Generalized Reference

Referring with and Without Language by Matching, Pointer, or Address

Roland Hausser(✉)

Universität Erlangen-Nürnberg (em.), Erlangen, Germany
rrh@linguistik.uni-erlangen.de

Abstract. Medieval logic defined reference as a relation between language and objects in the world. Recently, however, the term "representational token" has been used instead of language (Reimer and Michaelson 2014). This allows for reference with and without language. In a similar vein, Database Semantics (DBS) has implemented concept-based reference as a matching between two contents. If a content is attached to a language surface it is called the literal meaning$_1$ of the surface.

Referring with a content (as a representational token), regardless of whether or not it is attached to a surface, leads to a generalized notion of reference (Sect. 6.3). An example of reference without language is identifying a current nonlanguage recognition with something seen before. Another example is identifying a nonlanguage recognition with an earlier language content, e.g. something read (for example, in a guide book) or heard about.

In addition to the concept-based reference mechanism of (i) symbols (We follow the terminology used by Peirce (CP 2.228, 2.229, 5.473) for his theory of signs.) (Sect. 6), natural language uses the reference mechanisms of (ii) indexicals (Sect. 7) based on pointers, and of (iii) names (Sect. 8) based on acts of generalized baptism and coreference by address. This paper systematically reconstructs the mechanisms of reference as they function with and without language in an agent-based computational framework.

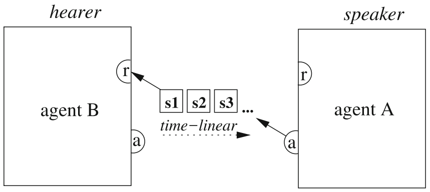
Keywords: Agent-based · Matching, pointing, address · Hear mode, speak mode · Time-linear derivation order · Declarative specification · Talking robot

1 Agent-Based Ontology

When two agents communicate with each other by means of a natural language, the speaker uses its external action interface to produce a sequence of language surfaces while the hearer uses its external recognition interface to identify the elements of the sequence. The sequence is time-linear in the sense that it is linear like time and in the direction of time. In accordance with the Western writing convention, the progression of time is shown in the direction from left to right.

M. Kurosu (Ed.): HCI 2017, Part I, LNCS 10271, pp. 427–446, 2017.
DOI: 10.1007/978-3-319-58071-5_32

1.1 Physical Framework of Communication



The **r**ecognition and **a**ction interfaces of the agents are indicated by half circles marked with r and a. The language surfaces are represented by boxes containing s1, s2, s3, ... As agent-external modality-dependent sound waves (speech), dots on paper (writing), or gestures (signing), the surfaces may be measured and described with the methods of the natural sciences, but have no meaning and no grammatical properties whatsoever.

The first surface leaving the speaker is the first to reach the hearer. The last surface leaving the speaker is the last to reach the hearer.[1] All other aspects of language communication are agent-internal, modality[2]-independent, and cognitive.

Modality-independence may be illustrated by the basic operations of arithmetic, i.e., addition, subtraction, multiplication, and division. They exist at a level of abstraction which may be realized equivalently as the operations (i) of a human, (ii) a mechanical calculator, or (iii) an electronic computer.[3]

With autonomous robots still absent in today's computational linguistics, the external framework Sect. 1.1 may be simulated, using the keyboard and the screen of standard computers as primitive recognition and action components. This, however, works only for the transfer of surfaces. It does not work for nonlanguage recognition and action, which are required for a cognitive reconstruction of reference. For example, the agent's ability to refer to agent-external items is needed for fulfilling a request like *Pick up the blue square!* or to report how many blue squares there are in the agent's current task environment.

2 Elementary Concepts

The minimum in reconstructing higher-level cognition is (i) an agent-internal memory, (ii) a central control embedded into and interacting with memory, (iii) a mapping from the recognition interface to central control, and (iv) a mapping from central control to the action interface. The mappings between modality-dependent raw data and modality-independent concepts are formally

[1] In Sects. 1.1, 4.2, and 5.2 this is expressed graphically by placing the hearer to the left and the speaker to the right. If the order is reversed, the progression of time would have to be shown from right to left.

[2] The term *modality* is being used in several different fields of science. As employed here, modality is known as *sensory* modality (Chen 2006, Sect. 6.13.1).

[3] According to Wiener (1961, p. 132): "Information is information, and not matter or energy.".

based on the type-token distinction, familiar from philosophy.[4] The type of a concept describes the *necessary* properties, while an associated token is an instantiation with certain additional *accidental*[5] properties. As an example consider the recognition of colors (Hausser 1989, p. 296 ff). In physics, they are defined as intervals on the one-dimensional scales of electromagnetic wave length and frequency. Accordingly, the type and a token of the color blue may shown as follows.

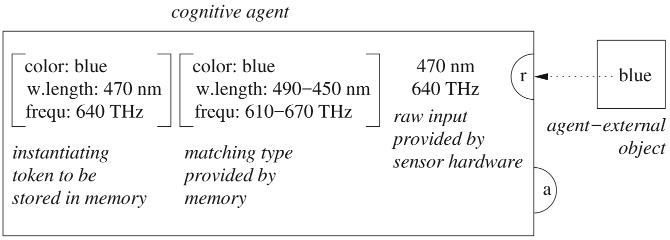
2.1 Type and Token of the Color Called blue

$$\begin{array}{ll} \textit{type} & \textit{token} \\ \begin{bmatrix}\text{wavelength: } \alpha \\ \text{frequency: } \beta\end{bmatrix} & \begin{bmatrix}\text{wavelength: 470 nm} \\ \text{frequency: 640 THz}\end{bmatrix} \\ \text{where } \alpha \epsilon \text{ 490–450 nm} & \\ \text{and } \beta \epsilon \text{ 610–670 THz} & \end{array}$$

The type specifies the wavelength and the frequency of the color *blue* by means of variables which are restricted to the corresponding intervals provided by physics. The token uses constants which lie within these intervals.

In the recognition of colors, the type provided by memory and the raw input data provided by a sensor interact as follows, resulting in a classified token.

