Natural Language Generation & Text Planning

Issues in natural language generation – architectures

Advanced search issues in text planning

TASKS IN GENERATION

Content determination

Choosing and accommodating information

Document structuring

Ordering and rhetorical relations between pieces of content

Lexicalisation

Choice of words for pieces of content

Generating referring expressions

Descriptions of objects

Aggregation

Sentence constructions, compositions

Linguistic and structural realisation

Mapping specifications onto pieces of text

SYSTEM ARCHITECTURE

Decomposition

- ***What* is said** **Course-graind planning (text planning)**
- ***When* it is said** **Fine-grained planning (sentence planning)**
- ***How* it is said** **Realization (syntactic generation)**

Interfaces of central importance

Precise decomposition into subprocesses unclear

Architectural models

- **Integrated – uniform, inefficient (historic)**
- **Sequential – practical, simplifying (currently the *standard* architecture)**
  However, no standards about the order of sentence planning tasks
- **Revision-based – theoretically best, hard to handle**
  Dedicated approaches according to demands of the genre

TEXT PLANNING

Subtasks

- **Building "messages"**
  - **from information sources**
- **Choosing those "messages"**
  - **which contribute to fulfilling the communicative intention**
- **Structuring the document**
  - **to obtain coherent and fluent text**

Organisation of the process

- **Subtasks intervowen in various ways**
- **Application-dependent**
  - **Knowledge sources, domain conventions, text genre**

REVISION-BASED PLANNING (Robin 1997)

Motivation – corpus observations from sports reports

Concise linguistic forms

Complex sentences (50 words, parse tree depth of 10)

Optional and background facts opportunistically slipped as modifiers

High paraphrasing power

Measurements

Increasing the number of content planning and linguistic realization options

Modeling the mutual constraints relating these options

Applying a dedicated two-pass procedure that

generates a sentence in a revision-based fashion

THE INFORMATION TO BE CONVEYED

All facts expressed in simple sentences

Charles Barkley scored 42 points. Those 42 points equal his best scoring performance of the season. Danny Ainge is a teammate of Barkley. They play for the Phoenix Suns. Ainge is a reserve player. Yet he scored 21 points. The high scoring performance of Barkley and Ainge helped the Suns defeat the Dallas Mavericks. The Mavericks played on their homecourt in Texas. They had already lost their 12 previous games there. No other team in the league has lost so many games in a row at home. The final score was 123-97. The game was played Sunday.

Assessment

- **Sounds odd and cumbersome, although expressed in a coherent discourse**
- **Relatively simple to generate, with limited lexical material**
- **Much lower quality than corpus texts**

GENERATION TECHNIQUES

Two-pass planning process

1. Simple draft sentences with obligatory information

2. Opportunistically adding new facts by applying revision rules

Revision rules

Complement an already included fact

Justify the relevance of a fact by providing its historical background

Some of these rules are non-monotonic ! (reword also the original fact)

Process control

Popping additional facts from a priority list to integrate them in the sentence

Stopping the process when empirically observed complexity limits are reached

INCREMENTAL REVISION (EXAMPLE) (I)

Initial draft

"Dallas, TX – Charles Barkley *scored* 42 points Sunday as the Phoenix Suns defeated the Dallas mavericks 133-97."

Adjunction of Created into Instrument

"Dallas, TX – Charles Barkley *tied a season high with* 42 points Sunday as the Phoenix Suns defeated the Dallas mavericks 133-97."

Coordination Conjoin of Clause

"Dallas, TX – Charles Barkley tied a season high with 42 points and *Danny Ainge added* 21 Sunday as the Phoenix Suns defeated the Dallas mavericks 133-97."

INCREMENTAL REVISION (EXAMPLE) (II)

Absorb of Clause in Clause as Result with Agent Control

"Dallas, TX – Charles Barkley tied a season high with 42 points and *Danny Ainge came off the bench to add* 21 Sunday as the Phoenix Suns defeated the Dallas Mavericks 133-97."

