

# Dynamic Typing and Non-monotonic Reasoning

## Principles for a Semantic Interpreter

Hans Rudolf Straub – Semfinder AG

# Content List

1. Background and History
2. Basic principles - Interpretation paradigm
3. Concept architecture and semantic space
4. Transformation process

# Background and History

Semfinder      Program for automatic ICD-10 / OPS coding of medical diagnoses and procedures in daily use in more than 300 hospitals

1989            First Program for automatic ICD-coding (Clinic St. Leonhard, St. Gallen)

1996            Starting point: concept molecules

1999            Foundation of the Semfinder company

2009            300      hospitals in Germany  
                  25        hospitals in Switzerland

# Basic Principles

## → Interpretation Paradigm

Not a copy

→ Models are an interpretation of reality

Information reduction

→ Models contain less information than reality

No model is complete

→ Different models are possible

# Basic Principles

# → Interpretation Paradigm

Not a copy

→ Models are an interpretation of reality

Information reduction

→ Models contain less information than reality

No model is complete

→ Different models are possible

## Consequences

A) Look for the perfection of your model

→ static view

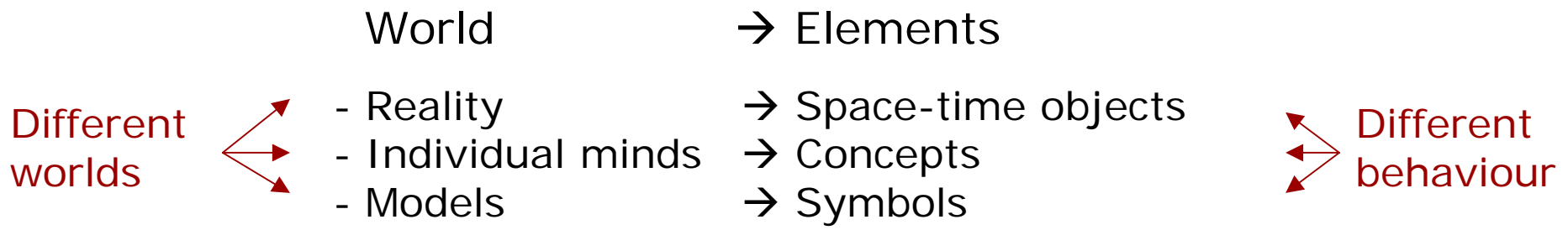
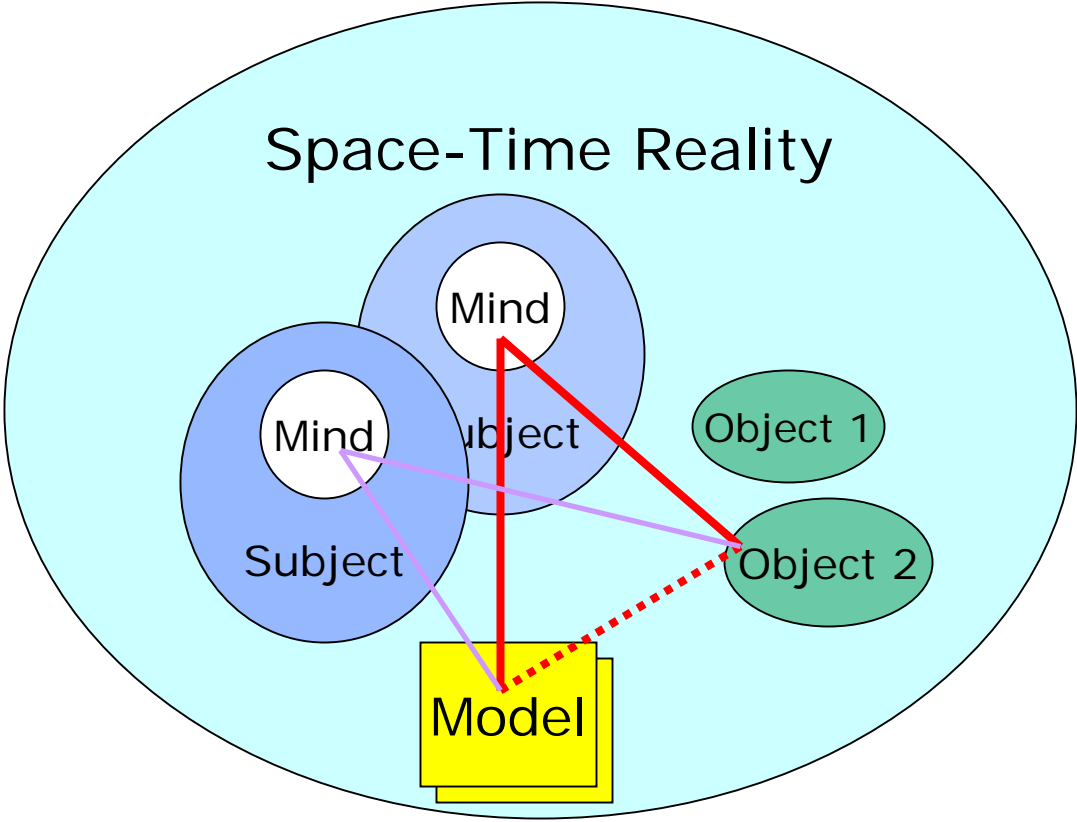
→ monotonic reasoning (DL, FOL)

B) Look for a way to transform models

→ dynamic view

→ non-monotonic reasoning

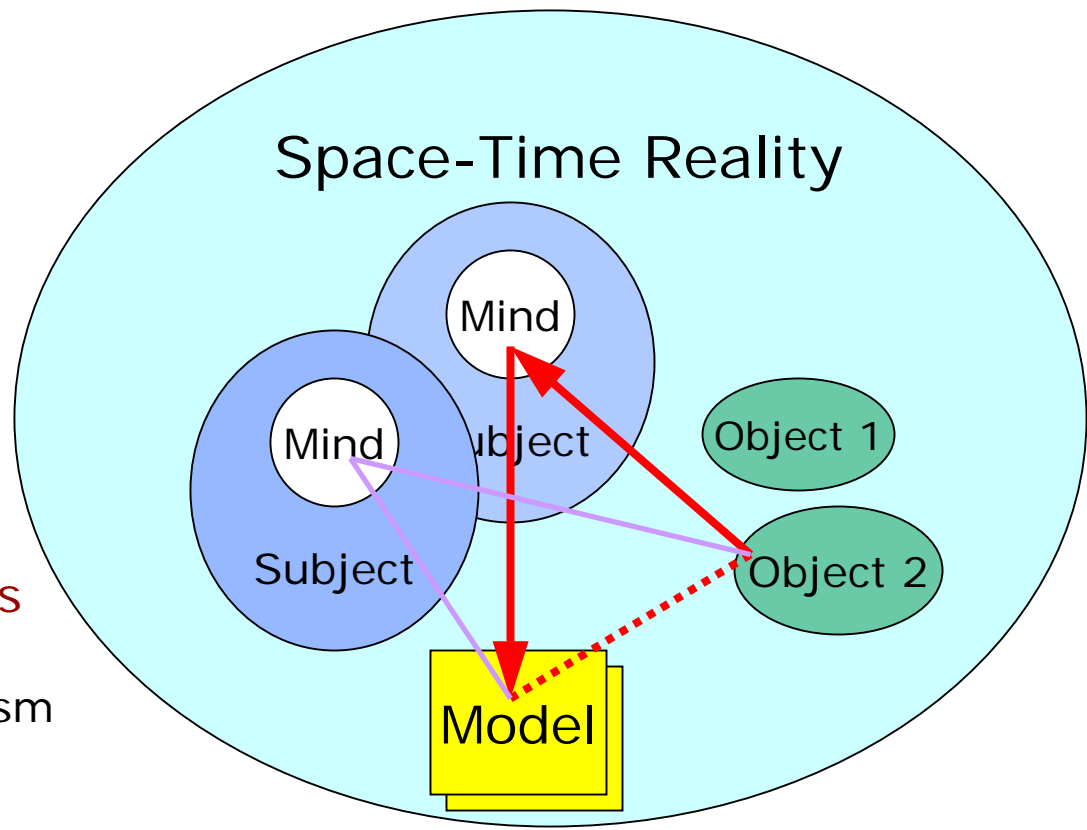
# Basic Principles → Semiotic Triangle



# Basic Principles → Semiotic Triangle

Mind follows reality  
= realism

Model follows mind  
= conceptualism



- World → Elements
- Reality → Space-time objects
- Individual minds → Concepts
- Models → Symbols

# Basic Principles → concept ≠ object

	Objects	Concepts
World	space/time reality	mental space (mind)
Visibility	visible	invisible
Boundaries	<b>disjunct</b> , concrete	<b>overlapping</b> , diffuse (fields)
Existence	<b>autonomous</b>	<b>context dependent</b>
Composition	<b>addition</b>	<b>complex mix</b>
Composed of	similar, but smaller objects	different (and bigger) concepts

# Basic Principles → concept ≠ object

Example (context dependency of concepts)

foot (as a body part)

lower limb (as a body part)

foot (as a location)

lower limb (as a location)

# Basic Principles → concept ≠ object

Example (context dependency of concepts)

foot	(as a body part)	is_part_of
lower limb	(as a body part)	

foot	(as a location)	is_a
lower limb	(as a location)	

# Basic Principles → concept ≠ object

Example (context dependency of concepts)