2.2 Concept Type and Token in Color Recognition



A sensor measures the wavelength 470 nm and frequency 640 THz in an agent-external object. These values lie within the intervals 490–450 nm and 610–6700 THz of the color blue and thus match the type. In the instantiating token, the wavelength and frequency intervals of the type are replaced by the measured values. The feature structures representing types and tokens may be extended as needed, for example, with an additional attribute for color intensity.

Next consider the type and the token of a two-dimensional geometric object.

[4] The type-token distinction was introduced by Peirce (CP 4: 537).

[5] The term accidental is used here in the philosophical tradition of Aristotle, who distinguishes in his Metaphysics, Books ζ and η, between the necessary and the accidental (incidental or coincidental – kata sumbebêkos) properties of an object in nature.

2.3 Concept Type and Token of the Concept square

type

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

where α is a length

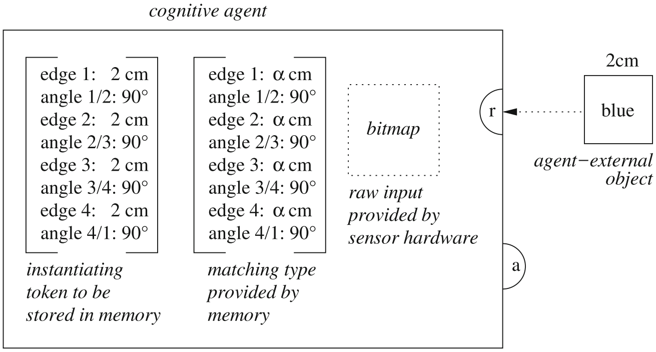
token

$$\begin{bmatrix}\text{edge 1: 2 cm}\\ \text{angle 1/2: }90^{\mathrm{O}}\\ \text{edge 2: 2 cm}\\ \text{angle 2/3: }90^{\mathrm{O}}\\ \text{edge 3: 2 cm}\\ \text{angle 3/4: }90^{\mathrm{O}}\\ \text{edge 4: 2 cm}\\ \text{angle 4/1: }90^{\mathrm{O}}\end{bmatrix}$$

Here, the type and the token share attributes which specify (i) the number of equally long edges and (ii) the angle of their intersections. The type and the token differ only in their edge lengths. The latter is accidental in that the type matches an infinite number of square tokens with different edge lengths.[6]

In analogy to Sect. 2.2, recognition of a square may be shown as follows.

2.4 Type and Token in Recognizing a Square



The type matches the outline of all kinds of different squares, whereby its variables are instantiated in the resulting tokens.

Today, there exist pattern recognition programs which are already quite good at recognizing geometric objects.[7] They differ from our approach in that they are based almost completely on statistics. However, even if the terms of the type and the token may not be found in their theoretical descriptions, the type-token distinction is nevertheless implicit in any pattern recognition processing. Furthermore, the rule-based, incremental procedures[8] of pattern recognition presented in Hausser (2005) are well-suited to be combined with statistical methods.[9]

[6] In an artificial agent, the type may be implemented as a pattern-matching software which recognizes tokens by approximating raw bitmap outlines Hausser (1999, 3.2.1).

[7] As shown by the work of Steels (1999), suitable algorithms may evolve new types automatically from similar data by abstracting from what they take to be accidental.

[8] They are based on an incremental, memory-based procedure of pattern recognition using geons (Biederman 1987).

[9] For building a talking robot, the automatic evolution of types has to result in concepts which correspond to those of the intended language community. This may be achieved by presenting the artificial agent with properly selected data in combination with human guidance (guided patterns method, Hausser 2011, Sect. 6.2).

The elementary concepts of nonlanguage recognition are complemented by those of action. For example, the concept *take* is defined as the type of a gripping action which is instantiated as a token to be realized as raw data. The token differs from the type in that it is adapted to a specific gripping occasion. It holds in general for recognition that raw data are classified by a type and instantiated as a token, while in action a type is specialized into a token which is passed to a suitable action component for realization as raw data (Hausser 1999, 3.3.5).

The interaction between the agent's external interfaces, the types, the tokens, and the memory must be hand in glove. For example, if the agent has no sensor for measuring electromagnetic wavelength/frequency, colors cannot be recognized – even if the proper types were available from memory. Conversely, without the types the raw data provided by a suitable sensor cannot be classified and instantiated as tokens. Also, without a memory the types cannot be provided for recognition and action, and the tokens cannot be stored.

3 Data Structure and Database Schema

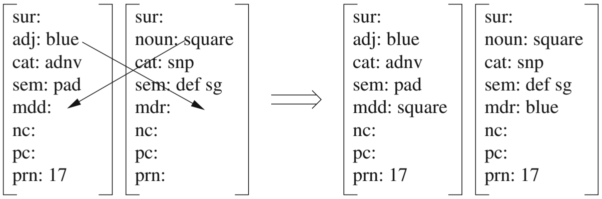
The concepts defined in Sects. 2.1 and 2.3 constitute elementary cognitive contents, but they do not provide any means for being connected, as in blue_square. For this, DBS lexically embeds the concepts as *core values* into nonrecursive[10] feature structures with ordered attributes, called *proplets* (because they are the elementary building blocks of propositions, in analogy to droplet). A feature structure is built from features. In computer science, a feature is defined as an attribute-value pair (avp), e.g. [noun: square], with noun: as the attribute and square as the value.

The embedding of core values into proplets allows their concatenation by means of value copying. For example, the proplets *blue* and *square* may be connected into the content of blue_square as follows.[11]

[10] A feature structure is nonrecursive if there is no recursive embedding of feature structures as values. Recursive feature structures are unsuitable for (i) contents with a coordination structure, (ii) the pattern matching needed for (a) modeling reference and (b) applying operation patterns to input, and (iii) storage and retrieval in a database. Unordered attributes are inefficient for computers and humans alike. Recursive feature structures with unordered attributes are not used in DBS.

[11] The algorithm used for connecting (hear mode) and activating (think mode) proplets is time-linear Left-Associative Grammar (Hausser 1992). The sur attribute takes the language dependent surface as value. For a detailed description of the attributes and values used in proplets for describing English see Hausser (2006) Appendix A3.

