A Computational Model of Natural Language Communication

Interpretation, Inference, and Production in Database Semantics

ROLAND HAUSSER
Computational Linguistics
Friedrich Alexander-Universität Erlangen-Nürnberg
Germany

Part I.
The Communication Mechanism of Cognition

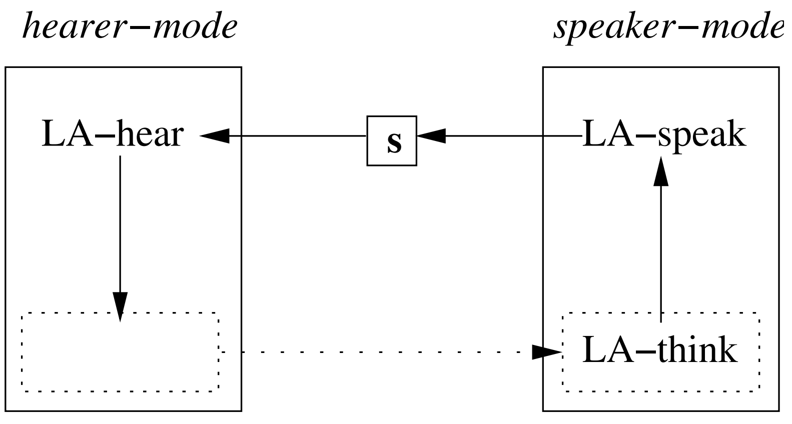
1. Matters of Method

1.1 Sign- or Agent-Oriented Analysis of Language?

The goal of Database Semantics is a theory of natural language communication which is complete with respect to function and data coverage, of low mathematical complexity, and suitable for an efficient implementation on the computer. The central question of Database Semantics is

How does communicating with natural language work?

1.1.1 THE BASIC MODEL OF TURN-TAKING

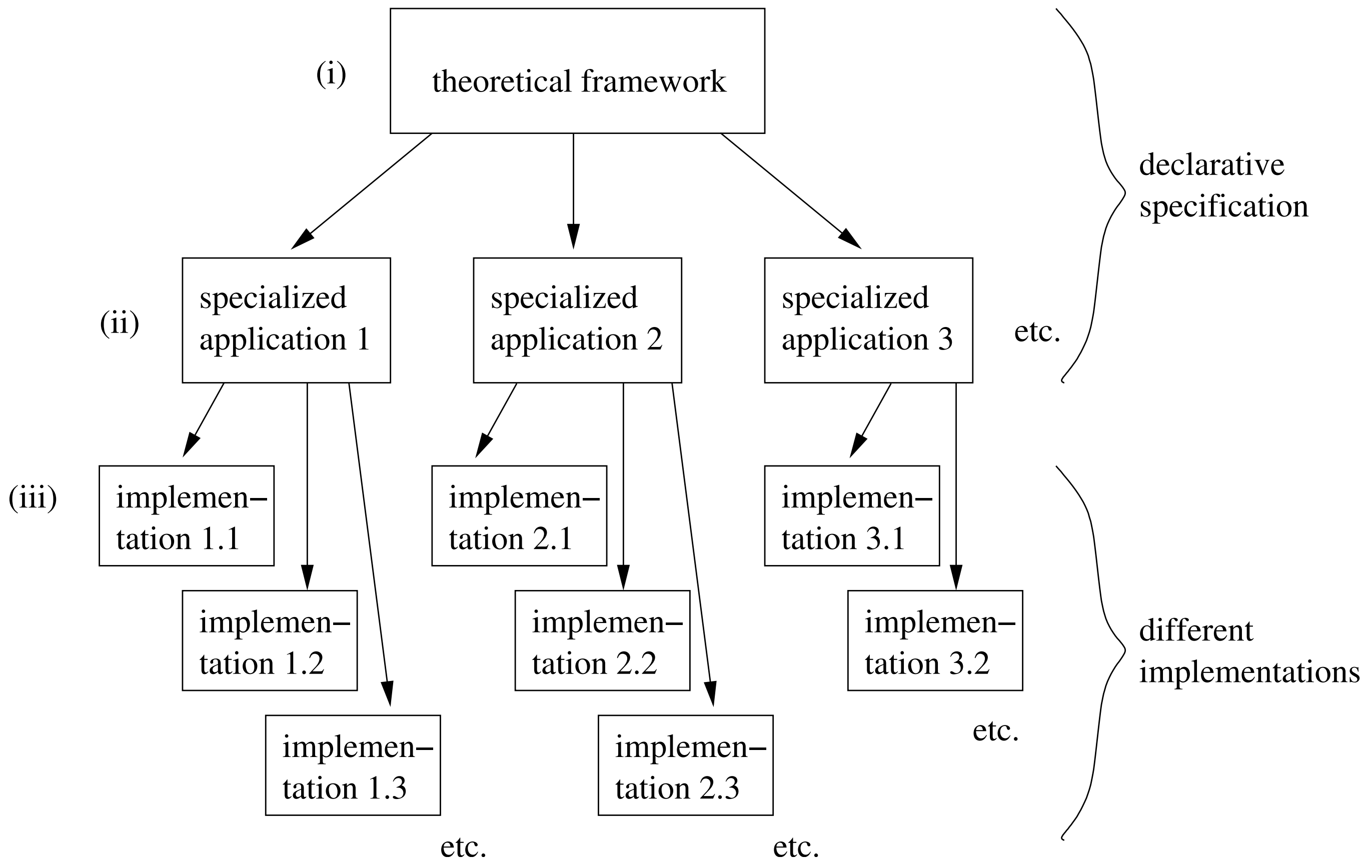


1.1.2 Two views of turn-taking

1. *Viewed from the outside:*
   Two communicating agents are observed as they are taking turns. This is represented by 1.1.1 when the two boxes are taken to be two different agents, one in the hearer- and the other in the speaker-mode.
2. *Viewed from the inside:*
   One communicating agent is observed as it switches between being the speaker and the hearer. This is represented by 1.1.1 when the two boxes are taken to be the same agent switching between the speaker- and the hearer-mode (with the dotted right-hand arrow indicating the switch).

1.2 Verification Principle

1.2.1 Correlation of declarative specification and implementations



1.3 Equation Principle

1.3.1 The equation principle of Database Semantics

1. The more realistic the reconstruction of cognition, the better the functioning of the model.
2. The better the functioning of the model, the more realistic the reconstruction of cognition.

1.4 Objectivation Principle

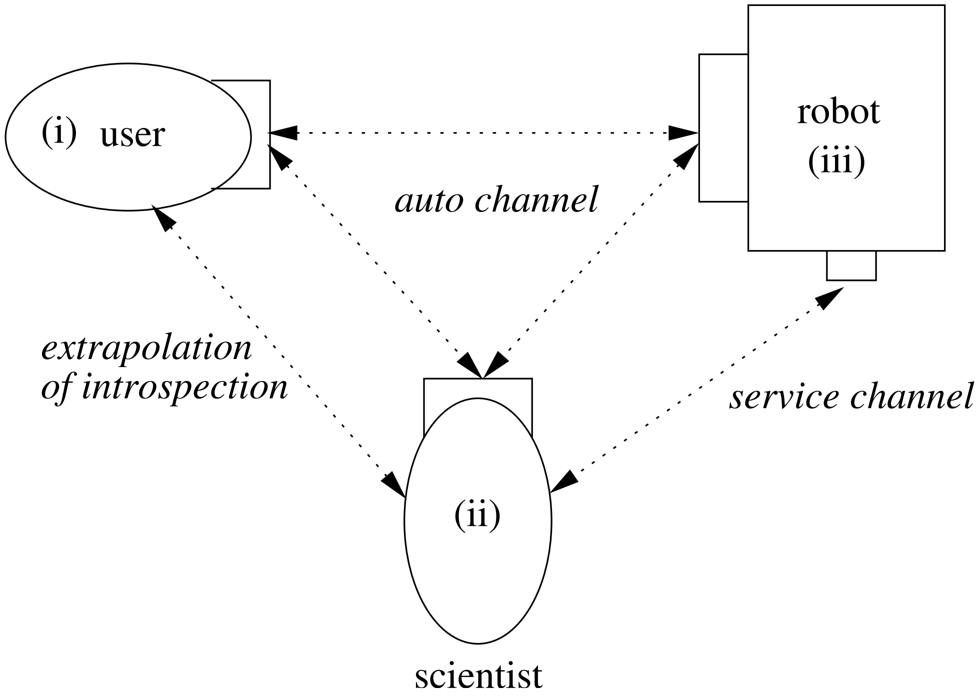
1.4.1 Constellations providing different kinds of data

1. Interaction between (i) the user and (iii) the robot
2. Interaction between (i) the user and (ii) the scientist
3. Interaction between (ii) the scientist and (iii) the robot

1.4.2 Data channels of communicative interaction

1. The *auto-channel*
   processes input automatically and produces output autonomously, at the context as well as the language level. In natural cognitive agents, i.e. the user and the scientist, the auto-channel is present from the very beginning in its full functionality. In artificial agents, in contrast, the auto-channel must be reconstructed – and it is the goal of Database Semantics to reconstruct it as realistically as possible.
2. The *extrapolation of introspection*
   is a specialization of the auto-channel and results from the scientists' effort to improve man-machine communication by taking the view of the human user. This is possible because the scientist and the user are natural agents.
3. The *service channel*
   is designed by the scientist for the observation and control of the artificial agent. It allows direct access to the robot's cognition because its cognitive architecture and functioning is a construct which in principle may be understood completely by the scientist.

1.4.3 Interaction between user, robot, and scientist



1.5 Equivalence Principles for Interfaces and for Input/Output

The methodological principles of Database Semantics presented so far, namely

1. the *Verification Principle*
   i.e. the development of the theory in the form of a declarative specification which is continuously verified by means of an implemented prototype (cf. Section 1.2),
2. the *Equation Principle*
   i.e. the equating of theoretical correctness with the behavioral adequacy of the prototype during longterm up-scaling (cf. Section 1.3), and
3. the *Objectivation Principle*
   i.e. the establishing of objective channels for observing language communication between natural and artificial agents (cf. Section 1.4),

are constrained by

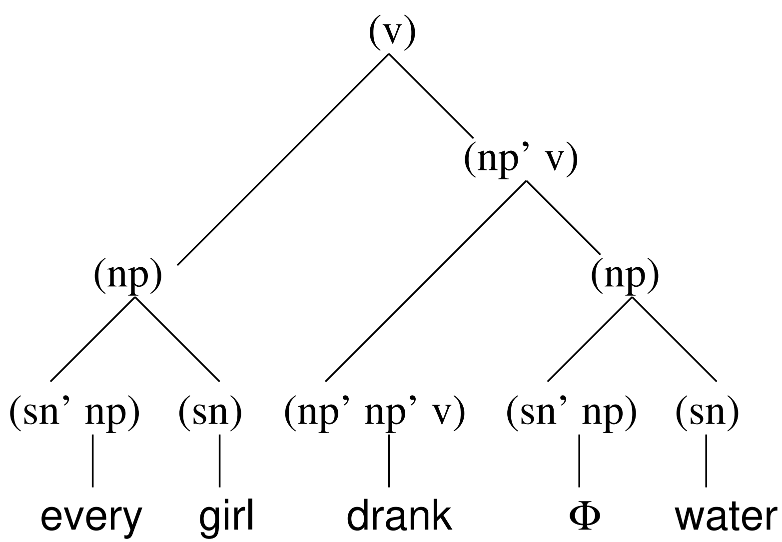
4. the *Interface Equivalence Principle* and
5. the *Input/Output Equivalence Principle.*

1.6 Surface Compositionality and Time-Linearity

1.6.1 Surface Compositionality

A grammatical analysis is surface compositional if it uses only the concrete word forms as the building blocks of composition, such that all syntactic and semantic properties of a complex expression derive systematically from the syntactic category and the literal meaning of the lexical items.

1.6.2 Analysis violating Surface Compositionality



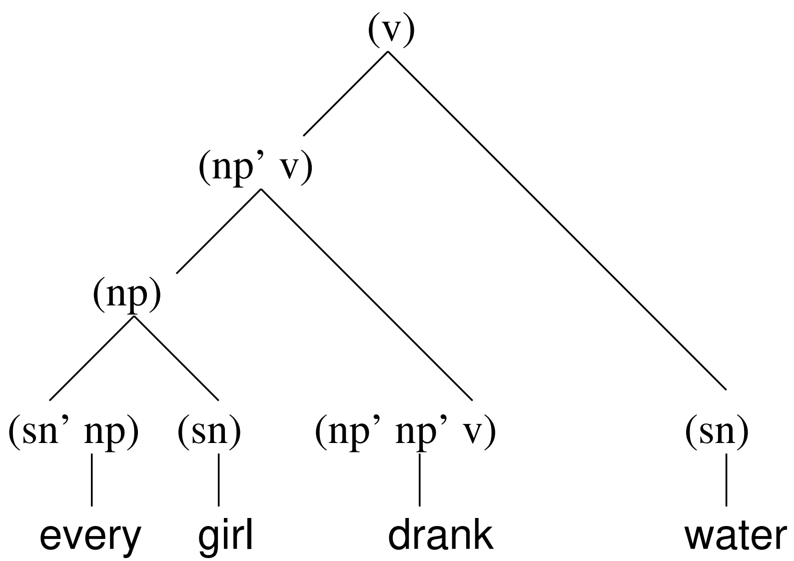
1.6.3 The categories of 1.6.2

(sn' np) = *determiner*, takes a singular noun sn' and makes a noun phrase np.
(sn) = *singular noun*, fills a valency position sn' in the determiner.
(np' np' v)= *transitive verb*, takes a noun phrase np and makes a (np' v).
(np) = *noun phrase*, fills a valency position np' in the verb.
(np' v) = *intransitive verb*, takes a noun phrase np and makes a (v).
(v) = *verb* with no open valency positions (sentence).

