
the *S4.3* past fragment

$\boxed{\mathrm{P}} \Phi \Rightarrow \langle\mathrm{P}\rangle \Phi$
$\boxed{\mathrm{P}} \Phi \Rightarrow \boxed{\mathrm{P}} \boxed{\mathrm{P}} \Phi$
$\langle\mathrm{P}\rangle \Phi \land \langle\mathrm{P}\rangle \Psi \Rightarrow \langle\mathrm{P}\rangle(\langle\mathrm{P}\rangle \Phi \land \Psi) \lor \langle\mathrm{P}\rangle(\Phi \land \Psi) \lor \langle\mathrm{P}\rangle(\Phi \land \langle\mathrm{P}\rangle \Psi)$
the *KD4.3'* past fragment

$\boxed{\mathrm{P}}_r \Phi \Leftrightarrow \boxed{\mathrm{P}} \Phi \land \Phi$
the combination of past operators

$\Phi \Rightarrow \boxed{\mathrm{F}}_r \langle\mathrm{P}\rangle_r \Phi$
$\Phi \Rightarrow \boxed{\mathrm{P}}_r \langle\mathrm{F}\rangle_r \Phi$
$\Phi \Rightarrow \boxed{\mathrm{F}} \langle\mathrm{P}\rangle \Phi$
$\Phi \Rightarrow \boxed{\mathrm{P}} \langle\mathrm{F}\rangle \Phi$
the past–future relation

Table 5.1: $K_t T4.3 \oplus K_t D4.3'$ Axioms

Now we should have a look at the "Earlier-Later" properties that are induced by this axiomatization. First we note that the final two axioms from above again guarantee that the (proper) "Earlier-Later" is just the converse of the (proper) "Later-Earlier". We make use of this fact by considering only the (proper) "Earlier-Later"; describing the corresponding properties of the other relation in terms of this one. Note that in the following set of accessibility relation properties the symbol R^* refers to the operator $\boxed{\mathrm{F}}_r$ and R to $\boxed{\mathrm{F}}$.

$$
\begin{array}{l}
R^*(u,u) \\
R^*(u,u\!:\!x^*_F) \\
R^*(u\!:\!x^*_P,u) \\
R^*(u,v) \wedge R^*(v,w) \Rightarrow R^*(u,w) \\
R^*(u,v) \wedge R^*(u,w) \Rightarrow R^*(v,w) \vee R^*(w,v) \\
R^*(u,w) \wedge R^*(v,w) \Rightarrow R^*(u,v) \vee R^*(v,u) \\
R(u,u\!:\!x_F) \\
R(u\!:\!x_P,u) \\
R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\
R(u,v) \wedge R(u,w) \Rightarrow R(v,w) \vee v = w \vee R(w,v) \\
R(u,w) \wedge R(v,w) \Rightarrow R(u,v) \vee u = v \vee R(v,u) \\
R(u,v) \Rightarrow R^*(u,v) \\
R^*(u,v) \Rightarrow R(u,v) \vee u = v
\end{array}
$$

This clause set is quite complicated. However, as already mentioned earlier, there are some redundancies that can be eliminated and, moreover, further simplifications are possible which make use of the connectedness assumption.

LEMMA 5.5.1
The $K_tT4.3 \oplus K_tD4.3$' background theory consists of the following set of clauses:

$$
\begin{array}{l}
R^*(u,u\!:\!x^*_F) \\
R^*(u\!:\!x^*_P,u) \\
R^*(u,v) \vee R^*(v,u) \\
R(u,u\!:\!x_F) \\
R(u\!:\!x_P,u) \\
R(u,v) \wedge R(v,w) \Rightarrow R(u,w) \\
R(u,v) \Rightarrow R^*(u,v) \\
R^*(u,v) \Rightarrow R(u,v) \vee u = v
\end{array}
$$

Proof: This could be checked with any standard theorem prover provided it has been shown already that the various linearity properties can be simplified to $R^*(u,v) \vee R^*(v,u)$ and $R(u,v) \vee u = v \vee R(v,u)$ respectively. The proof of this fact is exactly as for the temporal and modal logics we considered earlier and is therefore omitted here.

What the other "missing" properties is concerned: they can easily be proved from the ones given here. So, for instance, reflexivity of R^* follows immediately from the strong linearity of R^* and the linearity of R can directly be derived from $R^*(u,v) \vee R^*(v,u)$ and $R^*(u,v) \Rightarrow R(u,v) \vee u = v$. Finally, the transitivity of R^* follows from the transitivity of R and the fact that R^* denotes the reflexive closure of R.

LEMMA 5.5.2
The saturation of the $K_tT4.3 \oplus K_tD4.3'$ background theory is determined by

$$R^*(u_1:\overline{x_1}, u_2:\overline{y_2}) \vee \ldots \vee R^{(*)}(u_{n-1}:\overline{x_{n-1}}, u_n:\overline{y_n}), R^{(*)}(u_n:\overline{x_n}, u_1:\overline{y_1})$$
with $n \geq 1$ *and* $\overline{x_i} \in (F_{R_P} \cup F_{R_P^*})^*$, $\overline{y_i} \in (F_{R_F} \cup F_{R_F^*})^*$ *for all* $1 \leq i \leq n$

$$R(u_1:\overline{x_1}, u_2:\overline{y_2}) \vee \ldots \vee R(u_{n-1}:\overline{x_{n-1}}, u_n:\overline{y_n}), R(u_n:\overline{x_n}, u_1:\overline{y_1})$$
with $n \geq 1$ *and* $\overline{x_i} \in (F_{R_P} \cup F_{R_P^*})^*$, $\overline{y_i} \in (F_{R_F} \cup F_{R_F^*})^*$ *for all* $1 \leq i \leq n$
but $\overline{x_i} \notin (F_{R_P})^*$ *or* $\overline{y_i} \notin (F_{R_F})^*$ *for some* $1 \leq i \leq n$

$$R(u_1:\overline{x_1}:\overline{x_1}', u_2:\overline{y_2}:\overline{y_2}') \vee \ldots \vee R(u_n:\overline{x_n}:\overline{x_n}', u_1:\overline{y_1}:\overline{y_1}') \vee u_i:\overline{x_i} = u_j:\overline{y_j}$$
with $n \geq 1$ *and* $\overline{x_i}:\overline{x_i}' \in (F_{R_P^*})^*$ *and* $\overline{y_i}:\overline{y_i}' \in (F_{R_F^*})^*$

Proof: Can be performed exacly as in the corresponding proof for the logic *S4.3⊕KD4.3'* in Lemma 5.4.17 on page 128. The only particularities are that – in addition – clauses of the form

$$R(u,v) \Rightarrow R(u:\overline{x}, v:\overline{y}) \qquad \text{with } \overline{x} \in (F_{R_P} \cup F_{R_P^*})^* \text{ and } \overline{y} \in (F_{R_F} \cup F_{R_F^*})^*$$

can be derived from the background theory; all the rest remains essentially the same.

Note that here – just as in case of *S4.3⊕KD4.3'* – the third clause schema could equally be replaced by two less general schemata. We shall make use of this in the definition of the $K_tT4.3 \oplus K_tD4.3'$ Inference System.

DEFINITION 5.5.3 (THE $K_tT4.3 \oplus K_tD4.3'$ INFERENCE SYSTEM)
The $K_tT4.3 \oplus K_tD4.3'$ Inference System *consists of the $K_tT4 \oplus K_tD4$ Inference System together with the following inference rules*

$$\frac{\neg R^{(*)}(\alpha, \beta_1:\beta_2), C_1 \qquad \neg R^{(*)}(\gamma_1:\gamma_2, \delta), C_2}{\neg R^*(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2}$$

where $\sigma = mgu(\gamma_1, \beta_1)$, $\beta_2 \in (F_{R_F} \cup F_{R_F^*})^*$, $\gamma_2 \in (F_{R_P} \cup F_{R_P^*})^*$ *and either* $\beta_2 \notin (F_{R_F})^*$ *or* $\gamma_2 \notin (F_{R_P})^*$ *or at least one of the parents is an* R^**-literal*

$$\frac{\neg R(\alpha, \beta_1:\beta_2), C_1 \qquad \neg R(\gamma_1:\gamma_2, \delta), C_2}{\neg R(\sigma\alpha, \sigma\delta), \sigma C_1, \sigma C_2}$$

$\sigma = mgu(\gamma_1, \beta_1)$ *and* $\beta_2 \in (F_{R_F^*})^*$, $\gamma_2 \in (F_{R_P^*})^*$

$$\frac{\neg R(\alpha, \beta), C_1 \qquad \neg R(\gamma, \delta), C_2}{\sigma\alpha = \sigma\beta, \sigma C_1, \sigma C_2}$$

$\sigma = mgu((\gamma, \beta), (\alpha, \delta))$

THEOREM 5.5.4
The $K_tT4.3 \oplus K_tD4.3'$ Inference System is sound and refutation complete.

Proof: In full analogy to the corresponding proofs of Lemmata 5.4.20 and 5.4.21.

5.6 *Until*- and *Since*-Operators

The temporal logics examined so far can all be viewed as certain special multi-modal logics since the operators under consideration are in fact modal $\Box$ and $\Diamond$ operators (though we had several of them). Now point oriented temporal logics differ from modal logics in two respects. First, there are some particular accessibility relation properties like the linearity which seldomly occur with other interpretations than the temporal one, and second, there are certain *temporal* operators which wouldn't make much sense in a general modal logic setting. Two of the most interesting of such operators are the *Until* and the *Since* as they are described in the introductory chapter on temporal logics. Both of these operators (and their respective variants) are not definable in terms of the other temporal operators we examined up to now[9]. Nevertheless, their semantics can be described in terms of the "Earlier-Later" relation and therefore it is also possible to define the relational translation for these operators:

$$\begin{aligned}
\lceil \Phi\, \mathbf{U}\, \Psi \rceil_u &= \exists v\, R(u,v) \wedge \lceil \Psi \rceil_v \quad \wedge \\
&\qquad \forall w\, R(u,w) \wedge R(w,v) \Rightarrow \lceil \Phi \rceil_w \\
\lceil \Phi\, \mathbf{S}\, \Psi \rceil_u &= \exists v\, R(v,u) \wedge \lceil \Psi \rceil_v \quad \wedge \\
&\qquad \forall w\, R(v,w) \wedge R(w,u) \Rightarrow \lceil \Phi \rceil_w
\end{aligned}$$

Also, it is pretty easy to find the semi-functional translation for these operators, namely

$$\begin{aligned}
\lfloor \Phi\, \mathbf{U}\, \Psi \rfloor_u &= \exists x_F\, \lfloor \Psi \rfloor_{u:x_F} \quad \wedge \\
&\qquad \forall w\, R(u,w) \wedge R(w,u\!:\!x_F) \Rightarrow \lfloor \Phi \rfloor_w \\
\lfloor \Phi\, \mathbf{S}\, \Psi \rfloor_u &= \exists x_P\, \lfloor \Psi \rfloor_{u:x_P} \quad \wedge \\
&\qquad \forall w\, R(u\!:\!x_P,w) \wedge R(w,u) \Rightarrow \lfloor \Phi \rfloor_w
\end{aligned}$$

And indeed the semi-functional translation for such positive occurrences of the *Until* and *Since* operators result in formula without any positive R-literals and this was one of our main requirements for the saturation method to be applied.

Unfortunately, some complications arise when such an *Until* or a *Since* formula occurs within an odd number of negation signs. From the view of the pure relational translation approach there are still no problems for the negation can be moved inward the first-order formula just as usual. In the semi-functional translation approach, however, we end up with something like

$$\begin{aligned}
\lfloor \neg(\Phi\, \mathbf{U}\, \Psi) \rfloor_u &= \forall v\, R(u,v) \Rightarrow \lfloor \neg\Psi \rfloor_v \quad \vee \\
&\qquad \exists x_F\, R(u\!:\!x_F,v) \wedge \lfloor \neg\Phi \rfloor_{u:x_F} \\
\lfloor \neg(\Phi\, \mathbf{S}\, \Psi) \rfloor_u &= \forall v\, R(v,u) \Rightarrow \lfloor \neg\Psi \rfloor_v \quad \vee \\
&\qquad \exists x_P\, R(v,u\!:\!x_P) \wedge \lfloor \neg\Phi \rfloor_{u:x_P}
\end{aligned}$$

i.e. there are still positive occurrences of R-literals remaining and this seems to be somewhat perturbing.