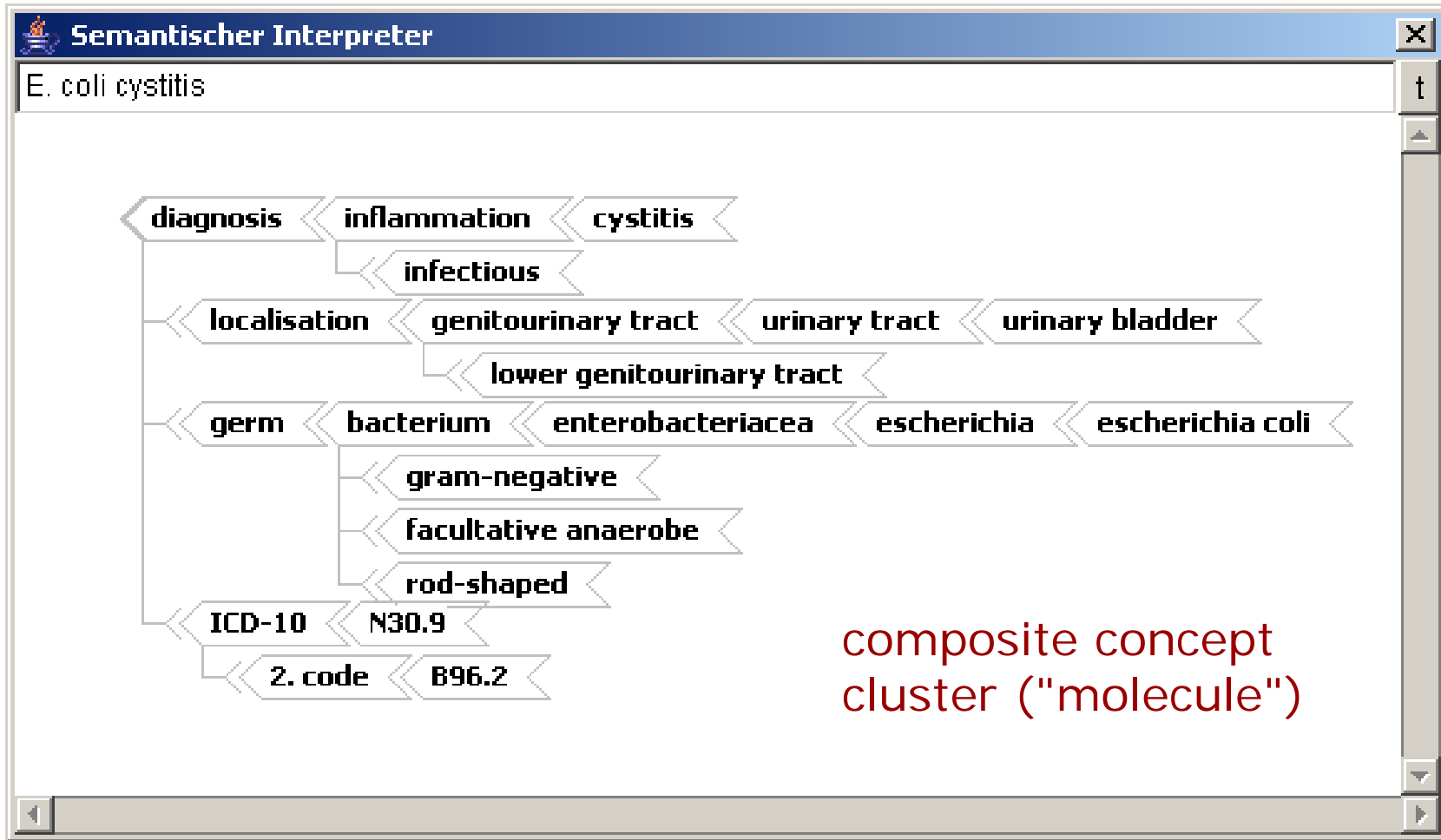
foot	(as a body part)	is_part_of
lower limb	(as a body part)	

foot	(as a location)	is_a
lower limb	(as a location)	

If concepts are not typed, false conclusions are inherent.  
The same concept (foot) can have different types.  
Different Relationships follow.

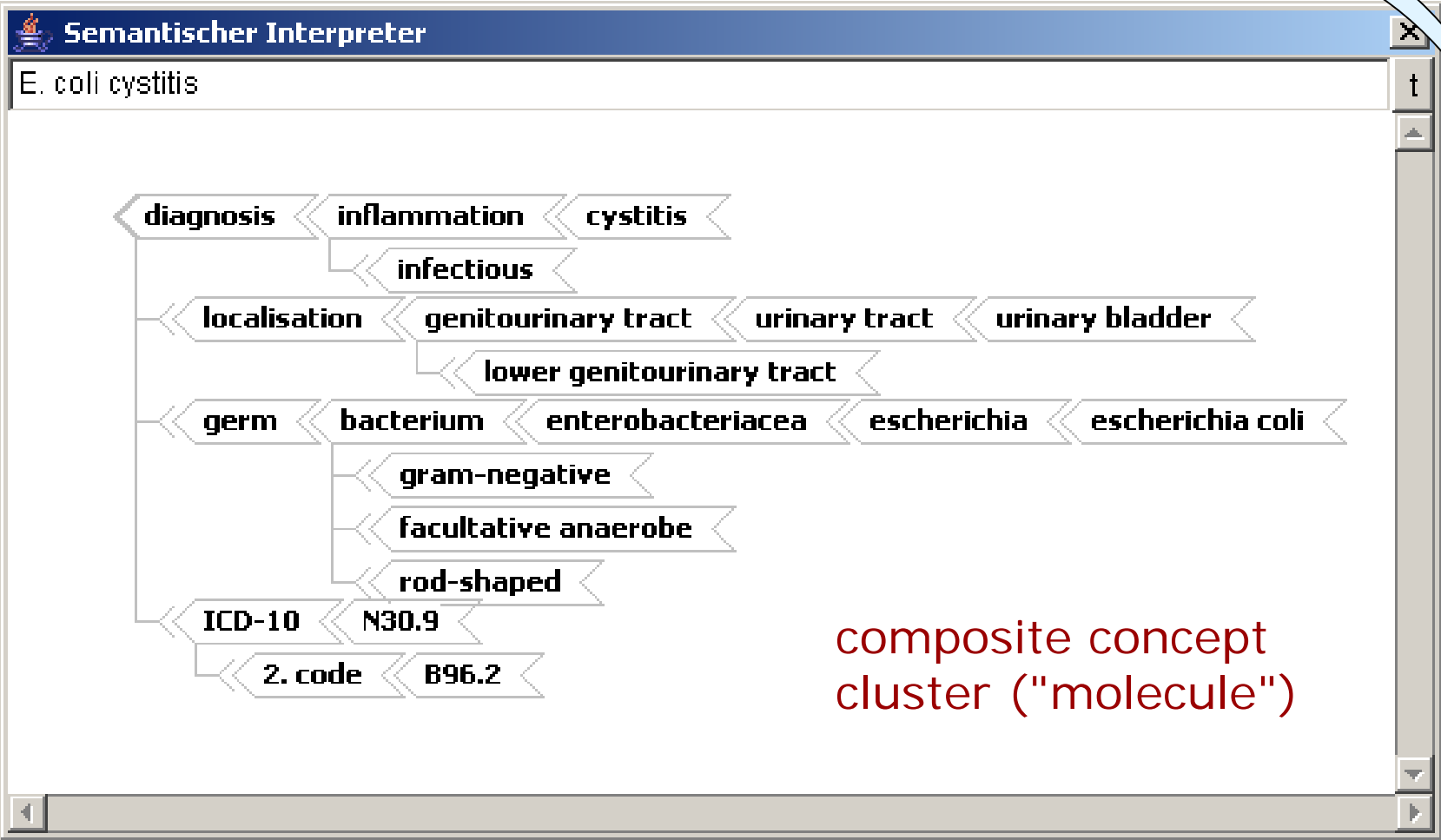
→ **allow dynamic typing!** ← use typed (composite) concepts  
(not untyped, unlinked objects)