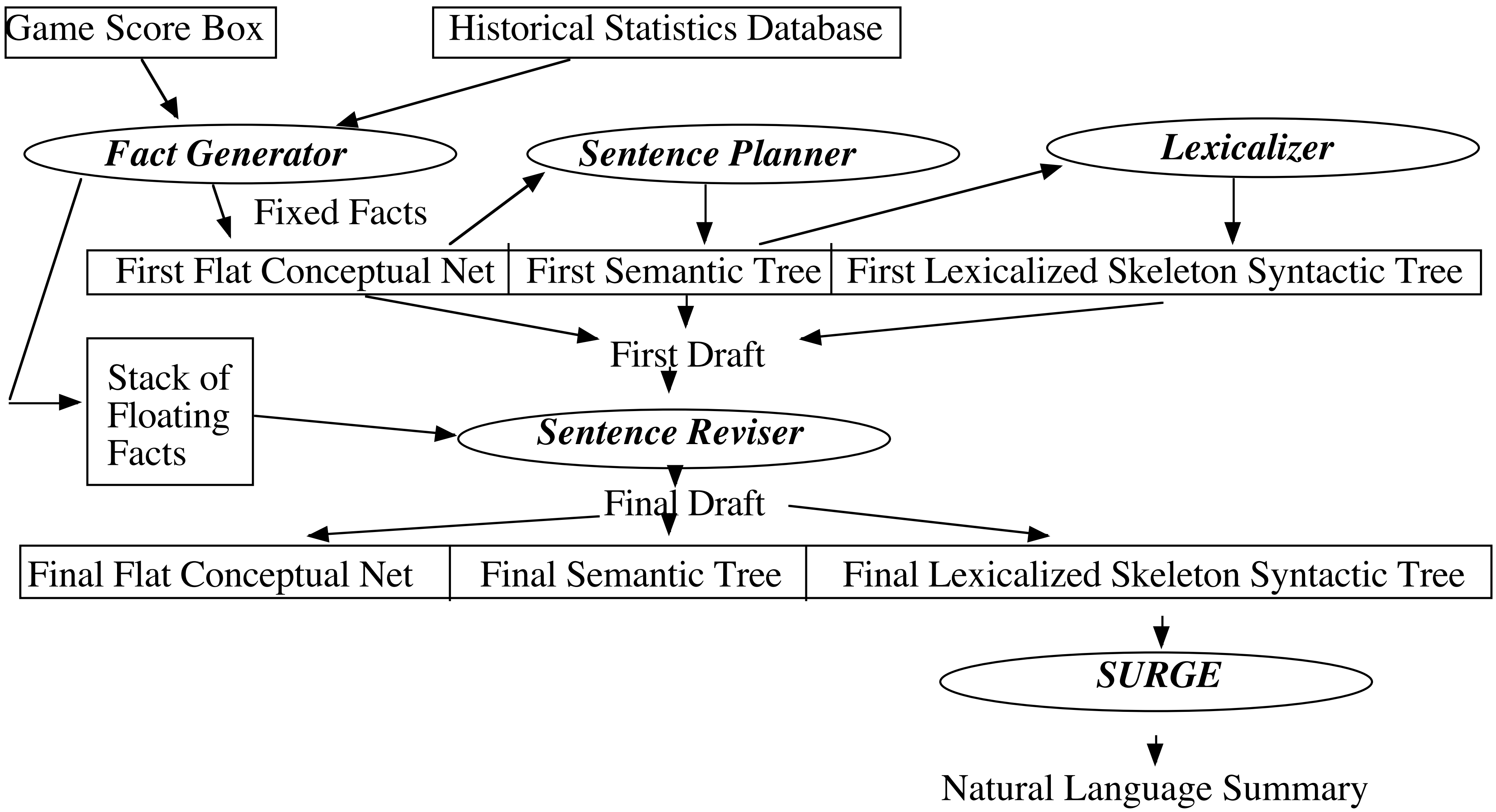
Nominalization with Ordinal Adjoin

"Dallas, TX – Charles Barkley tied a season high with 42 points and Danny Ainge came off the bench to add 21 Sunday as the Phoenix Suns handed the Dallas Mavericks *their 13th straight home defeat* 133-97."