A way to overcome this problem is to change the definition of the *Until* and the *Since* slightly such that the translation of such negative occurrences does not produce any positive R-literals and the general idea behind this little change is the equalization of $R^*(\alpha,\beta)$ with $\neg R(\beta,\alpha)$ under the linearity assumption. The semi-functional translation then gets

$$\begin{aligned}
\lfloor \neg(\Phi\, \mathbf{U}\, \Psi) \rfloor_u &= \forall v\, R(u,v) \Rightarrow \lfloor \neg\Psi \rfloor_v \quad \vee \\
&\qquad \exists x_F\, \neg R^*(v,u\!:\!x_F) \wedge \lfloor \neg\Phi \rfloor_{u:x_F} \\
\lfloor \neg(\Phi\, \mathbf{S}\, \Psi) \rfloor_u &= \forall v\, R(v,u) \Rightarrow \lfloor \neg\Psi \rfloor_v \quad \vee \\
&\qquad \exists x_P\, \neg R^*(u\!:\!x_P,v) \wedge \lfloor \neg\Phi \rfloor_{u:x_P}
\end{aligned}$$

[9]This was already shown by Hans Kamp in (Kamp 1968).

and the original requirement is saved. However, there remains the question to be answered whether we are really allowed to do this since in the given properties of the R predicate it is nowhere stated that R is irreflexive and this irreflexivity seems a crucial property for the equalization of the two possible *not-Until* translations. In fact we are not even able to guarantee irreflexivity because this (purely negative) property is not axiomatizable as has been shown in Section 3.6. Nevertheless, we are allowed to change the definition, for the problem is not really found in the fact that irreflexivity cannot be axiomatized but the in reason why this is so. Recall that we found out in Section 3.6 that any purely negative accessibility relation property that is at all consistent with the other given accessibility relation properties has no effect what the satisfiablity and the unsatisfiablity of a modal or temporal logic formula is concerned and *therefore* cannot be axiomatized. In particular this means that adding the irreflexivity or not makes no difference; modal and temporal logics cannot distinguish. In this light we may assume that R is irreflexive and therefore we are allowed to perform this tiny change in the translation of a negative *Until*-formula.

To state it otherwise, suppose we had defined a new (kind of negative *Until*) operator, say $\Phi\,\overline{\mathbf{U}}\,\Psi$, in the above sense, i.e. which gets translated into

$$\begin{array}{rcl}\lfloor \Phi\,\overline{\mathbf{U}}\,\Psi \rfloor_u & = & \forall v\ R(u,v) \Rightarrow \lfloor \neg\Psi \rfloor_v \quad \vee \\ & & \qquad \exists x_F\ \neg R^*(v, u\!:\!x_F) \wedge \lfloor \neg\Phi \rfloor_{u:x_F}\end{array}$$

In order to shown that there is no essential difference between the $\neg\mathbf{U}$ and the $\overline{\mathbf{U}}$ we consider the property that characterizes the axiom schema

$$\neg(\Phi\,\mathbf{U}\,\Psi) \Leftrightarrow (\Phi\,\overline{\mathbf{U}}\,\Psi)$$

and we do so by applying the second order quantifier elimination technique from Section 3.1.2. To this end we have to negate the axiom schema (better the two implication directions) and therefore try to find a first-order equivalent for both $\neg(\Phi\,\mathbf{U}\,\Psi) \wedge \neg(\Phi\,\overline{\mathbf{U}}\,\Psi)$ and $\Phi\,\mathbf{U}\,\Psi \wedge \Phi\,\overline{\mathbf{U}}\,\Psi$ with existentially quantified Φ and Ψ respectively. So, let us first have a look at $\Phi\,\mathbf{U}\,\Psi \wedge \Phi\,\overline{\mathbf{U}}\,\Psi$ which gets (relationally) translated into

$$\exists u, \Phi, \Psi \left[\begin{array}{c}\exists v\ R(u,v) \wedge \Psi(v) \wedge \forall w\ R(u,w) \wedge R(w,v) \Rightarrow \Phi(w) \\ \wedge \\ \forall v\ R(u,v) \wedge \Psi(v) \Rightarrow \exists w\ R(u,w) \wedge \neg R^*(v,w) \wedge \neg\Phi(w)\end{array}\right]$$

First we try to eliminate the Ψ. To this end the above formula is transformed into

$$\exists u, \Phi, \Psi \left[\begin{array}{c}\forall v\ \neg\Psi(v) \vee \neg R(u,v) \vee \exists w\ R(u,w) \wedge \neg R^*(v,w) \wedge \neg\Phi(w) \\ \wedge \\ \forall v\ R(u,v) \wedge \Psi(v) \Rightarrow \exists w\ R(u,w) \wedge \neg R^*(v,w) \wedge \neg\Phi(w)\end{array}\right]$$

which is equivalent to (according to the quantification elimination theorem)

$$\exists u, \Phi, v \left[\begin{array}{c}R(u,v) \\ \wedge \\ \neg R(u,v) \vee \exists w\ R(u,w) \wedge \neg R^*(v,w) \wedge \neg\Phi(w) \\ \wedge \\ \forall w\ R(u,w) \wedge R(w,v) \Rightarrow \Phi(w)\end{array}\right]$$

This can slightly be simplified and, since we intend to eliminate the Φ as well, has to be transformed into

$$\exists u, \Phi, v \left[\begin{array}{c} \forall w\ \Phi(w) \vee \neg R(u,w) \vee \neg R(w,v) \\ \wedge \\ R(u,v) \\ \wedge \\ \exists w\ R(u,w) \wedge \neg R^*(v,w) \wedge \neg \Phi(w) \end{array} \right]$$

A second application of the elimination theorem then results in

$$\exists u, v, w\ R(u,v) \wedge R(u,w) \wedge \neg R^*(v,w) \wedge \neg R(w,v)$$

Thus the original implication $\Phi\,\overline{\mathbf{U}}\,\Psi \Rightarrow \neg(\Phi\,\mathbf{U}\,\Psi)$ is characterized by the first order formula

$$\forall u, v, w\ R(u,v) \wedge R(u,w) \Rightarrow R^*(v,w) \vee R(w,v)$$

a formula that is not only consistent with the background theory we are considering but even is implied by it.

Next we have to proceed analogously for the other direction of the given equivalence, i.e. we try to find a first-order equivalent to $\neg(\Phi\,\mathbf{U}\,\Psi) \wedge \neg(\Phi\,\overline{\mathbf{U}}\,\Psi)$ with existentially quantified Φ and Ψ. Relational translation results in

$$\exists u, \Phi, \Psi \left[\begin{array}{c} \forall v\ R(u,v) \wedge \Psi(v) \Rightarrow \exists w\ R(u,w) \wedge R(w,v) \wedge \neg \Phi(w) \\ \wedge \\ \exists v\ R(u,v) \wedge \Psi(v) \wedge \forall w\ R(u,w) \wedge \neg R^*(v,w) \Rightarrow \Phi(w) \end{array} \right]$$

Eliminating Ψ then leads to

$$\exists u, \Phi \left[\begin{array}{c} \exists v\ R(u,v) \\ \wedge \\ \neg R(u,v) \vee \exists w\ R(u,w) \wedge R(w,v) \wedge \neg \Phi(w) \\ \wedge \\ \forall w\ R(u,w) \wedge \neg R^*(v,w) \Rightarrow \Phi(w) \end{array} \right]$$

which can be transformed into

$$\exists u, v, \Phi \left[\begin{array}{c} \forall w\ \Phi(w) \vee \neg R(u,w) \vee R^*(v,w) \\ \wedge \\ R(u,v) \\ \wedge \\ \exists w\ R(u,w) \wedge R(w,v) \wedge \neg \Phi(w) \end{array} \right]$$

Now we can eliminate the Φ and we end up with

$$\exists u, v, w\ R(u,v) \wedge R(u,w) \wedge R(w,v) \wedge R^*(v,w)$$

Thus the given implication $\neg(\Phi\,\mathbf{U}\,\Psi) \wedge \Phi\,\overline{\mathbf{U}}\,\Psi$ is characterized by

$$\forall u, v, w\ \neg R(u,v) \vee \neg R(u,w) \vee \neg R(w,v) \vee \neg R^*(v,w)$$

a formula that is consistent with the background theory under consideration and moreover that is purely negative.

We thus may use $\Phi \, \overline{\mathbf{U}} \, \Psi$ whenever there occurs a $\neg(\Phi \, \mathbf{U} \, \Psi)$ provided we add the characteristic formulae just obtained. However, the first of these two formulae was implied by the background theory anyway and the second is purely negative. Therefore both cannot play any role what the (un-)satisfiability of translated temporal logic formulae is concerned and hence are redundant.

The same result can be obtained for the other possible *Until*-operators and also for the various *Since*-operators and we finally can conclude this section with the following lemma.

LEMMA 5.6.1
The negation normal form for $\neg(\Phi \, \mathbf{U} \, \Psi)$ is $\Phi' \, \overline{\mathbf{U}} \, \Psi'$ where Φ' (Ψ') denotes the negation normal form of $\neg\Phi$ ($\neg\Psi$ respectively).
This way the semi-functional translation of any temporal logic formula results in a first-order formula which does not contain any positive $R-$ or R^*-literal.

Note that this construction absolutely requires the linearity assumption. Otherwise, the first of the two characteristic properties we obtained by second-order quantifier elimination would have to be added to the background theory and this would inevitably have a considerable effect on the saturation process.

6

A Short Digression to Interval Logics

The basic entities of the temporal logics we considered up to now were *time instants* and various *Earlier-Later-relations*. This choice is certainly not the only possible. We might equally think of *temporal intervals* and suitable relations between these. Still, even after we have made a decision for the one we do not really exclude the possibility of talking about the other. So, for instance, we might view intervals as convex sets of instants or instants as indivisible intervals. The latter is commonly used in the field of processes, actions and events which are assumed to have a certain *duration* in the sense that an event which occurs in some time interval does not occur in any of its subintervals. The other possibility also occurs frequently. Even if there is no real interest in talking about instants explicitly, intervals are represented by an ordered pair of instants which is to be interpreted as the set of moments that *exist between* these two instants (see e.g. (Allen 1981b), (Allen 1983), (Allen 1984), (Allen and Hayes 1985a), (Allen and Hayes 1985b), (Allen and Hayes 1985c), (Allen and Hayes 1987), (Allen 1981c), (Allen 1981a), (McDermott 1982), (Halpern and Shoham 1986), (Shoham 1987), (Shoham 1986), (Shoham 1988) on both issues[1]).

In this chapter I will briefly describe how the techniques devoloped in the earlier sections can be utilized for both, intervals as basic entities and intervals as convex sets of instants.

[1]One might also consider other possibilities. So, for instance, one might think of events as the basic primitives together with some causality relation and define both intervals and instants in terms of causal relationships between events. In (Winnie 1977) instants are defined as *maximal sets of pairwise simultaneous events* where two events are *simultaneous* if neither is a (direct or indirect) cause of the other. In this sense an interval could be defined as a subset of an instant.

6.1 Intervals as Convex Sets of Instants

A first very simple way of talking about intervals in a point oriented setting would be to translate operators of the kind $\langle L\rangle$ with the intended meaning: $\langle L\rangle\Phi$ *iff there exists a future interval – i.e. a convex set of later instants – such that* Φ *holds in every moment of this interval*, in terms of *Until* and *Since* operators. For the above operator $\langle L\rangle$ this would mean:

$$\lfloor\langle L\rangle\Phi\rfloor_u = \lfloor\diamondsuit_F(\Phi\,\mathbf{U}\,\top)\rfloor_u$$

The disadvantage of this approach is that formulae are still to be evaluated with respect to single moments of time and therefore processes and events cannot be represented. For instance, if Φ describes the event *I am walking from the bus stop to the cinema* and if Φ indeed holds for a certain time interval then it would not make very much sense to interpret this as Φ *holds in every moment of this interval.* One might accept that *I am walking* on every such moment but certainly not that *I am walking from the bus stop to the cinema* on every such moment.

