

# Strukturierungsmittel zur Behandlung komplexer Beweisaufgaben

Dieter Hutter

Deutsches Forschungszentrum  
für Künstliche Intelligenz GmbH

16. November 2000

Saarbrücken

Source: Hutter



# Deduktionssysteme

- Ursprünglich:
  - Common Sense Reasoning als zentraler Bestandteil der künstlichen Intelligenz (Advice Taker, McCarthy59)
- Heute:
  - Zentraler Bestandteil der Formalen Methoden
  - Programmiersprachen Paradigmen
  - Mathematisches Assistenzsystem



# Automatisches Beweisen

- Kalküle: z.B. Resolution, Paramodulation, Superposition
- Vermeidung von Redundanzen im Kalkül (Basic Superposition, Restart Model Elimination) → verfeinerte Kalküle
- Effiziente Verfahren (Normalformen, Indexing)
- “Strategiewissen” ist hauptsächlich im Kalkül eingebaut



# Taktisches Theorembeweisen

- Kalküle: Sequenzkalküle, ND-Kalküle
- Ineffiziente (aber „verständliche“) Kalküle
- Kalkül als abstrakte Maschine
- Taktiken als Makros / Programme auf der abstrakten Maschine
- Wissen wird in Taktiken kodiert, Kalkül bleibt fest
- Benutzer als oberste Strategie



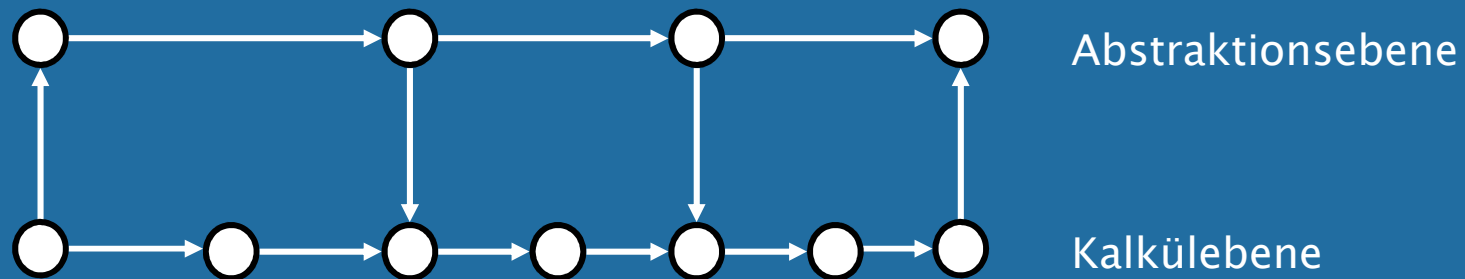
# Beweisplanen

- Planung des Beweises mit Hilfe von Taktiken
- Zusätzliche deklarative Beschreibung der Taktiken erforderlich
- Methoden = Taktiken + Vor- und Nachbedingungen
- Problem der schwachen Nachbedingungen oftmals nur Typinformationen:  $\Gamma \vdash \Delta$   
→ gemeinsame Plangenerierung und Planausführung
- mögliche Abhilfe: Abstraktionen



# Abstraktionen im Beweisen

- Abstraktionen
  - Formulierung von Vor- und Nachbedingungen
  - Planen von abstrakten Zwischenzielen



# PI-Abstraktionen

- PI-Abstraktionen:  
*„Die Abstraktion eines Beweises ist ein abstrakter Beweis“*
- Annahme: Termersetzungssysteme auf konkreter und abstrakter Ebene
- Korrespondenz von konkreten zu abstrakten Regeln
- → Abstraktionen und Substitution kommutieren
- → Abstraktionen und Subtermrelation kommutieren
- ⇒ nur Abstraktionen auf Signatur möglich