1.6.4 Rules computing possible substitutions for deriving 1.6.2

(v) $\rightarrow$ (np) (np' v)
(np) $\rightarrow$ (sn' np) (sn)
(np' v) $\rightarrow$ (np' np' v) (np)
(sn' np) $\rightarrow$ every, Φ
(sn) $\rightarrow$ girl, water
(np' np' v) $\rightarrow$ drank

1.6.5 Satisfying Surface Compositionality and Time-Linearity



1.6.6 Rules computing the possible continuations for deriving 1.6.5

(VAR' X) (VAR) $\longrightarrow$ (X)
(VAR) (VAR' X) $\longrightarrow$ (X)

1.6.7 Application of a rule computing a possible continuation

	ss		*nw*		*ss'*	
rule patterns	(VAR'	X)	(VAR)	→	(X)	
						matching and binding
categories	(sn'	np)	(sn)		(np)	
surfaces	every		girl		every girl	

1.6.8 Variable definition of the time-linear rules for deriving 1.6.5

If VAR' is sn', then VAR is sn. (*identity-based agreement*)
If VAR' is np', then VAR is np, sn, or pn.(*definition-based agreement*)

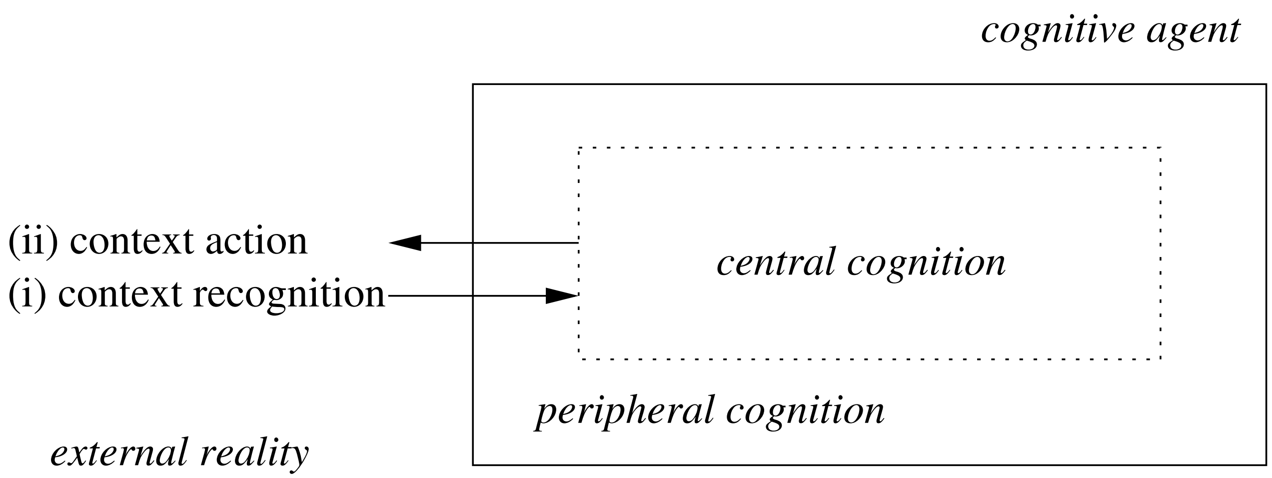
2. Interfaces and Components

2.1 Cognitive Agents with and without Language

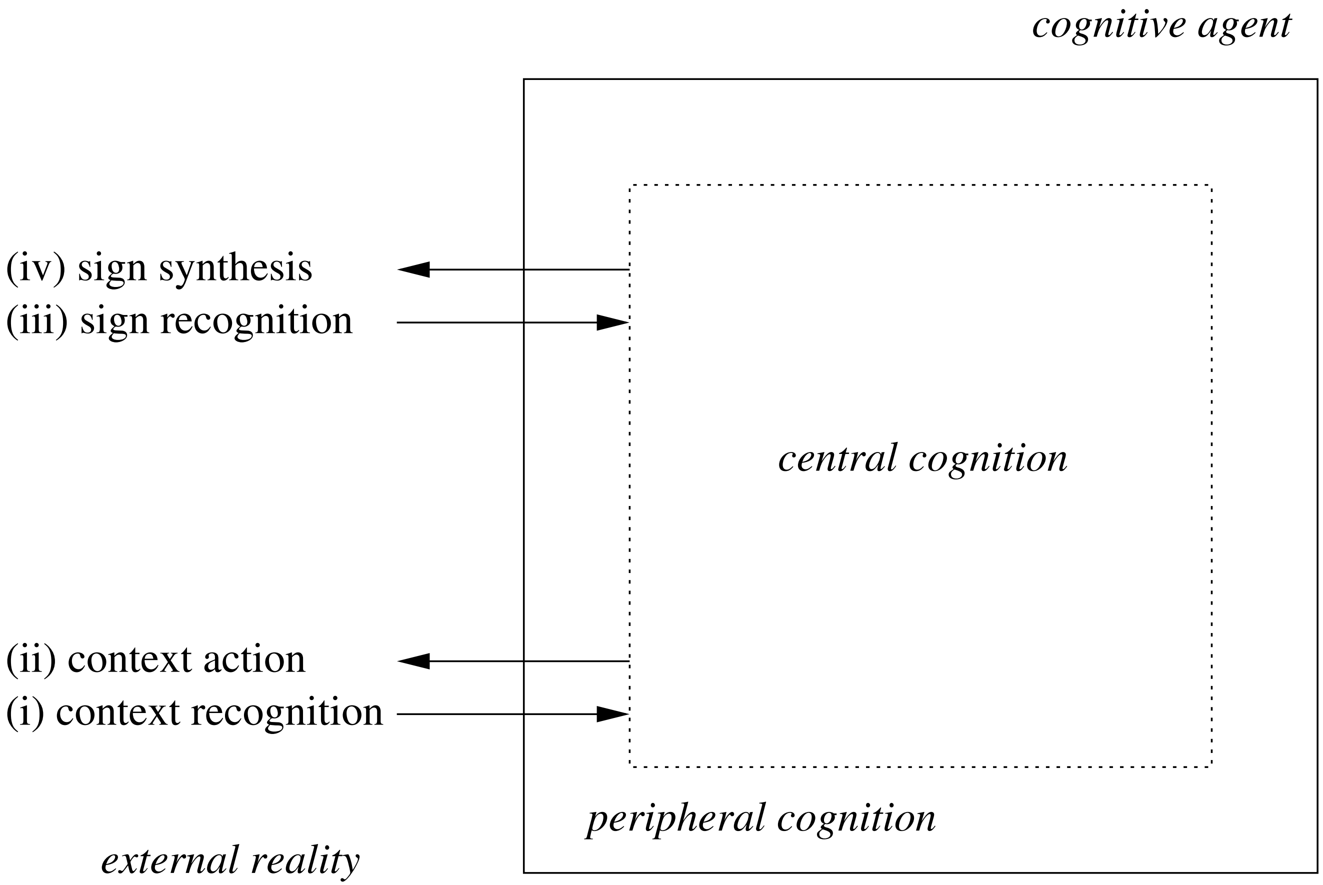
2.1.1 Support for building the context component first

1. Constructs from the context-level may be re-used at the language level. This holds for (i) the concepts, as types and as token, (ii) the external interfaces for input and output, (iii) the data structure, (iv) the algorithm, and (v) the inferences.
2. The context is universal – in the sense of being independent of a particular language, yet all the different languages may be interpreted relative to the same kind of context component.
3. In phylogeny (evolution) and ontogeny (child development) the context component comes first.

2.1.2 External interfaces of a cognitive agent without language

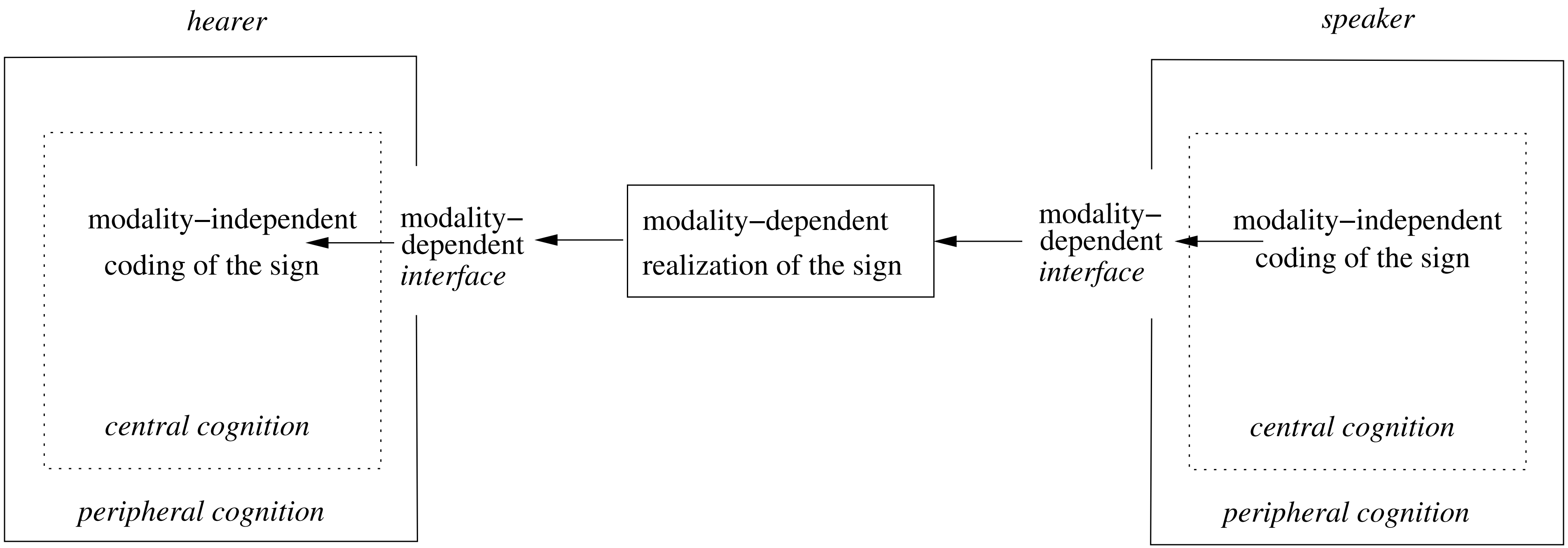


2.1.3 External Interfaces of a cognitive agent with language



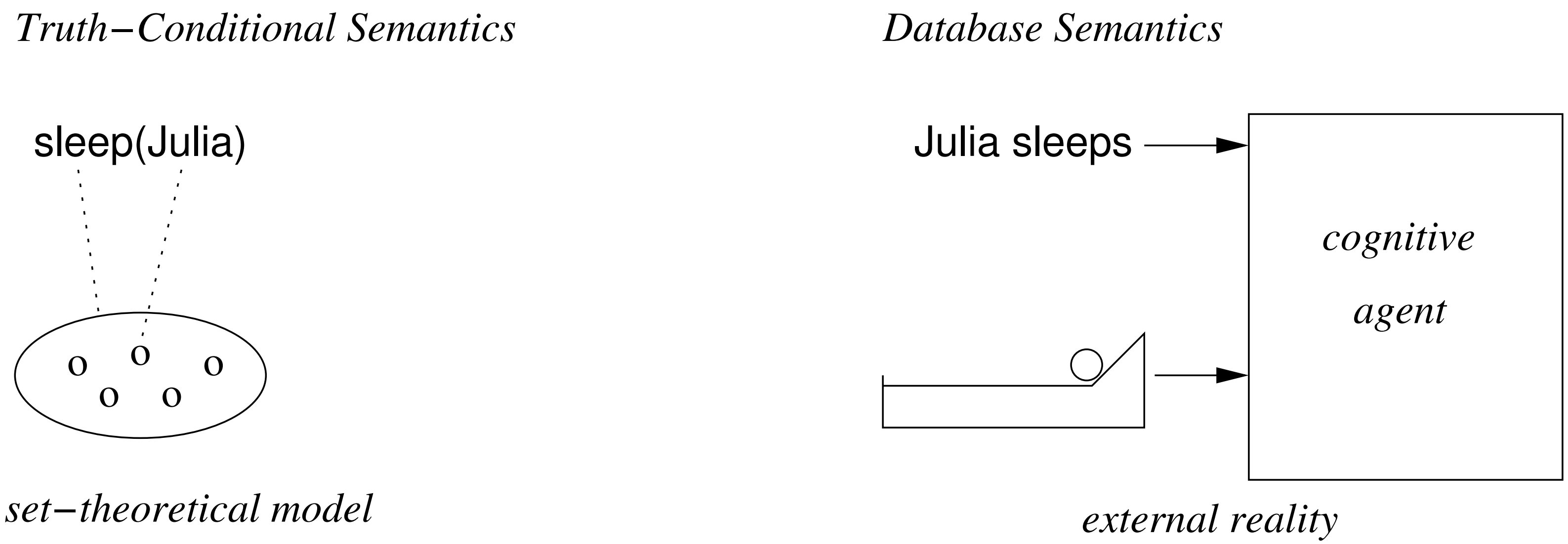
2.2 Modalities and Media

2.2.1 Modality-independent and modality-dependent coding



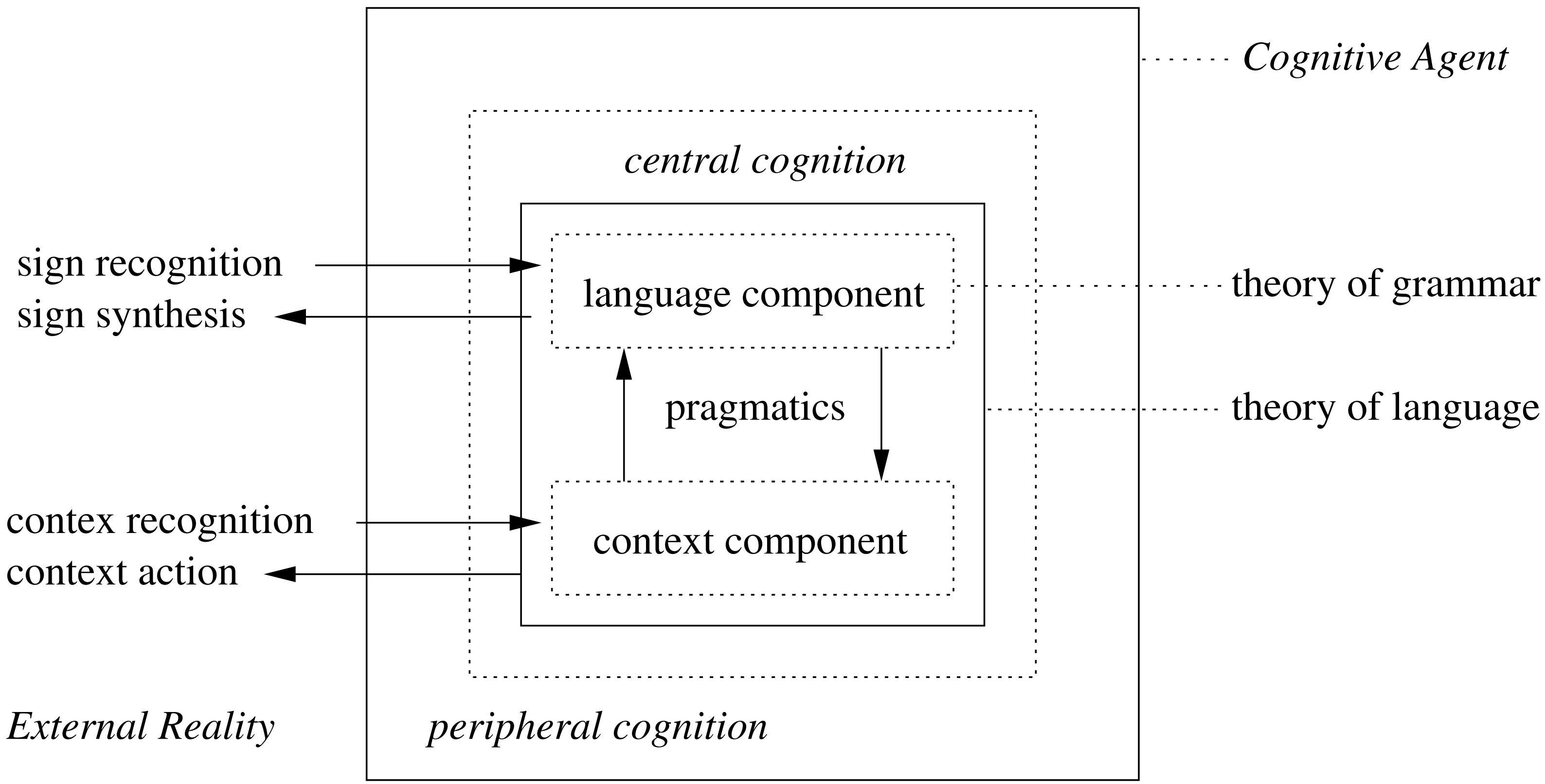
2.3 Alternative Ontologies for Referring with Language

