# 5. Forms of Thinking

Section 5.1 shows that the DBS data structure is well-suited for question answering. Furthermore, the data structure is easily extended[1] to the distinction between episodic and absolute propositions (Sect. 5.2), which in turn is the basis for implementing inferences, here *modus ponens* (Sect. 5.3). This inference is employed for implementing an indirect use of language (Sect. 5.4). Finally, it is shown how the intuitive meaning of a word is composed of several ingredients, some of which are derived from the agent's personal experiences and socio-cultural background (Sects. 5.5 and 5.6).

## 5.1 Retrieving Answers to Questions

So far, the data structure of a Word Bank, consisting of ordered token lines listing connected proplets with the same core value (cf. 3.3.1, 3.6.6), has been used (i) for the storage in the hearer mode and (ii) for the retrieval of successor proplets in the most basic kind of the think mode, namely free association based on fixed patterns, serving as an example of the speaker's conceptualization. We turn now to another kind of thinking supported by this data structure, namely (iii) moving along a token line.

Consider an agent thinking about girls. This means activating the corresponding token line, such as the following example:

### 5.1.1 Example of a token line

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

As indicated by the fnc and mdr values of the connected proplets (member records), the agent happened to observe or hear about a young girl walking, a blonde girl sleeping, a small girl eating, and a smart girl reading.

Traversing the token line of any given concept is powered by the following LA-think grammar, called LA-think.LINE:

[1] That a framework developed for certain purposes turns out to be suitable for other purposes as well is a kind of convergence. Convergence is one of most significant indicators in science for being on the right track (cf. FoCL'99, Sect. 9.5).

### 5.1.2 Definition of **LA-think.LINE**

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

$r_{forwd}$ [RA.1: $\alpha$ / prn: $n$] [RA.1: $\alpha$ / prn: $n+1$] output position nw {$r_{forwd}$}

$r_{bckwd}$ [RA.1: $\alpha$ / prn: $n$] [RA.1: $\alpha$ / prn: $n-1$] output position nw {$r_{bckwd}$}

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

The core attribute is represented by the replacement variable RA.1 (cf. 4.1.3, C.3.2), which is restricted to the attributes noun, verb, or adj. The prn values *n, n+1* and *n–1* follow the convention that *n+1* stands for the proplet immediately following the current proplet *n* in the token line, and *n–1* for the proplet immediately preceding. The rule $r_{forwd}$ traverses a token line from left to right (forward), following the temporal order, while the rule $r_{bckwd}$ moves right to left in the antitemporal order (backward).

Consider the application of $r_{forwd}$ to the proplet with the prn value 12 in 5.1.1:

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

| | *rule name* | *ss-pattern* | *nw-pattern* | *operations* | *rule package* |
|---|---|---|---|---|---|
| *rule level* | $r_{forwd}$: | [RA.1: $\alpha$ / prn: n] | [RA.1: $\alpha$ / prn: n+1] | output position nw | {$r_{forwd}$} |
| *proplet level* | | [noun: girl / fnc: sleep / mdr: blonde / prn: 12] | | | |

During matching, the replacement variable RA.*n* is substituted by the attribute noun, and the binding variables $\alpha$ and *n* are bound to the values girl and 12, respectively. The application of this rule to the token line example 5.1.1 has the following result:

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

[noun: girl / fnc: sleep / mdr: blonde / prn: 12] [noun: girl / fnc: eat / mdr: small / prn: 15]

Such moving along the token line of an activated concept is naturally extendable to answering questions. There are two kinds of basic query, called (i) *wh questions* and (ii) *yes/no questions*,[2] illustrated below with examples relating to the token line example 5.1.1.

### 5.1.5 Basic kinds of query in natural language

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

[2] The expressions used in these questions are of the sentence mood *interrogative*. Explicit syntactic–semantic derivations of interrogatives, including long distance dependency, are shown in Sect. 9.5.

In Database Semantics, these questions translate into the following proplet patterns:

### 5.1.6 Search proplets illustrating the two basic types of questions

| *wh question* | *yes/no question* |
|---|---|
| noun:*girl*<br>fnc: walk<br>mdr: $\sigma$<br>prn: $n$ | noun:*girl*<br>fnc: walk<br>mdr: young<br>prn: $n$ |

In the search proplet of the wh question, the mdr attribute has the variable $\sigma$ and the prn attribute the variable $n$ as value; in the search proplet of the yes/no question, only the prn attribute has a variable as value. Technically, the answer to a question consists in binding the variables of the search proplet to suitable values. Consider the application of the wh question search pattern in 5.1.6 to the token line 5.1.1:

### 5.1.7 Wh search pattern checking a token line

| | | | | | |
|---|---|---|---|---|---|
| | | | | noun:*girl*<br>fnc: walk<br>mdr: $\sigma$<br>prn: n | *search pattern* |
| | | | | *matching?* | |
| noun: *girl* | noun: girl<br>fnc: walk<br>mdr: young<br>prn: 10 | noun: girl<br>fnc: sleep<br>mdr: blonde<br>prn: 12 | noun: girl<br>fnc: eat<br>mdr: small<br>prn: 15 | noun: girl<br>fnc: read<br>mdr: smart<br>prn: 19 | *token line* |

The indicated attempt at matching fails because the fnc values of the proplet pattern (i.e., walk) and of the proplet token (i.e., read) are incompatible. The same holds after moving the pattern one proplet to the left. Only after reaching the leftmost proplet is the matching successful. Now the variable $\sigma$ is bound to young and the variable $n$ to 10. Accordingly, the answer provided to the question Which girl walked? is The young girl (walked). This procedure is formalized by the following LA-think grammar:

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

| | | | |
|---|---|---|---|
| $ST_S$: {( | noun: RV.1<br>fnc: ¬ RV.2<br>mdr: $\sigma$<br>prn: $n$ | {$r_1$, $r_2$}), ( | noun: RV.1<br>fnc: RV.2<br>mdr: $\sigma$<br>prn: $n$ { }) } |
| $r_1$: | noun: RV.1<br>fnc: ¬ RV.2<br>mdr: $\sigma$<br>prn: $n$ | noun: RV.1<br>fnc: ¬ RV.2<br>mdr: $\sigma$<br>prn: *n–1* | output position nw {$r_1$ $r_2$} |
| $r_2$: | noun: RV.1<br>fnc: ¬ RV.2<br>mdr: $\sigma$<br>prn: $n$ | noun: RV.1<br>fnc: RV.2<br>mdr: $\sigma$<br>prn: *n–1* | output position nw { } |

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

The proplets in the start states, the rules, and the final states represent search patterns, whereby RV.1 and RV.2 are replacement variables for a core and a continuation value, respectively, while $\sigma$, *n*, and *n–1* are binding variables. For example, applied to the wh search pattern of 5.1.6, RV.1 is replaced with girl and RV.2 with walk.

For the feature [fnc: ¬ RV.2] to match it must *not* be compatible with the input. In the definition of LA-think.Q1 in 5.1.8, this is specified for the first start state, the sentence start and the next word of $\mathrm{r}_1$, the sentence start of $\mathrm{r}_2$, and the first final state. In contrast, the corresponding feature [fnc: RV.2] in the second sentence start, the next word of $\mathrm{r}_2$, and second final state, does have to be compatible to match the input.

If the last proplet of a token line does not match the search pattern, the rules $\mathrm{r}_1$ and $\mathrm{r}_2$ are both activated (cf. rule package of the first start state). If the next to last proplet does not match either, $\mathrm{r}_1$ is successful; otherwise $\mathrm{r}_2$ is successful. As long as there are proplets in the token line for $\mathrm{r}_1$ to apply successfully (i.e., matching fails), the derivation continues; when no token remains, the grammar algorithm enters the first final state, realized in English as I don't know. If the derivation completes with a successful application of $\mathrm{r}_2$, the grammar enters the second final state, realized in English as a noun phrase based on the matching proplet, here the young girl.

A yes/no question is handled in a similar manner. For example, Did the young girl walk?, based on applying the yes/no search pattern of 5.1.6 to the token line 5.1.1, results in the answer yes, due to a successful matching between the search pattern and the proplet with the prn value 10. The procedure is formalized as LA-think.Q2:

### 5.1.9 Definition of **LA-think.Q2** (*yes/no question*)

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

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

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

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

In LA-think.Q2, RV.1, RV.2, and RV.3 are replacement variables, while *n* and *n–1* are binding variables. The answers are based on the final states $\mathrm{ST}_F$, the first of which represents the answer no and the second the answer yes. For example, in the attempt

to answer the question Did the blonde girl walk? relative to the token line 5.1.1, the search will be in vain; having applied $r_1$ to all proplets in the token line, LA-think.Q2 terminates in the first final state, which is realized in English as the answer no. The question Did the young girl walk?, in contrast, results in the second final state, which is realized in English as the answer yes. For further discussion see Sect. 9.5.

## 5.2 Episodic versus Absolute Propositions

Another kind of thinking, besides (i) navigating along interproplet relations (cf. 3.5.1) and (ii) moving along token lines (cf. 5.1.3), are (iii) inferences. In Database Semantics, inferences are based on *absolute* propositions, as opposed to *episodic* propositions (cf. FoCL'99, Sect. 22.4; AIJ'01).