Therefore, if we want to use intervals as the primitives we talk about but nevertheless want to represent intervals by means of instants and Earlier-Later-Relations we are forced to change the temporal logic semantics slightly. Fortunately, this change is not very difficult. It mainly consists of exchanging the *current instant* by a pair $[t_1, t_2]$ which is supposed to represent the *current interval* and to interpret (and thus also to translate) formulae with respect to such pairs of instants. All the rest essentially remains as before.

Now let us assume a set of modalities in the sense of (Allen 1984) and (Shoham 1988), i.e. we assume the *Diamond*-operators $\langle M\rangle$, $\langle B\rangle$, $\langle E\rangle$, $\langle\overline{M}\rangle$, $\langle\overline{B}\rangle$, and $\langle\overline{E}\rangle$, with the intended meaning

$\langle M\rangle\Phi$	Φ holds at some interval beginning immediately after the current one
$\langle B\rangle\Phi$	Φ holds at some interval during (and beginning with) the current one
$\langle E\rangle\Phi$	Φ holds at some interval during (and ending with) the current one
$\langle\overline{M}\rangle\Phi$	Φ holds at some interval ending immediately before the current one
$\langle\overline{B}\rangle\Phi$	Φ holds at some interval of which the current one is a beginning
$\langle\overline{E}\rangle\Phi$	Φ holds at some interval of which the current one is an end

A possible semi-functional translation into first-order predicate logic could then look like this:

$$\begin{array}{lcl}
\lfloor P(\ldots,\mathrm{arg}_i,\ldots)\rfloor_{[t_1,t_2]} & = & P(t_1,t_2,\ldots\lfloor\mathrm{arg}_i\rfloor_{[t_1,t_2]},\ldots)\\
\lfloor\langle M\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_F\ \lfloor\Phi\rfloor_{[t_2,t_2:x_F]}\\
\lfloor\langle B\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_F^*\ \neg R^*(t_2,t_1:x_F^*)\wedge\lfloor\Phi\rfloor_{[t_1,t_1:x_F^*]}\\
\lfloor\langle E\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_F\ \neg R^*(t_2,t_1:x_F)\wedge\lfloor\Phi\rfloor_{[t_1:x_F,t_2]}\\
\lfloor\langle\overline{M}\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_P\ \lfloor\Phi\rfloor_{[t_1:x_P,t_1]}\\
\lfloor\langle\overline{B}\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_F\ \lfloor\Phi\rfloor_{[t_1,t_2:x_F]}\\
\lfloor\langle\overline{E}\rangle\Phi\rfloor_{[t_1,t_2]} & = & \exists x_P\ \lfloor\Phi\rfloor_{[t_1:x_P,t_2]}
\end{array}$$

The respective translations of the corresponding dual operators should be obvious. For example the operator $[\overline{M}]$, i.e. the operator which is dual to $\langle\overline{M}\rangle$, would have to be translated into

$$\lfloor[\overline{M}]\Phi\rfloor_{[t_1,t_2]} = \forall u\ \neg R(u,t_1)\vee\lfloor\Phi\rfloor_{[u,t_1]}$$

Note that many other interesting interval operators can be described with the help of the above primitive operators. As an example consider the operator $\langle\overline{D}\rangle$ such that a formula $\langle\overline{D}\rangle\Phi$ has the informal meaning: Φ *holds in a proper superinterval of the current interval.* This operator can easily be defined by

$$\langle\overline{D}\rangle\Phi \Leftrightarrow \langle\overline{B}\rangle\langle\overline{E}\rangle\Phi$$

such that it is to be translated into

$$\lfloor\langle\overline{D}\rangle\Phi\rfloor_{[t_1,t_2]} = \exists x_P, y_F \; \lfloor\Phi\rfloor_{[t_1:x_P,t_2:y_F]}$$

An adequate background theory for the R-predicates is then K_t*T4.3*$\oplus$ K_t*D4.3'* and the inference system for this logic forms a sound and refutation complete inference system (under the above translation) for the given interval logic.

6.2 Intervals as Basic Temporal Entities

As it has been mentioned in the introductory chapter on temporal logic the achievements gained in the area of interval logics have not yet gone as far as for instant temporal logics at least what a Priorian setting is concerned. I.e. that the Priorian principle of considering basic temporal entities as primitives for the interpretation of certain temporal operators has not been taken over in many interval logics that occur in the temporal logic literature.

The interval logics that are examined in this section are the ones presented in (Humberstone 1979) and (van Benthem 1990). The basic entities are intervals rather than instants and the operators as they have been used in the instant temporal logics are taken over in order to be able to express an ordering on such intervals. Whereas instants were intuitively assumed to be indivisible primitives such an assumption cannot be made for intervals with a clear conscience. I.e. in addition to the temporal order, as it has been examined for instant temporal logics, there is a need for representing subintervals. As a matter of fact nothing prevents us from introducing new operators whose semantics is defined in terms of such subinterval relations and this is exactly what Humberstone and van Benthem did in their examinations.

The main idea behind the formal semantics definition for interval logics is thus to assume frames which consist of intervals instead of instants and a temporal as well as a subinterval relation on such primitives. However, an immediate complication arises when it becomes necessary to define what it means that a certain formula Φ holds over an interval I. It has to be agreed whether this means that Φ's truth stretches over the interval or not. Humberstone indeed made the assumption that Φ is true over an interval if and only if Φ is true throughout this interval although this is a rather intuitive interpretation for it is not obvious what "throughout an interval" could possible mean. Since we are not able to talk about *the elements* of an interval in the sense of moments in time such a description remains pretty vague. Another problem which immediately came into Humberstone's mind was about the interpretation of the negation $\neg$. If $\neg$ is to be interpreted classically, then $\neg\Phi$ holds over an interval iff it is *not* the case that Φ holds over this very interval. This is definitely weaker than stating that Φ is False throughout the interval and there seems no obvious solution to the problem of being able to express complete absense of Φ on an interval. Humberstone's idea to cope with this difficulty was to introduce another *strong* negation (called NOT by van Benthem) with some intuitionistic flavour and he axiomatized this new operator by[2]

$$\vdash \mathrm{NOT}\,\Phi \Rightarrow \neg\Phi$$
$$\text{if } \vdash \mathrm{NOT}\,\Phi \Rightarrow \neg\Psi \text{ then } \vdash \mathrm{NOT}\,\Phi \Rightarrow \mathrm{NOT}\,\Psi$$

This axiomatization is not quite given in a Priorian style but it can be reformulated in terms of Priorian operators if we define

$$\Im[\tau] \models \Box_{\sqsubseteq}\Phi \quad \text{iff} \quad \Im[\xi] \models \Phi \quad \text{for every subinterval } \xi \text{ of } \tau$$

Then a formula $\mathrm{NOT}\,\Phi$ can be represented as $\Box_{\sqsubseteq}\neg\Phi$ where the $\neg$ remains classical and the above axiomatization gets reformulated into

$$\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Phi$$
$$\text{if } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Psi \text{ then } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$$

What the first part of this axiomatization is concerned: we are already familiar with such a schema for it obviously amounts to the reflexivity of the subinterval relation. The second part, however, still looks rather unusual although, as we see below, it merely describes transitivity.

LEMMA 6.2.1
The Hilbert style rule

$$\textit{if } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Psi \textit{ then } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$$

characterizes the transitivity of the subinterval relation.

Proof: This is another case for the second-order quantifier elimination. Suppose that $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Psi$ but *not* $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$ for some Φ and Ψ, i.e.

$$\exists\Phi,\Psi \left[\begin{array}{c} \forall u\ \lfloor\Box_{\sqsubseteq}\Phi \Rightarrow \Psi\rfloor_u \\ \wedge \\ \exists u\ \lfloor\Box_{\sqsubseteq}\Phi \wedge \Diamond_{\sqsubseteq}\neg\Psi\rfloor_u \end{array}\right]$$

where the translation function $\lfloor\,\rfloor_u$ is to be defined as for the relational translation for instant temporal logics by

$$\begin{array}{rcl} \lfloor\Box_{\sqsubseteq}\Phi\rfloor_u &=& \forall v\ v \sqsubseteq u \Rightarrow \lfloor\Phi\rfloor_v \\ \lfloor\Diamond_{\sqsubseteq}\Phi\rfloor_u &=& \exists v\ v \sqsubseteq u \wedge \lfloor\Phi\rfloor_v \end{array}$$

We thus get

$$\exists\Phi,\Psi \left[\begin{array}{c} \forall u\ \Psi(u) \vee \exists v\ v \sqsubseteq u \wedge \neg\Phi(v) \\ \wedge \\ \exists u\ (\forall v\ v \sqsubseteq u \Rightarrow \Phi(v) \wedge \exists w\ w \sqsubseteq u \wedge \neg\Psi(w)) \end{array}\right]$$

Now the Elimination Theorem of Section 3.1.2 can be applied and this results in

$$\exists u\ \exists\Phi \left[\begin{array}{c} \forall v\ v \sqsubseteq u \Rightarrow \Phi(v) \\ \wedge \\ \exists w,v'\ w \sqsubseteq u \wedge v' \sqsubseteq w \wedge \neg\Phi(v')) \end{array}\right]$$

[2]Actually, instead of an axiomatization, Humberstone defines natural deduction sequents for the operators to be introduced. The rules given here are therefore rather "translations" from Humberstone's original formulation into a Hilbert calculus.

A second application of the Elimination Theorem then leads to

$$\exists u, v, w \; v \sqsubseteq w \land w \sqsubseteq u \land v \not\sqsubseteq u$$

and since the original input had to be negated we have to negate once again and we end up with

$$\forall u, v, w \; u \sqsubseteq v \land v \sqsubseteq w \Rightarrow u \sqsubseteq w$$

Hence, instead of the Hilbert rule

$$\text{if } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Psi \text{ then } \vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$$

we can alternatively switch to the axiom schema

$$\Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi$$

without causing any difficulties. Note that this result could also be obtained as follows:

- $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi$ follows from the rule by letting Ψ be $\Box_{\sqsubseteq}\Phi$.
- On the other hand, if we have that $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi$ then the rule can be derived from the axiomatization, for assume that we have $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Psi$. From this we can show (with necessitation rule amd K-axiom for $\Box_{\sqsubseteq}$ plus Modus Ponens) that $\vdash \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$. This – together with $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi$ and standard propositional calculus – then yields $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Psi$ and we are done.

In particular this means that a completeness result under $\vdash \Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi$ also serves as a completeness result under this rule.

Quite a big part of Humberstone's interval logic is axiomatized now, however, it should be evident that still something is missing, namely a correlation between the temporal order operators and the subrelation operator. The mixing postulates as they are suggested by Humberstone are

$$\langle F \rangle \, \Phi \Rightarrow \mathrm{NOT}\neg \langle F \rangle \, \Phi$$
$$\langle F \rangle \, \neg\mathrm{NOT}\, \Phi \Rightarrow \langle F \rangle \, \Phi$$

and in terms of the $\Box_{\sqsubseteq}$ operator these can be reformulated as

$$\langle F \rangle \, \Phi \Rightarrow \Box_{\sqsubseteq} \langle F \rangle \, \Phi$$
$$\langle F \rangle \, \Diamond_{\sqsubseteq}\Phi \Rightarrow \langle F \rangle \, \Phi$$

where this second schema is equivalent to

$$\langle P \rangle \, \Phi \Rightarrow \Box_{\sqsubseteq} \langle P \rangle \, \Phi$$

These axioms suit our intuition that any interval later (earlier) than the current interval is also later (earlier) than an arbitrary subinterval of the current interval. Moreover, these schemata guarantee that "later" intervals cannot overlap with the current interval.

Now we have reached a stage were we can define a minimal interval logic (which van Benthem called K_i) by

DEFINITION 6.2.2 (VAN BENTHEM'S MINIMAL INTERVAL LOGIC K_i)
The minimal interval logic K_i *is axiomatized by the minimal temporal logic* K_t *together with the necessitation rule and the K-Axiom for* $\Box_{\sqsubseteq}$ *and the axioms*

$$\begin{aligned}
&\Box_{\sqsubseteq}\Phi \Rightarrow \Phi\\
&\Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi\\
&\Diamond_F\, \Phi \Rightarrow \Box_{\sqsubseteq} \Diamond_F\, \Phi\\
&\Diamond_P\, \Phi \Rightarrow \Box_{\sqsubseteq} \Diamond_P\, \Phi
\end{aligned}$$

Van Benthem also defines the logic K_p which in addition contains the axiom $\boxed{F}\Phi \Rightarrow \boxed{F}\boxed{F}\Phi$, hence assumes transitivity of the Earlier-Later relation. In what follows we shall examine a resolution based calculus in the lines of the earlier chapters for both K_i and (slightly modified) K_p.