2.3.1 Reference in alternative ontologies



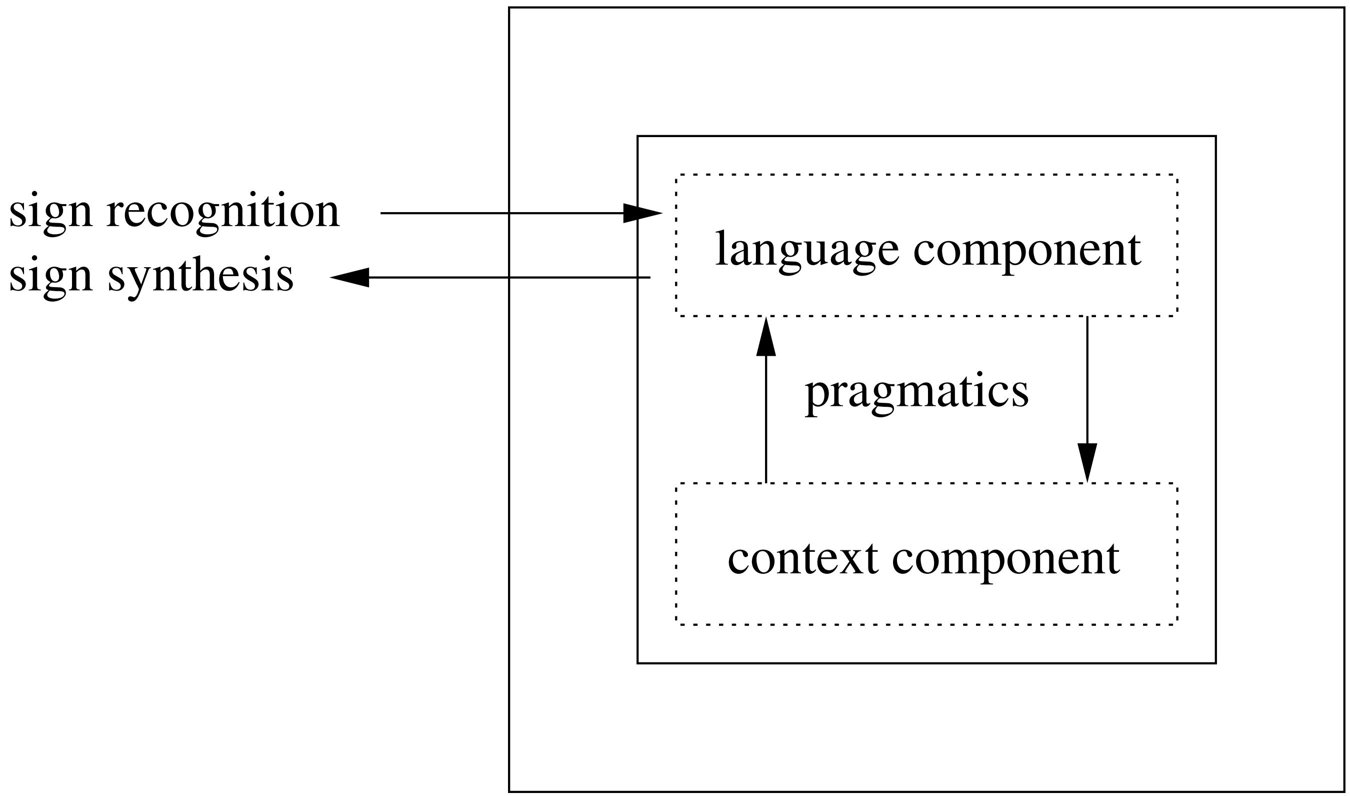
2.4 Theory of Language and Theory of Grammar

2.4.1 STRUCTURING CENTRAL COGNITION IN AGENTS WITH LANGUAGE



2.5 Immediate Reference and Mediated Reference

2.5.1 USE OF EXTERNAL INTERFACES IN MEDIATED REFERENCE



2.5.2 Disadvantages of not having contextual interfaces

1. *The conceptual core of language meanings remains undefined.*
   Most basic concepts originate in agents without language as recognition and action procedures of their contextual interfaces, and are re-used as the core of language meanings in agents with language.Therefore, agents with language but without contextual interfaces use meanings which are void of a conceptual core – though the relations between the concepts, represented by place holder words, may still be defined, both absolutely (for example in the *is-a* or *is-part-of* hierarchies) and episodically.

2. *The coherence or incoherence of content cannot be judged autonomously.*
   The coherence of stored content originates in the coherence of the external world.Therefore, only agents with contextual interfaces are able to relate content 'imported' by means of language to the data of their own experience. An agent without contextual interfaces, in contrast, has nothing but imported data – which is why the responsibility for their coherence lies solely with the users who store the data in the agent.

2.6 The SLIM Theory of Language

S = **S**urface Compositionality (methodological principle, cf. 1.6.1):

Syntactic-semantic composition assembles only concrete word forms, excluding the use of zero-elements, identity mappings, or transformations.

L = time-**L**inearity (empirical principle, cf. 1.6.5):

Interpretation and production of utterances are based on a strictly time-linear derivation order.

I = **I**nternal (ontological principle, cf. 2.3.1):

Interpretation and production of utterances are analyzed as cognitive procedures located inside the speaker-hearer.

M = **M**atching (functional principle, cf. 3.2.3):

Referring with language to past, current, or future objects and events is modeled in terms of pattern matching between language meanings and a context.

2.6.1 First principle of pragmatics (PoP-1)

> The speaker's utterance meaning$_2$ is the use of the sign's literal meaning$_1$ relative to an internal context.

Meaning$_1$ is the *literal meaning* of the sign, meaning$_2$ is the *speaker meaning* of an utterance in which the sign's meaning$_1$ is used. Even in the hearer-mode, meaning$_2$ is called the 'speaker meaning' because communication is successful only if the hearer uses the sign's meaning$_1$ to refer to the same objects or events as the speaker.

A sign can only be used successfully if the context of interpretation has been determined (and delimited) correctly. Finding the context is based on the sign's STAR:

S = space (location where the sign has been uttered)
T = time (time when the sign has been uttered)
A = author (agent who produced the sign)
R = recipient (agent intended to receive the sign)

2.6.2 Second principle of pragmatics (PoP-2)

A sign's STAR determines the *entry context* of production and interpretation in the contextual databases of the speaker and the hearer.

2.6.3 Third principle of pragmatics (PoP-3)

The matching of signs with their respective sub-contexts is incremental, whereby in production the elementary signs follow the time-linear order of the underlying thought path, while in interpretation the thought path follows the time-linear order of the incoming elementary signs.

2.6.4 Fourth principle of pragmatics (PoP-4)

The reference mechanism of a *symbol* is based on a meaning$_1$ which is defined as a concept type. Symbols refer from their place in a positioned sentence by matching their meaning$_1$ with corresponding concept tokens at the context level.

2.6.5 Fifth principle of pragmatics (PoP-5)

The reference mechanism of an *indexical* is based on a meaning$_1$ which is defined as one of two characteristic pointers. The first points into the agent's context and is called the context pointer or C. The second points at the agent and is called the agent pointer or A.

2.6.6 Indexicals as nouns and adjectives with A and C pointers

noun A	*noun C*	*adj A*	*adj C*
I, we	you	here	there
	he, she, it	now	then
	this, they		

2.6.7 Sixth principle of pragmatics (PoP-6)

The reference mechanism of a *name* is based on a private marker which matches a corresponding marker contained in the cognitive representation of the object referred to.

2.6.8 Seventh principle of pragmatics (PoP-7)

Symbols occur as verbs, adjectives, and nouns. *Indexicals* occur as adjectives and nouns. *Names* occur only as nouns.

2.6.9 Relation between the kinds of sign and the parts of speech

name	Fido		
indexical	this	here	
symbol	dog	black	see
	noun	*adj.*	*verb*

2.6.10 Correlation of part of speech and kind of sign in sentences

	noun	verb	adjective
name	John		
symbol		slept	
symbol			in the kitchen
indexical	he		
symbol		slept	
indexical			there

3. Data Structure and Algorithm

3.1 Proplets for Coding Propositional Content

3.1.1 Context proplets representing *dog barks. (I) run.*

sur:	sur:	sur:
noun: *dog*	verb: *bark*	verb: *run*
fnc: bark	arg: dog	arg: moi
prn: 22	nc: 23 run	pc: 22 bark
	prn: 22	prn: 23

3.1.2 Coding of relations between concepts via proplets

context level:

sur:	sur:	sur:
noun: *dog*	verb: *bark*	verb: *run*
fnc: bark	arg: dog	arg: moi
prn: 22	nc: 23 run	pc: 22 bark
	prn: 22	prn: 23

3.2 Internal Matching between Language and Context Proplets

3.2.1 LANGUAGE PROPLETS REPRESENTING *dog barks. (I) run.*

$$
\begin{bmatrix} \text{sur: } \mathsf{Hund} \\ \text{noun: } \textit{dog} \\ \text{fnc: bark} \\ \text{prn: 122} \end{bmatrix}
\quad
\begin{bmatrix} \text{sur: } \mathsf{bellt} \\ \text{verb: } \textit{bark} \\ \text{arg: dog} \\ \text{nc: 123 run} \\ \text{prn: 122} \end{bmatrix}
\quad
\begin{bmatrix} \text{sur: } \mathsf{fliehe} \\ \text{verb: } \textit{run} \\ \text{arg: moi} \\ \text{pc: 122 bark} \\ \text{prn: 123} \end{bmatrix}
$$

3.2.2 KEYS FOR LEXICAL LOOKUP IN THE SPEAKER- AND THE HEARER-MODE

$$
\begin{bmatrix} \text{sur: Hund} \\ \text{noun: dog} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}
\begin{array}{l} \leftarrow \textit{key for lexical lookup in the hearer-mode} \\ \leftarrow \textit{key for lexical lookup in the speaker-mode} \\ \\ \\ \end{array}
$$

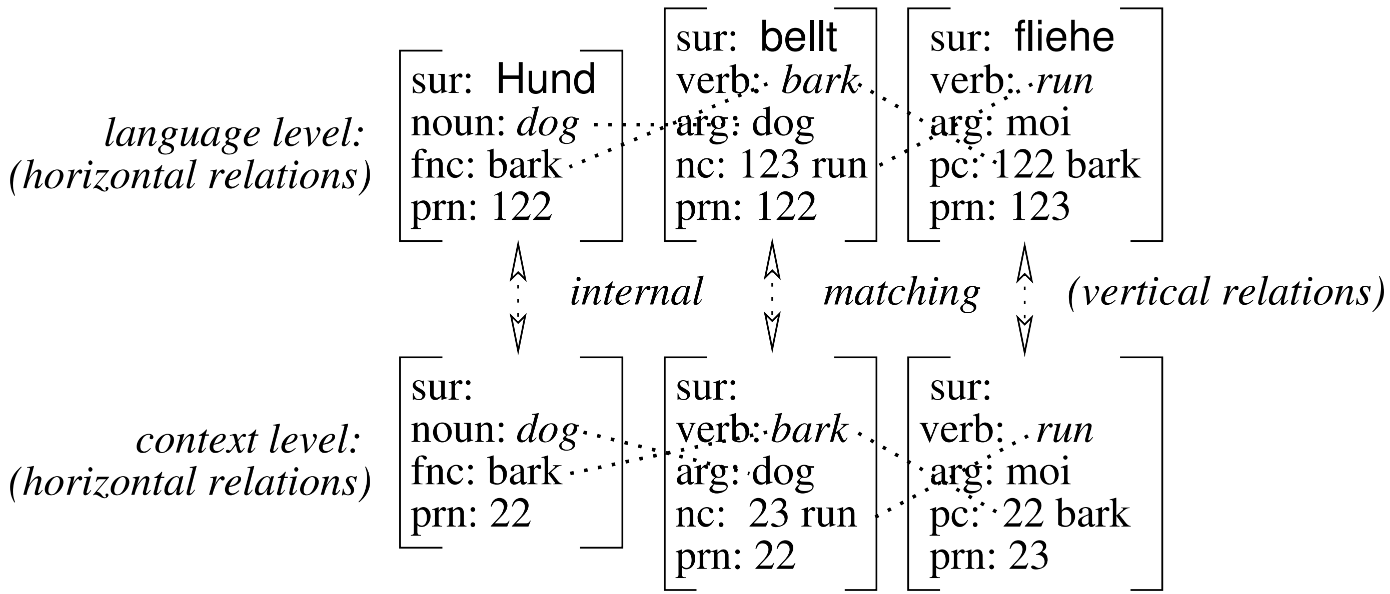
3.2.3 Conditions on successful matching

1. *Attribute condition*
   The matching between two proplets A and B requires that the intersection of their attributes contains a predefined list of attributes regarded as relevant:

   $$\{\text{list}\} \subseteq \{\{\text{proplet-A-attributes}\} \cap \{\text{proplet-B-attributes}\}\}$$

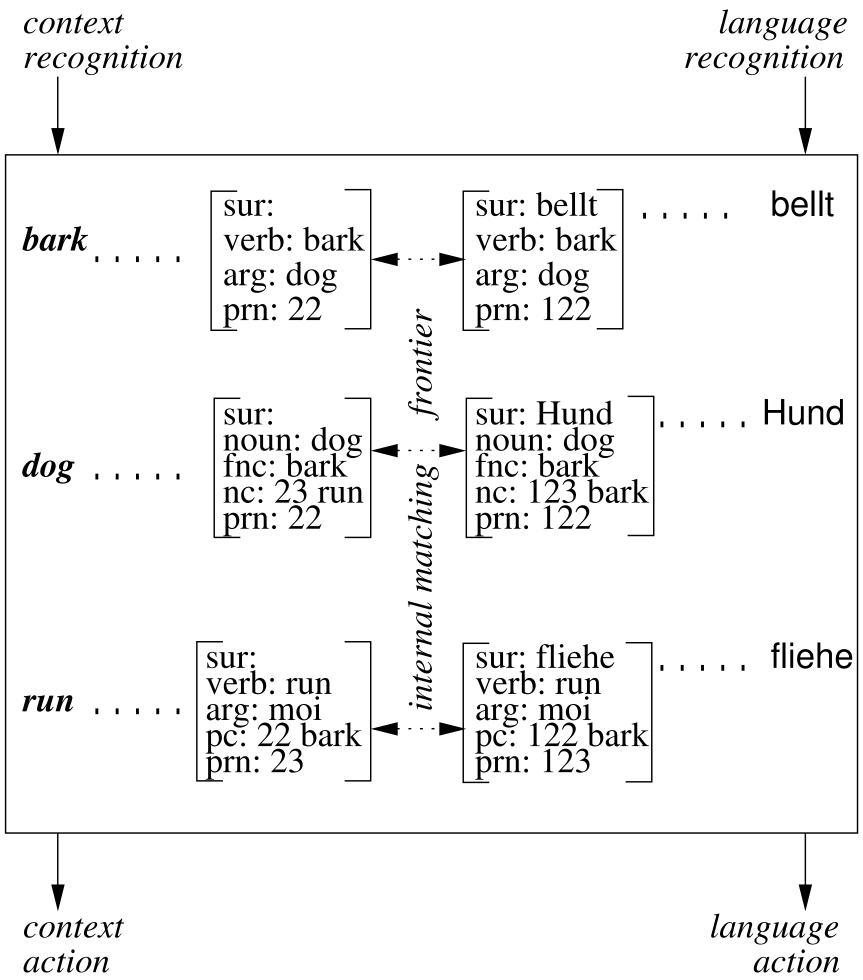
2. *Value condition*
   The matching between two proplets requires that the variables (and a fortiori the constants) of their common attributes are compatible.