# Basic Principles → concept ≠ object



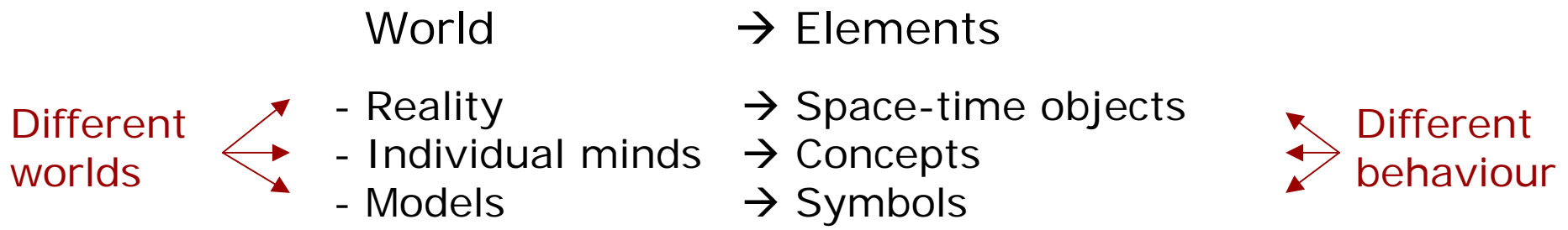
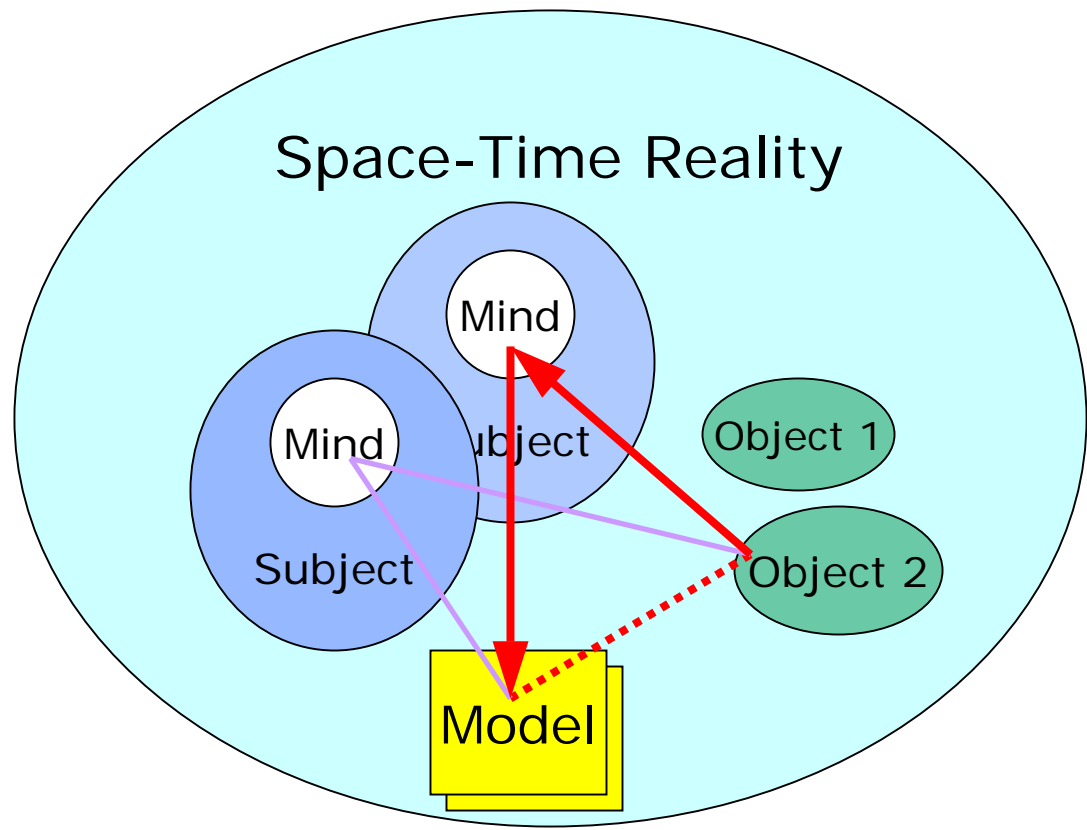
- use composition (postcoordination) whenever possible
- keep the atomic concepts inside the clusters
- give the cluster an adequate structure

# Basic Principles → concept ≠ object



- use composition (postcoordination) whenever possible!
- keep the atomic concepts inside the clusters!
- give the cluster an adequate structure!

# Basic Principles → Semiotic Triangle



# Content List

1. Background and History
2. Basic principles - Interpretation paradigm
3. Concept architecture and semantic space
4. Transformation process

# How to structure concepts? → 3 formal elements

injury

①

atomic concept

# How to structure concepts? → 3 formal elements



①

atomic concept

An atom is equally  
type and expression

→ Bifaciality

# How to structure concepts? → 3 formal elements



① atomic concept

An atom is equally type and expression

→ Bifaciality



← Hierarchic chain

# How to structure concepts? → 3 formal elements



① atomic concept

An atom is equally type and expression

→ Bifaciality



② hierarchic relationship

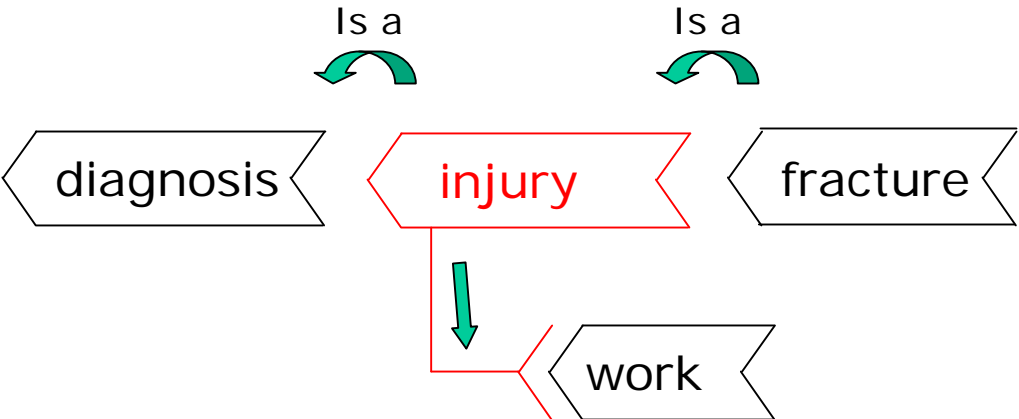
# How to structure concepts? → 3 formal elements



① atomic concept

An atom is equally type and expression

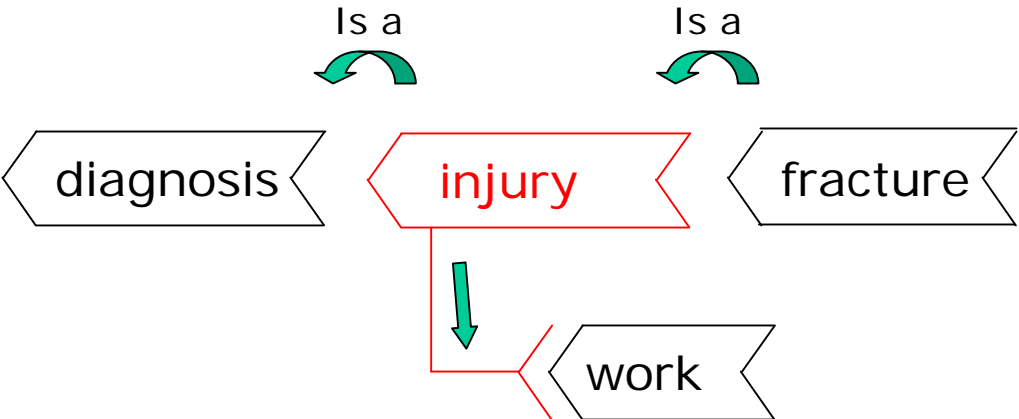
→ Bifaciality



② hierarchic relationship

③ attributive relationship

# How to structure concepts? → 3 formal elements

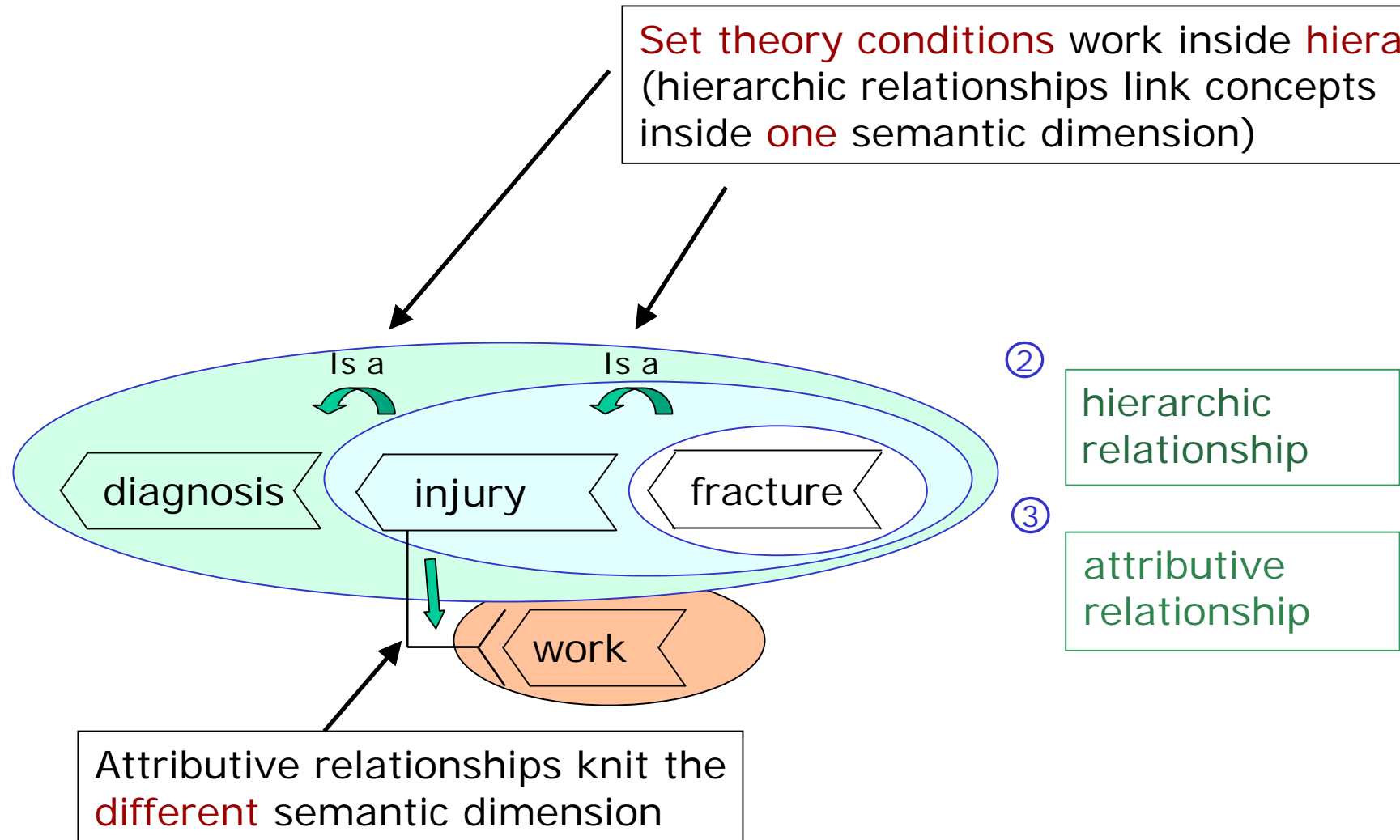


- ② hierarchic relationship
- ③ attributive relationship

The relationships are invisible on screen, but their positions indicate and distinguish them → Economy