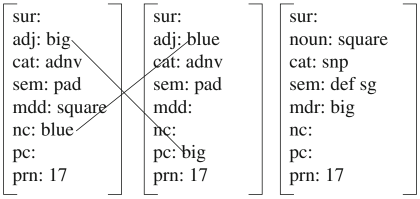
3.1 Concatenation by Cross-Copying



The nature of the semantic relation between *blue* and *square* is characterized by the attributes mdr (modifier) and mdd (modified). The relation is implemented by copying the core value of *square* into the mdd slot of *blue* and the core value of *blue* into the mdr slot *square*. In addition, the prn value of *blue*, here 17, is copied into the prn slot of the next word proplet *square*.

Next consider extending Sect. 3.1 to an intrapropositional coordination.

3.2 Coordination in *big blue square*



The relation of intrapropositional coordination is coded by the nc (next conjunct) and pc (previous conjunct) attributes of the conjoined adjectives.

The diagonal lines in Sect. 3.2 are intended as optical support for the reader. Technically, however, they are redundant and may be omitted. The real method of establishing semantic relations in DBS is by addresses coded declaratively as values and implemented procedurally as pointers. This method makes the proplets forming a complex content order-free, allowing the database to store them independently of the semantic relations between them.

For example, no matter where the storage mechanism of the database puts the adnominal *big*, its modified may be found via the primary key consisting of the mdd value square and the prn value 17. Similarly, no matter where the noun *square* is stored, its modifier may be found via the mdr value big and the prn value 17. And accordingly for the intrapropositional coordination in Sect. 3.2.

As another example consider the content of Julia knows John., represented as the following set of connected proplets.

3.3 Content of Julia Knows John. As a Set of Proplets

$$\begin{bmatrix}\text{sur: Julia}\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{sur: knows}\\ \text{verb: know}\\ \text{cat: \#s3}' \ \#a' \ \text{decl}\\ \text{arg: Julia John}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{sur: John}\\ \text{noun: John}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix}$$

The simplified proplets are held together by a common prn value, here 625. The functor-argument is coded solely in terms of attribute values. For example, the *Julia* and *John* proplets specify their functor as know, while the *know* proplet specifies Julia and John as its arguments. Because of their nonempty sur(face) slots, the proplets are language proplets, in contradistinction to the proplets in Sects. 3.1 and 3.2, which are context proplets.

For storage and retrieval, a proplet is specified uniquely[12] by its core and prn values (primary key). This suggests a two-dimensional database schema, as in a classic network database (Elmasri and Navathe 2010). However, instead of using member and owner records, DBS uses member proplets and owner values.

The result is called a word bank. Its database schema consists of a column of owner values in their alphabetical order (vertical). Each owner value is preceded by an empty slot, called the now front, and a list of member proplets (horizontal); together they constitute a *token line*.[13]

As an example, consider storing a nonlanguage content.

3.4 Storing the Proplets of Sect. 3.3 in a Word Bank

$$\begin{array}{rlll} & \textit{member proplets} & \textit{now front} & \textit{owner values}\\ \ldots & \begin{bmatrix}\text{noun: John}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix} & & \text{John}\\ \ldots & \begin{bmatrix}\text{noun: Julia}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix} & & \text{Julia}\\ \ldots & \begin{bmatrix}\text{verb: know}\\ \text{cat: \#n-s3}'\text{\#a}'\text{decl}\\ \text{arg: Julia John}\\ \text{prn: 625}\end{bmatrix} & & \text{know}\end{array}$$

The proplets in a token line all have the same core value and are in the temporal order of arrival, reflected by their prn values (Hausser 2006, Sects. 11.2, 11.3).

In contrast to the task of designing a practical schema for arranging the books in a private library, the sorting of proplets into a word bank is simple and mechanical. The letter sequence of a proplet's core value completely determines

[12] Propositions containing two or more proplets with the same values, as in Suzy loves Suzy, require extra attention. They constitute a special case which (i) rarely occurs and (ii) is disregarded here because it may be easily handled by the software.

[13] The token line of a core value is found with a trie structure (Briandais 1959). The search for a proplet within a token line may use the prn value of the address in relation to the linear increasing prn values. As pointed out by J. Handl, this may be based on binary search, in time O(log(n)) (Cormen et al. 2009), or interpolation, in time O(log(log(n))) (Weiss 2005), where n is the length of the token line.

its token line for storage: the storage location for any new arrival is the penultimate position (now front) in the corresponding token line. When this slot is filled, the now front is reopened by moving the owner value one slot to the right (or, equivalently, pushing the member proplets one slot to the left, as in a push-down automaton).

By storing content like *sediment*, the stored data are never modified and any need for checking consistency is obviated. Changes of fact are written to the now front, like diary entries recording changes of temperature. Current data which refer to old ones use addresses as core values, implemented as pointers.

4 Cycle of Natural Language Communication

The transfer mechanism of content from the speaker to the hearer is based on external surfaces which have neither a meaning nor any grammatical properties (Sect. 1.1). They must, however, belong to a language which the speaker and the hearer have each learned.

The learning enables the hearer to (i) recognize surfaces, (ii) use the recognized but otherwise unanalyzed surfaces for looking up lexical entries which provide the meaning and the grammatical properties, and (iii) connect them with the semantic relations of functor-argument and coordination. The learning enables the speaker to (i) navigate along the semantic relations between proplets, (ii) produce language-dependent word form surfaces from the core values of proplets traversed, and (iii) handle function word[14] precipitation, micro word order, and agreement.

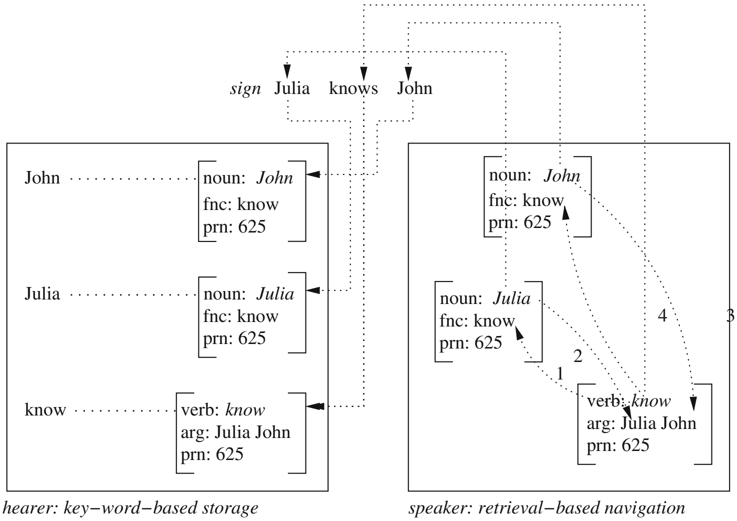
4.1 Definition of Successful Communication

> Natural language communication is successful if the content, mapped by the speaker into a sequence of external word form surfaces, is reconstructed and stored equivalently by the hearer.