LEMMA 6.2.3
The K_i background theory (in light of the semi-functional translation) is given by

$$\begin{aligned}
&u \sqsubseteq u\\
&u\!:\!x_{\sqsubseteq} \sqsubseteq u\\
&u \sqsubseteq v \wedge v \sqsubseteq w \Rightarrow u \sqsubseteq w\\
&N_F(u) \Rightarrow R(u, u\!:\!x_F)\\
&N_P(u) \Rightarrow R(u\!:\!x_P, u)\\
&R(u,v) \wedge w \sqsubseteq u \Rightarrow R(w,v)\\
&R(v,u) \wedge w \sqsubseteq u \Rightarrow R(v,w)
\end{aligned}$$

where it is already taken into account that the Later-Earlier relation can be described in terms of the Earlier-Later relation.

Proof: The top five of these clauses are obvious. Note that in K_i neither the seriality of R_F nor of R_P is assumed; therefore the "normality" predicates have to be introduced. The other two can be shown by an application of the Elimination Theorem. Since the respective proofs of the two clauses are very similar it suffices to go through one of the two here.

Consider the axiom $\Diamond_P\, \Phi \Rightarrow \Box_{\sqsubseteq} \Diamond_P\, \Phi$. After negation and relational translation we get

$$\exists u \exists \Phi \left[\begin{array}{c} \exists v\ R(v,u) \wedge \Phi(v)\\ \wedge\\ \exists w\ w \sqsubseteq u \wedge \forall x\ R(x,w) \Rightarrow \neg\Phi(x) \end{array}\right]$$

This has to be brought into the form that fits with the requirements of the Elimination Theorem, thus

$$\exists u \exists \Phi \left[\begin{array}{c} \forall y\ \Phi(y) \vee y \neq v\\ \wedge\\ R(v,u)\\ \wedge\\ \exists w\ w \sqsubseteq u \wedge \forall x\ R(x,w) \Rightarrow \neg\Phi(x) \end{array}\right]$$

Now the Elimination Theorem can be applied:

$$\exists u, v\ R(v,u) \wedge \exists w\ w \sqsubseteq u \wedge \forall x\ R(x,w) \Rightarrow x \neq v$$

and after a tiny simplification and a final negation we end up with the last one of the clauses from above.

According to the techniques developed for modal and (instant–based) temporal logics the saturation of the respective background theories has to be determined now.

LEMMA 6.2.4
The saturation of the K_i background theory is described by

$$\begin{array}{l} u:\overline{x} \sqsubseteq u \\ N_F(u) \Rightarrow R(u:\overline{x}, u:y_F:\overline{z}) \\ N_P(u) \Rightarrow R(u:y_P:\overline{x}, u:\overline{z}) \end{array}$$

with $\overline{x}, \overline{z} \in (F_{\sqsubseteq})^*$.

Proof: From $u:\overline{x} \sqsubseteq u$ and the transitivity of $\sqsubseteq$ it follows immediately that $u \sqsubseteq v \Rightarrow u:x_{\sqsubseteq} \sqsubseteq v$. By a trivial induction over the length of $\overline{x}$ we get that every unit of the form $u:\overline{x} \sqsubseteq u$ can be derived. Moreover, no other $\sqsubseteq$-literals are derivable for this could only happen with the help of the transitivity clause. However, any resolution step between the transitivity clause and the unit schema $u:\overline{x} \sqsubseteq u$ produces literals that are already in the form $u:\overline{x} \sqsubseteq u$.

The two monotonicity clauses can therefore be simplified to

$$\begin{array}{l} R(u,v) \Rightarrow R(u:\overline{x}, v) \\ R(u,v) \Rightarrow R(u, v:\overline{x}) \end{array}$$

from which the other clause schemata follow by some simple resolution steps. Furthermore, no resolution step between these schemata and the simplified monotonicity clause leads to anything new and we are done.

Interestingly, we would obtain the same result if we chose the schema $\Diamond_{\sqsubseteq} \boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\Box_{\sqsubseteq}\Phi$ instead of the two monotonicity axioms.

LEMMA 6.2.5
The axiom schema

$$\Diamond_{\sqsubseteq} \boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\Box_{\sqsubseteq}\Phi$$

characterizes both forms of monotonicity at once.

Proof: We determine the first-order equivalent of $\Diamond_{\sqsubseteq} \boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\Box_{\sqsubseteq}\Phi$ by applying the Elimination Theorem. To this end consider the negation and relational translation of this schema.

$$\exists u \exists \Phi \left[\begin{array}{c} \exists v\ v \sqsubseteq u \wedge \forall w\ R(v,w) \Rightarrow \Phi(w) \\ \wedge \\ \exists v'\ R(u,v') \wedge \exists w'\ w' \sqsubseteq v' \wedge \neg\Phi(w') \end{array} \right]$$

After some simple transformations the Elimination Theorem can be applied and we get

$$\exists u, v\ v \sqsubseteq u \wedge \exists v'\ R(u,v') \wedge \exists w'\ w' \sqsubseteq v' \wedge \neg R(v,w')$$

The final negation then leads us to

$$\forall u, v, w, x\ R(u,v) \wedge w \sqsubseteq u \wedge x \sqsubseteq v \Rightarrow R(w,x)$$

which obviously is equivalent (under the reflexivity assumption for $\sqsubseteq$) to the conjunction of the two monotonicity properties.

Since we know now about the saturation of the K_i background theory it might be worth having a look for alternative clause sets with an identical saturation.

LEMMA 6.2.6
The saturation of the clause set

$$\begin{array}{l} u \sqsubseteq u \\ u\!:\!x_{\sqsubseteq} \sqsubseteq u \\ u \sqsubseteq v \Rightarrow u\!:\!x_{\sqsubseteq} \sqsubseteq v \\ N_F(u) \wedge w \sqsubseteq u \wedge v \sqsubseteq u\!:\!x_F \Rightarrow R(w,v) \\ N_P(u) \wedge w \sqsubseteq u\!:\!x_P \wedge v \sqsubseteq u \Rightarrow R(w,v) \end{array}$$

is identical to the saturation of the K_i background theory.

Proof: Follows immediately by a simple induction argument.

This alternative clause set is interesting for two reasons. First, and not very surprising, $\sqsubseteq$ turns out to be an *S4* relation and second, the clauses containing an R are not recursive, i.e. they behave essentially like unit clauses. Hence, an inference system for K_i can easily be obtained by

LEMMA 6.2.7
If Φ is a K_i-unsatisfiable formula in negation normal form then

$$\lfloor\Phi\rfloor_\tau \cup \left\{ \begin{array}{l} N_F(u) \wedge w \sqsubseteq u \wedge v \sqsubseteq u\!:\!x_F \Rightarrow R(w,v) \\ N_P(u) \wedge w \sqsubseteq u\!:\!x_P \wedge v \sqsubseteq u \Rightarrow R(w,v) \end{array} \right\}$$

can be refuted by resolution, factorization and the rule

$$\frac{\alpha\!:\!\beta \not\sqsubseteq \gamma, C}{\sigma C}$$

where σ is the most general unifier of α and γ, and $\beta \in (F_{\sqsubseteq})^*$.
The formula translation is thereby defined as:

$$\begin{array}{lcl} \lfloor\Box_{\sqsubseteq}\Phi\rfloor_u & = & \forall v\; v \sqsubseteq u \Rightarrow \lfloor\Phi\rfloor_v \\ \lfloor\Diamond_{\sqsubseteq}\Phi\rfloor_u & = & \exists x \in F_{\sqsubseteq}\; \lfloor\Phi\rfloor_{u:x} \end{array}$$

The other cases remain as before.

Proof: $\Box_{\sqsubseteq}$ is an *S4* operator. The proof thus works exactly as the corresponding proof for the *S4* Inference System.

Now let us have a look at K_p, i.e. K_i with the additional assumption of transitivity for the Earlier-Later relation. However, I would like to modify K_p slightly for simplicity reasons. Recall that in the semi-functional translation method we prefer serial modalities over non-serial ones because of the annoying *Normality* predicates which cover the cases of non-normal worlds. Therefore let us assume seriality, and the interval logic we are considering, which we call K_p^s in the sequel, is thus axiomatized by

$$
\begin{array}{l}
\boxed{\mathrm{F}}\Phi \Rightarrow \Diamond_{\mathrm{F}}\Phi \\
\boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\boxed{\mathrm{F}}\Phi \\
\boxed{\mathrm{P}}\Phi \Rightarrow \Diamond_{\mathrm{P}}\Phi \\
\boxed{\mathrm{P}}\Phi \Rightarrow \boxed{\mathrm{P}}\boxed{\mathrm{P}}\Phi \\
\Box_{\sqsubseteq}\Phi \Rightarrow \Phi \\
\Box_{\sqsubseteq}\Phi \Rightarrow \Box_{\sqsubseteq}\Box_{\sqsubseteq}\Phi \\
\Diamond_{\sqsubseteq}\boxed{\mathrm{F}}\Phi \Rightarrow \boxed{\mathrm{F}}\Box_{\sqsubseteq}\Phi
\end{array}
$$

and the background theory we obtain from this axiomatization is

$$
\begin{array}{l}
R(u, u : x_F) \\
R(u : x_P, u) \\
R(u, v) \wedge R(v, w) \Rightarrow R(u, w) \\
u \sqsubseteq u \\
u : x_{\sqsubseteq} \sqsubseteq u \\
u \sqsubseteq v \wedge v \sqsubseteq w \Rightarrow u \sqsubseteq w \\
R(u, v) \wedge x \sqsubseteq u \wedge y \sqsubseteq v \Rightarrow R(x, y)
\end{array}
$$

where again it is already taken into account that R_P can be described in terms of R_F by $R_F(u, v) \Leftrightarrow R_P(v, u)$. Now, saturating this background theory is not a very difficult task.

LEMMA 6.2.8
The saturation of the K_p^s background theory consists of all unit clauses of the form

$$
\begin{array}{rl}
u : \overline{x} \sqsubseteq u & \textit{with } \overline{x} \in (F_{\sqsubseteq})^* \\
R(u : x_P : \overline{y}, u : \overline{z}) & \textit{with } \overline{y} \in (F_{\sqsubseteq} \cup F_{R_P})^* \\
R(u : \overline{y}, u : x_F : \overline{z}) & \textit{and } \overline{z} \in (F_{\sqsubseteq} \cup F_{R_F})^*
\end{array}
$$

Proof: For the first clause schema this should be obvious. For the others note that the clauses

$$
R(u, v) \Rightarrow \begin{cases} R(u, v : x_F) \\ R(u : x_P, v) \\ R(u, v : x_{\sqsubseteq}) \\ R(u : x_{\sqsubseteq}, v) \end{cases}
$$

can be derived from the background theory. By induction it then follows that

$$
R(u, v) \Rightarrow R(u : \overline{y}, v : \overline{z})
$$

with $\overline{y} \in (F_{\sqsubseteq} \cup F_{R_P})^*$ and $\overline{z} \in (F_{\sqsubseteq} \cup F_{R_F})^*$. The two R-schemata can then be derived by performing a resolution step with the R-unit-clauses from the background theory.
In order to show that no more clauses are derivable it suffices to resolve the schemata with the transitivity and the monotonicity clauses. This is trivial in case of the $\sqsubseteq$ predicate. The transitivity of the R predicate works almost as easy. We illustrate this for the case where we only consider the schema $R(u : \overline{y}, v : x_F : \overline{z})$ A first resolution step results in

$$
R(v : x_F : \overline{z}, w) \Rightarrow R(v : \overline{y}, w)
$$

with $\overline{z} \in (F_{\sqsubseteq} \cup F_{R_F})^*$ and $\overline{y} \in (F_{\sqsubseteq} \cup F_{R_P})^*$. A further resolution step with $R(u:\overline{y}, v:x_F:\overline{z})$ is only possible if u gets instantiated with $v:x_F:\overline{z_1}$. The resulting unit clause fits the pattern of the second clause schema and thus the transitivity of the R predicate is shown redundant. The monotonicity clause causes even less problems. It merely adds further $F_{\sqsubseteq}$ variables to either argument of an R predicate and therefore produces nothing new.