3.2.4 IMPACT OF INTER-PROPLET RELATIONS ON MATCHING



3.3 Storage of Proplets in a Word Bank

3.3.1 DATA STRUCTURE OF A WORD BANK

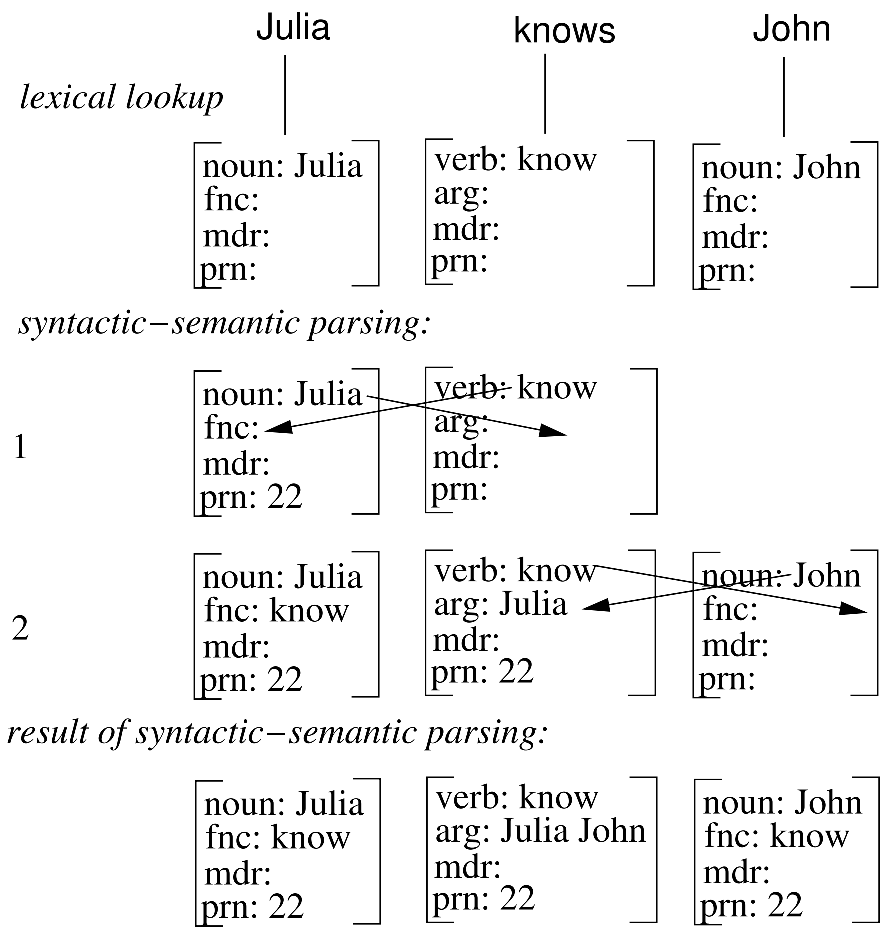


3.4 Time-linear Algorithm of LA-Grammar

3.4.1 Applying the Input/Output Equivalence Principle to language

1. Input and output at the language level are signs of natural language, such as phrases, sentences, or texts.
2. The parts into which the signs of natural language disassemble during intake and discharge are word forms.
3. The order of the parts during intake and discharge is time-linear.

3.4.2 Time-linear Derivation (Principle of Possible Continuations)



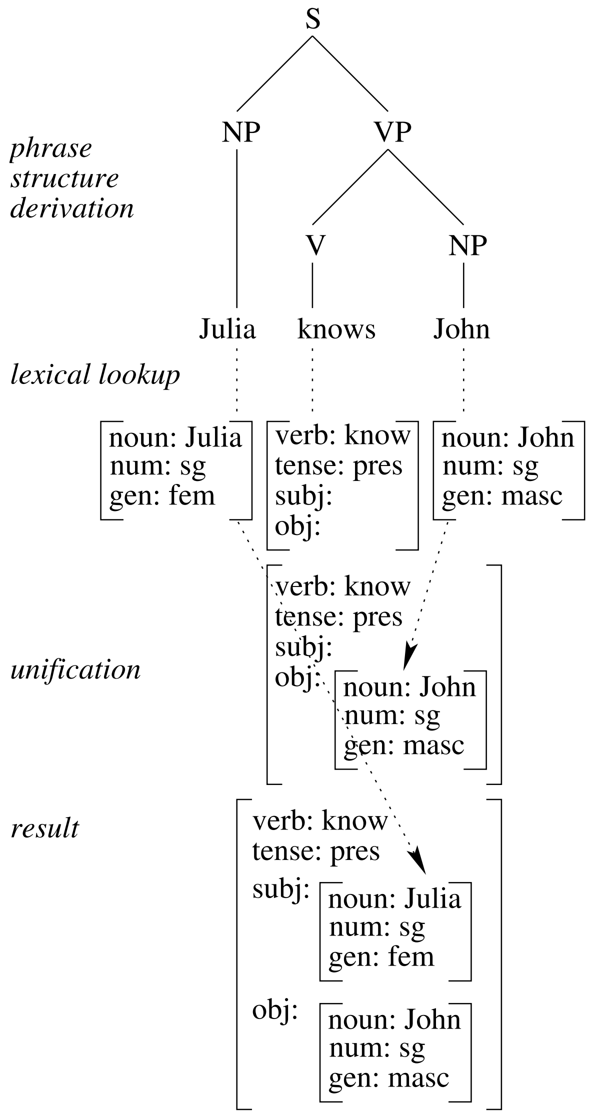
3.4.3 Example of an **LA-hear** rule application

	rule name	*ss pattern*	*nw pattern*	*operations*	*rule package*
rule level	NOM+FV:	$\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: }\end{bmatrix}$	$\begin{bmatrix}\text{verb: } \beta \\ \text{arg: }\end{bmatrix}$	copy α nw.arg copy β ss.fnc	{FV+OBJ, ...}
proplet level		$\begin{bmatrix}\text{noun: Julia} \\ \text{fnc: } \\ \text{mdr: } \\ \text{prn: 22}\end{bmatrix}$	$\begin{bmatrix}\text{verb: know} \\ \text{arg: } \\ \text{mdr: } \\ \text{prn: }\end{bmatrix}$		

3.4.4 Result of the **LA-hear** rule application

$$\begin{bmatrix}\text{noun: Julia} \\ \text{fnc: know} \\ \text{mdr: } \\ \text{prn: 22}\end{bmatrix} \quad \begin{bmatrix}\text{verb: know} \\ \text{arg: Julia} \\ \text{mdr: } \\ \text{prn: 22}\end{bmatrix}$$

3.4.5 Non-time-linear Derivation (Principle of Poss. Substitutions)



3.5 Cycle of Natural Language Communication

3.5.1 Example of an **LA-think** rule application

	rule name	ss pattern	nw pattern	operations	rule package
rule level	V_N_V:	$\begin{bmatrix}\text{verb: } \beta \\ \text{arg: X } \alpha \text{ Y} \\ \text{prn: k}\end{bmatrix}$	$\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta \\ \text{prn: k}\end{bmatrix}$	output position ss mark α ss	{V_N_V, ...}
proplet level		$\begin{bmatrix}\text{verb: know} \\ \text{arg: Julia John} \\ \text{mdr:} \\ \text{prn: 22}\end{bmatrix}$			

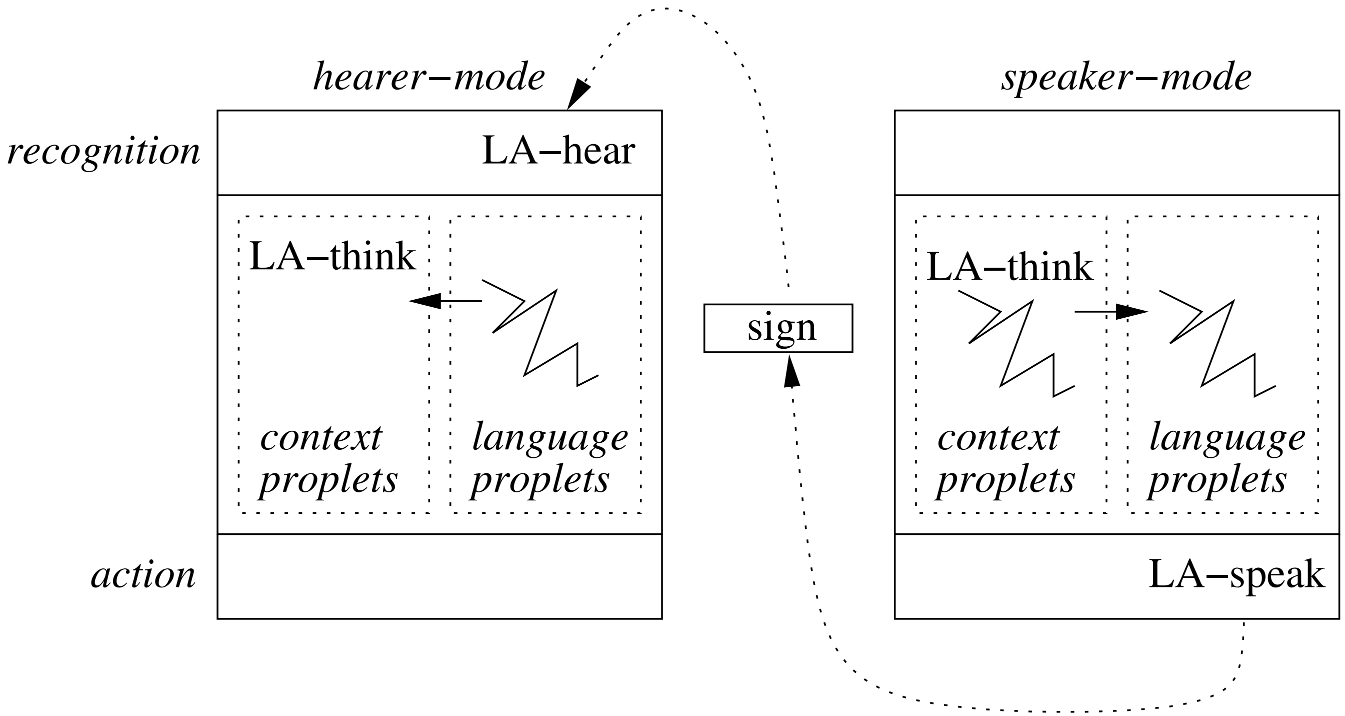
3.5.2 Result of the **LA-think** rule application

$$\begin{bmatrix}\text{verb: know} \\ \text{arg: !Julia John} \\ \text{mdr:} \\ \text{prn: 22}\end{bmatrix} \begin{bmatrix}\text{noun: Julia} \\ \text{fnc: know} \\ \text{mdr:} \\ \text{prn: 22}\end{bmatrix}$$

3.5.3 SCHEMATIC PRODUCTION OF Julia knows John.

activated sequence					*realization*
i					
	V				
i.1		n			n
	V	N			
i.2	fv	n			n fv
	V	N			
i.3	fv	n	n		n fv n
	V	N	N		
i.4	fv p	n	n		n fv n p
	V	N	N		

3.5.4 The cycle of natural language communication



3.6 A Bare Bone Example of Database Semantics: DBS-letter

3.6.1 Isolated proplets representing the letters A, E, L, O, S, V

$$\begin{bmatrix}\text{lett: A}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}\begin{bmatrix}\text{lett: E}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}\begin{bmatrix}\text{lett: L}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}\begin{bmatrix}\text{lett: O}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}\begin{bmatrix}\text{lett: S}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}\begin{bmatrix}\text{lett: V}\\\text{prev:}\\\text{next:}\\\text{wrd:}\end{bmatrix}$$

3.6.2 DEFINITION OF **LA-letter-IN** FOR CONNECTING ISOLATED PROPLETS

$\text{ST\_}S$ = {(lett: α, {r-in})}

$$\text{r-in:} \begin{bmatrix}\text{lett: }\alpha\\\text{prev:}\\\text{next:}\end{bmatrix}\begin{bmatrix}\text{lett: }\beta\\\text{prev:}\\\text{next:}\end{bmatrix} \begin{array}{l}\text{copy } \alpha \text{ nw.prev}\\\text{copy } \beta \text{ ss.next}\end{array} \quad \{\text{r-in}\}$$