The transfer of information from the speaker to the hearer, based solely (i) on unanalyzed external surfaces, (ii) the data structure of proplets, (iii) the database schema of a word bank, and (iv) the content Sect. 3.4, may be shown schematically as follows.

[14] Examples of function words in English are determiners like a(n), the, some, every, all, prepositions like in, on, above, below, auxiliaries like be, have, do, coordinating conjunctions like and, or, and subordinating conjunctions like that, who, which, when, because.

4.2 Natural Language Transfer Mechanism



The speaker's navigation through a set of connected proplets serves as the conceptualization (*what to say*) and as the basic serialization (*how to say it*) of natural language production (McKeown 1985; Kass and Finin 1988). The hearer's interpretation consists in deriving a corresponding set of proplets, based on automatic word form recognition and syntactic-semantic parsing. The time-linear order of the sign induced by the speaker's navigation is eliminated in the hear mode, allowing storage of the proplets in accordance with the database schema of the content-addressable[15] word bank. When the agent switches into the speak mode, order is reintroduced by navigating along the semantic relations between the proplets.

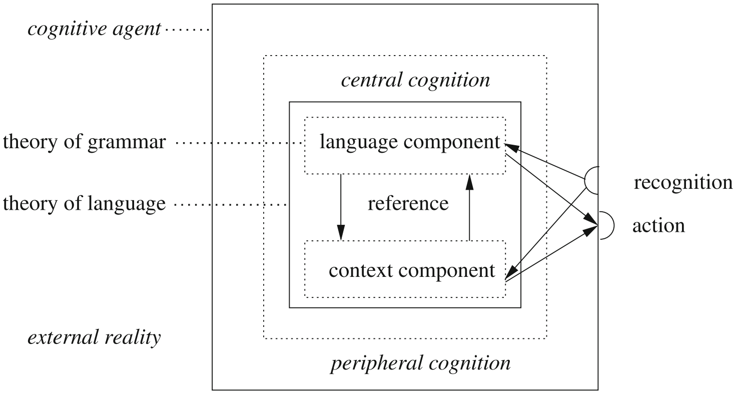
5 Conceptual Reconstruction of Reference

In DBS, a cognitive content is defined as a set of proplets connected by address. Proplets with a non-empty sur(face) slot (Sect. 3.3) represent a language content. Proplets with an empty sur slot (Sect. 3.1) represent a context content. Otherwise, language and context proplets are alike. This holds specifically for their storage and retrieval in a word bank, which is based solely on their core value and order of arrival.

Conceptually, however, reference may be modeled by (i) separating the levels of language and context, (ii) introducing the place of pragmatics as an interaction between the two levels, and (iii) distinguishing peripheral and central cognition.

[15] As a database, a word bank is content-addressable because it does not use an index (inverted file), in contrast to the widely used coordinate-addressable databases (RDBMS). See Chisvin and Duckworth (1992) for an overview.

5.1 Conceptual View of Interfaces and Components



Externally, the agent's interfaces for language and nonlanguage recognition are the same, as are those for language and nonlanguage action.[16] Internally, however, raw input data are separated by peripheral cognition into language and nonlanguage content (diagonal input arrows). Conversely in action, which realizes a content as raw output data regardless of whether it originated at the language or at the context level (diagonal output arrows).

For example, as a sound pattern the surface blue square will have a meaningful interpretation at the language level by someone who has learned English, but be treated as an uninterpreted noise at the context level by someone who has not. Conversely, even though the action of denying entrance may be realized by telling to go away (originating at the language level) or by slamming the door (originating at the context level), both result in raw output data.

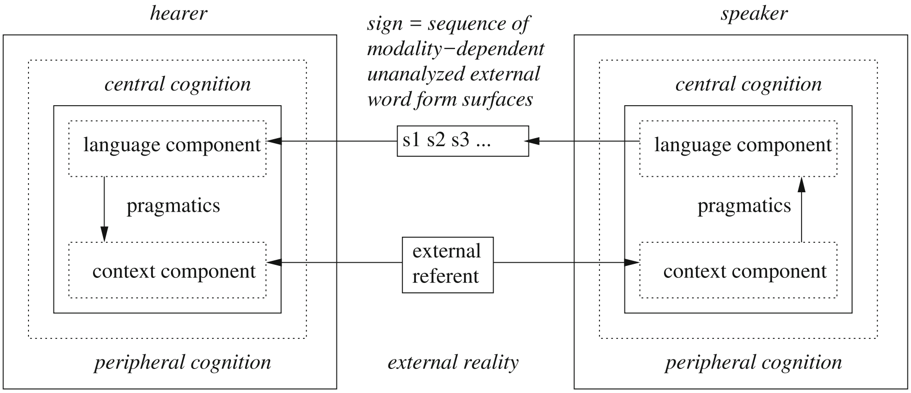
The distinction between the language and the context component provides a cognitive treatment of reference. Reference to an object in the agent's current environment is called *immediate* reference, while reference to cognitive content existing only in the agents' memory, for example, J.S. Bach, is called *mediated* reference. For mediated reference, the agent-based ontology of DBS (Sect. 1) is essential.

As an example of immediate reference consider a speaker and a hearer in a common task environment (Newell and Simon 1972) and looking at a blue square. If the speaker says Take the blue square, the noun phrase refers to the object in question. Similarly for the hearer, for whom fulfilling the request requires reference to the same object.