Adjoin of Classifier to NP

"Dallas, TX – Charles Barkley tied a season high with 42 points and Danny Ainge came off the bench to add 21 Sunday as the Phoenix Suns handed the Dallas Mavericks their *league worst* 13th straight home defeat 133-97."

STREAK'S ARCHITECTURE



TEXT PLANNING AS OPTIMIZATION

Motivation

Some applications produce structured information instead of list of facts

(inference-rich discourse, argumentation, proof presentation)

Input typically interpretable as rhetorically inadequate text plan

Methods applied

Rewrite rules for compactifying subtrees (modus ponens -> modus brevis)

Rewrite rules for recasting subtrees (sort of aggregation related)

Problems

Complexity of tree portions covered

Dependency of context (suitability for other rhetorical concerns)

Control of processing (interdependencies with other operations)

DEGREES OF EXPLICITNESS IN ARGUMENTATION

A example

Some extra copies of the Spring 1984 issue of AI Magazine

are available in the library.

This issue includes a "Research in Progress" report on AI research at ISI.

[Matthiessen, Thompson 1987]

The problem

- **Formally interpreted as an ELABORATION, but intended as a MOTIVATION**
- **No involvement of addressee expressed to justify a MOTIVATION interpretation**

UNFOLDING ARGUMENTS IN INCREASING DETAIL (1)

The implicit variant

MOTIVATION

N

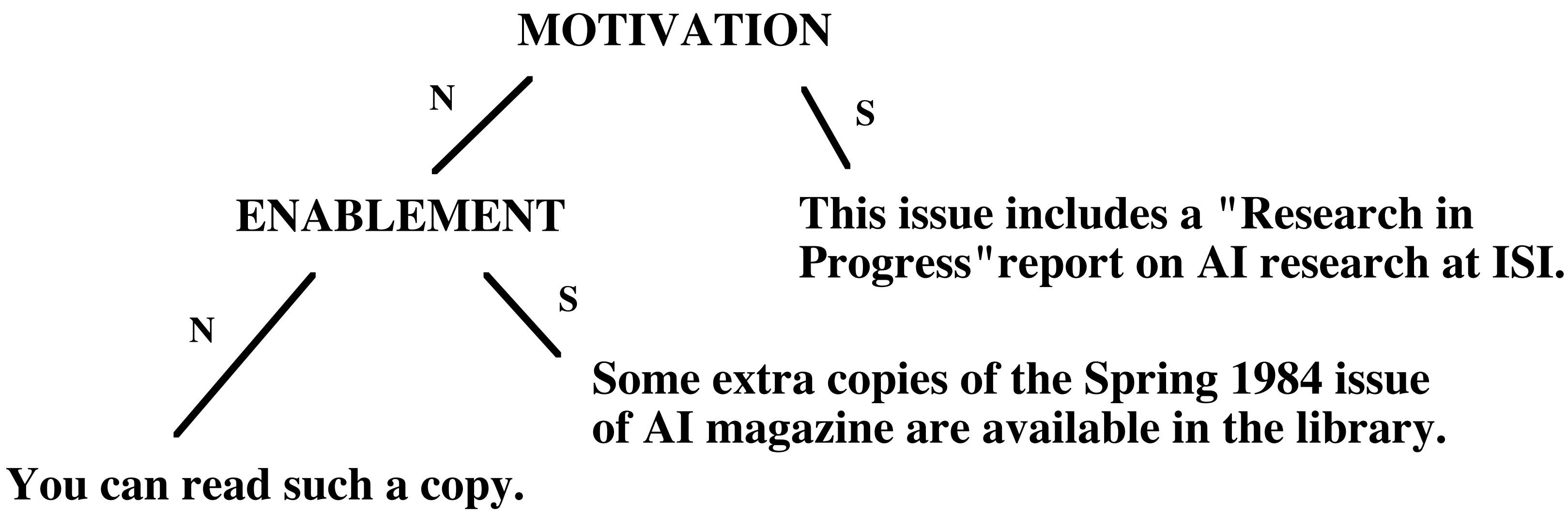
S

This issue includes a "Research in Progress"report on AI research at ISI.