$\text{ST\_}F$ = {(lett: β, rp_{r-in}) }

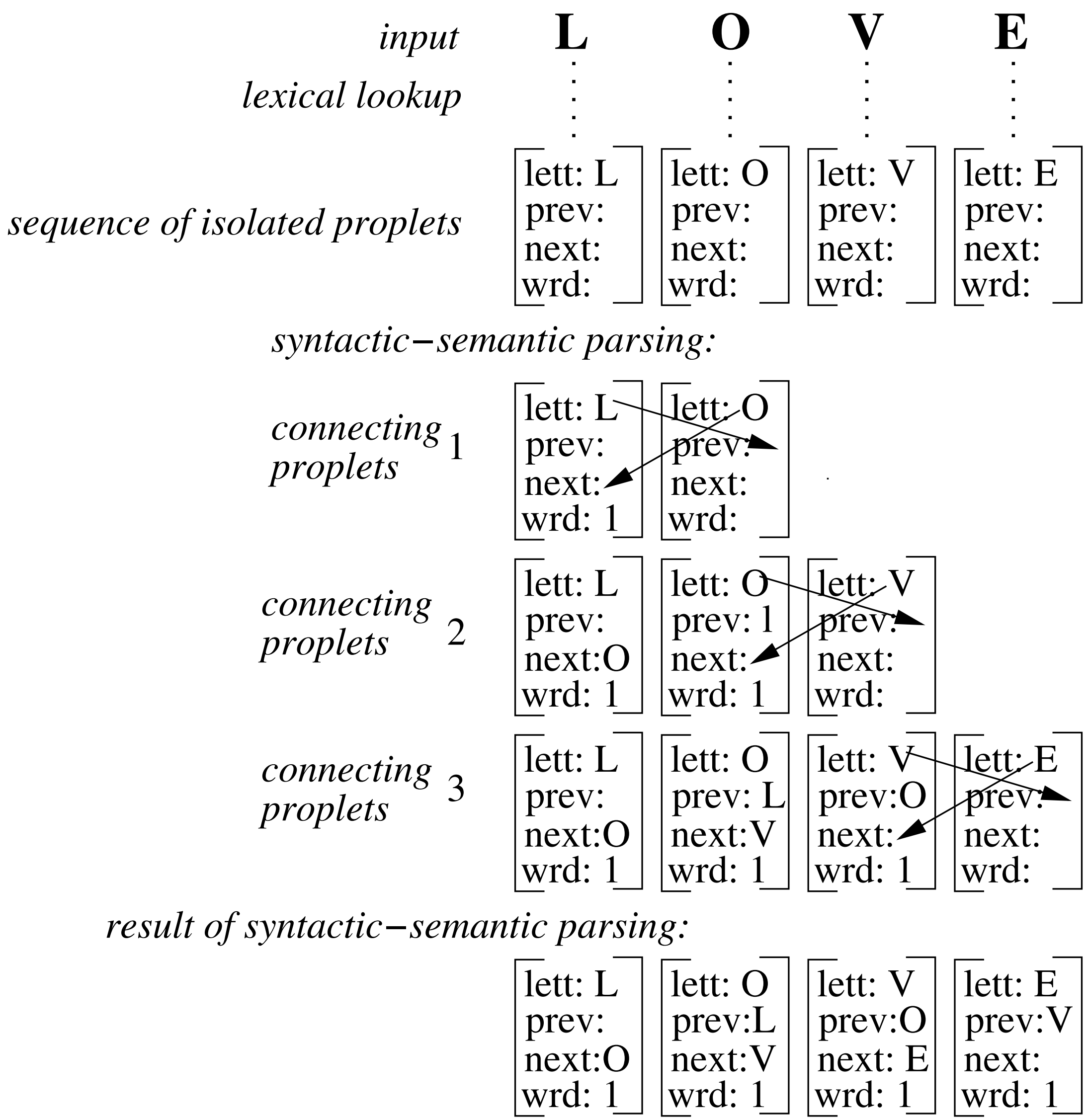
3.6.3 Example of an LA-letter-IN rule application

	rule name	*ss pattern*	*nw pattern*	*operations*	*rule package*
rule level	r-in:	$\begin{bmatrix}\text{lett: } \alpha \\ \text{prev:} \\ \text{next:}\end{bmatrix}$	$\begin{bmatrix}\text{lett: } \beta \\ \text{prev:} \\ \text{next:}\end{bmatrix}$	copy α nw.prev copy β ss.next	{r-in}
proplet level		$\begin{bmatrix}\text{lett: L} \\ \text{prev:} \\ \text{next:} \\ \text{wrd:}\end{bmatrix}$	$\begin{bmatrix}\text{lett: O} \\ \text{prev:} \\ \text{next:} \\ \text{wrd:}\end{bmatrix}$		

3.6.4 Result of the rule application 3.6.3

$$\begin{bmatrix}\text{lett: L} \\ \text{prev:} \\ \text{next: O} \\ \text{wrd: 1}\end{bmatrix} \begin{bmatrix}\text{lett: O} \\ \text{prev: L} \\ \text{next:} \\ \text{wrd: 1}\end{bmatrix}$$

3.6.5 Time-linear derivation connecting the letters of love



3.6.6 Proplets for LOVE, LOSS, and ALSO in a Word Bank

owner records	*member records*		
$\begin{bmatrix}\text{lett: A}\end{bmatrix}$			$\begin{bmatrix}\text{lett: A}\\ \text{prev:}\\ \text{next: L}\\ \text{wrd: 3}\end{bmatrix}$
$\begin{bmatrix}\text{lett:E}\end{bmatrix}$	$\begin{bmatrix}\text{lett: E}\\ \text{prev: V}\\ \text{next:}\\ \text{wrd: 1}\end{bmatrix}$		
$\begin{bmatrix}\text{lett:L}\end{bmatrix}$	$\begin{bmatrix}\text{lett: L}\\ \text{prev:}\\ \text{next: O}\\ \text{wrd: 1}\end{bmatrix}$	$\begin{bmatrix}\text{lett: L}\\ \text{prev:}\\ \text{next: O}\\ \text{wrd: 2}\end{bmatrix}$	$\begin{bmatrix}\text{lett: L}\\ \text{prev: A}\\ \text{next: S}\\ \text{wrd: 3}\end{bmatrix}$

owner records	*member*	*records*		
[lett: O]	lett: O prev: L next: V wrd: 1	lett: O prev: L next: S wrd: 2		lett: O prev: S next: wrd: 3
[lett: S]		lett: S prev: O next: S wrd: 2	lett: S prev: S next: wrd: 2	lett: S prev: L next: O wrd: 3
[lett: V]	lett: V prev: O next: E wrd: 1			

3.6.7 DEFINITION OF **LA-letter-OUT** FOR TRAVERSING CONNECTED PROPLETS

ST_S = {(lett: α, { r-out})}

r-out: $\begin{bmatrix}\text{lett: } \alpha \\ \text{next: } \beta \\ \text{wrd: k}\end{bmatrix}$ $\begin{bmatrix}\text{lett: } \beta \\ \text{prev: } \alpha \\ \text{wrd: k}\end{bmatrix}$ output position nw {r-out}

ST_F = {(lett: β, rp$_{r-out}$) }

3.6.8 Example of an **LA-letter-OUT** rule application

	rule name	*ss pattern*	*nw pattern*	*operations*	*rule package*
rule level	r-out:	$\begin{bmatrix}\text{lett: } \alpha \\ \text{next: } \beta \\ \text{wrn: k}\end{bmatrix}$	$\begin{bmatrix}\text{lett: } \beta \\ \text{prev: } \alpha \\ \text{wrn: k}\end{bmatrix}$	output position nw	{r-out}
proplet level		$\begin{bmatrix}\text{lett: L} \\ \text{prev: } \\ \text{next: O} \\ \text{wrn: 1}\end{bmatrix}$			

3.6.9 Result of the **LA-letter-OUT** rule application

$$\begin{bmatrix}\text{lett: L} \\ \text{prev: } \\ \text{next: O} \\ \text{wrn: 1}\end{bmatrix} \quad \begin{bmatrix}\text{lett: O} \\ \text{next: V} \\ \text{prev: L} \\ \text{wrn: 1}\end{bmatrix}$$

3.6.10 Extensions needed for natural language communication

1. *Automatic word form recognition and production*
   Instead of LA-letter-IN recognizing only elementary letters like L or O, a full system must recognize complex word forms in different languages, and similarly for LA-letter-OUT and word form production.

2. *Separation of navigation and language realization*
   Instead of LA-letter-OUT treating the core values of proplets in the database and the surface items of the output as the same, a full system must handle the functions of navigation and language realization separately by means LA-think and LA-speak, respectively (cf. 3.5.4). This requires a distinction between the core and the surface attributes of proplets (cf. 3.2.2).

3. *Distinction between language and context data*
   The distinction between language and context requires a division of the Word Bank into a context and a language area (compare 3.6.6 and 3.3.1). Reference to the external world requires that the input-output component of the language level is complemented with an input-output component at the context level (compare 2.5.1 and 2.4.1).

4. *Extending the navigation into a control structure*
   Instead of LA-letter-OUT merely following the continuations coded into the proplets, a full system must extend the navigation into a method of inferencing. This requires a distinction between absolute propositionsand episodic propositions (cf. Section 5.2). After complementing the agent with a value structure, the inferencing must be extended into a control structure.

4. Concept Types and Concept Tokens

4.1 Kinds of Proplets

4.1.1 Examples of the three main kinds of proplets

noun proplet	*verb proplet*	*adj proplet*
sur:	sur:	sur:
noun: book	verb: read	adj: blue
cat: sn	cat: decl	cat: adn
sem: def sg	sem: pres	sem:
mdr: blue	mdr:	mdr: B
fnc: buy	arg: John book	mdd: book 3
idy: 3	pc: 15 sit	idy: B
nc:	nc: 17 sleep	nc:
pc:	prn: 16	pc:
prn: 11		prn: 20

4.1.2 Proplet attributes and their values

1. Surface attribute: sur
   All proplets have a surface attribute. If it has the value NIL, the proplet is a context proplet. In language proplets, it gets its non-NIL value from the lexicon.

2. Core attributes: noun, verb, adj
   The core attribute of a proplet gets its unique value from the lexicon. From a sign-theoretic point of view,the core values may be a *concept*, a *pointer*, or a *marker*, which corresponds to the sign kinds of *symbol, indexical* and *name*.

3. Grammatical attributes: cat, sem
   The grammatical attributes cat (category) and sem (semantics) get their initial values from the lexicon; they may be modified during the derivation.

4. Intra-propositional continuation attributes: fnc, arg, mdd, mdr
   The intra-propositional continuation attributes get their value(s) by copying during the time-linear composition of proplets (cf. 3.4.2). The values consist of characters (char), which represent the names of other proplets. In complete propositions, the values of fnc (functor), arg (argument), and mdd (modified) must be non-NIL (obligatory continuation attributes), while that of mdr (modifier) may be NIL (optional continuation attribute).

5. Extra-propositional continuation attributes: nc, pc, idy
   The extra-propositional continuation attributes nc (next conjunct), pc (previous conjunct), and idy (identity) are used extra- and intra-propositionally. Their extra-propositional use is obligatory (in a text, at least one of these attributes must have a non-NIL value), while their intra-propositional use is optional. Like intra-propositional continuation attributes, they get their values by copying.

6. Book-keeping attribute: prn
   The book-keeping attribute prn (proposition number) gets its value from the control structure of the parser and consists of a number (integer). Additional book-keeping attributes are wrn (word number), and trc (transition counter), which serve in the implementation.

4.1.3 Substituting a replacement variable with a core value

semantic core	*proplet shell*	*lexical proplet*
apple	$\begin{bmatrix} \text{noun: RV.1} \\ \text{fnc:} \\ \text{mdr:} \\ \text{prn:} \end{bmatrix}$	$\begin{bmatrix} \text{noun: apple} \\ \text{fnc:} \\ \text{mdr:} \\ \text{prn:} \end{bmatrix}$

4.1.4 Relating proplet shells, lexical items, and rule patterns

$$
\begin{array}{lcll}
 & \nearrow & \textit{rule pattern} & \begin{bmatrix} \text{noun:} & \alpha \\ \text{fnc:} & \\ \text{mdr:} & \\ \text{prn:} & \end{bmatrix} \\
\textit{proplet shell} \; \begin{bmatrix} \text{noun:} & \text{RV.1} \\ \text{fnc:} & \\ \text{mdr:} & \\ \text{prn:} & \end{bmatrix} & & & \textit{compatible for matching} \\
 & \searrow & \textit{lexical item} & \begin{bmatrix} \text{noun:} & \text{apple} \\ \text{fnc:} & \\ \text{mdr:} & \\ \text{prn:} & \end{bmatrix}
\end{array}
$$

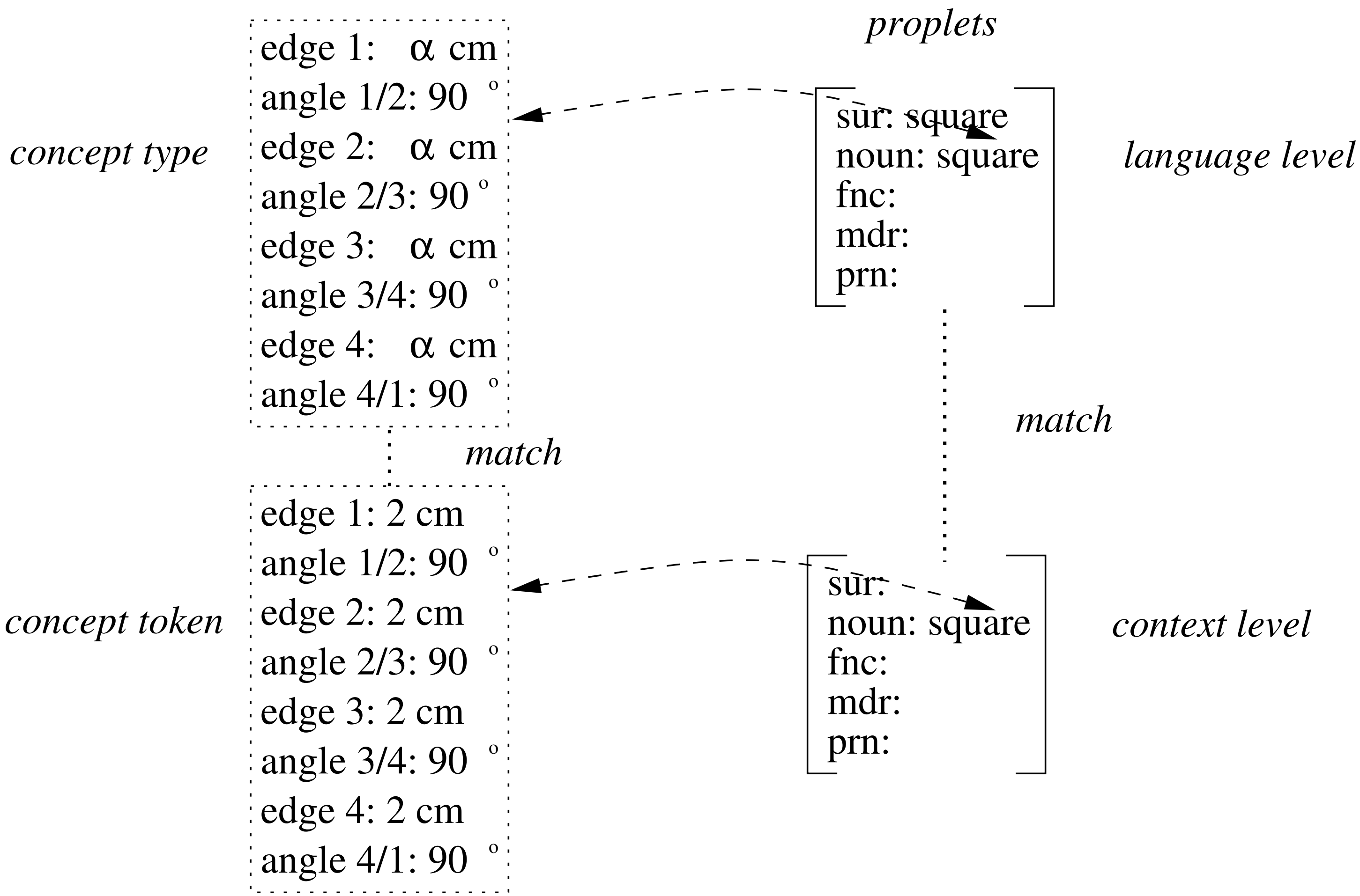
4.2 Type-Token Relation for Establishing Reference

4.2.1 TYPE AND TOKEN OF THE CONCEPT *square*

type *token*

$$\begin{bmatrix} \text{edge 1: } \alpha \\ \text{angle 1/2: } 90^o \\ \text{edge 2: } \alpha \\ \text{angle 2/3: } 90^o \\ \text{edge 3: } \alpha \\ \text{angle 3/4: } 90^o \\ \text{edge 4: } \alpha \\ \text{angle 4/1: } 90^o \end{bmatrix} \qquad \begin{bmatrix} \text{edge 1: 2cm} \\ \text{angle 1/2: } 90^o \\ \text{edge 2: 2cm} \\ \text{angle 2/3: } 90^o \\ \text{edge 3: 2cm} \\ \text{angle 3/4: } 90^o \\ \text{edge 4: 2cm} \\ \text{angle 4/1: } 90^o \end{bmatrix}$$