Absolute propositions represent content which holds independently of any STAR (cf. 2.6.2), such as scientific or mathematical content, but also personal beliefs like Mary takes a nap after lunch. Episodic propositions, in contrast, require specification of the STAR in order for the content to be coded correctly.[3]

In DBS, episodic and absolute propositions are distinguished by the value of their prn attribute, as illustrated by the following example showing the proplets of the propositions A dog is an animal (absolute) and the dog is tired (episodic):

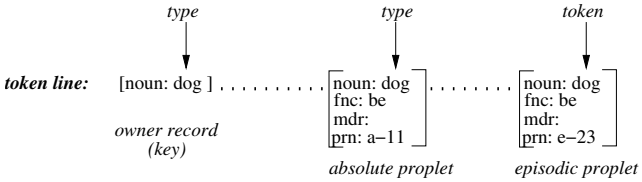
5.2.1 ABSOLUTE AND EPISODIC PROPLETS IN A WORD BANK (context level)

| *owner records* | *absolute proplets* | *episodic proplets* |
|---|---|---|
| [noun: animal] | … [noun: animal<br>fnc: is<br>mdr:<br>prn: a-11] | |
| [verb: be] | … [verb: be<br>arg: dog animal<br>mdr:<br>prn: a-11] | … [verb: be<br>arg: dog<br>mdr: tired<br>prn: e-23] |
| [noun: dog] | … [noun: dog<br>fnc: be<br>mdr:<br>prn: a-11] | … [noun: dog<br>fnc: be<br>mdr:<br>prn: e-23] |
| [adj: tired] | … | … [adj: tired<br>mdd: be<br>mdr:<br>prn: e-23] |

In addition to the different prn values, here a-11 versus e-23, absolute and episodic proplets differ in that the core value of absolute proplets is a concept type, while the core value of episodic proplets is a concept token:

[3] Absolute propositions of the context level are coded in language as *generic* sentences which express their atemporal character by using present tense. Episodic propositions of the context level are coded in language as sentences which usually contain temporal and local adjectives, and a verb which is not in present tense.

### 5.2.2 Core values as concept types and concept tokens




Thus, at the context level the core values of episodic proplets are typed indirectly by the associated owner record.

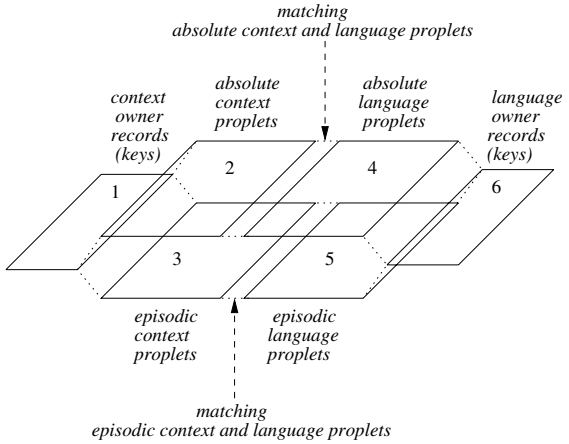
The distinction between absolute and episodic proplets leads to an additional division of fields in a Word Bank, both in the area of context and of language proplets:

### 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 |

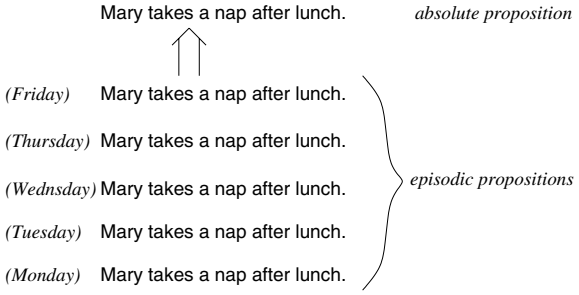
To obtain matching frontiers (cf. 3.3.1) between the areas of context and language for the absolute as well as the episodic propositions, absolute and episodic proplets are stored conceptually in two levels, as indicated by the following representation:

### 5.2.4 Matching frontiers in a three-dimensional representation




Presenting the absolute proplets above the episodic ones is motivated by the transition of frequent episodic propositions of the same content into an absolute proposition, as illustrated below:

### 5.2.5 From episodic propositions to an absolute proposition




Episodic propositions represent isolated events, bound to a particular STAR. If their STAR grounding turns out to be redundant (because it does not seem to matter), they may be summarized as an absolute proposition, in which the STAR is omitted.

