3. Data Structure and Algorithm

3.1 Proplets for Coding Propositional Content

3.1.1 CONTEXT PROPLETS REPRESENTING *dog barks. (I) run.*

$$\begin{bmatrix}\text{sur:}\\ \text{noun: } \textit{dog}\\ \text{fnc: bark}\\ \text{prn: 22}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: } \textit{bark}\\ \text{arg: dog}\\ \text{nc: 23 run}\\ \text{prn: 22}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: } \textit{run}\\ \text{arg: moi}\\ \text{pc: 22 bark}\\ \text{prn: 23}\end{bmatrix}$$

3.1.2 CODING OF RELATIONS BETWEEN CONCEPTS VIA PROPLETS

$$\textit{context level:}\quad \begin{bmatrix}\text{sur:}\\ \text{noun: } \textit{dog}\\ \text{fnc: bark}\\ \text{prn: 22}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: } \textit{bark}\\ \text{arg: dog}\\ \text{nc: 23 run}\\ \text{prn: 22}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: } \textit{run}\\ \text{arg: moi}\\ \text{pc: 22 bark}\\ \text{prn: 23}\end{bmatrix}$$

3.2 Internal Matching between Language and Context Proplets

3.2.1 Language proplets representing *dog barks. (I) run.*

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

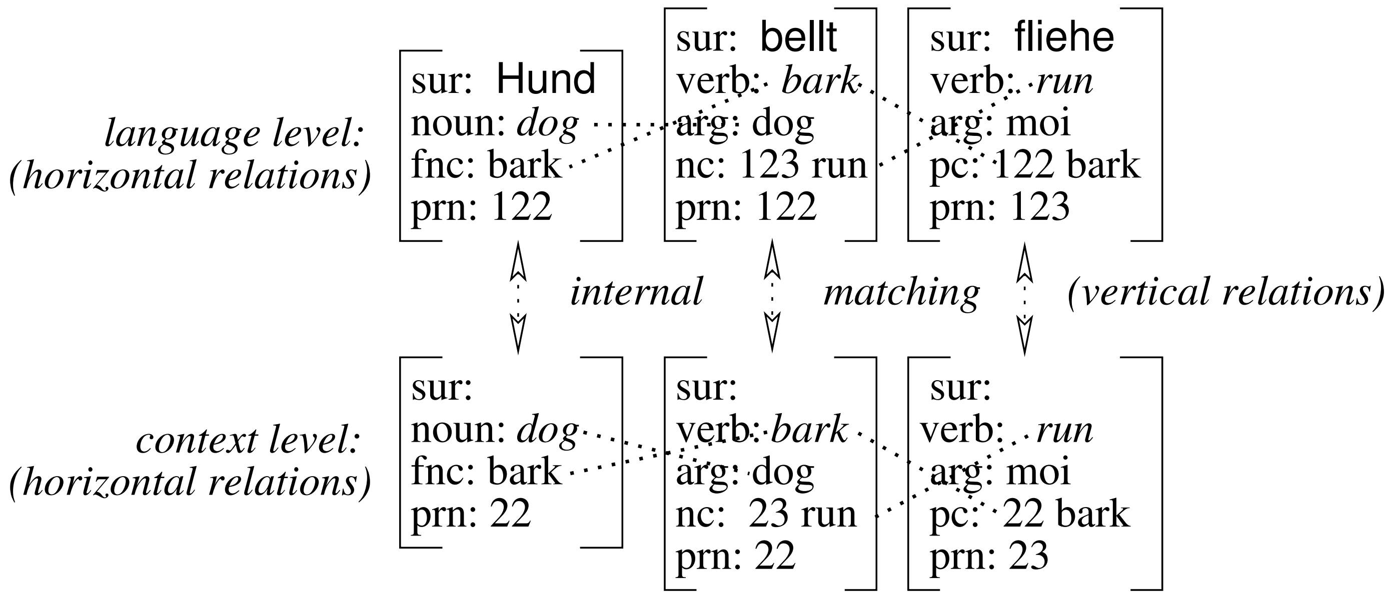
3.2.2 Keys for lexical lookup in the speaker- and the hearer-mode