4.2.2 CONCEPTS AS VALUES OF THE CORE ATTRIBUTES

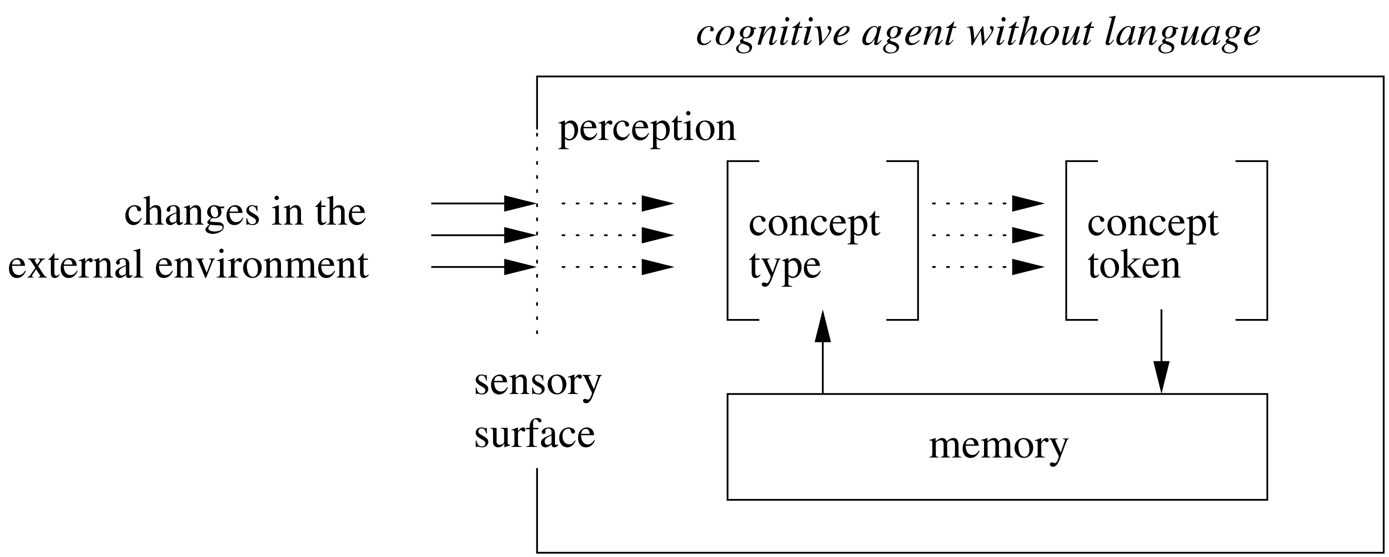


4.2.3 Why the type-token relation is important

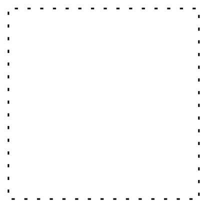
Type-token relations based on feature structures with variables and constants are easily computed. Procedures matching the concept type of a language proplet with the concept token of a context proplet enable the language proplet to refer in different utterance situations to different context proplets, including reference to items never encountered before.

4.3 Context Recognition

4.3.1 CONCEPT TYPE AND CONCEPT TOKEN IN CONTEXTUAL RECOGNITION



4.3.2 Concept type and concept token in recognizing a square



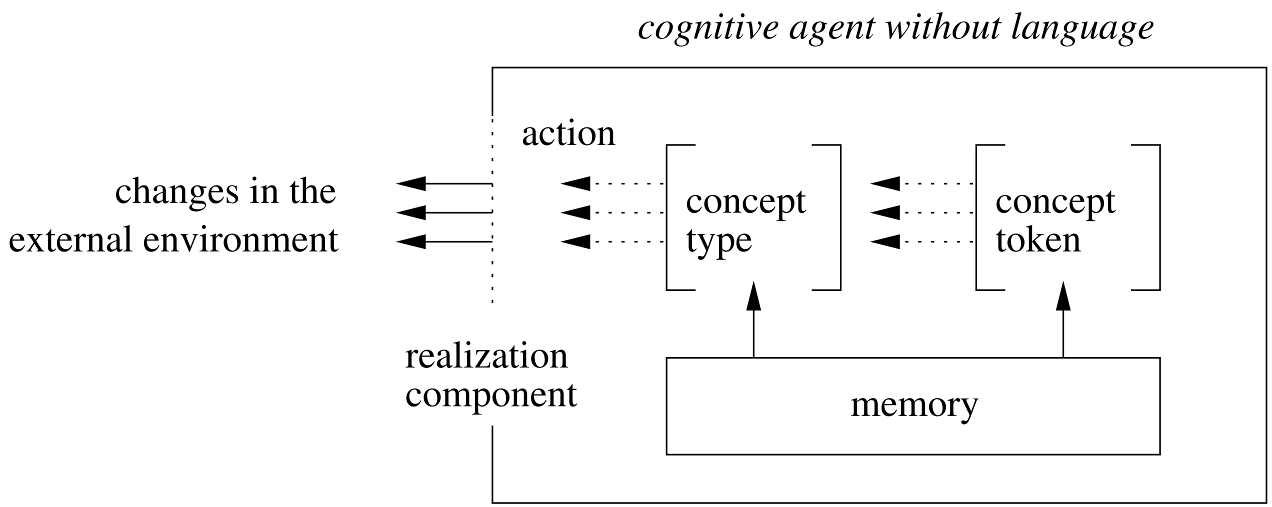
bitmap outline

concept type	concept token
edge 1: α cm	edge 1: 2 cm
angle 1/2: 90°	angle 1/2: 90°
edge 2: α cm	edge 2: 2 cm
angle 2/3: 90°	angle 2/3: 90°
edge 3: α cm	edge 3: 2 cm
angle 3/4: 90°	angle 3/4: 90°
edge 4: α cm	edge 4: 2 cm
angle 4/1: 90°	angle 4/1: 90°

concept type   *concept token*

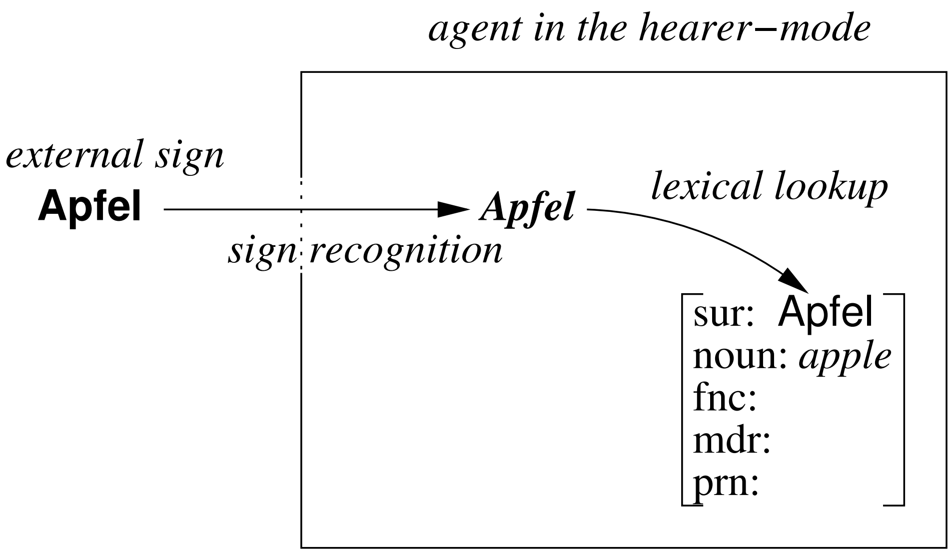
4.4 Context Action

4.4.1 Concept types and concept tokens in action

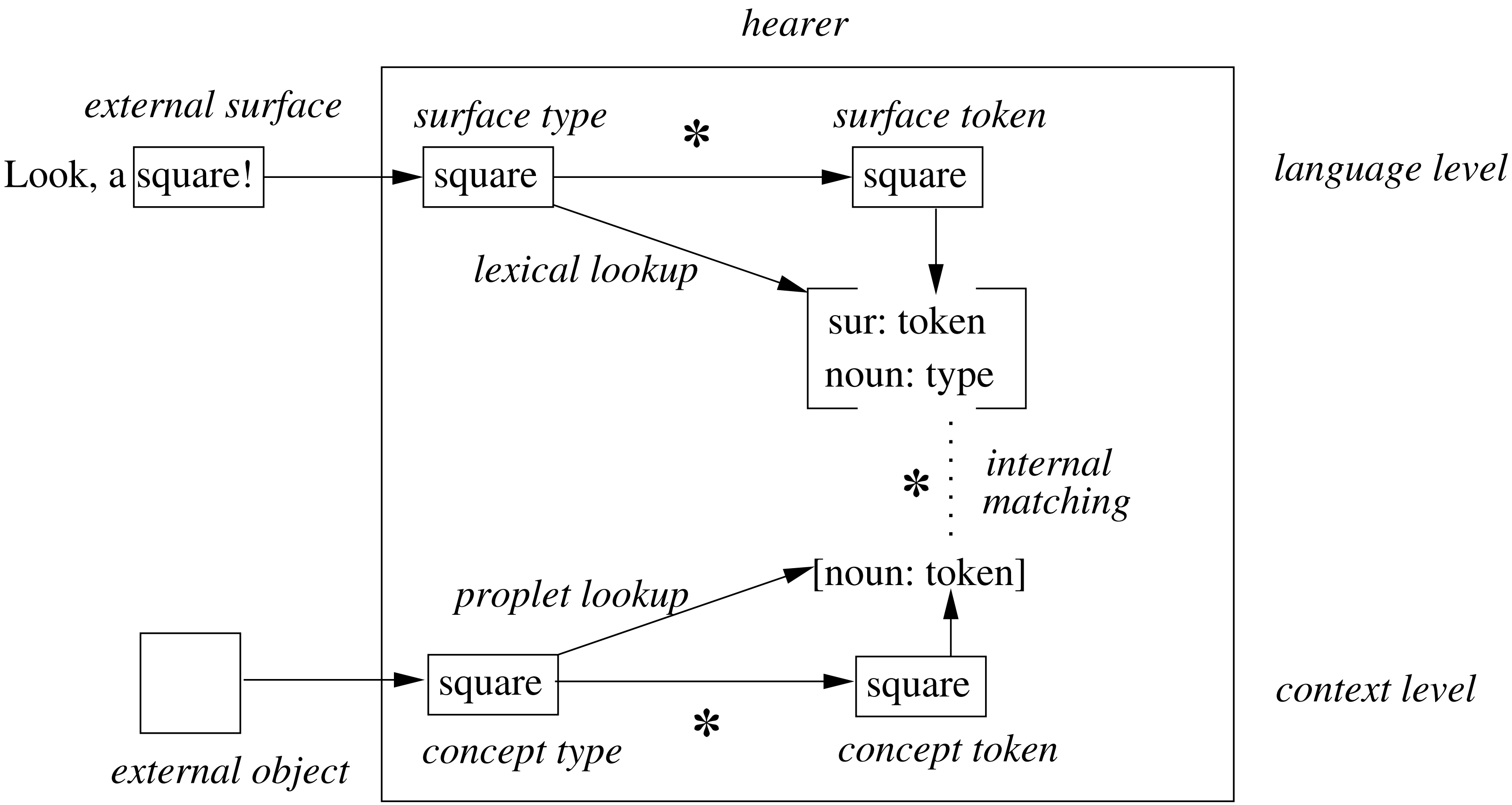


4.5 Sign Recognition and Production

4.5.1 LEXICAL LOOKUP OF GERMAN Apfel IN THE HEARER-MODE

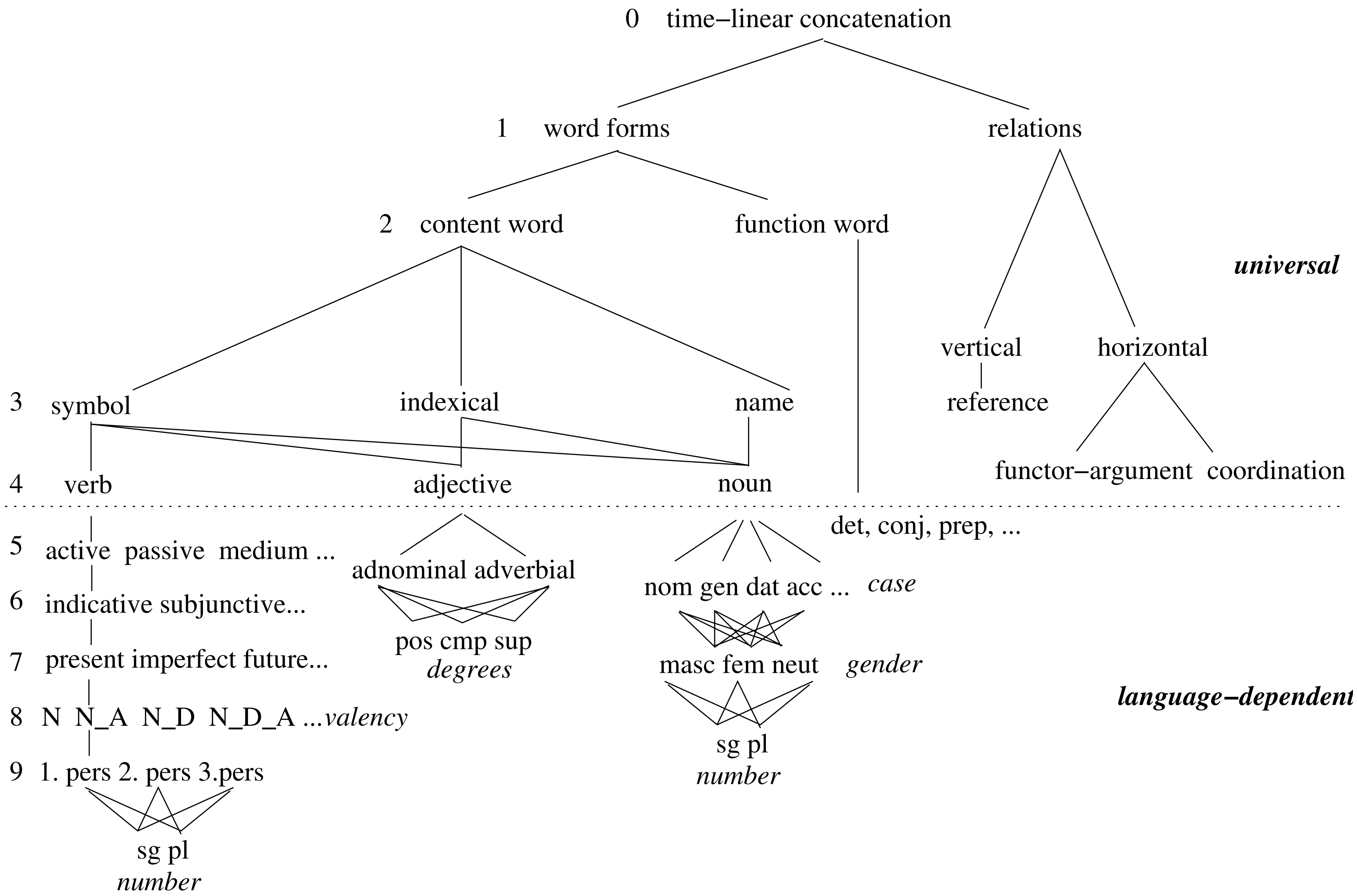


4.5.2 THREE TYPE-TOKEN RELATIONS IN THE CYCLE OF COMMUNICATION



4.6 Universal versus Language-Dependent Properties

4.6.1 Hierarchy of notions and distinctions in Database Semantics



5. Forms of Thinking

5.1 Retrieving Answers to Questions

5.1.1 EXAMPLE OF A TOKEN LINE

owner record	*member records*			
[noun: *girl*]	[noun: girl fnc: walk mdr: young prn: 10]	[noun: girl fnc: sleep mdr: blond prn: 12]	[noun: girl fnc: eat mdr: small prn: 15]	[noun: girl fnc: read mdr: smart prn: 19]