From a technical point of view, it must be emphasized that the structuring of a Word Bank indicated in 5.2.4 is purely conceptual and does not restrict the storage locations in the actual implementation of the database. This is because relations between proplets are coded solely in terms of attributes and their values.

## 5.3 Inference: Reconstructing *Modus Ponens*

Before we turn to the handling of inferences in Database Semantics, let us consider the classical inferences of Propositional Calculus. In the following summary, $A$ and $B$ are true propositions, $\neg$ is sentential negation, & is logical and, $\vee$ is logical or, $\rightarrow$ is logical implication, the expressions above the horizontal line are the premise or premises, and the expression below is the conclusion, indicated by $\vdash$:

### 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. $\dfrac{\neg\neg A}{\vdash\ A}$

In the inference schema 1, called *conjunction*, the truth of two arbitrary propositions $A$ and $B$ implies the truth of the complex proposition $A\&B$. If the inference of *conjunction* were reconstructed in Database Semantics, it would amount to an operation which establishes new extrapropositional relations between arbitrary propositions, based on the conjunction and. This operation may be characterized as the following LA-grammar rule.

### 5.3.2 LA-RULE FOR THE INFERENCE OF *conjunction* (HYPOTHETICAL)

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

The patterns to the left of the arrow each match the verb proplet of some arbitrary proposition, thereby binding the variables $\alpha, \beta$, *m*, and *n*. To the right of the arrow the same patterns are shown with the additional feature [cnj: *m* and *n*]. Thus, inf1 produces the extrapropositional relation of coordination between two arbitrary propositions *m* and *n* by adding cnj attributes with values to their proplets in the word bank. These new connections enable new navigations, from any proposition to any other proposition asserted in the Word Bank.

In its Truth-Conditional Semantics interpretation, the inference in question expresses a conjunction of truth and is, as such, intuitively obvious. From this point of view, it is not relevant which propositions are conjoined. The only condition for the inference to be valid is that the two propositions in the premise happen to be true.

From the view point of Database Semantics, in contrast, there arises the question of *why* two – previously unconnected – propositions should be concatenated with and. For example, even though conjoining Lady Marion smiled at Robin Hood and The glove compartment was open might not result in asserting a falsehood, it would make little sense. Because an uncontrolled application of $\text{inf}_1$ would create new, unmotivated connections between hitherto unconnected propositions, the coherence of content in a word bank would be destroyed by it.

The example of *conjunction* has shown how a classical inference can be reconstructed in Database Semantics. We have also seen that the classical inference schemata may change their character substantially when transferred to Database Semantics. Therefore they should not be transferred blindly.

Let us turn now to *modus ponens*. Its inference schema (cf. 3 in 5.3.1) may be illustrated in Propositional Calculus as follows:

### 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. | $\underline{A\quad}$ |
| Conclusion: | Mary takes a walk. | $\vdash B$ |

Because Propositional Calculus treats propositions as unanalyzed constants. e.g., $A$ and $B$, this form of *modus ponens* is suitable to assert the inference relation between sentences, but not between individuals. Let us therefore consider *modus ponens* in Predicate Calculus (cf. Bochenski 1961, 16.07 ff., 43.16 ff.), which analyzes the internal structure of propositions using the connectives $\neg$, &, $\vee$, and $\rightarrow$ of Propositional Calculus plus one- and two-place functor constants like f, g, and h, and the quantifiers $\exists$ and $\forall$ horizontally binding the variables x, y, z:

### 5.3.4 *modus ponens* IN PREDICATE CALCULUS

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

Thus, if f is realized as dog, g as animal, and h as tired, the inference would read:

| | | |
|---|---|---|
| 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. | $\vdash \exists x[animal(x)\&tired(x)]$ |

Because the binding of variables in Prepositional Calculus is horizontal (e.g., ∃x [... x]), but vertical in Database Semantics, and because the parts of a logical formula are ordered, while proplets are unordered, this inference cannot be transferred to Database Semantics directly. Instead, *modus ponens* may be reconstructed by representing (i) the premise with the universal quantifier as an absolute proposition, e.g., A dog is an animal, (ii) the premise with the existential quantifier as an episodic proposition, e.g., The dog was tired, and (iii) the conclusion as the following rule:

### 5.3.5 INFERENCE RULE INF$_2$ FOR RECONSTRUCTING *modus ponens*

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

The proplets with the prn value *e-n* represent the episodic premise, the one with the prn value *a-m* the absolute premise, and the operation realizes the conclusion.