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

3.2.3 Conditions on successful matching

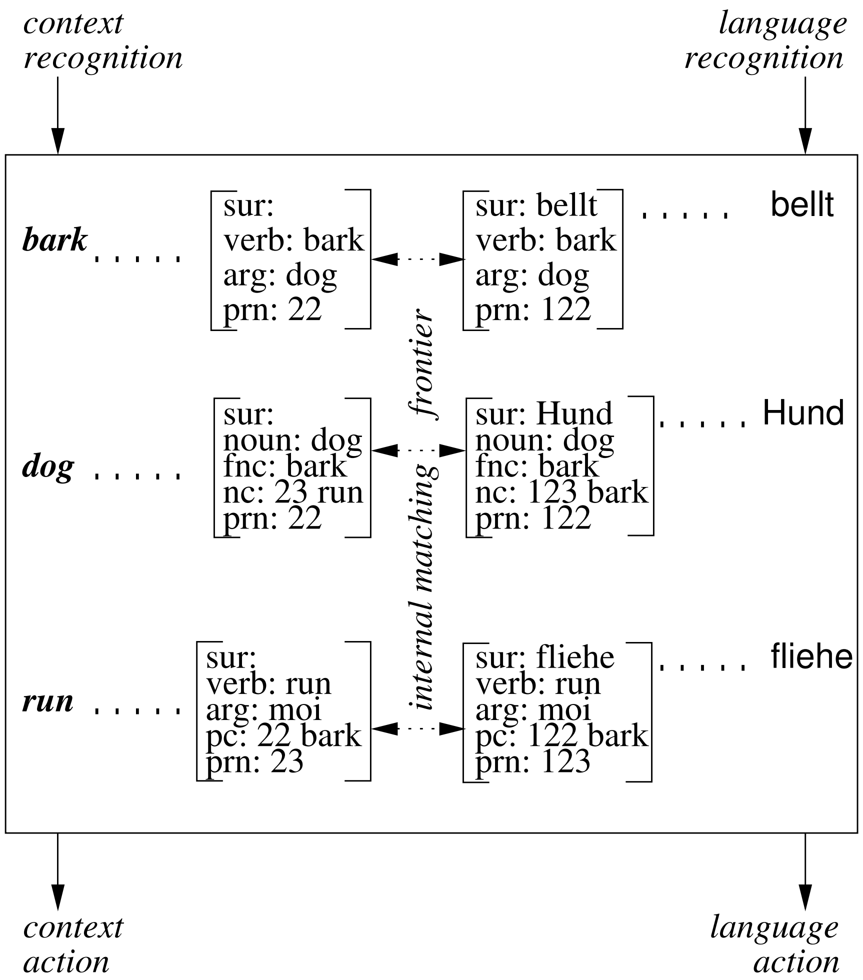
1. *Attribute condition*
   The matching between two proplets A and B requires that the intersection of their attributes contains a predefined list of attributes regarded as relevant:
   $$\{\text{list}\} \subseteq \{\{\text{proplet-A-attributes}\} \cap \{\text{proplet-B-attributes}\}\}$$
2. *Value condition*
   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)

	Julia	knows	John

lexical lookup

noun: Julia	verb: know	noun: John
fnc:	arg:	fnc:
mdr:	mdr:	mdr:
prn:	prn:	prn:

syntactic−semantic parsing:

1

noun: Julia	verb: know
fnc:	arg:
mdr:	mdr:
prn: 22	prn:

2

noun: Julia	verb: know	noun: John
fnc: know	arg: Julia	fnc:
mdr:	mdr:	mdr:
prn: 22	prn: 22	prn:

result of syntactic−semantic parsing:

noun: Julia	verb: know	noun: John
fnc: know	arg: Julia John	fnc: know
mdr:	mdr:	mdr:
prn: 22	prn: 22	prn: 22

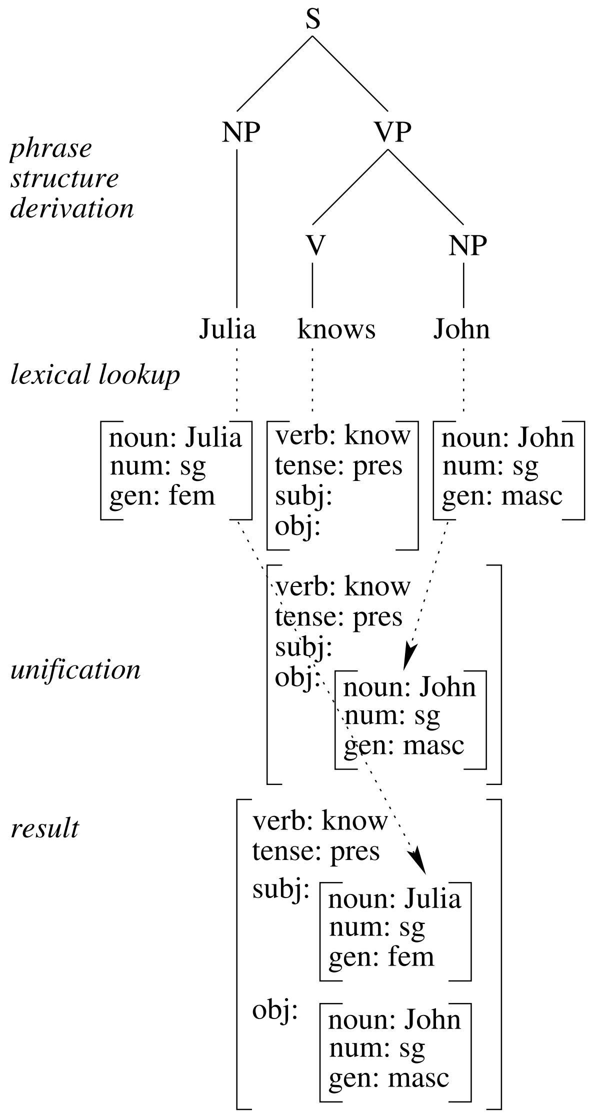
3.4.3 Example of an **LA-hear** rule application