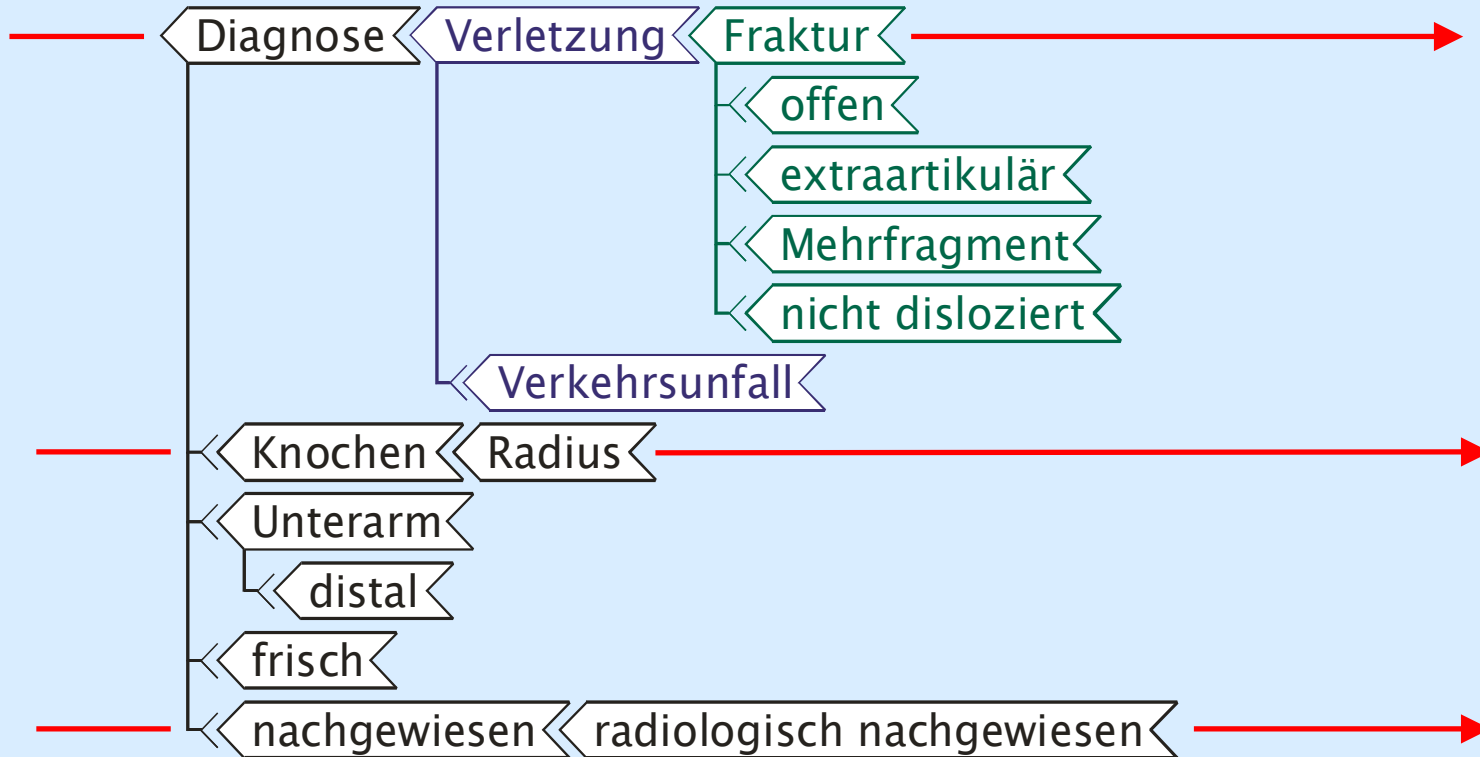
# The two relationships span the semantic space

Set theory conditions work inside hierarchies (hierarchical relationships link concepts inside **one** semantic dimension)



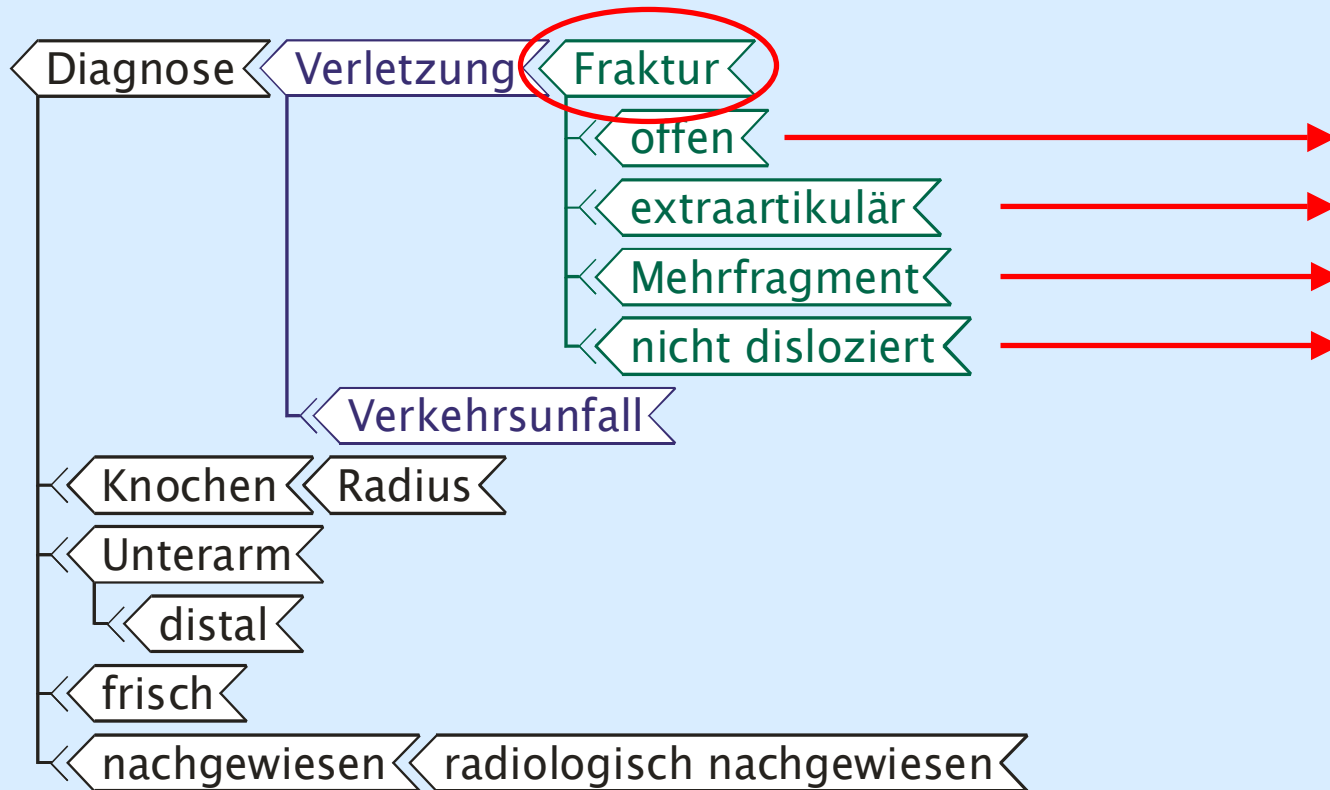
# The two relationships span the semantic space