[16] A differentiation into the sensory modalities (vision, audition, locomotion, manipulation) is omitted – not only for simplicity, but also because the meaning of a word or expression is independent of the modality of its external surface. For example, the meaning of the word square (Sect. 2.3) is the same regardless of whether its surface is realized in speech, writing, or signing. A nonlanguage concept like the shape, color, taste, etc., of a blueberry may also be assumed to be independent from the modalities of its recognition (Hausser 2011, Sects. 2.2–2.4).

Postulating an external relation between a surface and its referent would be a reification fallacy. Instead we reconstruct immediate reference cognitively.

5.2 Immediate Reference as a Purely Cognitive Procedure



Immediate reference relies on the agents' action and recognition interfaces for language (upper level) and the recognition of nonlanguage content (lower level). Mediated reference, in contrast, relies on language action and recognition (upper level) and the existence of corresponding content in the agent's memory. While immediate reference may be regarded as prototypical for the origin of language, it is a special case of mediated reference in that it has the additional requirement of context recognition (Hausser 2006, Sect. 2.5).

Terminological Remark

Computer Science uses the term "reference" differently from philosophy and linguistics. A computational reference is an address in a storage location. This may be coded as (i) a symbolic address (declarative) or as (ii) a pointer to a physical storage location in the memory hardware (procedural). The term "generalized reference" is used in image reconstruction (computer vision).

In DBS, the term "reference" is used in the sense of philosophy and linguistics. However, the term is generalized insofar as no agent-external "representational token" is required (Sect. 6.3, constellations 1 and 3). Recanati (1997), Pelczar and Rainsbury (1998), and others use "generalized reference" for an analysis of the sign kind name which allows the surface Mary, for example, to refer to several individuals, in contradistinction to Russell's (1905) definite description analysis of "proper" names, which requires a unique referent.

The DBS analysis of names also allows different referents (Sect. 8.4). However, while the "generalized reference" of Recanati, Pelczar et al., and others is based on assimilating names to indexicals, the DBS analysis is based on an act of baptism which is generalized in that it may occur implicitly as well as explicitly. Moreover, generalized reference in DBS is not limited to names, but includes reference by means of matching concepts (symbol) and pointing (indexical).

6 Reference by Matching (Symbol)

The reference mechanism based on matching uses the type-token relation (Sects. 2.1, 2.3) and is associated with the sign kind *symbol*. For example, the terms a blue square and blue squares in the sentence sequence John saw a blue square. ... Blue squares are rare. are related as follows.

6.1 Reference with Language Proplets in Token Lines

member proplets			now front	owner values
sur: adj: blue (*token*) mdd: square prn: 41	...	sur: blue adj: blue (*type*) mdd: square prn: 48		blue
...				
sur: noun: square (*token*) sem: indef sg fnc: see mdr: blue prn: 41	...	sur: square noun: square (*type*) sem: indef pl fnc: rare mdr: blue prn: 48		square

The vertical relation between the language and the context component shown in Sect. 5.1 reappears as a horizontal relation between proplets within token lines. The language proplets with the prn value 48 have non-empty sur slots, while sur slots of the context with the prn value 41 are empty. Reference by matching holds between the two *blue* proplets with the prn values 41 and 48 and similarly between the two square proplets. The distinction between the type and the token, here indicated after the core values, is usually left implicit.

The combination of the proplets *blue* and *square* by means of a functor-argument relation is coded by the features [mdd: square] and [mdr: blue], respectively. The noun proplet with the feature [sem: indef sg] is an indefinite singular, that with the feature [sem: indef pl] is an indefinite plural.

Next consider the same reference relation without language.

6.2 Reference by Matching Without Language

member proplets			now front	owner values
sur: adj: blue (*token*) mdd: square prn: 41	...	sur: ∅ adj: blue (*type*) mdd: square prn: 48		blue
...				
sur: noun: square (*token*) sem: indef sg fnc: see mdr: blue prn: 41	...	sur: ∅ noun: square (*type*) sem: indef pl fnc: rare mdr: blue prn: 48		square

Here the reference relation holds between two nonlanguage contents – and not between a language content (meaning$_1$) and a nonlanguage content, as in Sect. 6.1.

Even though the reference relation is established between two individual proplet pairs in the same token lines, the combination into the complex content corresponding to blue square is accommodated as well[17]: in order to match, the two *blue* proplets must not only have the same[18] core value, but also the same

[17] Apparently, Aristotle struggled to reconcile reference with content combination (Modrak 2001).

[18] Disregarding the type-token distinction.

mdd continuation value, here square, and correspondingly for the mdr values of the two *square* proplets. Their fnc and prn values, however, are different.

Generalizing reference by matching to include referring with nonlanguage content results in the following constellations.

6.3 Constellations of Generalized Reference

1. *Nonlanguage content referring to nonlanguage content*
   Example: Agent identifies something seen with something seen before.
2. *Language content referring to nonlanguage content*
   Example: Agent describes a landscape in speak mode.
3. *Nonlanguage content referring to language content*
   Example: Agent identifies a current nonlanguage recognition with something it has read (for example, in a guide book) or heard about before.
4. *Language content referring to language content*
   Example: Agent describes what it has heard or read.

Cognitive agents without language are capable of reference constellation 1 only, while agents with language may use all four.

7 Reference by Pointing (Indexical)

The second reference mechanism of cognition is based on pointing. In natural language, it is illustrated by the indexical signs, such as the pronouns. The first step toward a computational implementation is the linguistic observation that the indexicals point at only five different parameters, namely (1) first person, (2) second person, (3) third person, (4) place, and (5) time.

In English, the pronouns I, me, mine, we, and us point at the parameter for first person, you points at the parameter for second person, and he, him, his, she, her, it, they, them point at the parameter for third person. The indexical adjs here and there point at the parameter for place. The indexical adjs now, yesterday, and tomorrow point at the parameter for time.

The indexical nouns pointing at the parameters of first, second, and third person are varied by grammatical distinctions. Consider the following examples illustrating grammatical variation in 1st person pronouns of English.