Some extra copies of the Spring 1984 issue of AI magazine are available in the library.

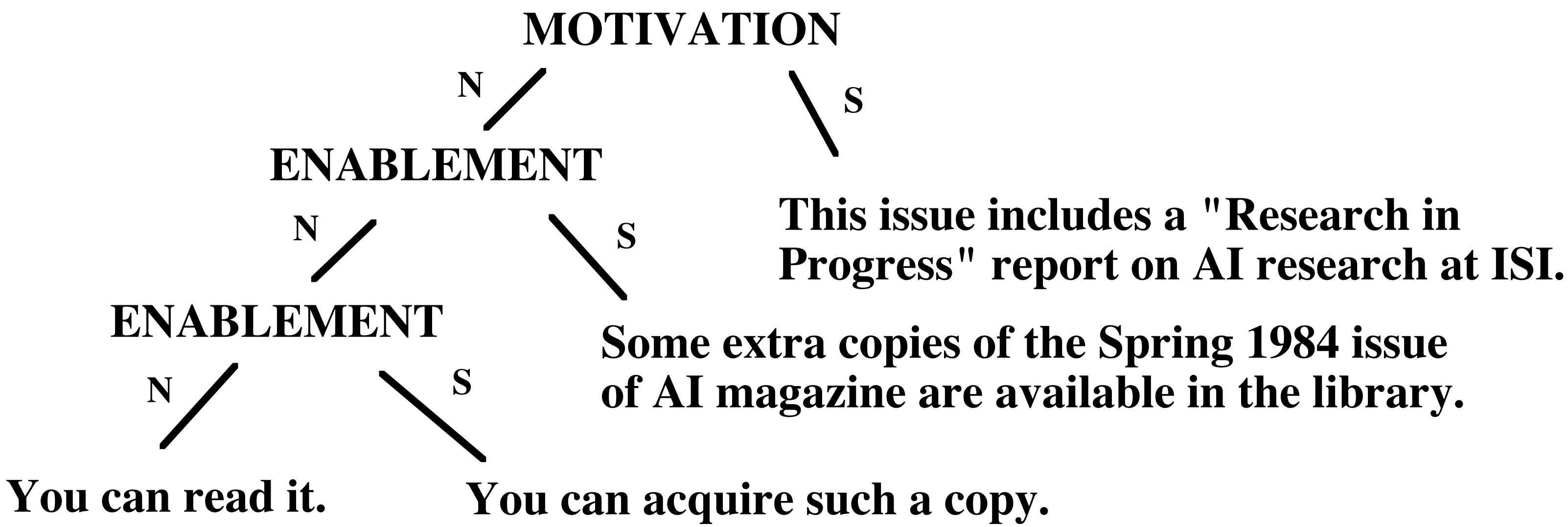
UNFOLDING ARGUMENTS IN INCREASING DETAIL (2)

A more explicit variant



UNFOLDING ARGUMENTS IN INCREASING DETAIL (3)

A fully explicit variant



DEGREES OF EXPLICITNESS IN ARGUMENTATION

A example

Some extra copies of the Spring 1984 issue of AI Magazine

are available in the library.

This issue includes a "Research in Progress" report on AI research at ISI.