## Compositionality with different dimensions



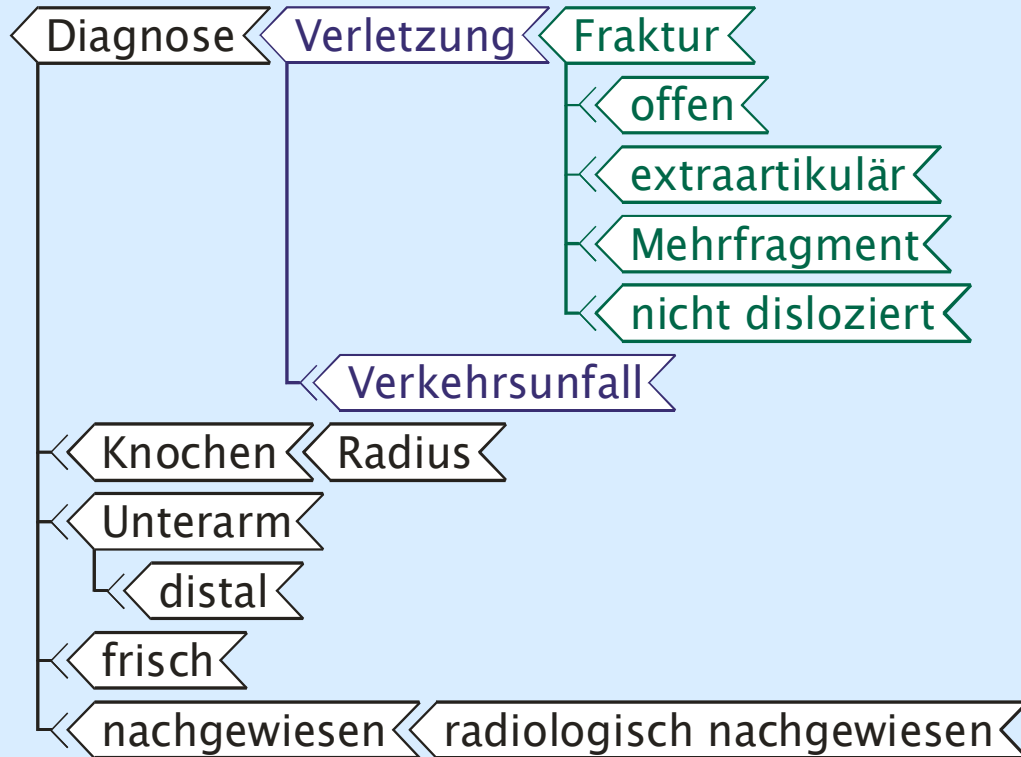
# The two relationships span the semantic space

## Focus with encapsulated dimensions



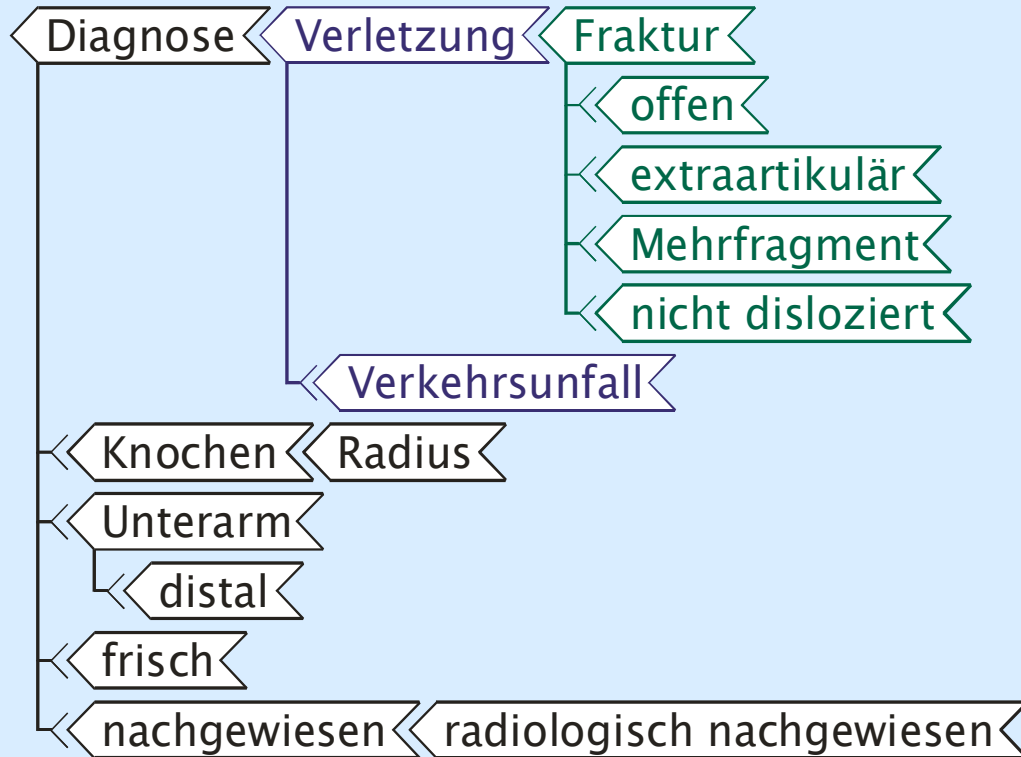
# The two relationships span the semantic space

## Pseudotree structure helps computability



# The two relationships span the semantic space

Only 1 formal element visible → readability



# Top Level Ontology

We distinguish between **formal** and **content** TLO:

## Formal TLO

→ 3+1 Elements

1 atomic concept + 2 relationships + 1 multiplication element

These 4 Elements are not to change,  
but can be filled (named) with any content

## Content TLO

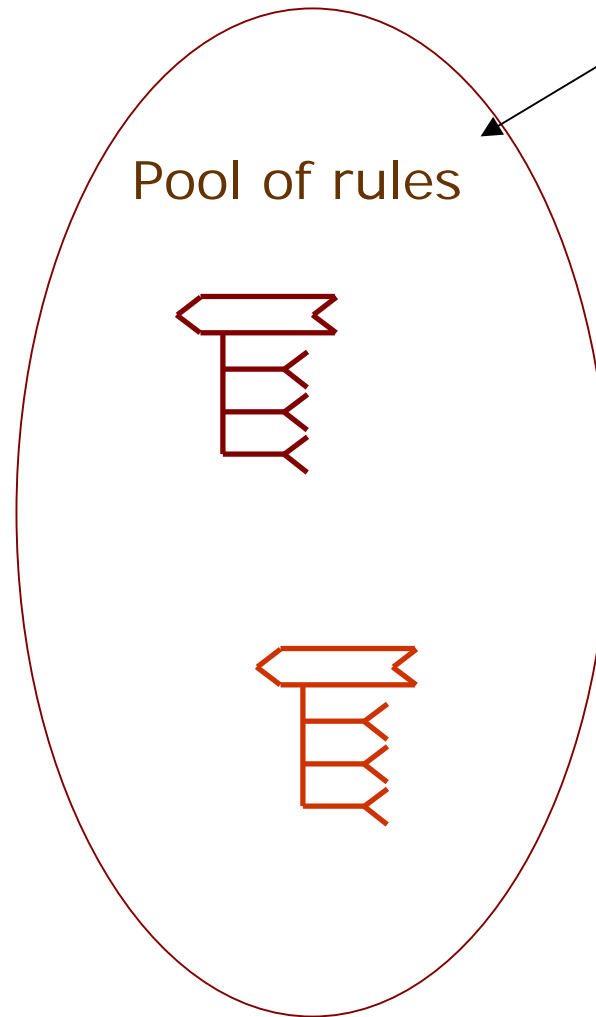
Content TLO is not proprietary in our system

In principle any content can be given to the four formal elements

Thus we can model other systems and transform them  
(semantic interoperability)

# Transformation process (inference process)

# Transformation process



## Intelligence

The rule pool represents the intelligence of the system (includes thus the human engineers brain)

## Form of rules

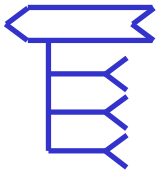
Rules have the same multidimensional, strict composite and pseudohierarchical form like all other concepts

## No definitions

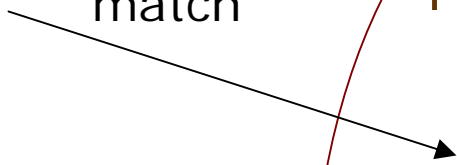
The non-monotonic system comprises only rules (which can be overruled) and no definitions

# Transformation process

Input



match



Pool of rules

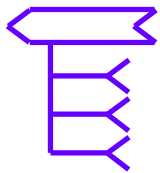
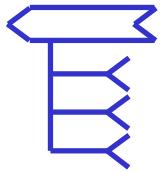


## Match

The composite and multidimensional form of both input and rules allow a **fast** match of the most **appropriate** rule