7.1 1st Person Pronouns Distinctions

$$
\begin{bmatrix} \text{sur: I} \\ \text{noun: pro1} \\ \text{cat: s1} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur: me} \\ \text{noun: pro1} \\ \text{cat: obq} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur: we} \\ \text{noun: pro1} \\ \text{cat: p1} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur: us} \\ \text{noun: pro1} \\ \text{cat: obq} \\ \text{sem: pl} \\ \text{mdr:} \\ \text{fnc:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
$$

The proplets all share the indexical pointer pro1 as their core value. The different cat values s1 (first person singular), p1 (first person plural), and obq (oblique) control verb agreement, preventing, for example, *Me saw a tree or *Peter saw we. *I sees a tree and *he see a tree are prevented by using the different cat values s1 (singular 1st person) and s3 (singular 3rd person).

Indexical nouns combine in the same way into propositions as proplets of the sign kind symbol or name. Consider the DBS analysis of English I heard you..

7.2 Representing I heard you. As a Language Content

$$\begin{bmatrix}\text{sur: I}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{sem: sg}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 71}\end{bmatrix}
\begin{bmatrix}\text{sur: heard}\\ \text{verb: hear}\\ \text{cat: }\#n'\#a'\text{ decl}\\ \text{sem: past}\\ \text{arg: pro1 pro2}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 71}\end{bmatrix}
\begin{bmatrix}\text{sur: you}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 71}\end{bmatrix}$$

The question raised by this example is how the indexical pointers pro1 and pro2 are to be interpreted pragmatically relative to a context of use.

This leads to the second step of modeling the indexical reference mechanism. It is based on combining a propositional content with a cluster of parameter values of the agent's current STAR (Hausser 1999 Sect. 5.3). The STAR is an acronym for (i) location (Space), (ii) time (Time), (iii) self-identity (Agent), and (iv) intended addressee (Recipient).

The STAR has two functions: (a) keeping track of the agent's current situation (orientation) and (b) providing referents for indexicals occurring in contents.[19] A STAR is coded as a proplet, with the A value serving as the core value and as the owner. In a word bank, a temporal sequence of STARs records the output of the agent's on-board orientation system and is listed as a token line.[20]

7.3 Token Line Example of STARs Defined as Proplets

$$\begin{array}{cccccc}
 & & \textit{member proplets} & & \textit{now front} & \textit{owner value}\\
\begin{bmatrix}\text{S: kitchen}\\ \text{T: } t_1 - t_2\\ \text{A: Sylvester}\\ \text{R: Speedy}\\ \text{3rd:}\\ \text{prn: 63–70}\end{bmatrix} &
\begin{bmatrix}\text{S: kitchen}\\ \text{T: } t_{2+1} - t_3\\ \text{A: Sylvester}\\ \text{R: Speedy}\\ \text{3rd:}\\ \text{prn: 71–78}\end{bmatrix} &
\begin{bmatrix}\text{S: living room}\\ \text{T: } t_{3+1} - t_4\\ \text{A: Sylvester}\\ \text{R: Tweety}\\ \text{3rd: Speedy}\\ \text{prn: 79–82}\end{bmatrix} &
\begin{bmatrix}\text{S: garden}\\ \text{T: } t_{4+1} - t_5\\ \text{A: Sylvester}\\ \text{R: Hector}\\ \text{3rd:}\\ \text{prn: 83–87}\end{bmatrix} &
 & \text{Sylvester}
\end{array}$$

In addition to attributes represented by the letters of the STAR, there is a fifth, called 3rd, for third person indexicals. Though not required for the agent's

[19] Integrating the interpretation of indexicals into the agent's on-board STAR orientation may be seen as an enhancement of Montague (1973), who's sign-oriented approach uses arbitrary parameter values, i.e. $i \in I$ for a moment of time and $j \in J$ for a possible world (space, location), superscripted at the end of a lambda expression. Introducing additional parameters for 1st, 2nd, and 3rd person, as has been suggested, has been made light of by Cresswell's (1972, p. 4) joking proposal of a "next drink parameter." The parameter approach has resurfaced as "Variablism," i.e. the view that names and pronouns should be treated semantically as variables (Cumming 2008).

[20] Instead of the names John, Mary, etc., usually employed in linguistic examples, Sect. 7.3 uses the animation characters of Sylvester the cat, Speedy Gonzales the mouse, Tweety the bird, and Hector the dog, familiar from TV, as an aid to distinguish the individuals pointed at by indexicals.

basic orientation, 3rd is needed to provide the referent for items which are neither 1st nor 2nd person.[21] As indicated by the prn values, e.g. [prn: 63–70], several consecutive propositions may share the same STAR.

In communication, three perspectives on content must be distinguished (Hausser 2011, Chaps. 10, 11). The STAR-0 is the agent's perspective onto its current environment; it need not involve language. The STAR-1 is the agent's speak mode perspective onto stored content as required for language production; if ongoing events are reported directly, the STAR-1 equals the STAR-0. The STAR-2 is the agent's hear mode perspective onto language content as needed for the correct interpretation of indexicals. As an example of a STAR-0 perspective, consider the non-language content corresponding to I hear you.

7.4 Anchoring a Content to a STAR-0

$$\begin{bmatrix}\text{sur:}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{sem: sg}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 63}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: hear}\\ \text{cat: \#n}'\text{\#a}'\text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 63}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 63}\end{bmatrix} \begin{bmatrix}\text{S: kitchen}\\ \text{T: 2013-09...}\\ \text{A: Sylvester}\\ \text{R: Speedy}\\ \text{3rd:}\\ \text{prn: 63}\end{bmatrix}$$

This content differs from Sect. 7.2 because (i) it is nonlanguage (no sur values), (ii) the sem value of the verb is pres (present tense) rather than past, and (iii) a STAR is attached by having the same prn value as the content, here 63.

The STAR-0 shows the perspective of the agent Sylvester on his current environment. The S value specifies the location as the kitchen, the pres value of the verb points at the T value, the indexical pro1 points at the A value Sylvester the cat, and pro2 points at the R value Speedy the mouse.

Next Sylvester realizes the content in language by saying to Speedy I heard you. As time has moved, the language content Sect. 7.2 is anchored to a second, later STAR-0 with the prn value 71 (Sect. 7.3). From these two STAR-0, the agent computes the following STAR-1 perspective for the language content Sect. 7.2.