The simplicity of the K_p^s background theory immediately gives rise to the following inference system.

DEFINITION 6.2.9 (THE K_p^s INFERENCE SYSTEM)
The K_p^s Inference System *consists of the classical resolution and factorization rule together with*

$$\frac{\alpha:\beta \not\sqsubseteq \gamma, C}{\sigma C} \qquad \textit{where } \sigma = mgu(\alpha,\gamma) \textit{ and } \beta \in (F_{\sqsubseteq})^*$$

$$\frac{\neg R(\alpha:\beta, \gamma:\delta), C}{\sigma C} \qquad \textit{where } \sigma = mgu(\alpha,\gamma), \beta \in (F_{\sqsubseteq} \cup F_{R_P})^*, \delta \in (F_{\sqsubseteq} \cup F_{R_F})^* \textit{ and either } \beta \textit{ starts with a } F_{R_P} \textit{ term or } \delta \textit{ starts with a } F_{R_F} \textit{ term}$$

$$\left.\begin{array}{c} \dfrac{\neg R(\alpha,u), C}{C^u_{\alpha:x_F}} \\[2ex] \dfrac{\neg R(u,\alpha), C}{C^u_{\alpha:x_P}} \end{array}\right\} \quad \textit{provided } u \textit{ does not occur in } \alpha$$

LEMMA 6.2.10
The K_p^s Inference System is sound.

Proof: this follows trivially from the fact that every application of one of the two new inference rules corresponds to a resolution step with an instance of the background theory saturation and thus corresponds to a sequence of resolution steps with the K_p^s background theory clauses.

LEMMA 6.2.11
The K_p^s Inference System is refutation complete.

Proof: Evidently, it is possible to derive a single K_p^s-unsatisfiable clause C from an arbitrary K_p^s-unsatisfiable set of clauses. Now consider a minimal unsatisfiable ground instance σC of C. If none of the steps in σC can be lifted then each literal in C contains an overestimated variable. In analogy to the refutation completeness proof for the *S4* and the *KD4* Inference Systems we would be able to find arbitrarily many different interval variables and this is impossible. Hence, at least one step on the ground level must be liftable and what has been claimed follows by induction over the length of C.

The monotonicity schemata are the only axioms in K_i and K_p^s that express some interrelation between temporal ordering and subintervals. Several additional axioms and properties can be imagined, however. For instance, one might require some kind of linearity[3] or "super-interval property" which states that every two intervals have a common superinterval.

[3]In interval logics one would certainly not require that any two intervals are temporally ordered in the sense that one of them is (temporally) before the other. A more appropriate property would be: either one is before the other or the two intervals do overlap.

Another very interesting property would be to demand that intervals are convex. Convexity is usually defined in terms of points which form the interval. Such a description would not work here since there are no time instants available in the interval logics we are considering. Its formulation in terms of intervals could be: Any interval (temporally) inbetween two other intervals is part of the convex union of the other two, or, more formally,

$$\forall u, v, w, v' \; R(u,v) \wedge R(v,w) \wedge u \sqsubseteq v' \wedge w \sqsubseteq v' \Rightarrow v \sqsubseteq v'$$

It might be surprising at the first glance that the temporal logics we are considering cannot distinguish between arbitrary interval structures and convex interval structures.

LEMMA 6.2.12
Convexity does not contribute anything new to the saturation of the background theory of K_p^s.
Proof: Consider the convexity property

$$R(u,v) \wedge R(v,w) \wedge u \sqsubseteq v' \wedge w \sqsubseteq v' \Rightarrow v \sqsubseteq v'$$

and perform resolution steps between the $\sqsubseteq$-literals and the saturation of the K_p^s background theory. This results in the clause schema

$$R(v' : \overline{x}, v) \wedge R(v, v' : \overline{y}) \Rightarrow v \sqsubseteq v'$$

where both $\overline{x}$ and $\overline{y}$ belong to $(F_{\sqsubseteq})^*$. Now these two R-literals have to be simultaneously unified with the schemata of the form $R(u : \overline{x}, u : y_F : \overline{z})$ or $R(u : y_p : \overline{x}, u : \overline{z})$ from the given saturation. Such an attempt, however, must fail as exemplified by the following: If the literal $\neg R(v' : \overline{x}, v)$ is resolved with $R(u : y_p : \overline{x}, u : \overline{z})$ then we obtain

$$R(u : \overline{z}, u : y_P : \overline{x} : \overline{x}') \Rightarrow u : \overline{z} \sqsubseteq u : y_P : \overline{x}.$$

No further resolution steps are possible because of sort clashes and hence no new pure-positive clause can be derived. Therefore, the pure-positive clauses that are derivable with the additional convexity property are exactly the ones we already had.

COROLLARY 6.2.13
K_p^s is a logic of convex interval structures.

The fact that our temporal logics cannot distinguish between convex and non-convex structures has an immediate effect on the axiomatizability of this property.

COROLLARY 6.2.14
Convexity is not axiomatizable.

Proof: Since the saturation of the K_p^s background theory is not influenced by the convexity property we know that any formula which can be proved under the convexity assumption also follows from K_p^s alone. Now assume there were an axiom schema Φ for convexity. Since this property does not follow from K_p^s there must exist an instance of Φ which is valid under convexity but cannot be proved valid without this extra assumption. This, however, contradicts the fact that no such formula exists.

A final remark on convexity: Humberstone assumed a property which is somewhat related to convexity and linearity, namely[4]: Any interval later than some subinterval of the current interval is either later than, or overlaps with, the current interval. I.e.

$$\forall u, v, w \;\; v \sqsubseteq u \wedge R(v, w) \Rightarrow (R(u, w) \vee \exists x \; (x \sqsubseteq u \wedge x \sqsubseteq w))$$

A possible axiom which characterizes this property can be found in

$$\Diamond_{\sqsubseteq} \Diamond_{F} (\Box_{\sqsubseteq} \Phi \wedge \Psi) \Rightarrow (\Diamond_{\sqsubseteq} \Phi \vee \Diamond_{F} \Psi)$$

This property is not further examined here. It is interesting in so far as it disallows temporal branching within intervals.

[4]Actually, the sequent rule introduced by Humberstone is a bit different; it corresponds to the same first-order property, however.

7

Summary and Future Work

7.1 The Approach

7.1.1 The General Framework

Although only applied to modal and temporal logic theorem proving, the approach presented in this work is actually fairly general. It can be briefly summarized as follows: Given as input a set of clauses and some distinguished predicate symbol R that occurs within these clauses, we divide the input into two disjoint subsets such that one of the two consists of exactly those clauses which contain a positive occurrence of R. This set we call the positive theory of R[1]. This theory now has to be saturated, i.e. we look for a finite set of clause schemata which represents all the pure-R-positive clauses that are derivable within this theory. Provided we are successfull we can make use of the saturation result in two different ways: Either we try to find a finite set of R-positive clauses which is somewhat simpler than the theory of R but nevertheless generates the same saturation, or we transform the schemata into suitable inference rules which then may replace the theory.

Either possibility is fairly large-scale and it certainly would not make very much sense to perform the whole procedure for every theorem to be proved. If, however, the theory is known beforehand such that formulae cannot add anything further to it[2] then the procedure described above can be applied once and for all for the logic itself, and that independently of the theorems to be proved.

[1]Obviously, we might equally consider the negative occurrences and speak of the negative theory then.

[2]I.e. in case of a positive theory there are only (negative) R-constraints in the non-theory clauses.

7.1.2 Application to Modal and Temporal Logics

The naive relational translation into first-order predicate logic certainly does not help here since the accessibility relation symbol R (which is the only imaginable candidate for a background theory) may occur both positively and negatively in the translation result[3].

A way out of the dilemma can be found by applying the semi-functional translation approach. There are several advantages with this method. First of all the number of generated clauses is identical to the number of clauses generated by the functional translation approach and is therefore small compared to the relational translation result. Moreover, it does not produce any positive R-literals and this means that the background theory is strictly separated from the theorem to be proved. This often allows us to detect – with simple proof-theoretical means – the non-axiomatizability of accessibility relation properties . Also, and this might be interesting for people working in the area of modal and temporal logic programming, the translation output is in Horn form if and only if the input formulae are in Horn form.

Because of the strict separation of the logic background theory from the theorem to be proved, the semi-functional translation fits the pattern of the general framework.

There remains only one further question, namely how to obtain the background theories for the logics we are interested in. In cases were the logic is described in terms of modal interpretations – and thus properties of the underlying accessibility relation – this theory is already given. Otherwise certain axioms are provided such that the theory has to be obtained from such an axiomatization. In case of well-known modal and temporal logics the respective correspondences can be found in any related text-book. In all the other cases we have to apply a second-order quantifier elimination to obtain the corresponding property. One such possibility occurs in this text as the Elimination Theorem[4]. What the semi-functional approach additionally demands is the extra clause $R(u, u\!:\!x)$ (or $N(u) \Rightarrow R(u, u\!:\!x)$ for non-serial logics) and we are back in the general framework described above, i.e. we saturate the resulting theory and either find a simpler alternative theory with identical saturation or transform it into a suitable inference system.

7.2 Possible Caveats

Obviously, there are possibilities where the one or the other step in the informal procedure from above cannot be applied. For instance, it might happen that the given axiomatization results in properties which are not first-order, witness the discreteness property which is often used in connection with program verification. In such a case the approach wouldn't work. However, it is often possible to describe such properties in terms of infinitary logic (fixpoint calculus) as it is described in the Elimination Theorem. Although it is not clear yet how such fixpoint formulae can be used in a saturation process, it is at least worth an examination. Finally, there might be problems with the saturation process itself. As an example consider the modal logic *S4.2* which does not behave very nicely during the attempt to saturate its background theory. In such cases it is often possible to define some suitable auxiliary operators which are to be described in terms

[3]For some very sophisticated clause form transformations a suitable seperation would be possible, though. Nevertheless, the theory gets so complicated then that the saturation process turns out to be very difficult.

[4]Note that the completeness of the axiomatization with respect to the interpretations we obtain by finding correspondence axioms has to be shown seperately then.

of the already existing operators such that the resulting theory can be saturated more easily (as an example see again the modal logic *S4.2*).

Still, there are lots of possible sources for the one or the other technical problem, be it in finding suitable inference rules or proving the soundness and the completeness of the resulting inference system. There is no fully automatic system yet and, in fact, such a system cannot really be expected.

7.3 Other Properties and Operators

7.3.1 Accessibility Relation Properties

There are fairly interesting accessibility relation properties which are first-order definable but have not been examined in this work. One of these is *density*, i.e. the property that for two arbitrary worlds α and β with $R(\alpha,\beta)$ there is a third world γ with $R(\alpha,\gamma)$ and $R(\gamma,\beta)$. Saturating the resulting background theory is certainly possible, although with similar technical problems as in case of *S4.2*. There is some evidence however, that the definition of auxiliary operators helps here as well. In case of temporal logics we may consider the axiom $\langle F \rangle \boxed{P}\,\Phi \Rightarrow \langle F \rangle\,\Phi$ instead of the usual density axiom $\Box\Box\Phi \Rightarrow \Box\Phi$, define $\blacksquare$ as $\langle F \rangle \boxed{P}$, and proceed as in case of *S4.2*.

Quite opposite to density is *discreteness*, a property which is not first-order describable and therefore can only be handled by the proposed approach if there is a way to deal with fixpoints in the saturation process. Nothing along these lines has been done yet but the general idea is certainly worth an examination and that also with regard to such complicated properties.

7.3.2 Operators

An interesting operator which frequently occurs in the literature of temporal logic is the so called *Next*-operator. It often plays a role in applications like program specification and verification. The existence of this operator is evidently closely related to the discreteness of the underlying earlier-later relation, although not necessarily with all its inductive power. We can easily imagine some kind of weak discreteness which is first-order definable and for which a *Next*-operator would make sense. In this case the *Next* would have to be treated just as the individual belief operator in the examination of "mutual belief" in Section 3.8.