[Matthiessen, Thompson 1987]

Diagnosis

- **Some facts and connecting relations not explicitly expressed**
- **Missing content pragmatically implied according to conversational maximes**

Challenges

- **Reconstructing the fully explicit argumentative structure (in analysis)**
- **Presenting arguments in a way avoiding redundancy (in generation)**

EMPIRICAL MOTIVATIONS

Some sorts of logical consequences preferably conveyed implicitly

through discourse context and default expectations (e.g., 'direct' instantiations)

[Thüring, Wender 1985]

Modus ponens communicated as *Modus brevis* [Sadock 1977], [Cohen 1987]

Some kinds of "easy" inferable consequences

- **Taxonomic inferences (category memberships)**
- **Normal consequences of actions**
- **Contextually suitable instantiation of rules/regularities mentioned**
- **Responsible causes if sufficiently salient**

GENERATING ARGUMENTS FROM EXPLICIT FORMS

Techniques

Revision of a fully explicit, but rhetorically inadequate representation

Incrementally omitting contextually inferable information

[Horacek 1998]

Revision operations

Omission of a *proposition* (premise) or a *rule* appearing as a justification

(building variants of a *modus brevis* form)

Omission of an intermediate inference step

Knowledge and inferential skills of the addressee

- ***Awareness* of regularities and facts justifying an inference**

  (but also judging *memory limitations*)

- ***Coherence* in chaining elementary clues**

  (but assessing the complexity of *substitutions*)

CONTENT DETERMINATION

1. **Addressing *granularity* – Building *expansions***

   Expand assertion level steps into sequences of cognitively hard deductive syllogisms

2. **Addressing *explicitness* – Introducing *omissions* and *short-cuts***

   Modeled by user-adaptive, communicatively motivated presentation rules

   - ***Cut-prop*-rule**
     Omission of a *proposition* appearing as a justification – trivial facts, e.g. $0 < 1$

   - ***Cut-rule*-rule**
     Omission of a *rule* (axiom instance) appearing as a justification

   - ***Compactification*-rule**
     Short-cut by omitting an *intermediate* inference step
     $0 \neq a$ because of [$0 < a$ because of] $1 < a$ [and $0 < 1$]

PRESENTING A MACHINE-FOUND PROOF – EXAMPLE

***Theorem*: (theorem 1.1 [Lüneburg 1981]):**

Let K be an ordered field.
If $a \in K$, then $1 < a$ implies $0 < a^{-1} < 1$, and vice-versa.

***Lemma*: (lemma 1.10 [Lüneburg 1981]):**

Let K be an ordered field.
If $a \neq 0 \in K$, then $0 < a$ implies $0 < a^{-1}$, and vice-versa.

***Proof*: (according to [Lüneburg 1981])**

Let $1 < a$. According to Lemma 1.10 we then have $a^{-1} > 0$.
Therefore $a^{-1} = 1a^{-1} < aa^{-1} = 1$.

VARIANTS PRESENTING A MACHINE-FOUND PROOF

A fully explicit verbalization (boring and redundant)

(1) Let $1 < a$.

(2) Then $0 < a$, because '<' is transitive and $0 < 1$.

(3) $0 \neq a$ follows from the trichotomy of '<'.

(4) Lemma 1.10 implies $0 < a^{-1}$.

(5) Since '<' is monotone and $1 < a$, $1a^{-1} < aa^{-1}$.

(6) $a^{-1} < aa^{-1}$ because of the definition of the unit element of K.

(7) $aa^{-1} = 1$ because of the definition of the inverse element of K for $a \neq 0$.

(8) Hence $a^{-1} < 1$.

A concise, rhetorically adequate verbalization (without justifications for inequations)

(1) Let $1 < a$.

(4) Then Lemma 1.10 implies $0 < a^{-1}$.

(5-7) Therefore $a^{-1} = 1a^{-1} < aa^{-1} = 1$ holds.