7.5 Speak Mode Anchoring to a STAR-1

$$\begin{bmatrix}\text{sur: I}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{sem: sg}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 64}\end{bmatrix} \begin{bmatrix}\text{sur: heard}\\ \text{verb: hear}\\ \text{cat: \#n}'\ \text{\#a}'\text{ decl}\\ \text{sem: past}\\ \text{arg: pro1 pro2}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 64}\end{bmatrix} \begin{bmatrix}\text{sur: you}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 64}\end{bmatrix} \begin{bmatrix}\text{S: kitchen}\\ \text{T: 2013-09...}\\ \text{A: Sylvester}\\ \text{R: Speedy}\\ \text{3rd:}\\ \text{prn: 64}\end{bmatrix}$$

[21] According to King (2014), a context consists of time, location, agent, and world. In a STAR, the S corresponds to King's location, T to time, and A to agent. The counterpart to King's world are R (intended recipient, you) and 3rd (everyone and everything that is neither A nor R). However, DBS distinguishes between the STAR parameters as the agent's on-board orientation system and basis for interpreting indexicals, on the one hand, and the context as a selectively activated content in memory, on the other.

The agent's perspective is looking from his present situation back on the stored content Sect. 7.4 and encoding it in language. The content is connected to the agent's current STAR via the common prn value 64. The content differs from that of Sect. 7.4 in (i) the sem value past (rather than pres) of the verb proplet and (ii) the language-dependent sur values of the content proplets.

When the language content I heard you is interpreted by the addressee (recipient), Speedy the mouse uses the content of the language sign and its current STAR-0 to derive the STAR-2 perspective. The result is as follows.

7.6 STAR-2 Perspective in Hear Mode

$$\begin{bmatrix}\text{sur:}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 53}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: hear}\\ \text{cat: \#n}' \text{ \#a}' \text{ decl}\\ \text{sem: past}\\ \text{arg: pro2 pro1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 53}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{noun: pro1}\\ \text{cat: obl}\\ \text{sem: sg}\\ \text{fnc: hear}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 53}\end{bmatrix}
\begin{bmatrix}\text{S: kitchen}\\ \text{T: 2013-09...}\\ \text{A: Speedy}\\ \text{R: Sylvester}\\ \text{3rd:}\\ \text{prn: 53}\end{bmatrix}$$

Speedy as the hearer uses his personal prn value and a different STAR: compared to the STAR of Sylvester, the A and R values are reversed and Sylvester's I heard you is reinterpreted by Speedy's STAR-2 perspective as you heard me.

8 Reference by Generalized Baptism (Name)

In DBS, the reference mechanism of the sign kind name is also implemented as a cognitive operation. It consists of (i) establishing *object permanence*[22] and (ii) generalized *baptism* based on cross-copying between a name and its referent.

Object permanence is implemented as identity by address. It is coded by using an address as the core value of the non-initial proplets, pointing at the proplet representing the initial appearance of the referent.

8.1 Object Permanence by Using Address

$$\begin{array}{cccccc}
 & \textit{member proplets} & & & \textit{now front} & \textit{owner value}\\
\begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{. . .}\\ \text{prn: 83}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{noun: (dog 83)}\\ \text{. . .}\\ \text{prn: 99}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{noun: (dog 83)}\\ \text{. . .}\\ \text{prn: 112}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{noun: (dog 83)}\\ \text{. . .}\\ \text{prn: 131}\end{bmatrix} & & \text{dog}
\end{array}$$

The different prn values indicate that each member proplet is part of a different proposition, allowing different continuation values. The core values (dog 83) of the non-initial member proplets point at the initial proplet, which is the referent and formally recognizable by its *non-address* core value.

A token line like Sect. 8.1 may contain several initial dog referents, each referring to another individual. They are distinguished by their different prn values and the address numbers of the associated coreferent proplets. This is

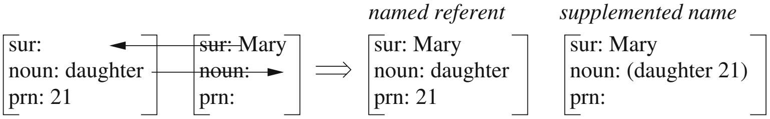
[22] This notion originated in cognitive psychology (Piaget 1954). It may be regarded as a non-truthconditional, non-modal counterpart to "rigid designators" (Kripke 1972).

sufficient for the agent to properly discriminate between different dog referents in cognition and between their sets of coreferent proplets, all in the same token line.

It is not sufficient, however, for language communication. This is because the prn values of referents are not synchronized between agent. What is needed is a name surface and an interagent consensus on which item(s) the name refers to. The consensus is simply achieved: the not yet initiated agent follows the practice observed because communication would break down otherwise.

The DBS implementation is based on (i) a lexical name proplet which has a sur(face) value but no core value and (ii) a referent proplet which has a core value but no sur value. The two proplets are supplemented by an event of generalized baptism which cross-copies the sur value of the name into the sur slot of the referent and the core value of the referent into core slot of the name.

8.2 Baptism as Cross-Copying



The named referent proplet is stored in the token line of the core value and used in the speak mode. The supplemented name proplet is stored in the token line of the surface and used in the hear mode.

The baptizing event is formalized as the following DBS inference.

8.3 Applying the Formal Baptizing Inference

$$\begin{array}{llllll}
\textit{pattern level} & \begin{bmatrix}\text{sur:}\\ \text{noun: } \alpha\\ \text{sem: X}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: } \beta\\ \text{noun:}\\ \text{sem: nm}\\ \text{prn:}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: } \beta\\ \text{noun: } \alpha\\ \text{sem: nm X}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: } \beta\\ \text{noun: } (\alpha \text{ K})\\ \text{sem: nm}\\ \text{prn:}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow \\
\textit{content level} & \begin{bmatrix}\text{sur:}\\ \text{noun: daughter}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: read}\\ \dots\\ \text{prn: 21}\end{bmatrix} & \begin{bmatrix}\text{sur: Mary}\\ \text{noun:}\\ \text{cat:}\\ \text{sem: nm}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{bmatrix} & & \begin{bmatrix}\text{sur: Mary}\\ \text{noun: daughter}\\ \text{cat: snp}\\ \text{s.: nm def sg}\\ \text{fnc: read}\\ \dots\\ \text{prn: 21}\end{bmatrix} & \begin{bmatrix}\text{sur: Mary}\\ \text{noun: (daughter 21)}\\ \text{cat: snp}\\ \text{sem: nm}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{bmatrix}
\end{array}$$

The third content proplet is the named referent, the fourth the supplemented name.