Further interesting operators can be found for the so called CTL-like structures (Emerson and Halpern 1986) where it is possible to explicitly quantify over the various branches within a temporal structure. With such languages it is possible to distinguish between the *inevitably* and the *eventually* and also between the *henceforth* and the *possibly forever*. CTL itself (and its even more complicated sibling CTL*) is defined on the basis of a discrete structure, however. Nevertheless, even if discreteness is not assumed, such operators definitly make sense and it would be worthwhile to examine how the approach presented in this thesis can possibly handle them.

7.4 Decidability of the Propositional Fragment

The logics considered in this work are all first-order and semi-decidable. However, most of the propositional fragments of the respective logics are even known to be decidable. Decidability has not played a role in this work, though. It would certainly be interesting to examine whether – and if so, how – the obtained calculi can be extended by ordering strategies and/or depth restrictions such that a derivation process definitly will stop eventually provided the logic under consideration is indeed decidable.

7.5 Comparison with Other Approaches

As the functional translation method is the "forefather" of the semi-functional approach it is not too surprising that there are the most common grounds. Essentially, what the two approaches have in common is that they use similar translations into first-order predicate logic such that the translation results of the one can quite easily be transformed into the translation result of the other. Both methods do not really need anything more since the clauses obtained after translation are (classically) satisfiable if and only if the original modal formula was (modal logic) satisfiable. At this stage both methods are not yet very convincing, though. It is therefore tried to make use of certain translational invariants and the special knowledge about the respective modal logic background theories. This is where the two approaches differ significantly. In the functional translation the background theory is described in terms of an equational system which sometimes can be transformed into a suitable unification algorithm provided the theory consists only of unit equations. However, if accessibility relation properties like *linearity* are to be considered then the equational system has to remain as is. This is a rather serious problem because it is not at all clear how an appropriate guidance through the search space could be described then. In the semi-functional translation approach the background theory does not change compared to the relational translation[5]. Nevertheless, the particular syntactic structure obtained this way has some nice and interesting properties which can be utilized in the saturation of the background theory.

A quite closely related approach had been defined in (chung Chan 1990). In this work the propositinal modal logic *S4.3* is examined with respect to the functional translation approach, and that quite similarly to Section 3.10, although with a different set of inference rules. For instance, according to the calculus defined in this thesis, it is possible to derive $P(\beta)$ from $P(\alpha : x)$ and $\neg R(\beta, \alpha)$[6]. Such a derivation is not possible within the *S4.3* Inference System as it is described in Section 5.1. However, from the *S4.3* Inference System it would be possible to derive the empty clause from $\neg P(\beta)$, $P(\alpha : x)$, and $\neg R(\beta, \alpha)$. Chan's system thus seems to be a bit more forward-directed than the *S4.3* Inference System. Unfortunately, (chung Chan 1990) lacks completeness results and even soundness is merely checked by some operational justification of pseudo-code.

Christoph Brzoska's translation approach (Brzoska 1993) is also related to some extent. His target language is not temporal logic in general, however. What he is rather interested in is temporal logic programming and therefore he assumes certain syntactic restrictions on the temporal logic formulae under consideration. The translation itself is nevertheless closely related to the functional translation method. A major difference can be found in the treatment of the background theory. In (Brzoska 1993) this theory is taken care of by some constraint solver which looks for integer solutions for some given integer constraint problem. The special syntactic

[5]There are some exceptions, witness *S4.2*. These are not crucial, however.

[6]Some liberties have been taken with the original formulation.

structure of translated formulae is thereby an additional source of simplification possibilities[7].

Although not directly related, known extensions of the tableau or Gentzen type sequent calculi should be compared as well (see for example (Rescher and Urquhart 1971) and (Goré 1993)). For modal logics like *S4* such tableau systems look fairly promising. For example, a typical *S4* tableau rule looks like this:

$$\frac{\Diamond A, \Diamond\Delta, \Box\Gamma, \Omega}{A, \Box\Gamma}$$

Informally, this rule is to be interpreted as follows: Having a tableau for some world in which A and formulae Δ are in the scope of some $\Diamond$s, formulae Γ are in the scope of $\Box$s and Ω denotes classical formulae which cannot be further split by the standard tableau rules, we switch to a new accessible world in which A is true (according to the $\Diamond A$) and guarantee that $\Box\Gamma$ holds there as well. This rule is motivated by the fact that in such a new world at least all the formulae in Γ should be true because these hold in every world accessible from the current one and thus in particular in the world accesses by the $\Diamond$ in $\Diamond A$. Moreover, because of the transitivity of the accessibility relation, we have that $\Box\Gamma$ holds there and by reflexivity it is even the case that Γ is subsumed by $\Box\Gamma$. We may thus ignore all the $\Diamond\Delta$ if we are able to close the tableau for $\Diamond A$ already, i.e. the formulae in Δs are of no further importance then.

This approach works in (possibly branching) tree structures because different accessible worlds do not necessarily have to be related. The situation changes considerably if linearity comes into play. Suddenly any two worlds with a common predecessor are comparable with respect to the earlier-later relation. Suppose the current tableau contains exactly two $\Diamond$-formulae, $\Diamond A$ and $\Diamond B$. From the *S4.3* axiom *3* we know that one of the two (or both) comes first. If the world in which A holds comes first then $\Diamond B$ holds there; and if the world in which B holds comes first then $\Diamond A$ holds there. This means that there are two different branches to be openend, one for each possible situation. This may not yet be too crucial for two $\Diamond$ formulae, but if there are more, say n, we see that $n!$ (in words: n faculty) such branches have to be opened and there seems no way to simplify or to reduce this. Even worse, if not *S4.3* but *KD4.3'* is considered then for any pair there are three possible branches to be opened and this increases the branching rate by another factor of n. As an example consider the following formula set

$$\begin{array}{l}\Diamond(P_1 \wedge \Box\neg P_2)\\ \Diamond(P_2 \wedge \Box\neg P_3)\\ \ldots\\ \Diamond(P_{n-1} \wedge \Box\neg P_n)\\ \Diamond(P_n \wedge \Box\neg P_1)\end{array}$$

This set is *S4.3*-unsatisfiable and what the tableau calculus has to do is to produce $n!$ branches each of which can be closed with the *S4* subsystem.

[7] In addition, Brzoska considers metric operators and dense structures.

Comparing this with the semi-functional translation approach we get:

$$
\begin{array}{l}
P_1(\iota : a_1) \\
\neg R(\iota : a_1, u) \vee \neg P_2(u) \\
P_2(\iota : a_2) \\
\neg R(\iota : a_2, u) \vee \neg P_3(u) \\
\ldots \\
P_n(\iota : a_n) \\
\neg R(\iota : a_n, u) \vee P_1(u)
\end{array}
$$

And after n resolution steps we end up with

$$
\begin{array}{l}
\neg R(\iota : a_1, \iota : a_2) \\
\neg R(\iota : a_2, \iota : a_3) \\
\ldots \\
\neg R(\iota : a_n, \iota : a_1)
\end{array}
$$

These resulting clauses can be refuted by n further *S4.3* Inference Rule steps and therefore only $2 \times n$ steps are necessary to derive the empty clause from the original formulae; a small number compared to $n!$.

A calculus somewhere in the middle between the functional translation and the above tableau system can be found in the so called *prefixed tableaux* (see (Fitting 1983), (Reddy 1995)). Here the idea is to associate prefixes with modal formulae which essentially represent worlds and to manipulate such extended modal formulae as follows[8]:

$$
\begin{array}{lcl}
\alpha : \boxed{F}\,\Phi & \rightarrow & \alpha\, g_i : \Phi \\
\alpha : \langle F \rangle\,\Phi & \rightarrow & \alpha\, f_i : \Phi \\
\alpha : \boxed{P}\,\Phi & \rightarrow & \alpha\, h_i : \Phi \\
\alpha : \langle P \rangle\,\Phi & \rightarrow & \alpha\, p_i : \Phi
\end{array}
$$

where each transformation gets a new index number. The rest of the tableau rules are just as usual with the exception that a tableau can only be closed if there are two complementary formulae in this tableau and the corresponding prefixes *codesignate*. Modal logics are distinguished by the different definitions for codesignation. It is not necessary here to provide with the full definition in case of temporal logics. It should suffice to note that essentially two prefixes codesignate if they are unifiable in the sense of the functional translation. The calculus developed by Reddy seems to be complete for K_tD4; there is no proof, however. The actual aim, namely the development of a calculus for discrete $K_tD4.3$', is not fulfilled, not even if discreteness is dropped. Reddy remarks that linearity cannot be handled that easily and proposes a, as he called it, *conditional codesignation*. The main idea is to close branches (which get a certain condition), to combine the resulting conditions, and to check whether this combination is universally valid (under linearity). This has not been worked out in the thesis, however.

[8]The choice of the small letters has its origin in the Priorian tense operators G, F, H, and P.

Bibliography

Åqvist, L. and Guenthner, F.: 1977, Tense logic, *Logique et Analyse* **80**.

Abadi, M.: 1987a, The power of temporal proofs, *Proceedings of the Symposium on Logic in Computer Science*, Ithaca, pp. 176–186.

Abadi, M.: 1987b, *Temporal Logic Theorem Proving*, PhD thesis, Stanford University, Department of Computing. Appeared as internal report STAN-CS-87-1151.

Abadi, M. and Manna, Z.: 1986a, Modal theorem proving, *Proceedings 8th CADE, LNCS 230*, Springer.

Abadi, M. and Manna, Z.: 1986b, A timely resolution, *1st Annual Symposium on Logic in Computer Science*, IEEE Computer Society.

Abadi, M. and Manna, Z.: 1989, Temporal logic programming, *Journal of Symbolic Computation* **8**.

Abadi, M. and Manna, Z.: 1990, Nonclausal deduction in first-order temporal logic, *Journal of the ACM* **37**(2).

Allen, J. F.: 1981a, A general model of action and time, *Technical Report TR97*, University of Rochester.

Allen, J. F.: 1981b, An interval based representation of temporal knowledge, *Proceedings 7th IJCAI*, pp. 221 – 226.

Allen, J. F.: 1981c, Maintaining knowledge about temporal intervals, *Technical Report TR86*, University of Rochester.

Allen, J. F.: 1983, Maintaining knowledge about temporal intervals, *Communications of the ACM* **26**(11), 832 – 843.

Allen, J. F.: 1984, Towards a general theory of action and time, *AI* **23**(2), 123 – 154.

Allen, J. F. and Hayes, P. J.: 1985a, A commonsense theory of time, *Proceedings of the 9th IJCAI*, pp. 528 – 531.

Allen, J. F. and Hayes, P. J.: 1985b, A commonsense theory of time: The longer paper, *Technical report*, University of Rochester.

Allen, J. F. and Hayes, P. J.: 1985c, Moments and points in an interval-based temporal logic, *Technical Report TR180*, University of Rochester, Department of Computer Science.

Allen, J. F. and Hayes, P. J.: 1987, Short-time periods, *Proceedings 10th IJCAI.*

Amir, A. and Gabbay, D.: 1987, Preservation of expressive completeness in temporal models, *Information and Computation* **72**, 66–83.

Auffray, H.: 1989, Résolution modale et logique des chemins, *Technical report*, University of Caen, Department of Informatics.

Auffray, Y. and Enjalbert, P.: 1992, Modal theorem proving: An equational viewpoint, *Journal of Logic and Computation* **2**(3), 247 – 295.

Bachmair, L. and Ganzinger, H.: 1990, On restrictions of ordered paramodulation with simplification, *Proceedings CADE'90.*

Bachmair, L. and Ganzinger, H.: 1994a, Rewrite-based equational theorem proving with selection and simplification, *Journal of Symbolic Computation* **4**(3), 217–247.

Bachmair, L. and Ganzinger, H.: 1994b, Rewrite techniques for transitive relations, *Proceedings 9th IEEE Symposium on Logic in Computer Science.*

Baudinet, M.: 1989, Temporal logic programming is complete and expressive, *Proceedings of the ACM Symposium on Principles of Programming Languages*, Austin, Texas.

Ben-Ari, M., Manna, Z. and Pnueli, A.: 1981a, The logic of nexttime, *Proceedings of the 8th Symposium Principles of Programming Languages*, ACM.

Ben-Ari, M., Manna, Z. and Pnueli, A.: 1981b, The temporal logic of branching time, *Proceedings of the 8th Symposium Principles of Programming Languages*, ACM.