This inference rule may be restated informally as the following abstract paraphrase:

### 5.3.6 ABSTRACT PARAPHRASE

1. Absolute premise: noun type $\delta$ is-a noun type $\gamma$.
2. Episodic premise: noun token $\alpha$ happens to be adjective $\beta$.
3. Episodic conclusion: If noun token $\alpha$ instantiates noun type $\delta$, then noun token $\gamma$ happens to be adjective $\beta$.

This reconstruction is based semantically on equating the grammatical terms *noun, verb*, and *adjective* with the logical terms *argument, functor*, and *modifier*, and the ontological terms *object/individual, relation*, and *property*, respectively (cf. FoCL'99, Sect. 3.4). Furthermore, while the two premises of *modus ponens* in Predicate Calculus are connected – roughly speaking – by a quantifier-kind of Substitutivity of Identicals (cf. FoCL'99, Sect. 20.1), in Database Semantics the connection is based on the type–token relation "*instantiates*." This is shown by the following example, which realizes the variables of the abstract paraphrases as concrete terms:

### 5.3.7 Concrete example based on the abstract paraphrase

1. Absolute 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*.

In the formal definition of the rule $\mathsf{inf}_2$, the distinction between concept tokens and concept types is left implicit for simplicity.[4]

Consider the application of the rule $\mathsf{inf}_2$ to proplets in the Word Bank 5.2.1:

### 5.3.8 Applying $\mathsf{INF}_2$ to the Word Bank 5.2.1

$\mathsf{inf}_2$: [verb: be; arg: $\alpha$; mdr: $\beta$; prn: *e-n*] [noun: $\alpha$; fnc: be; prn: *e-n*] [adj: $\beta$; mdd: be; mdr:; prn: *e-n*] [verb: is-a; arg: $\delta$ $\gamma$; prn: *a-m*] if $\alpha$ instantiates $\delta$, replace $\alpha$ with $\gamma$ and replace *e-n* with *e-n'* { . . . }

[verb: be; arg: dog; mdr: tired; prn: e-23] [noun: dog; fnc: be; mdr:; prn: e-23] [adj: tired; mdd: be; mdr:; prn: e-23] [verb: is-a; arg: dog animal; mdr:; prn: a-11]

In this example, the first three patterns of the rule are matched with the episodic proplets *be, dog,* and *tired*, thereby vertically binding the variable $\alpha$ to the value dog and the variable $\beta$ to the value tired. The fourth pattern is matched with an absolute proplet. The variables $\delta$ and $\gamma$ are vertically bound to the values dog and animal, respectively. Because the token bound to the variable $\alpha$ instantiates the type bound to variable $\delta$, the operation of the rule replaces $\alpha$ with $\gamma$, resulting in the following new proposition, which represents the conclusion of the inference:

### 5.3.9 Result of applying $\mathsf{INF}_2$ to the Word Bank 5.2.1

[verb: be; arg: animal; mdr: tired; prn: e-23'] [noun: animal; fnc: be; mdr:; prn: e-23'] [adj: tired; mdd: be; mdr:; prn: e-23']

In this way, the original episodic proposition the dog is tired with the prn value e-23 is complemented by the variant the animal is tired with the prn value e-23'.

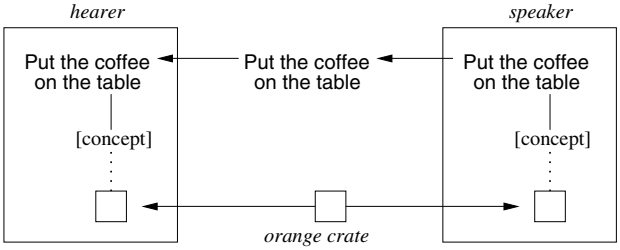
Therefore, prior to this application of *modus ponens*, querying the Word Bank 5.2.1 with the question Was the animal tired? would result in the answer no. Once the inference is applied, however, the answer is yes – which is in concord with intuition. Technically, the application of $\mathsf{inf}_2$ is an instance of nonmonotonic reasoning.

The method of inferring illustrated with $\mathsf{inf}_2$ may be extended to episodic propositions of other structures, e.g., one-place main verbs, two-place main verbs, etc., and applied to proplets of all absolute propositions with the verb is-a, i.e., to all the elements of the *is-a* hierarchy. Similar rules may be defined for the other hierarchies.

## 5.4 Indirect Uses of Language

The strict correspondence between language and context proplets illustrated in 3.2.4 raises the question of how to extend internal matching to nonliteral uses. As an example (borrowed from FoCL'99, p. 92), consider a hearer who has just entered a room containing only an orange crate. If the speaker commands Put the coffee on the table!, the hearer will infer that table refers to the orange crate. Given the small number of referential candidates in this limited context of use, the minimal meaning$_1$ of the word table best fits the structure of the orange crate (*best match*).