# Transformation process

Input



match



execution



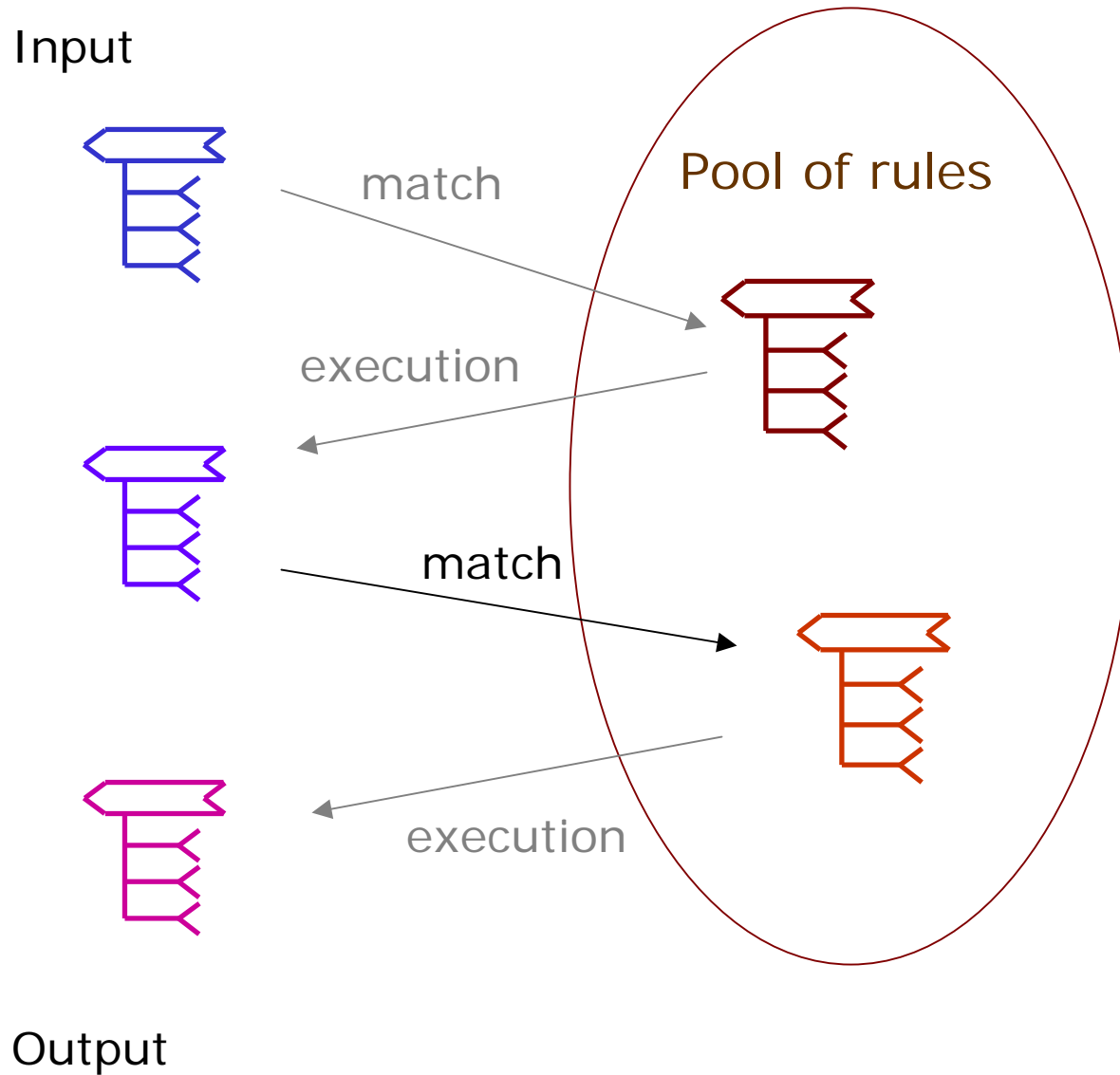
Pool of rules



## Execution

The rule contains  
**ifs** for the match  
and **thens** for the  
modification of the  
input

# Transformation process



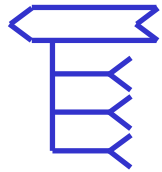
## Next rules

The modified input matches with a different rule which is then executed

Procedure continues until no more rules do match → output

# Transformation process

Input

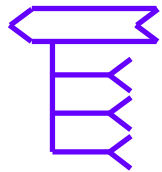


match

Pool of rules



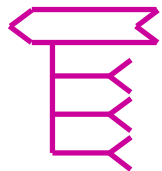
execution



match



execution



Output

## Semantic way of chronology control

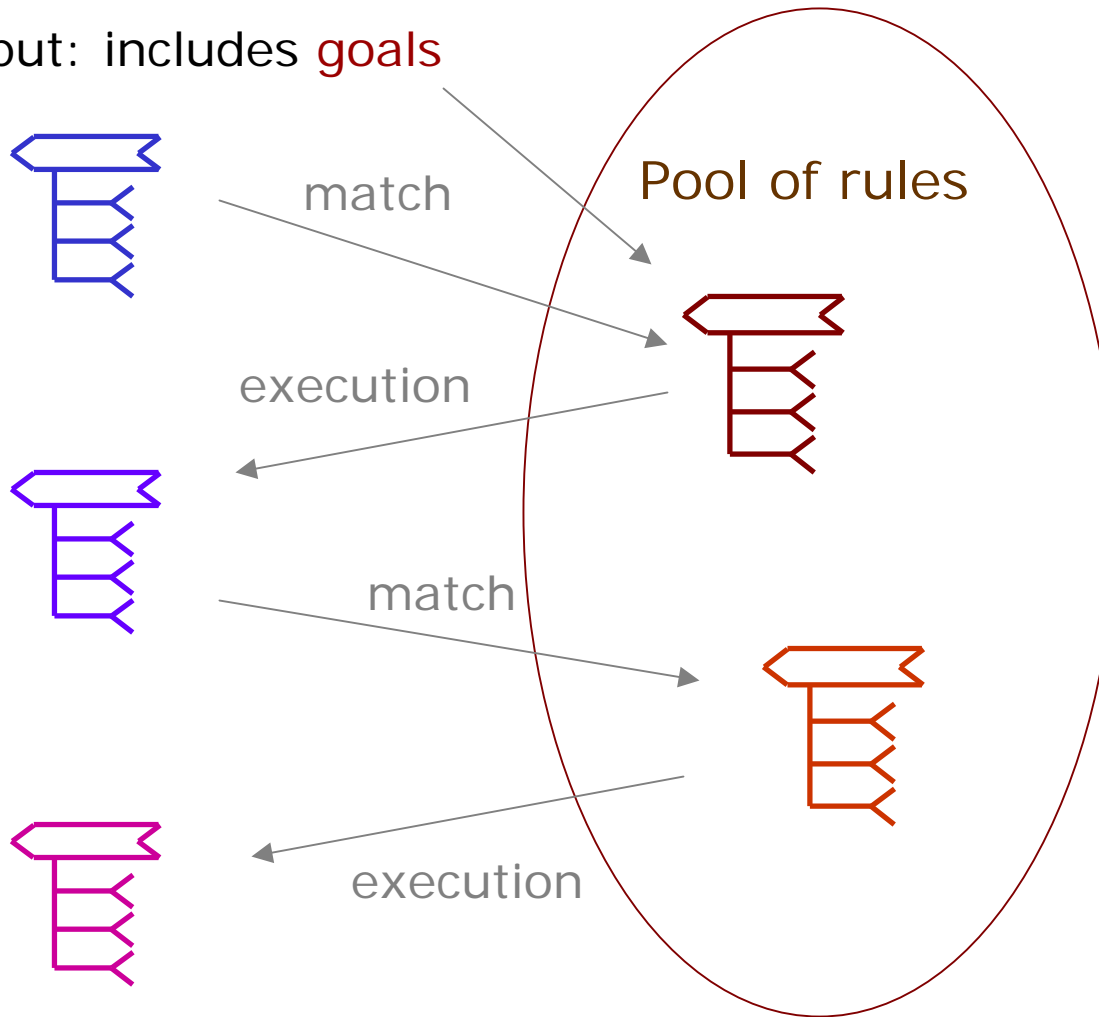
Which rule is executed?

Chronology is controlled by the information of the actual input and by dynamic triggers installed into the status during the execution of the process.

The control lies therefore in the concepts of input and rules (in the semantics) and not in algorithm steps of the program

# Transformation process

Input: includes **goals**



Output

## Semantic way of chronology control

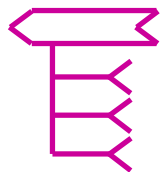
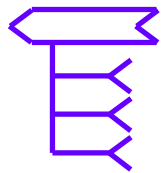
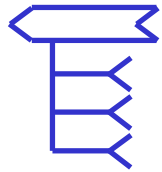
Which rule is executed?

The Input contains information about the **observations** to be processed (objective input) as well as information about the **goal** of the interpretation process (subjective input).

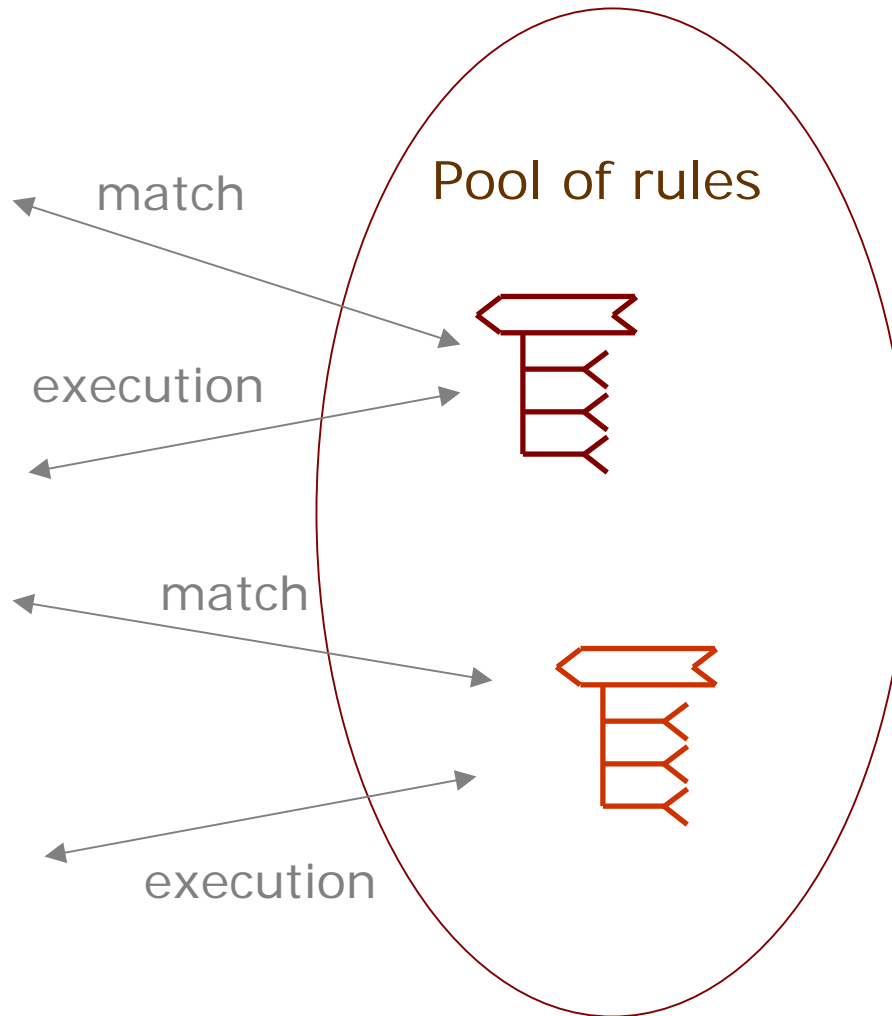
Both are given to match in the pool of rules and to select the rule to be executed

# Transformation process

Input



Output



## Semantic way of chronology control

Chronology of inference is controlled by the semantic concepts of input and rules.