5.1.2 DEFINITION OF **LA-think.LINE**

ST_S: $\{([\mathrm{RA.1}{:}\ \alpha]\{1\ \mathrm{r}_{forwd}\ 2\ \mathrm{r}_{bckwd}\})\}$

$$\mathrm{r}_{forwd} \quad \begin{bmatrix}\mathrm{RA.1}{:}\ \alpha \\ \mathrm{prn}{:}\ n\end{bmatrix} \begin{bmatrix}\mathrm{RA.1}{:}\ \alpha \\ \mathrm{prn}{:}\ n{+}1\end{bmatrix} \quad \text{output position nw} \quad \{\mathrm{r}_{forwd}\}$$

$$\mathrm{r}_{bckwd} \quad \begin{bmatrix}\mathrm{RA.1}{:}\ \alpha \\ \mathrm{prn}{:}\ n\end{bmatrix} \begin{bmatrix}\mathrm{RA.1}{:}\ \alpha \\ \mathrm{prn}{:}\ n{-}1\end{bmatrix} \quad \text{output position nw} \quad \{\mathrm{r}_{bckwd}\}$$

ST_F: $\{([\mathrm{RA.1}{:}\ \alpha]\ \mathrm{rp}_{forwd}), ([\mathrm{RA.1}{:}\ \alpha]\ \mathrm{rp}_{bckwd})\}$

5.1.3 Example of an **LA-think.LINE** rule application

	rule name	*ss pattern*	*nw pattern*	*operations*	*rule package*
rule level	r_{forwd}:	$\begin{bmatrix}\text{RA.1: } \alpha \\ \text{prn: n}\end{bmatrix}$	$\begin{bmatrix}\text{RA.1: } \alpha \\ \text{prn: n+1}\end{bmatrix}$	output position nw	$\{\mathrm{r}_{forwd}\}$
proplet level		$\begin{bmatrix}\text{noun: girl} \\ \text{fnc: sleep} \\ \text{mdr: blonde} \\ \text{prn: 12}\end{bmatrix}$			

5.1.4 Result of the **LA-think.LINE** rule application

$$\begin{bmatrix}\text{noun: girl} \\ \text{fnc: sleep} \\ \text{mdr: blonde} \\ \text{prn: 12}\end{bmatrix} \begin{bmatrix}\text{noun: girl} \\ \text{fnc: eat} \\ \text{mdr: small} \\ \text{prn: 15}\end{bmatrix}$$

5.1.5 Basic kinds of query in natural language

wh-question	*yes/no-question*
Which girl walked?	Did the young girl walk?

5.1.6 Search proplets illustrating the two basic types of questions

wh-question

$$\begin{bmatrix} \text{noun:}\textit{girl} \\ \text{fnc: walk} \\ \text{mdr: } \sigma \\ \text{prn: } n \end{bmatrix}$$

yes/no-question

$$\begin{bmatrix} \text{noun:}\textit{girl} \\ \text{fnc: walk} \\ \text{mdr: young} \\ \text{prn: } n \end{bmatrix}$$

5.1.7 wh-search pattern checking a token line

$$\begin{bmatrix} \text{noun:}\textit{girl} \\ \text{fnc: walk} \\ \text{mdr: } \sigma \\ \text{prn: n} \end{bmatrix} \quad \textit{search pattern}$$

matching?

$$\begin{bmatrix} \text{noun: } \textit{girl} \end{bmatrix} \begin{bmatrix} \text{noun: girl} \\ \text{fnc: walk} \\ \text{mdr: young} \\ \text{prn: 10} \end{bmatrix} \begin{bmatrix} \text{noun: girl} \\ \text{fnc: sleep} \\ \text{mdr: blonde} \\ \text{prn: 12} \end{bmatrix} \begin{bmatrix} \text{noun: girl} \\ \text{fnc: eat} \\ \text{mdr: small} \\ \text{prn: 15} \end{bmatrix} \begin{bmatrix} \text{noun: girl} \\ \text{fnc: read} \\ \text{mdr: smart} \\ \text{prn: 19} \end{bmatrix} \quad \textit{token line}$$

5.1.8 Definition of LA-think.Q1 (*wh-question*)

$$\text{ST}_S\text{: }\{(\begin{bmatrix}\text{noun: RV.1}\\ \neg\text{ fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\end{bmatrix}\{\text{r}_1, \text{r}_2\}), (\begin{bmatrix}\text{noun: RV.1}\\ \text{fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\end{bmatrix}\{\ \}) \}$$

$$\text{r}_1\text{: }\begin{bmatrix}\text{noun: RV.1}\\ \neg\text{ fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\end{bmatrix}\begin{bmatrix}\text{noun: RV.1}\\ \neg\text{ fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\text{-}1\end{bmatrix}\text{ output position nw }\{\text{r}_1\ \text{r}_2\}$$

$$\text{r}_2\text{: }\begin{bmatrix}\text{noun: RV.1}\\ \neg\text{ fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\end{bmatrix}\begin{bmatrix}\text{noun: RV.1}\\ \text{fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\text{-}1\end{bmatrix}\text{ output position nw }\{\ \}$$

$$\text{ST}_F\text{: }\{(\begin{bmatrix}\text{noun: RV.1}\\ \neg\text{ fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\text{-}1\end{bmatrix}\text{rp}_1), (\begin{bmatrix}\text{noun: RV.1}\\ \text{fnc: RV.2}\\ \text{mdr: }\sigma\\ \text{prn: }n\text{-}1\end{bmatrix}\text{rp}_2) \}$$

5.1.9 DEFINITION OF **LA-think.Q2** (*yes/no-question*)

$$\mathrm{ST}_S\text{: } \{(\begin{bmatrix} \text{noun: RV.1} \\ \neg\ \text{fnc: RV.2} \\ \neg\ \text{mdr: RV.3} \\ \text{prn: } n \end{bmatrix} \{\mathrm{r}_1, \mathrm{r}_2\}),\ (\begin{bmatrix} \text{noun: RV.1} \\ \text{fnc: RV.2} \\ \text{mdr: RV.3} \\ \text{prn: } n \end{bmatrix} \{\ \})\ \}$$

$$\mathrm{r}_1\text{: } \begin{bmatrix} \text{noun: RV.1} \\ \neg\ \text{fnc: RV.2} \\ \neg\ \text{mdr: RV.3} \\ \text{prn: } n \end{bmatrix} \begin{bmatrix} \text{noun: RV.1} \\ \neg\ \text{fnc: RV.2} \\ \neg\ \text{mdr: RV.3} \\ \text{prn: } n\text{-}1 \end{bmatrix} \quad \text{output position nw} \ \ (\{\mathrm{r}_1\ \mathrm{r}_2\})$$

$$\mathrm{r}_2\text{: } \begin{bmatrix} \text{noun: RV.1} \\ \neg\ \text{fnc: RV.2} \\ \neg\ \text{mdr: RV.3} \\ \text{prn: } n \end{bmatrix} \begin{bmatrix} \text{noun: RV.1} \\ \text{fnc: RV.2} \\ \text{mdr: RV.3} \\ \text{prn: } n\text{-}1 \end{bmatrix} \quad \text{output position nw} \ \ (\{\ \})$$

$$\mathrm{ST}_F\text{: } \{(\begin{bmatrix} \text{noun: RV.1} \\ \neg\ \text{fnc: RV.2} \\ \neg\ \text{mdr: RV.3} \\ \text{prn: } n\text{-}1 \end{bmatrix} \mathrm{rp}_1),\ (\begin{bmatrix} \text{noun: RV.1} \\ \text{fnc: RV.2} \\ \text{mdr: RV.3} \\ \text{prn: } n\text{-}1 \end{bmatrix} \mathrm{rp}_2)\ \}$$

5.2 Episodic versus Absolute Propositions

5.2.1 Absolute and episodic proplets in a Word Bank (context level)

$$
\begin{array}{llll}
\textit{owner records} & \textit{absolute proplets} & & \textit{episodic proplets} \\
\begin{bmatrix}\text{noun: animal}\end{bmatrix} & \ldots \begin{bmatrix}\text{noun: animal}\\ \text{fnc: is}\\ \text{mdr:}\\ \text{prn: a-11}\end{bmatrix} & & \\
\begin{bmatrix}\text{verb: be}\end{bmatrix} & \ldots \begin{bmatrix}\text{verb: be}\\ \text{arg: dog animal}\\ \text{mdr:}\\ \text{prn: a-11}\end{bmatrix} & \ldots & \begin{bmatrix}\text{verb: be}\\ \text{arg: dog}\\ \text{mdr: tired}\\ \text{prn: e-23}\end{bmatrix} \\
\begin{bmatrix}\text{noun: dog}\end{bmatrix} & \ldots \begin{bmatrix}\text{noun: dog}\\ \text{fnc: be}\\ \text{mdr:}\\ \text{prn: a-11}\end{bmatrix} & \ldots & \begin{bmatrix}\text{noun: dog}\\ \text{fnc: be}\\ \text{mdr:}\\ \text{prn: e-23}\end{bmatrix} \\
\begin{bmatrix}\text{adj: tired}\end{bmatrix} & \ldots & \ldots & \begin{bmatrix}\text{adj: tired}\\ \text{mdd: be}\\ \text{mdr:}\\ \text{prn: e-23}\end{bmatrix}
\end{array}
$$

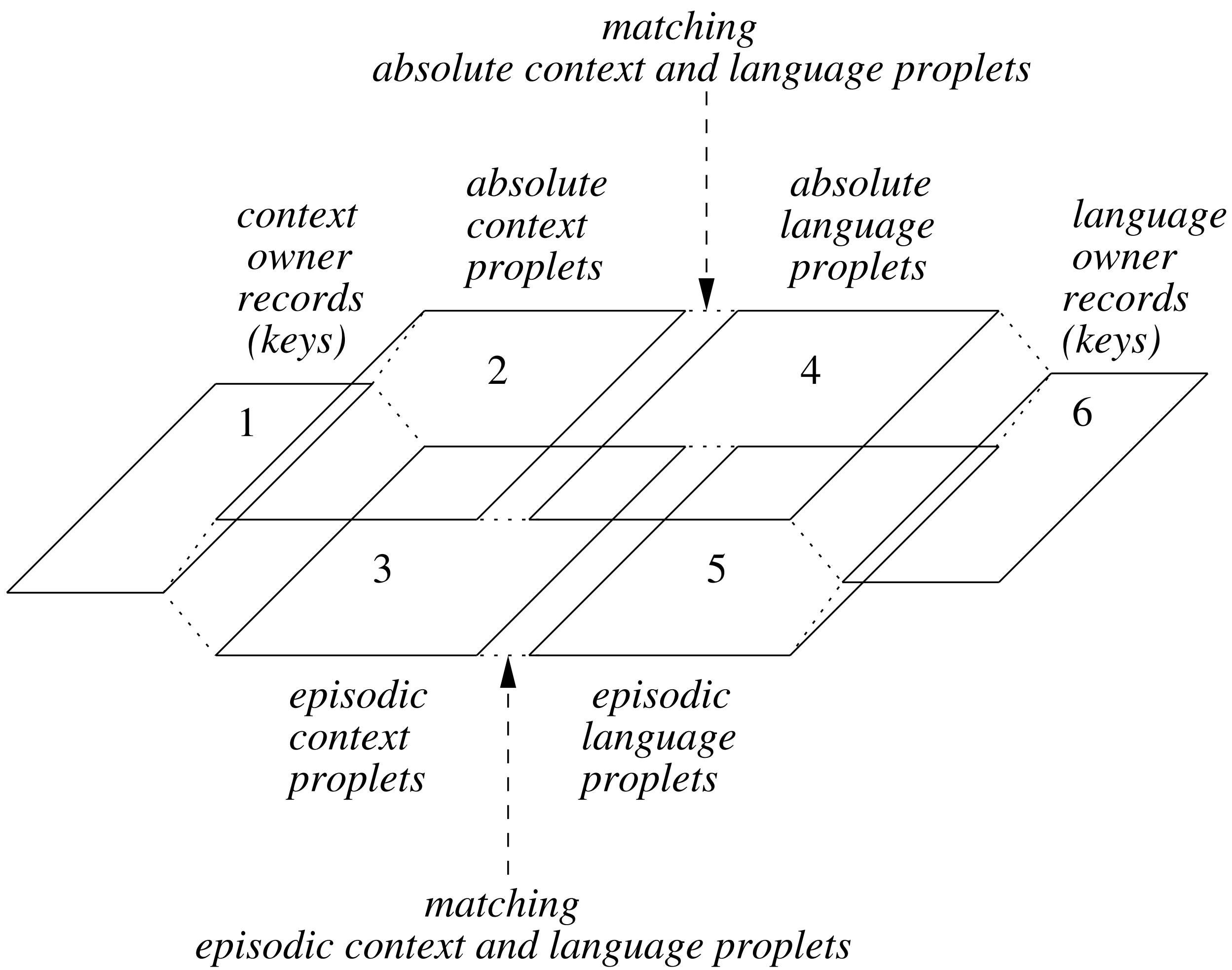
5.2.2 Core values as concept types and concept tokens

	type	*type*	*token*
token line:	[noun: dog]	[noun: dog fnc: be mdr: prn: a–11]	[noun: dog fnc: be mdr: prn: e–23]
	owner record (key)	*absolute proplet*	*episodic proplet*

5.2.3 Areas of episodic vs. absolute proplets in a Word Bank

context owner records	*absolute context proplets*	*episodic context proplets*	*episodic language proplets*	*absolute language proplets*	*language owner records*
1	2	3	4	5	6

5.2.4 3-dimensional representation indicating matching frontiers



5.2.5 From episodic propositions to an absolute proposition

Mary takes a nap after lunch. *absolute proposition*

$\Uparrow$

(Friday) Mary takes a nap after lunch.

(Thursday) Mary takes a nap after lunch.

(Wednsday) Mary takes a nap after lunch. *episodic propositions*

(Tuesday) Mary takes a nap after lunch.

(Monday) Mary takes a nap after lunch.