### 5.4.1 Nonliteral use of the word table




However, if a prototypical table were placed next to the orange crate, the hearer would interpret the sentence differently, putting the coffee not on the orange crate, but on the table. This is not caused by a change in the meaning$_1$ of table, but by the fact that the context of use has changed, providing an additional candidate for best match.[5]

The importance of metaphor as a basic principle of natural language communication has been emphasized by Lakoff and Johnson (1980) and subsequent work. However, their approach is quite different from ours because it is based on the following meaning definition by Grice:

### 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.[6]

One problem with this definition is that it is not suitable for a computational model.[7]

[4] Whether a concept happens to be a type or a token may be inferred from the distinction between absolute and episodic propositions, formally marked by the prn values.

[5] The principle of best match can only function properly if the choice of possible candidates is restricted. Therefore the selection and delimitation of the context of use based on the sign's STAR is crucial for the successful interpretation of natural language. Cf. 2.6.2 above. For a more detailed discussion see FoCL'99, pp. 93 ff., Sect. 5.3.

[6] Cf. Grice (1957) and (1965).

[7] For a more detailed discussion see FoCL'99, pp. 84–86.

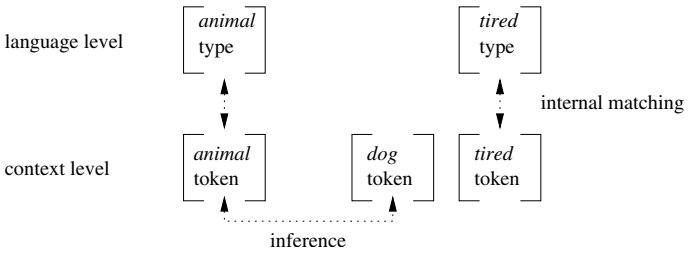
Regarding metaphor, the difference between the two approaches may be demonstrated with the following example from Lakoff and Johnson (1980, p. 12). They present the utterance Please sit in the apple juice seat as referring to the seat with the apple juice setting, and continue: "In isolation this sentence has no meaning at all, since the expression 'apple juice seat' is not a conventional way of referring to any kind of object." For the SLIM theory of language, the question is whether Lakoff and Johnson use the term "meaning" as the meaning$_1$ of the sign, or the meaning$_2$ of the utterance (cf. 2.6.1). If "meaning" is to be interpreted as the literal meaning$_1$ of the sign apple juice seat, then the sentence clearly does have such a meaning, compositionally derived from the meaning$_1$ of its words. If "meaning" is to be interpreted as meaning$_2$, however, then it holds by definition that no sentence has a meaning$_2$ *in isolation*, i.e., as a sign type and without an associated context of interpretation.

According to our analysis, the metaphoric reference in this example comes about by matching the meaning$_1$ of the expression with a context containing a seat with an apple juice setting. The meaning$_1$ derived from the words apple, juice, and seat is minimal in the sense of *seat related to apple juice*. Given the limited set of referential candidates in the context in question, this literal meaning is sufficient to pick out the intended referent based on the principle of best match.

For Database Semantics, the crucial question raised by this analysis of indirect use is how the principle of best match should be implemented formally. Rather than loosening the matching conditions defined in 3.2.3 (as would be suggested by the intuitive approach illustrated in 5.4.1), Database Semantics handles nonliteral (indirect) language uses on the basis of inferencing.

Direct and indirect uses differ as follows: In direct use, the language and the context proplets match in accordance with the matching condition 3.2.3, as illustrated in 3.2.4, whereas in indirect use, inferences first map some of the context proplets into a *secondary coding* that is then matched by corresponding language proplets directly, again in accordance with the matching condition 3.2.3.[8] For example, if the contextual content *the dog is tired* is to be coded into language as the animal is tired, then is tired is a direct use while the animal is an indirect use based on the absolute proposition *a dog is an animal* (see 5.3.5–5.3.9):

### 5.4.3 CONTEXTUAL INFERENCE UNDERLYING A NONLITERAL USE



In the speaker mode, the proposition *dog tired* is extended at the context level into the secondary coding *animal tired*. This secondary coding is matched directly with language proplets. In the hearer mode, the language proplets *animal tired* are matched directly with corresponding context proplets. By looking for an appropriate instantiation of *animal* at the context level, the hearer reconstructs the primary coding from the secondary coding via an inference.