# Abstraktionen durch Zusatzinformationen

Rippling zur Steuerung von Induktionsbeweisen

Induktionshypothese:  $(x + y) + z = x + (y + z)$

Induktionsschluss:  $(s(x) + y) + z = s(x) + (y + z)$

Umformung:

$$(s(x) + y) + z = s(x) + (y + z)$$

$$s(x + y) + z = s(x) + (y + z)$$

$$s((x + y) + z) = s(x) + (y + z)$$

$$s((x + y) + z) = s(x + (y + z))$$





# Abstraktion durch Zusatzinformationen

- Abstraktion berücksichtigt Farben
- Wo befinden sich die roten Punkte?

$$((\cdot x + y) + z = (\cdot x + (y + z)$$

$$(\cdot (x + y) + z = (\cdot x + (y + z)$$

$$\cdot ((x + y) + z) = (\cdot x + (y + z)$$

$$\cdot ((x + y) + z) = \cdot (x + (y + z))$$

Abstrakte Regeln:  $(\cdot x + y) = \cdot (x + y)$



# Andere Informationsarten

- Zuordnung von Teilformeln:
  - Rippling, Differenz-Reduktion-Verfahren
  - Window-Techniken
  - Wiederverwendung von Beweisen
  - Origin-Tracking
- Sammeln von Informationen während der Beweissuche
  - Normalform von Teilformeln
  - Simplifikation



# Kodierung und Verwalten von Zusatzinformationen

- Abstraktionen werden definiert über Term + Zusatzinformation
  - Zusatzinformationen ändern sich während der Beweissuche
- ⇒ Integration von Termen und Zusatzinformationen
- ⇒ (mit Zusatzinformationen) annotierte Terme
- ⇒ Repräsentation der Zusatzinformation in der Logik
- ⇒ Verwaltung der Zusatzinformation in einem Kalkül



# Logik zur Wissensrepräsentation

- Wie repräsentiert man Zusatzinformationen in einer Logik?
- Labelled–Deductive Systems (Gabbay):
  - Eine Formel  $A$  besitzt Label  $t$   
Kalküle operieren auf Paaren  $t:A$
  - keine Labels für Substrukturen (z.B. Teilterme)
  - Sprache der Labels ist unbestimmt
  - keine generelle methodische Unterstützung