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

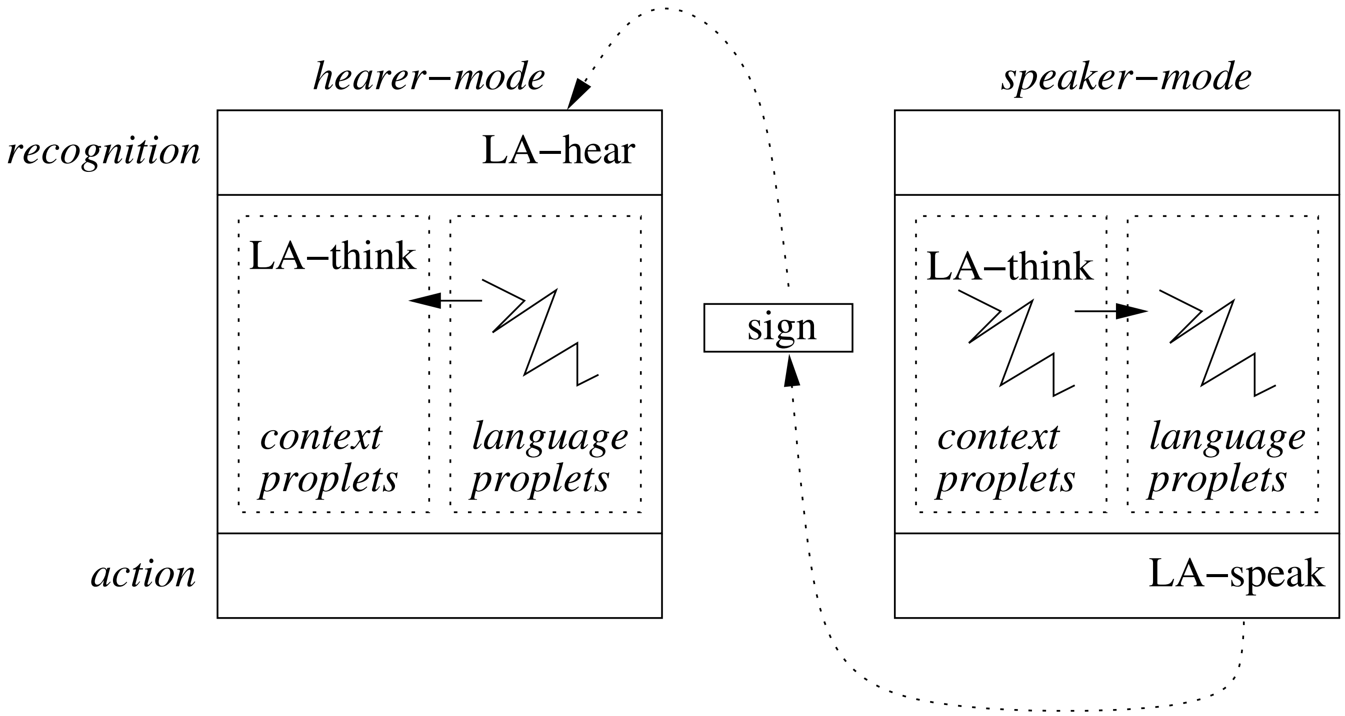
3.5.2 Result of the **LA-think** rule application