Benanav, D.: 1990, Simultaneous paramodulation, *Proceedings of the 10th International Conference on Automated Deduction.*

Blackburn, P.: 1989, *Reasoning in and about Time*, PhD thesis, Center for Cognitive Science, University of Edinburgh.

Boolos, G. and Sambin, G.: 1985, An incomplete system of modal logic, *Journal of Philosophical Logic* **14**.

Bowen, K. and de Jongh, D.: 1986, Some complete logics for branched time, *Technical Report Technical Report 86-05*, Institute for Language, Logic and Information.

Brown, M., Clarke, E. and Grumberg, O.: 1987, Characterizing kripke structures in temporal logic, *Proceedings Colloquium on Trees in Algebra and Programming*, Pisa.

Brzoska, C.: 1989, Temporal logic programming: A survey, *Technical report*, Universität Karlsruhe.

Brzoska, C.: 1990, Temporal logic programming and its relation to constraint logic programming, *Technical report*, University of Karlsruhe, Germany.

Brzoska, C.: 1992, Temporal logic programming with metric and past operators based on constraint logic programming, *Technical report*, University of Karlsruhe. second version.

Brzoska, C.: 1993, *Temporal-logisches Programmieren*, PhD thesis, Universität Karlsruhe, Germany.

Bull, R. A.: 1964, A note on the modal calculi S4.2 and S4.3, *Zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **10**.

Bull, R. A.: 1965, An algebraic study of diodorean modal systems, *Journal of Symbolic Logic* **30**.

Bull, R. A.: 1966, That all normal extensions of S4.3 have the finite model property, *Zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **12**.

Bull, R. A.: 1968, An algebraic study of tense logics with linear time, *Journal of Symbolic Logic* **33**(1).

Bull, R. A. and Segerberg, K.: 1984, Basic modal logic, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic, vol II*, Reidel, Dordrecht.

Bürckert, H.-J.: 1990, A resolution principle for clauses with constraints, *in* M. E. Stickel (ed.), *Proceedings of the 10th International Conference on Automated Deduction, LNAI 449*, Springer Verlag.

Bürckert, H.-J.: 1991, *A Resolution Principle for a Logic with Restricted Quantifiers*, Vol. 568 of *Lecture Notes in AI*, Springer Verlag.

Burgess, J. P.: 1979, Logic and time, *Journal of Symbolic Logic* **44**(4), 566–582.

Burgess, J. P.: 1980, Decidability for branching time, *Studia Logica* **39**, 203–218.

Burgess, J. P.: 1982a, Axioms for tense logic I: "since" and "until", *Notre Dame Journal of Formal Logic* **23**(2), 367–374.

Burgess, J. P.: 1982b, Axioms for tense logic II: Time periods, *Notre Dame Journal of Formal Logic* **23**(2), 375–383.

Burgess, J. P.: 1984a, Basic tense logic, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic, vol II*, Vol. II, Reidel, Dordrecht, pp. 89–113.

Burgess, J. P.: 1984b, Beyond tense logic, *Journal of Philosophical Logic* **13**, 235–248.

Burgess, J. P. and Gurevich, Y.: 1985, The decision problem for linear time temporal logic, *Notre Dame Journal of Formal Logic* **26**(2), 115–128.

Carnap, R.: 1946, Modalities and quantification, *Journal of Symbolic Logic* **11**(2).

Cavalli, A. and Fariñas del Cerro, L.: 1984, A decision method for linear temporal logics, *Lecture Notes in Computer Science 170*, Springer.

Chellas, B. F.: 1980, *Modal Logic, an introduction*, Cambridge University Press.

chung Chan, M.: 1990, *A Theory of Functional Unification: an Approach to Automated Reasoning in a Logic of Discrete and Linear Time*, PhD thesis, La Trobe University, Bundoora, Australia.

Clarke, E. and Sistla, A. P.: 1985, Complexity of propositional linear temporal logics, *Journal of the ACM* **32**, 733–749.

Clifford, J. E.: 1966, Tense logic and the logic of change, *Logique et Analyse* **9**.

Cocchiarella, N.: 1965, *Tense and Modal Logic: A Study of the Topology of Temporal Reference*, PhD thesis, University of California, Los Angeles.

Cocchiarella, N.: 1966, A completeness theorem for tense logic, *Journal of Symbolic Logic* **31**, 688–691.

Cresswell, M.: 1975, Frames and models in modal logic, *in* J.N.Crossley (ed.), *Algebra and Logic, Lecture Notes in Mathematics 450*, Springer.

Cresswell, M.: 1984, An incomplete decidable modal logic, *Journal of Symbolic Logic* **49**(2).

de Jongh, D. and Bowen, K.: 1986, Some complete logics for branched time, *Technical Report 86-05*, Institute for Logic, Language and Information, University of Amsterdam.

de Jongh, D., Feltman, F. and Verbrugge, R.: 1988, The logics of dense and discrete time, *Technical report*, Institute for Logic, Language and Information, University of Amsterdam.

de Jongh, D., Veltmann, F. and Verbrugge, R.: 1988, The logics of dense and discrete time, *Technical report*, Institute for Logic, Language and Information, University of Amsterdam.

Doets, K.: 1989, Monadic Π^1_1-theories of Π^1_1-properties, *Notre Dame Journal of Formal Logic* **30**, 224–240.

Dosen, K.: 1985, Sequent-systems for modal logic, *Journal of Symbolic Logic* **50**(1), 149–168.

Dummet, M. and Lemmon, E.: 1959, Modal logics between S4 and S5, *Zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **5**.

Echagüe, J. V.: 1991, A Gentzen-system for CTL, *RR 865-I*, LIFIA-IMAG, Grenoble.

Emerson, E. A.: 1983, Alternative semantics for temporal logics, *Theoretical Computer Science* **26**.

Emerson, E. A.: 1990, Temporal and modal logic, *in* van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Vol. B, Elsevier, chapter 16, pp. 996 – 1072.

Emerson, E. A. and Clarke, E.: 1982, Using branching time temporal logic to synthesize synchronization skeletons, *Science of Computer Programming* **2**.

Emerson, E. A. and Halpern, J.: 1982, Decision procedures and expressiveness in the temporal logic of branching time, *14th Symposium on Theory of Computing*, ACM.

Emerson, E. A. and Halpern, J.: 1983, Sometimes and not never revisited: On branching versus linear time, *10th Symposium on Principles of Programming Languages*, ACM.

Emerson, E. A. and Halpern, J.: 1986, "Sometimes" and "Not Never" revisited: On branching versus linear time, *Journal of the ACM* **33**(1), 151 – 178.

Emerson, E. A., Sadler, T. and Srinivasan, J.: 1989, Efficient temporal reasoning, *Proceedings 16th POPL*.

Emerson, E. A. and Sistla, A. P.: ??, Deciding full branching time logic, *Information and Control* **61**.

Emerson, E. A. and Sistla, A. P.: 1984, Deciding branching time logic, *16th Symposium on Theory of Computing*, ACM.

Emerson, E. A. and Srinivasan, J.: 1988, Branching time temporal logic, *LNCS 354*, Springer.

Fariñas del Cerro, L.: 1982, A simple deduction method for modal logic, *Information Processing Letters* **14**(2).

Fariñas del Cerro, L.: 1985, Resolution modal logics, *in* K.Apt (ed.), *Logics and Models of Concurrent Systems*, NATO ASI Series. Series volume 13.

Fariñas del Cerro, L. and Herzig, A.: 1988, Quantified modal logic and unification theory, *Rapport LSI 293*, Languages et Systèmes Informatique, Université Paul Sabatier, Toulouse.

Fine, K.: ??, Model theory for modal logic, *Journal of Philosophical Logic* **10**.

Fine, K.: 1971, The logics containing S4.3, *Zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **17**.

Fine, K.: 1978, Model theory for modal logic i: The 'de re'/'de dicto' distinction, *Journal of Philosophical Logic* **7**, 125–156.

Fine, K.: 1981, First-order modal theories, *Noûs* **15**.

Finger, M. and Gabbay, D.: 1991, Adding a temporal dimension to a logic, *Technical report*, Imperial College, London.

Fisher, M.: 1988, Implementing a prototype Metatem interpreter, *Technical report*, University of Manchester, Department of Computer Science.

Fitting, M.: 1983, *Proof Methods for Modal and Intuitionistic Logics*, Reidel.

Fitting, M.: 1989, Destructive modal resolution, Oxford University Press.

Gabbay, D.: 1969, Model theory for tense logics, *Scientific Report 1*, Hebrew University, Jerusalem.

Gabbay, D.: 1971, On decidable, finitely axiomatizable, modal and tense logics without the finite model property, *Israel Journal of Mathematics* **10**, 478–503.

Gabbay, D.: 1973, A survey of decidability results for modal, tense and intermediate logics, *in* P. Suppes (ed.), *Proceedings of the fourth International Congress on Logic, Methodology and Philosophy of Science*, North Holland, pp. 29–43.

Gabbay, D.: 1975, Model theory for tense logics, *Annals of Mathematical Logic* **8**, 185–235.

Gabbay, D.: 1976, *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics*, Vol. 92 of *Synthese Library*, Reidel, Dordrecht.

Gabbay, D.: 1981, An irreflexivity lemma with applications to axiomatizations of conditions on tense frames, *in* U. Mönnich (ed.), *Aspects of Philosophical Logic*, Reidel, Dordrecht, pp. 67–89.

Gabbay, D.: 1986, Modal and temporal logic programming, *Technical Report 86/15*, Imperial College of Science and Technology, London, GB.

Gabbay, D.: 1987, Executable temporal logic for interactive systems, *Technical report*, Imperial College of Science and Technology, London, GB.

Gabbay, D.: 1989a, The declarative past and imperative future, *Technical report*, Imperial College of Science and Technology, Department of Computing, London.

Gabbay, D.: 1989b, Temporal logics: Tense or non-tense, *in* R. Spencer-Smith and S. Torrance (eds), *Machinations, Computational Studies of Logic, Language and Cognition*, Ablex Publishing Company.

Gabbay, D.: 1992a, *Labelled Deductive Systems*, Oxford University Press.

Gabbay, D.: 1992b, Temporal logic: Mathematical foundations, *Technical Report MPI-I-92-213*, Max-Planck-Institut für Informatik.

Gabbay, D.: 1992c, *Temporal Logic: Mathematical Foundations and Computational Aspects*, Oxford University Press.

Gabbay, D. and Guenthner, F. (eds): 1984, *Handbook of Philosophical Logic*, Reidel.

Gabbay, D. and Hodkinson, I.: 1990, An axiomatization of the temporal logic with until and since over the real numbers, *Journal of Logic and Computation* **1**(2), 229–260.

Gabbay, D. M. and Ohlbach, H. J.: 1992a, Quantifier elimination in second–order predicate logic, *South African Computer Journal* **7**, 35–43.

Gabbay, D. M. and Ohlbach, H. J.: 1992b, Quantifier elimination in second–order predicate logic, *in* B. Nebel, C. Rich and W. Swartout (eds), *Principles of Knowledge Representation and Reasoning (KR92)*, Morgan Kaufmann, pp. 425–435.

Gabbay, D. and Moravcsik, J.: 1980, Verbs, events and the flow of time, *Time, Tense and Quantifiers: Proceedings of the Stuttgart Conference on the Logic of Tense and Quantification*, Max Niemayer Verlag, Tübingen, Germany.

Gabbay, D. and Ohlbach, H. J.: 1992c, From a hilbert calculus to its model theoretic semantics, *Technical report*, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Galton, A.: 1987a, Temporal logic and computer science: An overview, *in* A. Galton (ed.), *Temporal Logics and Their Applications*, Academic Press.

Galton, A.: 1990, A critical examination of J.F. Allen's Theory of Action and Time, *Artificial Intelligence* **42**, 159–188.

Galton, A. (ed.): 1987b, *Temporal Logics and Their Applications*, Academic Press, London, UK.

Garson, J.: 1984, Quantification in modal logic, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic*, Reidel.

G.Kutty, L.E.Moser, P.M.Melliar-Smith, Y.S.Ramakrishna and L.K.Dillon: 1994, Axiomatizations of Interval Logics, *Technical report*, University of California, Santa Barbera.

Goble, L. F.: 1974, Gentzen systems for modal logic, *ndjfl* **15**, 455–461.

Goldblatt, R.: 1975, First-order definability in modal logic, *Journal of Symbolic Logic* **40**.