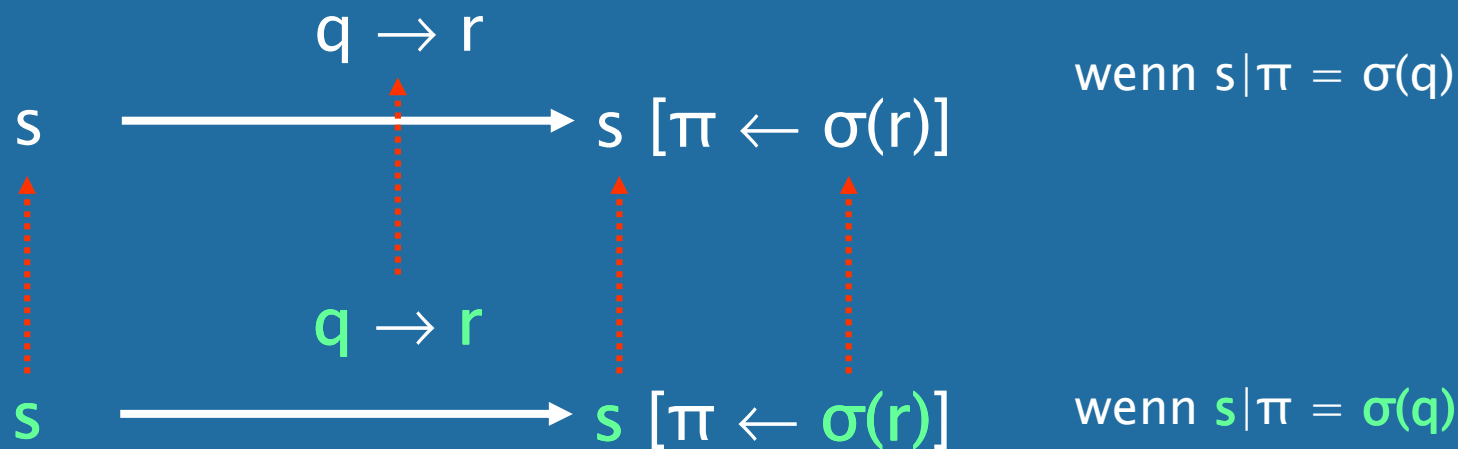
# Logik zur Wissensverwaltung

- Annotationen
  - Jedes Vorkommen eines Symbols kann Informationen tragen
  - Informationen sind prädikatenlogische Terme (mit eigener Signatur und Variablen)
  - Kodierung von Informationen in Annotationen einzelner Symbolvorkommen
  - uniforme Verwaltung durch eine entsprechend erweiterte Logik



# Vererbung von Informationen per Inferenz

- Idee: Informationen werden über einen erweiterten Inferenzmechanismus mitverwaltet



# Annotierte Substitution

- annotierte Substitutionen entsprechen Substitutionen im normalen Kalkül (Korrektheit):  
 $|\sigma(\mathbf{t})| = |\sigma|(|\mathbf{t}|) = \sigma(\mathbf{t})$
- annotierte Substitutionen instantiieren auch Annotationen (Zusatzinformationen)
- annotierte Substitution gliedert sich
  - in einen Teil für Termvariable
  - in einen Teil für Annotationenaber: Teile sind nicht unabhängig



# Informationsvererbung

- Annotation der Regel + Substitutionsbegriff bestimmen Art der Informationsvererbung
- $q$  und  $r$  beinhalten Informationspattern, die bei der Anwendung instantiiert werden

$$\begin{array}{c} q \rightarrow r \\ \swarrow \quad \searrow \\ P(\dots \sigma(q) \dots) \rightarrow P(\dots \sigma(r) \dots) \end{array}$$

$$\begin{array}{c} f(x^{h(\alpha)}) \rightarrow x^{k(\alpha)} \\ \swarrow \quad \searrow \\ g(f(a^{h(b)})) \rightarrow g(a^{k(b)}) \end{array}$$





# Kopplung von Annotations- und Termlogik

- Annotationslogik und Termlogik sind getrennt d.h. Annotationen und Terme können weitgehend unabhängig voneinander instantiiert werden
- Kopplung erfolgt durch Annotationen an Variablen (uniforme Vererbung)

$$\begin{array}{ccc} f(x^{h(\alpha)}) \rightarrow x^{k(\alpha)} & & \\ \swarrow \quad \downarrow \quad \searrow & \text{---} & \swarrow \quad \searrow \\ f(g^{h(c)}(a^{h(b)})) \rightarrow g^{k(c)}(a^{k(b)}) & & \end{array}$$



# Annotierte Logiken

## Formalisten:

- Prädikatenlogik + Farben  
(Hutter, Journal of Automated Reasoning 1997)
- Lambda-Kalkül + Farben  
(Hutter + Kohlhase, Journal of Automated Reasoning 2000)
- Prädikatenlogik (Lambda-Kalkül) + Termsprache  
(Hutter, Annals of Mathematics and Artificial Intelligence 2000)



# Anwendungen

- Deduktion:
  - Rippling, Differenz-Reduktion-Verfahren
  - Window-Techniken
  - Wiederverwendung von Beweisen
  - Simulation „effizienterer“ Kalküle
- Linguistik:
  - Verwaltung von Foci



# Zusammenfassung und Ausblick

- Generischer Mechanismus zur Integration von Zusatzwissen in der Deduktion (ohne individuelle Modifikation des Kalküls für einzelne Anwendungen)
- Definition der Informationsvererbung erfolgt durch Annotation der Regeln
  - unkritisch für die Korrektheit des Gesamtsystems
  - hohe Flexibilität ohne Änderung des Inferenzmechanismus
  - Logical Frameworks für Taktiken