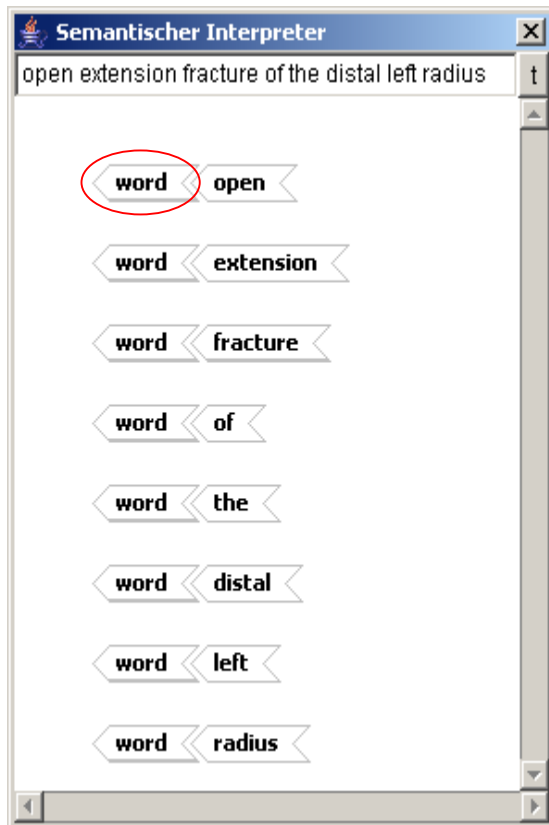
## Non-monotonism

The actual state of the transformation process determines which rule matches. Contradicting rules can thus coexist. Context decides which one is active.

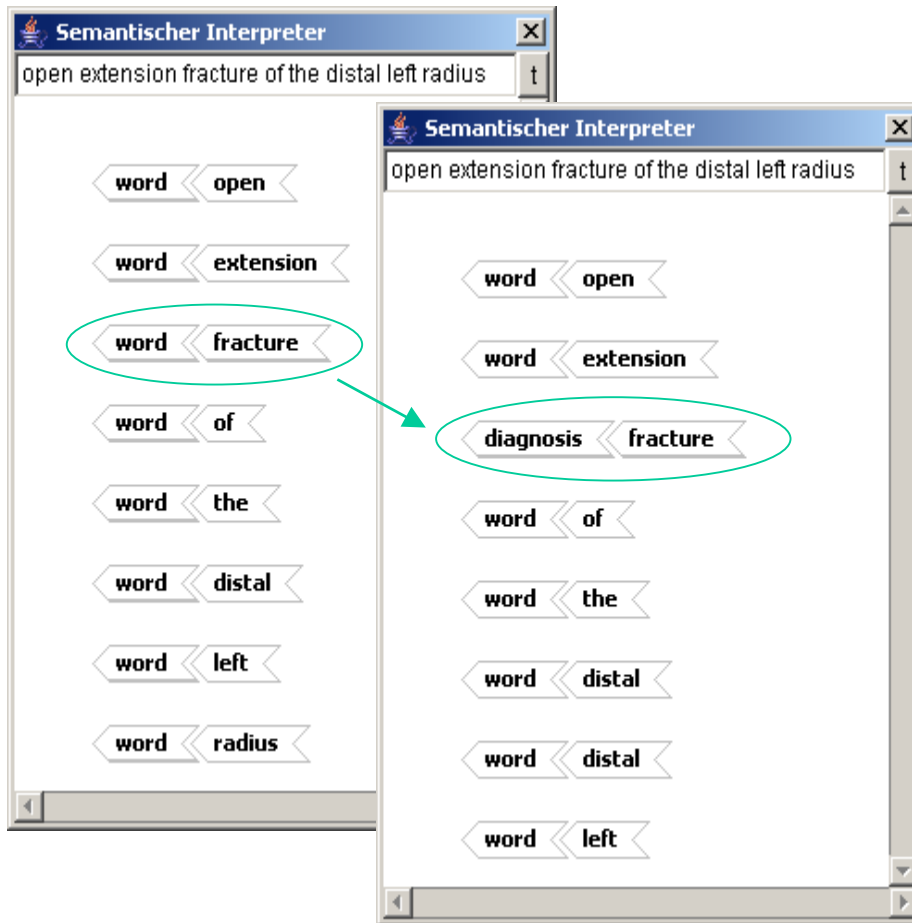
## Visibility

All rules and steps are visible to the human knowledge engineer.

# Transformation example: from words to meaning



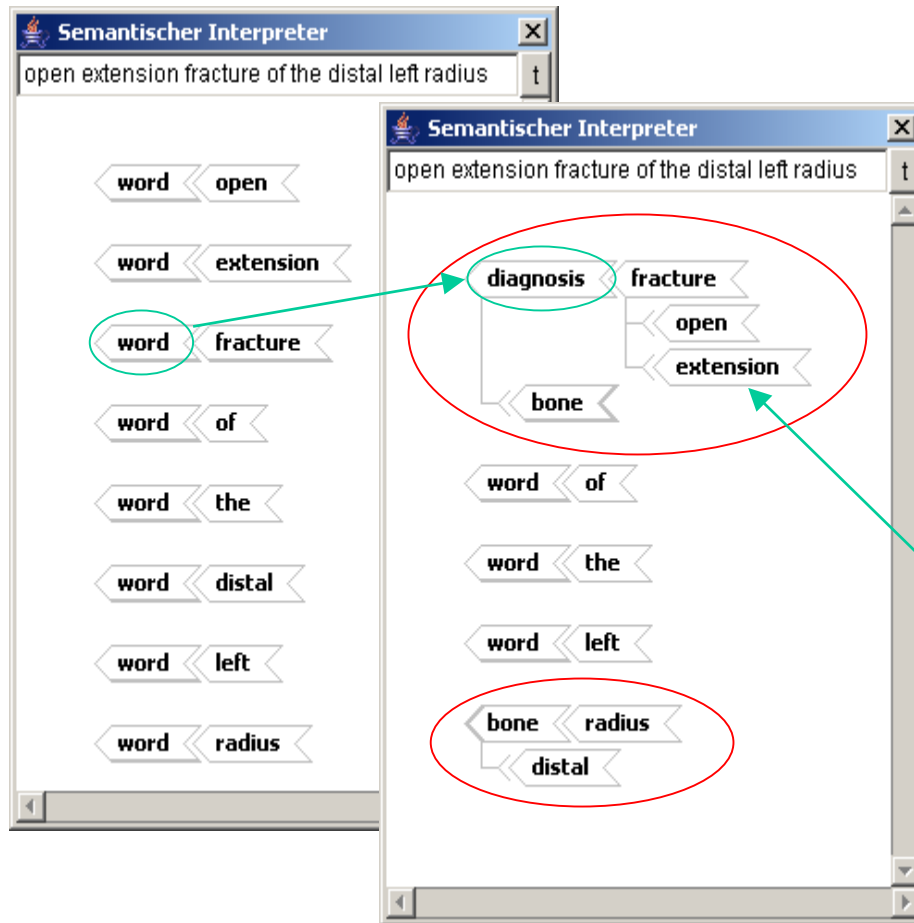
# Transformation example: from words to meaning