Goldblatt, R.: 1976, *Metamathematics of modal logic*, Vol. Reports on mathematical logic, Nr 6 and 7, Polish Scientific Publishers.

Goldblatt, R.: 1980, Diodorean modality in minkowski spacetime, *Studia Logica* **39**, 219–236.

Goldblatt, R.: 1987, *Logics of Time and Computation*, Vol. 7 of *CSLI Lecture Notes*, Chicago University Press, Chicago.

Goldblatt, R. and Thomason, S.: 1975, *Axiomatic Classes in Propositional Modal Logic*, Vol. Lecture Notes in Maths 480, Springer Verlag.

Goré, R.: 1992, Cut-free Sequent and Tableau Systems for Propositional Normal Modal Logics, *Technical Report 257*, Computer Laboratory, University of Cambridge, Cambridge, England.

Goré, R.: 1993, Cut-free Sequent and Tableau Systems for Propositional Diodorian Modal Logics, *Technical report*, Computer Laboratory, University of Cambridge, Cambridge, England.

Halpern, J. and Shoham, Y.: 1986, A propositional modal logic of time intervals, *Technical report*, IBM.

Hamblin, C.: 1971, Instants and intervals, *Studium Generale* **24**.

Harrison, C.: 1969, Kripke's semantics for modal logic, *in* J. D. et al (ed.), *Philosophical Logic*, Reidel.

Herzig, A.: 1989, *Raisonnement automatique en logique modale et algorithmes d'unification.*, PhD thesis, Université Paul-Sabatier, Toulouse.

Hughes, G. and Cresswell, M.: 1968, *An Introduction to Modal Logic*, Menthuen, London.

Hughes, G. and Cresswell, M.: 1984, *A Companion to Modal Logic*, Menthuen, London.

Humberstone, I. L.: 1979, Interval semantics for tense logic: Some remarks, *Journal of Philosophical Logic* **8**, 171–196.

Jackson, P. and Reichgelt, H.: 1987, A general proof method for first-order modal logic, *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, Milano.

Kamp, H.: 1968, *Tense Logic and the Theory of Linear Order*, PhD thesis, University of California, Los Angeles.

Kanger, S.: 1957, A note on quantification and modalities, *Theoria* **23**.

Kowalski, R. and Sergot, M.: 1986, A logic-based calculus of events, *New Generation Computing* **4**(1).

Kripke, S.: 1959, A completeness theorem in modal logic, *Journal of Symbolic Logic* **24**.

Kripke, S.: 1963, Semantical considerations on modal logics, *Acta Philosophica Fennica: Modal and Many-Valued Logics* **16**, 83–94.

Ladkin, P.: 1986, Time representation: A taxonomy of interval relations, *Proceedings AAAI-86, Philadelphia, PA*, pp. 360 – 366.

Ladkin, P.: 1987, Models of axioms for time intervals, *Proceedings AAAI*.

Lamport, L.: 1980, SOMETIME is sometimes NOT NEVER, *Communications of the ACM* .

Lamport, L.: 1983, What good is temporal logic?, *in* R. Manson (ed.), *Information Processing 83 IFIP*, North-Holland: Elsevier.

Lemmon, E. J.: 1977, *An Introduction to Modal Logic*, American Philosophical Quarterly Monograph Series 11, Basil Blackwell, Oxford.

L.E.Moser, P.M.Melliar-Smith, G.Kutty and Y.S.Ramakrishna: 1993, Completeness and Soundness of Axiomatizations for Temporal Logics without Next, *Technical report*, University of California, Santa Barbera.

Lewis, C.: 1912a, Implication and the algebra of logic, *Mind* **21**.

Lewis, C.: 1912b, A new algebra of strict implication, *Mind* **21**.

Lewis, C.: 1913, Interesting theorems in symbolic logic, *Journal of Philosophy* **10**.

Lewis, C.: 1914, The matrix algebra for implication, *Journal of Philosophy* **11**.

Li, R.-W.: 1987, A natural deduction system of temporal logic, *Technical Report AS-IS-87-2-1,XYZ Report No. 12*, nstitute of Software, Academia Sinica, Beijing, China.

Lundberg, B.: 1982, An axiomatization of events, *BIT* **3**.

Makinson, D.: 1966, Completeness theorem for modal logics, *Zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **12**.

McArthur, R.: 1976, *Tense Logic*, Synthese Library, Vol III, Reidel.

McDermott, D.: 1982, A temporal logic for reasoning about processes and plans, *Cognitive Science* **6**, 101 – 155.

Moore, R.: 1980, *Reasoning About Knowledge and Action*, PhD thesis, MIT, Cambridge.

Moszkowski, B. and Manna, Z.: 1983, Reasoning in Interval Temporal Logic, *Proceedings Workshop on Logics of Programs '83*, Vol. 164 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 371–383.

Nonnengart, A.: 1992, First-order modal logic theorem proving and standard PROLOG, *Technical Report MPI-I-92-228*, Max-Planck-Institute for Computer Science, Saarbrücken, Germany.

Nonnengart, A.: 1993, First-order modal logic theorem proving and functional simulation, *in* R. Bajcsy (ed.), *Proceedings of the 13th IJCAI*, Vol. 1, Morgan Kaufmann Publishers, pp. 80 – 85.

Nonnengart, A. and Szałas, A.: 1995, A fixpoint approach to second-order quantifier elimination with applications to correspondence theory, *Technical Report MPI-I-95-2-007*, Max-Planck-Institute for Computer Science, Saarbrücken, Germany.

Ohlbach, H. J.: 1988, A resolution calculus for modal logics, *in* E. Lusk and R. Overbeek (eds), *Proc. of* 9^{th} *International Conference on Automated Deduction, CADE-88 Argonne, IL*, Vol. 310 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, New York, pp. 500–516. extended version: SEKI Report SR-88-08, FB Informatik, Universität Kaiserslautern, 1988.

Ohlbach, H. J.: 1989, *A Resolution Calculus for Modal Logics*, PhD thesis, University of Kaiserslautern, Germany.

Ohlbach, H. J.: 1991, Semantics-based translation methods for modal logics, *Journal of Logic and Computation* **1**(5), 691–746.

Ohlbach, H. J. and Weidenbach, C.: 1995, A note on assumptions about skolem functions, *Forthcoming in the Journal of Automated Reasoning* .

Paech, B.: 1988, Gentzen-systems for propositional temporal logics, *Technical Report 88/01*, Ludwig-Maximilians-Universität München, Institut für Informatik, Theresientr.39, D-8000 München 2, Germany.

Petrovic, S.: 1988, The temporal logic of ai, *Automatika* **29**(5-6).

Prior, A.: 1957, *Time and Modality*, Clarendon Press, Oxford.

Prior, A.: 1967, *Past, Present and Future*, Clarendon Press, Oxford.

Prior, A.: 1968, *Papers on Time and Tense*, Clarendon Press, Oxford.

Quine, W.: 1960, *Word and Object*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts.

Reddy, P. V.: 1995, *Tableau Systems for Tense Logics – A Constraint Approach*, PhD thesis, Imperial College, London.

Rescher, N. and Urquhart, A.: 1971, *Temporal Logic*, Springer, Berlin.

Reynolds, M.: 1992a, Axiomatising first-order temporal logic: Until and since over linear time, *Technical report*, Imperial College, Department of Computing, London, UK.

Reynolds, M.: 1992b, An axiomatization for until and since over the reals without the irreflexivity rule, *Studia Logica* .

Reynolds, M.: 1992c, Axiomatizing until and since over integer time, *Technical report*, Imperial College, Department of Computing, London, UK.

Robinson, O. A.: 1979, A modal natural deduction system for s4, *ndjfl* **20**, 625–627.

Sadri, F.: 1987, Three recent approaches to temporal reasoning, *in* A. Galton (ed.), *Temporal Logics and Their Applications*, Academic Press.

Sahlqvist, H.: 1975, Completeness and correspondence in the first and second order semantics for modal logic, *in* S. Kanger (ed.), *Proceedings of the 3rd Scandinavian Logic Symposium*, North Holland, Amsterdam, pp. 110–143.

Sambin, G. and Vaccaro, V.: 1989, A new proof of Sahlqvist's theorem on modal definability and completeness, *Journal of Symbolic Logic* **54**, 992–999.

Scherl, R. B.: 1993, *A Constraint Logic Approach To Automated Modal Deduction*, PhD thesis, University of Illinois at Urbana – Champaign.

Scott, D.: 1965, The logic of tenses.

Segerberg, K.: 1971, An essay in classical modal logic, *Technical Report 13*, University of Uppsala, Filosofiska Studier. Volume 1-3.

Segerberg, K.: 1980, Applying modal logic, *Studia Logica* **39**.

Shetman, V. and Skvortsov, D.: 1990, Semantics of non-classical first-order predicate logics, *in* P. Petkov (ed.), *Mathematical Logic*, Plenum Press, New York.

Shoham, Y.: 1986, Reified temporal logics: Semantical and ontological considerations, *Proceedings of the 7th ECAI, Brighton, UK.*

Shoham, Y.: 1987, Temporal logics in AI: Semantical and ontological considerations, *Artificial Intelligence* **33**.

Shoham, Y.: 1988, *Reasoning About Change: Time and Causation from the Standpoint of AI*, PhD thesis, Yale University, Computer Science Department.

Strawson, P.: 1952, *Introduction to Logical Theory*, Barnes and Noble, New York.

Szałas, A.: 1987, A complete axiomatic characterization of first-order temporal logic of linear time, *Theoretical Computer Science* **54**, 199–214.

Thomason, S.: 1970, Some completeness results for modal predicate calculi, *in* K. Lambert (ed.), *Philosophical Problems in Logic*, Reidel.

Tichy, P.: 1985, Do we need interval semantics?, *Linguistics and Philosophy* **8**, 236–282.

Turner, R.: 1984, *Logics for Artificial Intelligence*, Ellis Horwood.

van Benthem, J.: 1977a, Modal logic as second order logic, *Technical Report 77-04*, Department of Mathematics, Universityof Amsterdam.

van Benthem, J.: 1977b, Tense logic and standard logic, *Logique et Analyse* **Themenheft: Tense Logic**(80), 47–83.

van Benthem, J.: 1977c, Tense logic and standard logic, *in* L. Åqvist and F. Guenthner (eds), *Tense Logic*, Nauwelaerts, Louvain.

van Benthem, J.: 1980, Points and periods, *in* Rohrer (ed.), *Time, Tense and Quantifiers*, Niemayer.

van Benthem, J.: 1984a, Correspondence theory, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic*, Reidel, Dordrecht, pp. 167–247.

van Benthem, J.: 1984b, Tense logic and time, *Notre Dame Journal of Formal Logic* **25**(1), 1–16.

van Benthem, J.: 1985, *Modal Logic and Classical Logic*, Atlantic Heights: Bibliopolis, The Humanities Press, Napoli.

van Benthem, J.: 1986, Tense and modality, *Essays in Logical Semantics*, Reidel, Dordrecht.

van Benthem, J.: 1988a, *A Manual of Intensional Logic*, Vol. 1 of *Lecture Notes*, Center for the Study of Languages and Information.

van Benthem, J.: 1988b, Time, logic and computation, *LNCS 354*, Springer.

van Benthem, J.: 1990, *The Logic of Time*, Reidel, Dordrecht.

Vardi, M.: 1988, A temporal fixed-point calculus, *Proceedings of the 13th ACM Symposium on Principles of Programming Languages*, pp. 250–259.

Venema, Y.: 1990, Expressiveness and completeness of an interval tense logic, *Notre Dame Journal of Formal Logic* **31**, 529–547.

von Wright, G.: 1968, *Time, Change and Contradiction*, A. S. Eddington Memorial Lecture No 22, Cambridge.

Wallen, L. A.: 1989, *Automated deduction in nonclassical logics*, MIT Press, Cambridge, Mass.

Winnie, J. A.: 1977, The causal theory of space-time, *in* J. E. et al (ed.), *Foundations of Space-Time Theories, Minnesota Studies in the Philosophy of Science*, University of Minnesota Press, pp. 134–205.

Wolper, P.: 1985, The tableau method for temporal logic: An overview, *Logique et Analyse* **110-111**.

Index

Symbols

A

I

L

M

N

O

P