With this approach, direct and indirect uses of language differ solely in whether or not the proplet representing the referent is the referent itself (primary coding) or instead some other proplet related to the referent via an inference (secondary coding). This way of handling inferences, secondary codings, and indirect language uses has the following advantages:

### 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.

Let us consider the inference, the secondary coding, and the indirect use of language in a few more examples. Another kind of indirect use, called *pars pro toto*, is based on the inference of choosing a prominent property of the referent, as when observing *girl with pink dress wants to sing* (context level) and expressing it with the sentence The pink dress wants to sing (language level).

Other inferences underlying a secondary coding select a characteristic behavior to refer to an individual, as in The man eater went back into the jungle, where the subject refers to a tiger preying on humans. Indirect uses arise also with absolute propositions. For example, The master of the animal kingdom lives in Africa (language level) may be used to express *The lion lives in Africa* (context level).

Indirect uses and the inferences on which they are based apply to all three parts of speech. An indirect use of a verb is illustrated by Julia did the dishes. Here an inference selects the verb from a higher place in the semantic hierarchy associated with *wash*, just as in our initial nominal example of referring to a dog with the word animal. The indirect use of an adjective, finally, is illustrated by easy money, where *easy* is a secondary coding expressing the manner in which the money is obtained.

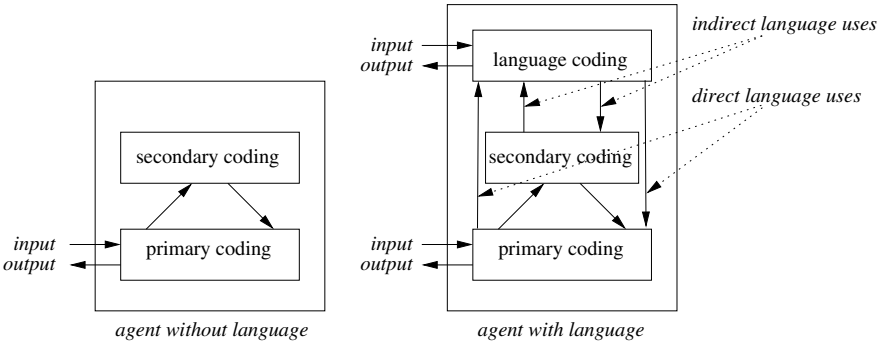
[8] For example, the inference underlying 5.4.1 is based on the absolute propositions A table has a flat surface and An orange crate has a flat surface. Rather than formally reconstructing this inference, we use the inference defined in the previous section for the following Example 5.4.3.

Indirect uses arise not only with the sign kind of symbols, as when referring to the context proplet *dog* using the symbolic language proplet animal, but also with indexicals and names. For example, there are indirect uses of referring with a concept to a referent represented by a name proplet, as in calling *Richthofen* the red baron, referring with an indexical to a name proplet, as in calling *Julia* she, referring with an indexical to a referent represented by a concept proplet, as in calling the book in front of me it, etc. All these forms of reference are based on a secondary coding with associated inferencing at the level of context.

## 5.5 Secondary Coding as Perspective Taking

The derivation of secondary codings, defined solely at the level of context, is independent of whether or not the agent has the language faculty. Accordingly, even a dog, for example, could view an entity from a certain subjective perspective, such as viewing the mail man as an enemy. The role of secondary codings in agents with and without language may be shown schematically as follows:

### 5.5.1 Coding levels in agents with and without language




In terms of evolution, the levels of primary and secondary coding in agents without language are being reused essentially unchanged in agents with language. The whole machinery of inferencing and secondary coding is already present when language is added. The adding of language is based on matching the primary and the secondary codings directly (vertical arrows).

The functions of the three kinds of coding may be summarized as follows:

### 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.

Primary and secondary codings are related to each other by inferences. They are related to the language coding by a strict method of internal matching.

The combination of primary and secondary coding allows content to be represented from different points of view at different levels of abstraction. Thereby the primary contextual coding can be left intact, at least in short-term memory. Long-term storage, however, may involve a fusion of primary and secondary coding into a condensed form – depending on the amount of content and the limits of storage.

While the primary coding of a given content is basically the same in different agents, their secondary codings may vary widely, depending on their knowledge and purposes. For example, an experienced scout and a greenhorn may see the same broken twig in primary coding. In secondary coding, however, the scout may spontaneously see the twig as a trace of whatever they pursue, while the greenhorn does not. Using language, however, the scout may adjust the greenhorn's secondary coding to his own.

## 5.6 Shades of Meaning

It seems to be taken for granted that a native speaker's intuitions about the meaning of a word could never be modeled in an artificial agent. This, however, is an unjustified assumption. In fact, the meaning of a word consists of several well-defined ingredients. That some of them consist of (i) personal experiences and (ii) implicit assumptions of the surrounding culture does not mean that they cannot be modeled.