5.3 Inference: Reconstructing *Modus Ponens*

5.3.1 Inference schemata of Propositional Calculus

1. $\dfrac{A, B}{\vdash\ A \& B}$
2. $\dfrac{A \vee B, \neg A}{\vdash\ B}$
3. $\dfrac{A \rightarrow B, A}{\vdash\ B}$
4. $\dfrac{A \rightarrow B, \neg B}{\vdash\ \neg A}$
5. $\dfrac{A \& B}{\vdash\ A}$
6. $\dfrac{A}{\vdash\ A \vee B}$
7. $\dfrac{\neg A}{\vdash\ A \rightarrow B}$
8. $\dfrac{\neg\neg A}{\vdash\ A}$

5.3.2 LA-rule for the inference of *conjunction* (hypothetical)

$$\text{inf}_1\text{:}\quad \begin{bmatrix}\text{verb: } \alpha \\ \text{prn: } m\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{prn: } n\end{bmatrix} \Longrightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{prn: } m \\ \text{cnj: } m \textit{ and } n\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{prn: } n \\ \text{cnj: } m \textit{ and } n\end{bmatrix}$$

5.3.3 *modus ponens* in Propositional Calculus

Premise 1:	If the sun is shining, Mary takes a walk.	$A \rightarrow B$
Premise 2:	The sun is shining.	$\underline{A \qquad}$
Conclusion:	Mary takes a walk.	$\vdash B$

5.3.4 *modus ponens* in Predicate Calculus

$$\frac{\forall x[\mathsf{f}(x) \rightarrow \mathsf{g}(x)],\ \exists x[\mathsf{f}(x)\ \&\ \mathsf{h}(x)]}{\vdash \exists x[\mathsf{g}(x)\ \&\ \mathsf{h}(x)]}$$

Premise 1:	For all x, if x is a dog, then x is an animal.	$\forall$x[dog(x) $\rightarrow$ animal(x)]
Premise 2:	There exists an x, x is a dog and x is tired.	$\exists$x[dog(x) & tired(x)]
Conclusion:	There exists an x, x is an animal and x is tired.	$\vdash$ $\exists$x[animal(x)&tired(x)]

5.3.5 Inference rule inf$_2$ for reconstructing *modus ponens*

$$\text{inf}_2\text{:}\ \begin{bmatrix}\text{verb: be}\\ \text{arg: } \alpha\\ \text{mdr: } \beta\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: be}\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{adj: } \beta\\ \text{mdd: be}\\ \text{mdr:}\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{verb: is-a}\\ \text{arg: } \delta\ \gamma\\ \text{prn: } \textit{a-m}\end{bmatrix} \begin{array}{l}\text{if } \alpha \text{ instantiates } \delta,\\ \text{replace } \alpha \text{ with } \gamma\\ \text{and replace } \textit{e-n} \text{ with } \textit{e-n'}\end{array} \quad \{\ \ldots\ \}$$

5.3.6 Abstract Paraphrase

1. Absolute premise: noun type δ is-a noun type γ.
2. Episodic premise: noun token α happens to be adjective β.
3. Episodic conclusion: If noun token α instantiates noun type δ, then noun token γ happens to be adjective β.

5.3.7 Concrete example based on the abstract paraphrase

1. Absolut premise: *dog* type is-a(n) *animal* type.
2. Episodic premise: *dog* token happens to be *tired*.
3. Episodic conclusion: If *dog* token instantiates *dog* type, then *animal* token happens to be *tired*

5.3.8 Applying inf_2 to the Word Bank 5.2.1

$$\text{inf}_2\text{:} \begin{bmatrix}\text{verb: be}\\ \text{arg: } \alpha\\ \text{mdr: } \beta\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: be}\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{adj: } \beta\\ \text{mdd: be}\\ \text{mdr:}\\ \text{prn: } \textit{e-n}\end{bmatrix} \begin{bmatrix}\text{verb: is-a}\\ \text{arg: } \delta\ \gamma\\ \text{prn: } \textit{a-m}\end{bmatrix} \begin{array}{l}\text{if } \alpha \text{ instantiates } \delta,\\ \text{replace } \alpha \text{ with } \gamma\\ \text{and replace } \textit{e-n} \text{ with } \textit{e-n}\text{'}\end{array} \quad \{\ldots\}$$

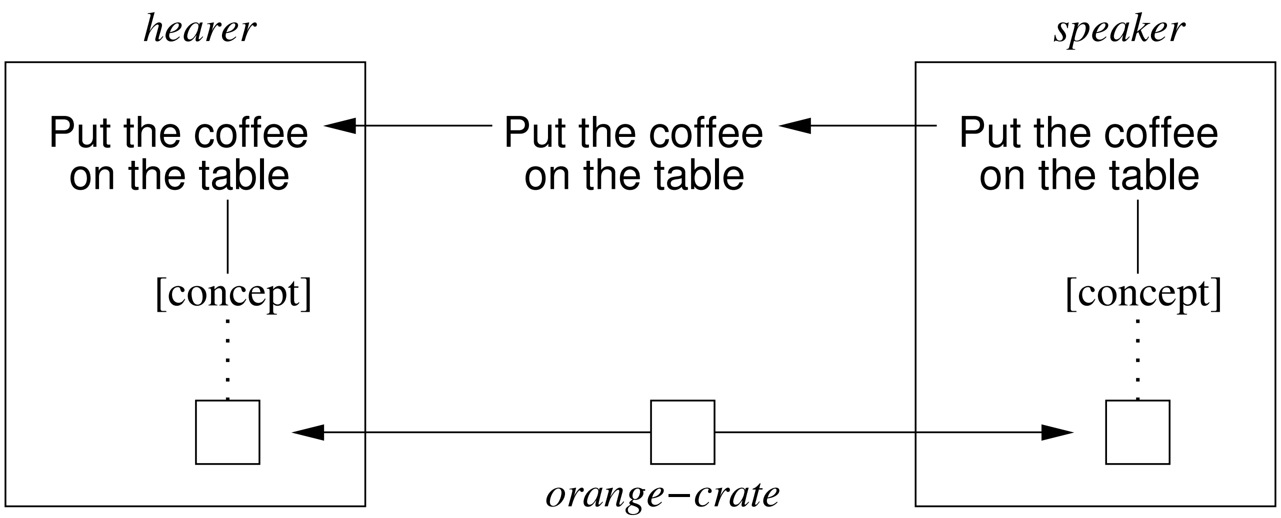
$$\begin{bmatrix}\text{verb: be}\\ \text{arg: dog}\\ \text{mdr: tired}\\ \text{prn: e-23}\end{bmatrix} \begin{bmatrix}\text{noun: dog}\\ \text{fnc: be}\\ \text{mdr:}\\ \text{prn: e-23}\end{bmatrix} \begin{bmatrix}\text{adj: tired}\\ \text{mdd: be}\\ \text{mdr:}\\ \text{prn: e-23}\end{bmatrix} \begin{bmatrix}\text{verb: is-a}\\ \text{arg: dog animal}\\ \text{mdr:}\\ \text{prn: a-11}\end{bmatrix}$$

5.3.9 Result of applying inf_2 to the Word Bank 5.2.1

$$\begin{bmatrix}\text{verb: be}\\ \text{arg: animal}\\ \text{mdr: tired}\\ \text{prn: e-23'}\end{bmatrix} \begin{bmatrix}\text{noun: animal}\\ \text{fnc: be}\\ \text{mdr:}\\ \text{prn: e-23'}\end{bmatrix} \begin{bmatrix}\text{adj: tired}\\ \text{mdd: be}\\ \text{mdr:}\\ \text{prn: e-23'}\end{bmatrix}$$

5.4 Indirect Uses of Language

5.4.1 NON-LITERAL USE OF THE WORD table

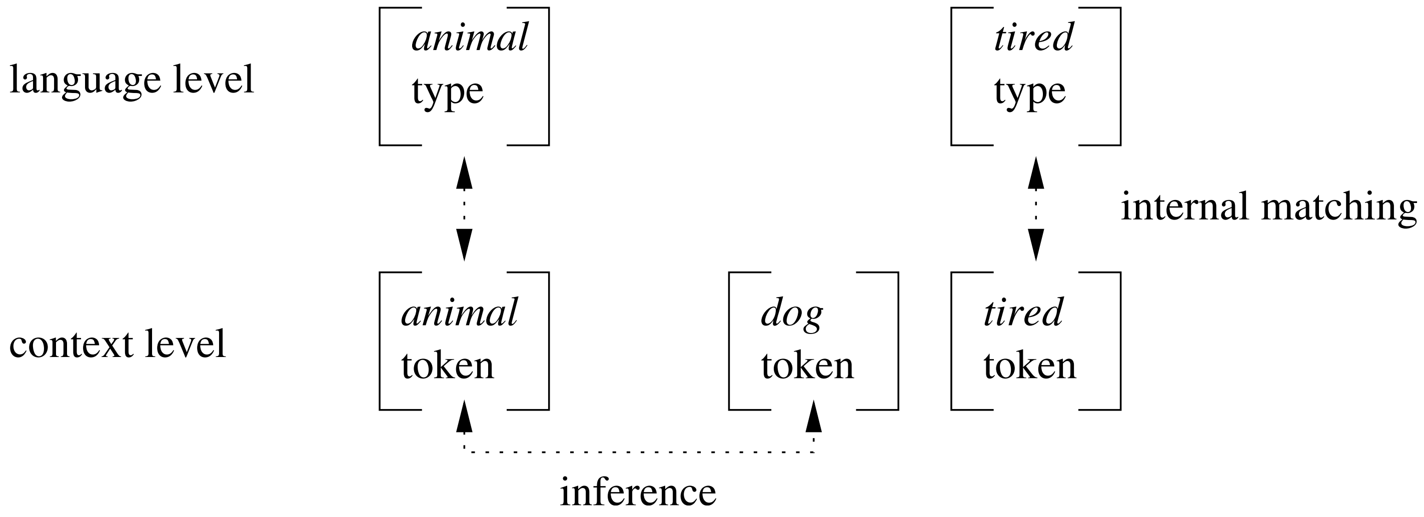


5.4.2 Grice's definition of meaning

Definiendum: U meant something by uttering x.

Definiens: For some audience A, U intends his utterance of x to produce in A some effect (response) E, by means of A's recognition of the intention.

5.4.3 CONTEXTUAL INFERENCE UNDERLYING A NON-LITERAL USE

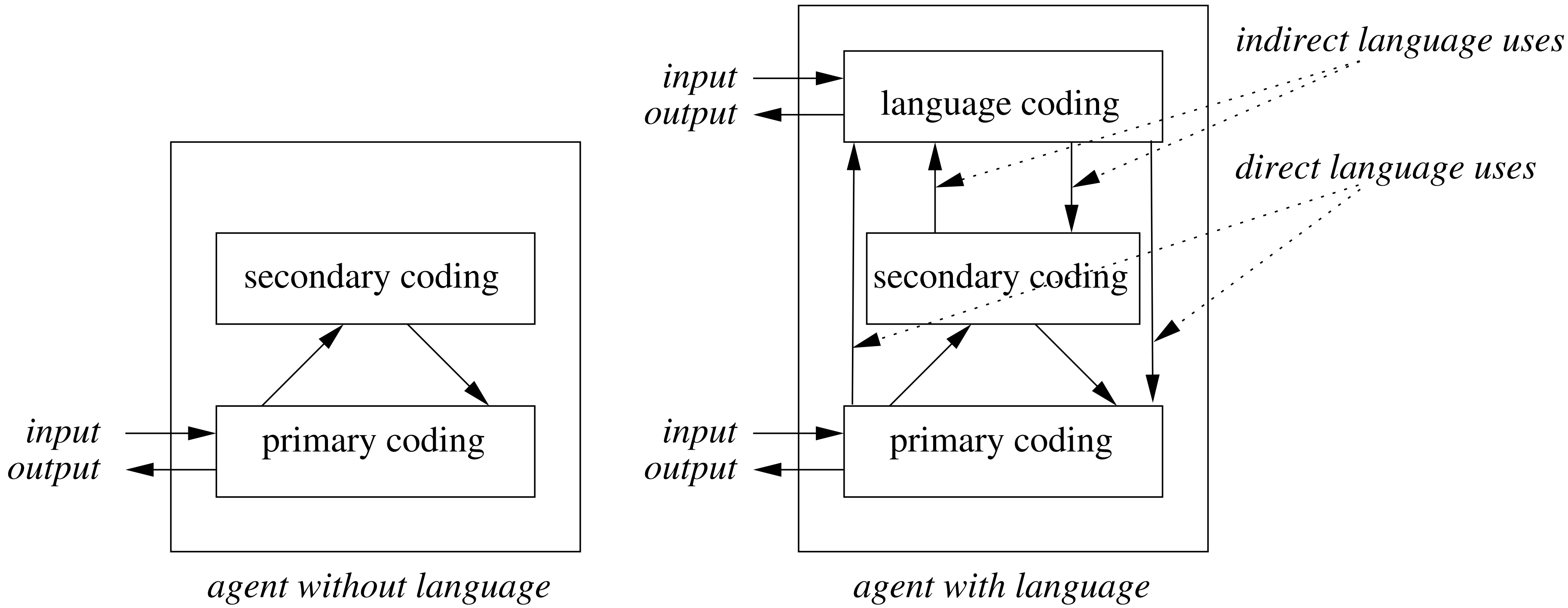


5.4.4 ADVANTAGES OF HANDLING INDIRECT USES VIA INFERENCES

1. Direct and indirect uses of language are based on the same method of strict internal matching (cf. 3.2.3), which greatly facilitates computational realization.
2. The inferencing underlying indirect uses is restricted to the level of context. Therefore, agents with and without language can use the same cognitive system.
3. Inferencing at the level of context is much more powerful and flexible than the traditional inferencing based on isolated signs of language.
4. Assuming that natural language directly reflects the contextual coding, the contextual inferences can be studied by analyzing their language reflections.

5.5 Secondary Coding as Perspective Taking

5.5.1 CODING LEVELS IN AGENTS WITH AND WITHOUT LANGUAGE



5.5.2 Functions of Coding Levels in Database Semantics

1. *Primary coding* at the context level
   Represents contextual recognition and action at a low level of abstraction in a simple standardized format in order to ensure veracity.
2. *Secondary coding* at the context level
   Consists of inferencing over the primary coding in order to obtain sufficient expressive power at varying levels of abstraction.
3. *Language coding*
   Represents primary and secondary context coding in a natural language.

5.6 Shades of Meaning

5.6.1 Lexical lookup of the word dog

$$\begin{bmatrix} \text{sur: dog} \\ \text{noun: } \textit{dog} \\ \text{fnc:} \\ \text{mdr:} \\ \text{prn:} \end{bmatrix}$$

5.6.2 Absolute part of the token line of *dog*

$$\begin{bmatrix} \text{noun: } \textit{dog} \end{bmatrix} \begin{bmatrix} \text{noun: dog} \\ \text{fnc: be} \\ \text{mdr:} \\ \text{prn: a-1} \end{bmatrix} \begin{bmatrix} \text{noun: dog} \\ \text{fnc: have} \\ \text{mdr:} \\ \text{prn: a-2} \end{bmatrix} \ldots$$

5.6.3 Complementation of a literal meaning$_1$ (concept type)