(with justifications for inequations)

(1)	**Let $1 < a$.**	
(4)	**Then Lemma 1.10 implies $0 < a^{-1}$.**	
(5)	**and $a^{-1} = 1a^{-1}$**	**because of the unit element of K**
(6)	**$< aa^{-1}$**	**since $1 < a$, $0 < a^{-1}$, and the monotony hold**
(7)	**$= 1$**	**because of the inverse element of K for $a \neq 0$.**

EXAMPLE – REORGANIZING CASE DISTINCTIONS

Text corresponding to the original (application-produced) structure

To prove $|ab| = |a||b|$, let us consider the cases where $a = 0$, $a > 0$, $a < 0$, resp.

Case 1: $a = 0$. Then $|ab| = |0b| = 0 = 0|b| = |a||b|$.

Case 2: $a > 0$. Let us consider the cases where $b = 0$, $b > 0$, $b < 0$, resp.
- **Case 2.1: $b = 0$. Then $|ab| = |a0| = 0 = |a|0 = |a||b|$.**
- **Case 2.2: $b > 0$. Then $|ab| = ab = |a||b|$.**
- **Case 2.3: $b < 0$. Then $|ab| = -ab = a(-b) = |a||b|$.**

Case 3: $a < 0$. Let us consider the cases where $b = 0$, $b > 0$, $b < 0$, resp.
- **Case 3.1: $b = 0$. Then $|ab| = |a0| = 0 = |a|0 = |a||b|$.**
- **Case 3.2: $b > 0$. Then $|ab| = -ab = (-a)b = |a||b|$.**
- **Case 3.3: $b < 0$. Then $|ab| = ab = (-a)(-b) = |a||b|$.**

Text corresponding to rhetorically improved structure

If either a or b is 0, then both $|ab|$ and $|a||b|$ are equal to 0.

If $a > 0$ and $b > 0$, then $ab > 0$ so that $|ab| = ab = |a||b|$.

If $a > 0$ and $b < 0$, then $ab < 0$ so that $|ab| = -ab = a(-b) = |a||b|$.

The other two cases are treated similarly.

PROPERTIES OF BOOK PROOFS

Complexities and use of text forms

Case distinctions mostly expressed implicitly by a conditional clause

(followed by a critical case: “it remains to show that … ”)

Case distinction expressed explicitly for

- **2 or more complex cases**
- **Untypically large expressions**

Number of cases rarely more than 3

METHODS FOR A TRANSFORMATION (I)

Goals

Avoid nested case analyses whenever possible

Reduce the number of cases

Produce structures that enable the use of implicit textual forms

Crucial parameters

The length of case analysis branches

The number of cases

The complexity of the case expressions

Operations

Case shrinking (pulling out statements independent of case distinction)

Case aggregation (putting several cases together)

Case unnesting (lifting embedded case distinctions)

EXAMPLE – DEPTH REDUCTION

Operation **(inverse to the two other reduction operations)**

Lifting an embedded case analysis

Moving down copies of the inferences of the embedding case

Merging case assumption of the embedding case with each embedded one

Application conditions

Expression types in case assumptions compatible

Number of cases and case expression remain within limits

Length increase tolerable or compensated by subsequent number reductions

INTERNAL DEPENDENCIES

Dependencies within each operation

Testing multiple applications of lifting inference subsequences

Determining subtree size of case branches

Dependencies across operations – Traversing proof graph starting from leaf nodes

1. **Combining case branches if possible**
2. **Lifting subsequences of inferences**
3. **Linearization of embedded case analyses** — **possible results**
4. **Lifting subsequences of inferences**

METHODS FOR A TRANSFORMATION (II)

Control structure

Starting bottom-up from embedded case analyses, continuing recursively

Applying case shrinking and aggregation operations, if applicable

Applying case unnesting operations, if “profitable”

Application in the rules in the example

1. Unnesting case 2

2. Unnesting (original) case 3)

3. Aggregating (original) cases 1, 2.1, 3.1

Control extensions to address dependencies

With widely independent operations (inferability) – strict order

Within rules – apply unnesting temporarily, try effect of other rules

With tightly dependent operations (verbalization) – check expected result