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

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

$\text{ST\_}S = \{(\text{lett: } \alpha, \{\text{r-in}\})\}$

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

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

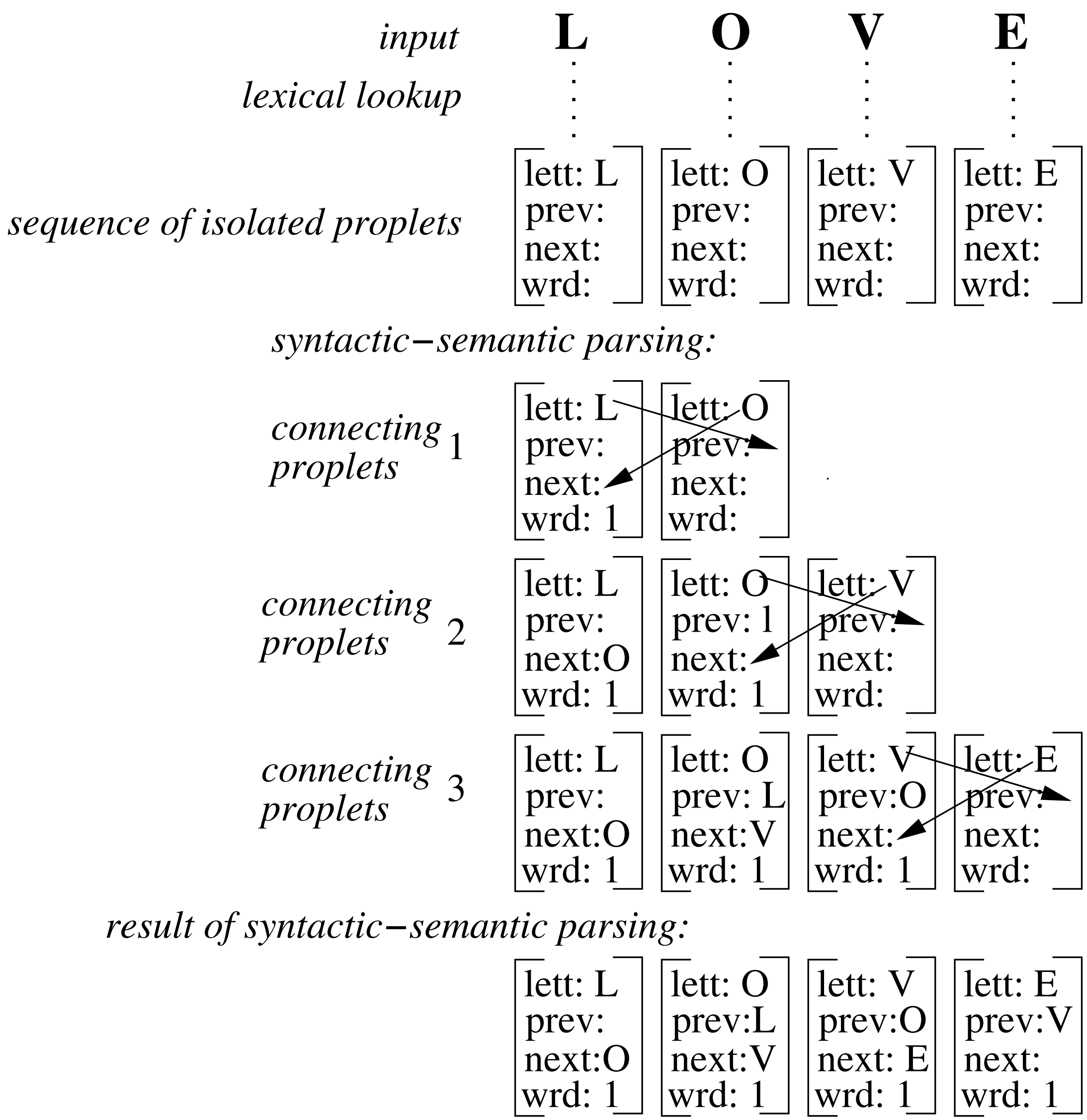
3.6.3 Example of an LA-letter-IN rule application

$$
\begin{array}{llllll}
 & \textit{rule name} & \textit{ss pattern} & \textit{nw pattern} & \textit{operations} & \textit{rule package} \\
\textit{rule level} & \text{r-in:} & \begin{bmatrix}\text{lett: } \alpha \\ \text{prev:} \\ \text{next:}\end{bmatrix} & \begin{bmatrix}\text{lett: } \beta \\ \text{prev:} \\ \text{next:}\end{bmatrix} & \begin{array}{l}\text{copy } \alpha \text{ nw.prev} \\ \text{copy } \beta \text{ ss.next}\end{array} & \{\text{r-in}\} \\
\textit{proplet level} & & \begin{bmatrix}\text{lett: L} \\ \text{prev:} \\ \text{next:} \\ \text{wrd:}\end{bmatrix} & \begin{bmatrix}\text{lett: O} \\ \text{prev:} \\ \text{next:} \\ \text{wrd:}\end{bmatrix} & &
\end{array}
$$

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

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.