The ingredients of a word meaning are based on the distinctions (i) between concept types and concept tokens, (ii) between episodic and absolute propositions, and (iii) between primary and secondary codings. Consider the word dog, for example.

First, there is the result of lexical lookup:

### 5.6.1 Lexical lookup of the word dog

| |
|---|
| sur: dog |
| noun: *dog* |
| fnc: |
| mdr: |
| prn: |

Lexical lookup provides the semantic core, here *dog*, defined as a concept type and serving as the meaning$_1$.

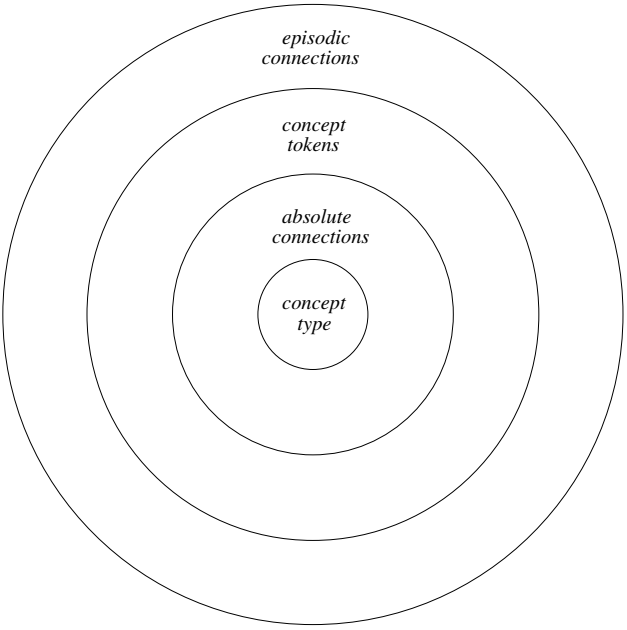
Then there are all the absolute proplets containing the concept type *dog* listed in the absolute part of the token line of this concept (cf. 5.2.3 and 5.2.4, field 2):

### 5.6.2 Absolute part of the token line of *dog*

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

These absolute proplets are connected: Based on their continuation values, all the absolute propositions about dogs may be activated, lighting up a subnetwork in the Word Bank. In this way, the semantic core of the lexical proplet (meaning$_1$) is complemented compositionally with the agent's general knowledge about dogs. This aspect of meaning is represented in the following schema as the second circle, called *absolute connections*:

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




In addition to complementing a literal meaning$_1$, represented by a concept type, with all the absolute connections related to it there are the episodic proplets in the token line of the concept (cf. 5.2.3 and 5.2.4, field 3). If the concept is *dog*, for example, then the episodic proplets in its token line complement the core value type of the lexical item, e.g., 5.6.1, with all the core value tokens of dogs the agent has encountered and stored so far. These core value tokens may have any degree of detail, such as the kind of dog, the color of fur, the sound of barking, etc., depending on the memory available to the agent as well as the individual importance of each token. Given that these tokens arise

at the level of context, they constitute an aspect of meaning which is generalized in the sense that it is independent of a particular utterance (cf. *concept tokens* in 5.6.3).

Finally, the episodic proplets in the token line of *dog* are the starting points of all the episodic propositions involving dogs, including secondary codings based on inferences. Using the retrieval mechanism of the Word Bank, the proplets of these propositions may be activated. In this way, the generalized aspect of meaning represented by the set of dog tokens is complemented compositionally with the agent's individual dog experiences (cf. *episodic connections* in 5.6.3).

Thus, the build-up of structure, from concepts types to absolute connections to concept tokens to episodic connections can well explain why a concept type, as the minimal literal meaning$_1$ used for matching between the language and the context level (cf. 4.2.2), is intuitively shrouded behind a cloud of individual differentiations. These consist of the agent's various tokens of the concept instantiating the type (content aspect of the word) and the various absolute and episodic connections (combinatorial aspects of the word). Due to the data structure of a Word Bank, they can all be activated with great ease, and jointly represent the associations an agent may have with a word's concept.[9]

[9] In some natural languages, e.g., German, the meaning of a sentence or phrase may be shaded by using a fairly large number of "particles." Examples in German (cf. Weydt (1969), Engel (1991), Ickler (1994)) are aber (`but`), auch (`also`), bloss (`merely`), as in Das ist aber schön! (`That is but beautiful!`), Sind Sie auch zufrieden? (`Are you also content?`), Was hat sie sich bloss gedacht?! (`What has she herself merely thought?!`), etc. Their main contribution to the overall interpretation is the triggering of certain inferences about the speaker's perspective.