First step

"Fracture" becomes a diagnosis  
= **conceptualisation**

# Transformation example: from words to meaning



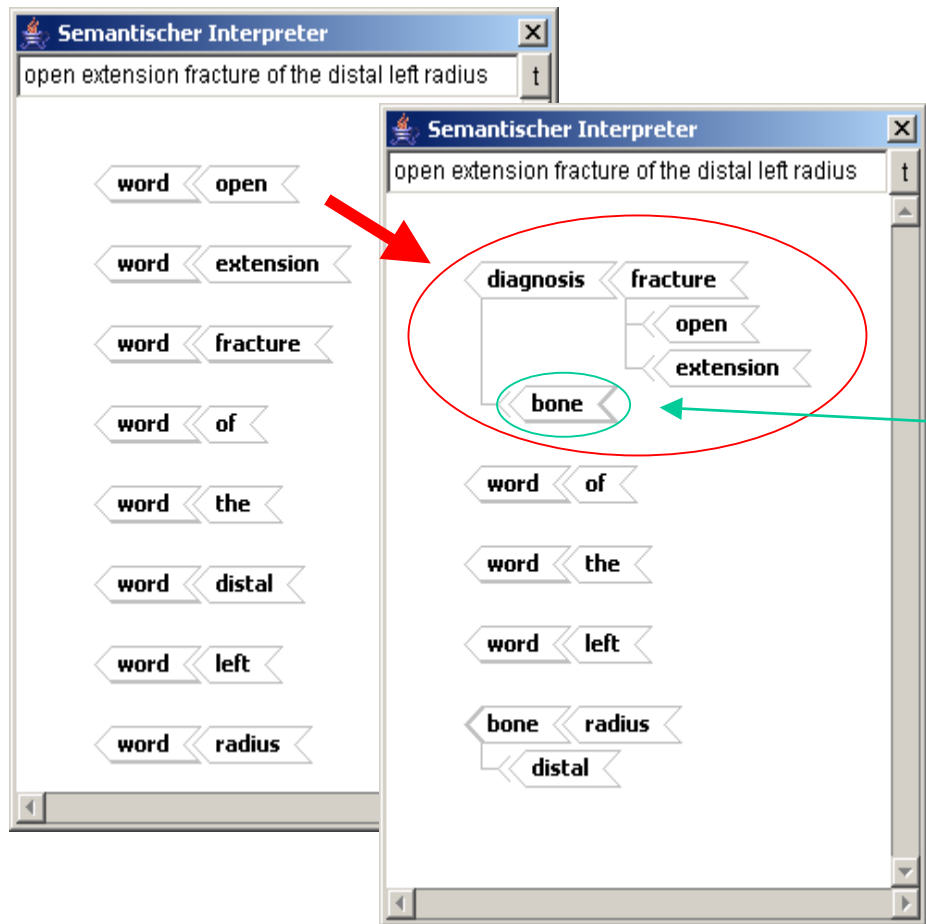
First step

"Fracture" becomes a diagnosis  
= conceptualisation

Second step

Attributes from the input  
are added

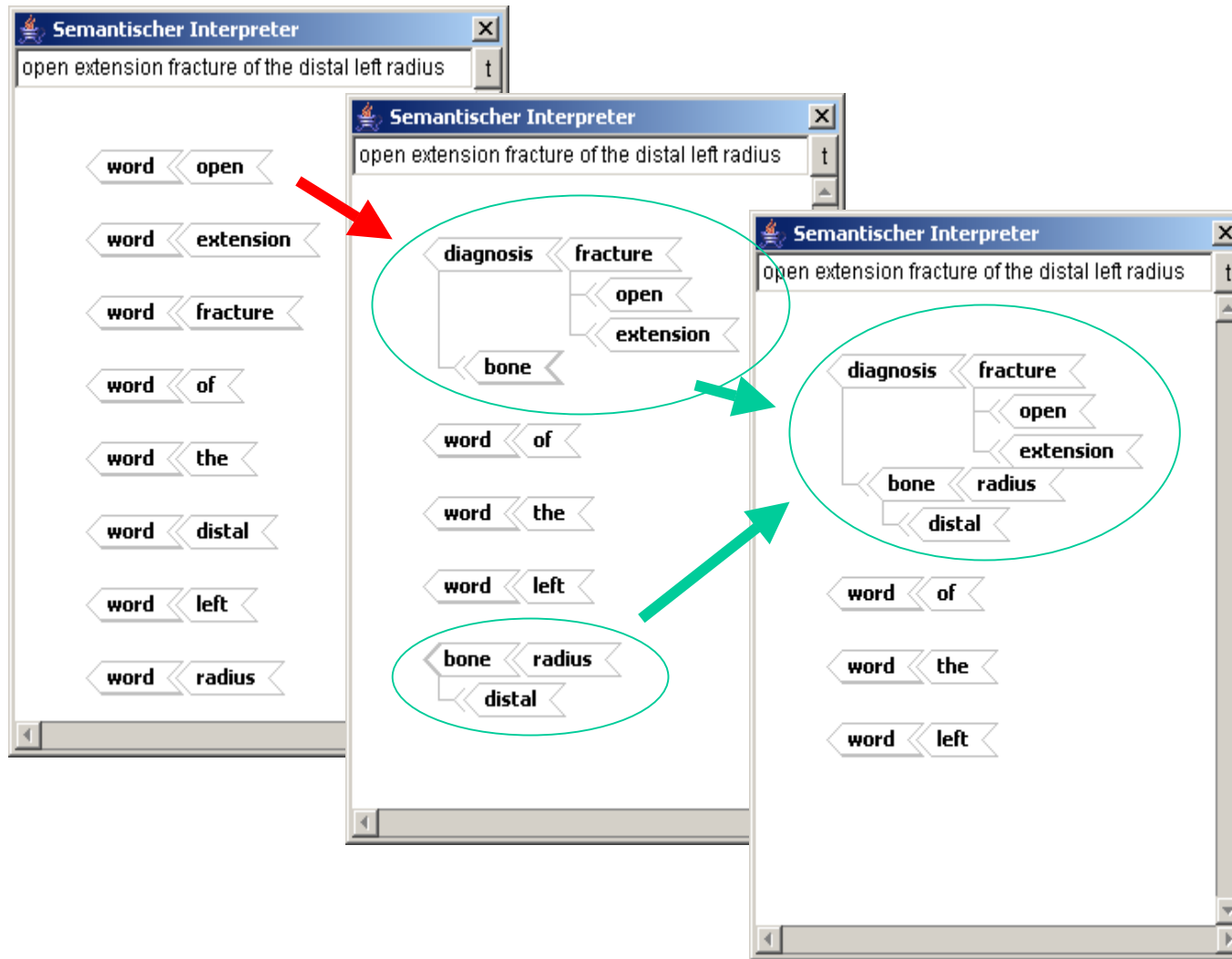
# Transformation example: from words to meaning



## Implicit information

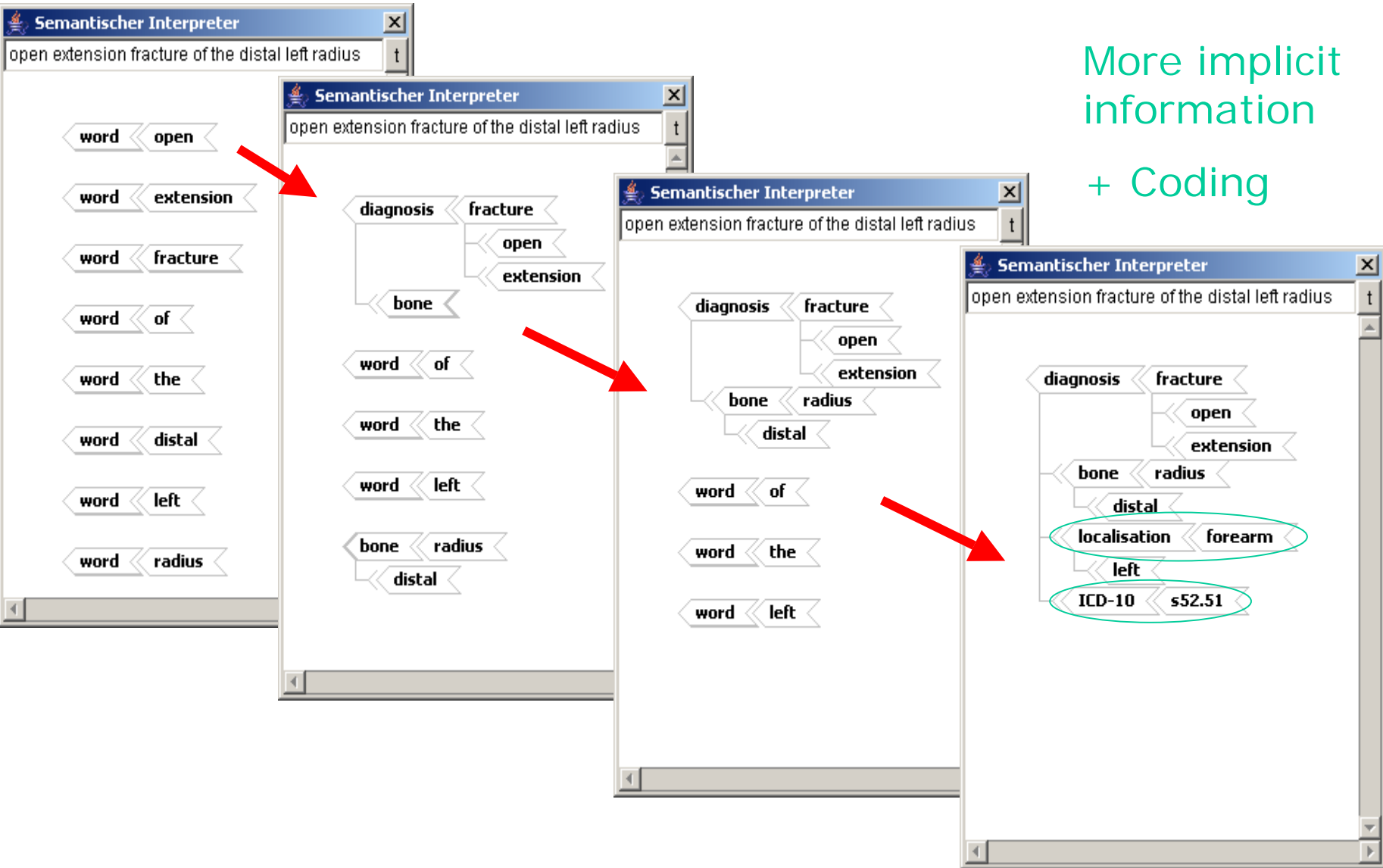
"Bone" was not mentioned in the input, but is implicitly deduced by the system

# Transformation example: from words to meaning



2 Molecules  
are joined

# Transformation example: from words to meaning



More implicit information  
+ Coding

# Conclusions

- It is possible to have a non-monotonic system working
- Concepts are not objects
- Use well-structured composite concept clusters

Thank you for your attention!