Consider the following word bank containing three referents named Mary, referring to the grandmother, the mother, and the daughter in a family. The token lines are in the alphabetical order *daughter, grandmother, Mary, mother.*

8.4 Name Referring with Multiple Referents

```
                          member proplets                      now front    owner values
                                          ⎡sur: Mary       ⎤
                                          ⎢noun: daughter  ⎥                daughter
                                          ⎢cat: snp        ⎥
                                          ⎢sem: nm f       ⎥
                                          ⎢fnc: sing       ⎥
                                          ⎢. . .           ⎥
                                          ⎣prn: 21         ⎦
⎡sur: Mary        ⎤
⎢noun: grandm.    ⎥                                                         grandmother
⎢cat: snp         ⎥
⎢sem: nm f        ⎥
⎢fnc: cook        ⎥
⎢. . .            ⎥
⎣prn: 7           ⎦
⎡sur: Mary          ⎤ ⎡sur: Mary          ⎤ ⎡sur: Mary             ⎤        Mary
⎢noun: (grandm. 7)  ⎥ ⎢noun: (mother 14)  ⎥ ⎢noun: (daughter 21)   ⎥
⎢cat: snp           ⎥ ⎢cat: snp           ⎥ ⎢cat: snp              ⎥
⎢sem: nm f          ⎥ ⎢sem: nm f          ⎥ ⎢sem: nm f             ⎥
⎢fnc:               ⎥ ⎢fnc:               ⎥ ⎢fnc:                  ⎥
⎢. . .              ⎥ ⎢. . .              ⎥ ⎢. . .                 ⎥
⎣prn:               ⎦ ⎣prn:               ⎦ ⎣prn:                  ⎦
                                          ⎡sur: Mary       ⎤
                                          ⎢noun: mother    ⎥                mother
                                          ⎢cat: snp        ⎥
                                          ⎢sem: nm f       ⎥
                                          ⎢fnc: read       ⎥
                                          ⎢. . .           ⎥
                                          ⎣prn: 14         ⎦
```

The member proplets show the result of three baptism inferences like Sect. 8.3. In the token line of *Mary*, each supplemented name proplet occurs only once.

Supplemented names are not written into the lexicon because a core value like (daughter 21) is not a convention of the natural language at hand. Instead it is the result of a *generalized* baptism: for applying the inference Sect. 8.3,[23] it is sufficient for the uninitiated agent to witness the use of a name. The supplemented name proplets in the *Mary* token line have a lexical quality insofar, however, as they have neither continuation nor prn values – like the lexical proplets resulting from automatic word form recognition.[24]

When the hearer interprets a sentence containing a name, the name activates the corresponding token line, here that of Mary. The choice between different referents, here the grandmother, the mother, and the daughter, may have one of the following results: (i) the chosen referent equals the one intended by the speaker (correct choice), (ii) does not equal the one intended by the speaker (incorrect choice), or (iii) no referent is chosen (inconclusive result). The choice between multiple potential name referents is usually not at random, however. Instead, the referent most suitable to the utterance situation will usually be the correct one. If uncertainty remains, the hearer may ask for clarification.

For an agent in the speak mode, there is no ambiguity.[25] Instead, the speaker selects the intended referent, e.g. (daughter 21). If the agent acquired

[23] As an agent-based approach, DBS does not use causal chains (Kripke 1972).

[24] Our computational reconstruction of reference with the sign kind of names, based on object permanence and generalized baptism, differs from other theories of naming such as the descriptive theory of proper names (Russell 1905) and the rigid designator analysis (Kripke 1972), which are based on a sign-based ontology (FoCL Sect. 20.4). The formal DBS treatment of name-based reference (Sect. 8.2) provides a simple, efficient procedural implementation suitable for building a talking robot.

[25] Even an ambiguity deliberately created in the speak mode ("diplomatic ambiguity," Pehar 2001) arises only for the hearer.

the appropriate name in the hear mode (Sect. 8.3), it is preserved in the word bank and may be used in the speak mode. If the agent is in the position to select and bestow a name, it is also available for realization.

9 Reference by Address (Coreference)

Coreference by address occurs with all three sign kinds. In name-based reference it is the only mechanism for relating the supplemented name to the named referent (Sect. 8.2). In reference by matching (symbol) and by pointing (indexical), in contrast, it is an additional method.

References

Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychol. Rev. **94**, 115–147 (1987)

de la Briandais, R.: File searching using variable length keys. In: Proceedings of Western Joint Computer Conference, vol. 15, pp. 295–298 (1959)

Burgess, A.: Metalinguistic descriptivism for Millians. Australas. J. Philos. **91**, 443–457 (2012)

Chen, F.: Human factors in speech interface design. In: Designing Human Interface in Speech Technology, pp. 167–224. Springer, New York (2006)

Chisvin, L., Duckworth, R.J.: Content-addressable, associative memory. In: Yovits, M.C., (ed.) Advances in Computer Science, 2nd edn, pp. 159–235. Academic Press (1992)

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press, Cambridge (2009)

Cresswell, M.: The world is everything that is the case. Australas. J. Philos. **50**, 1–13 (1972)

Cumming, S.: Variabilism. Philos. Rev. **117**(4), 525–554 (2008)

Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 6th edn. Benjamin-Cummings, Redwood City (2010)

Frege, G.: Über Sinn und Bedeutung. Zeitschrift für Philosophie und philosophische Kritik. **100**, 25–50 (1892)

Handl, J.: Entwurf und Implementierung einer abstrakten Maschine für die oberflächenkompositionale inkrementelle Analyse natürlicher Sprache, Diplom thesis, Department of Computer Science, University of Erlangen Nürnberg (2008)

Handl, J.: Inkrementelle Oberflächenkompositionale Analyse und Generierung Natürlicher Sprache. Inaugural dissertation, CLUE, Univ. Erlangen Nbg (2012). http://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/3933

Hausser, R.: Computation of Language, An Essay on Syntax, Semantics, and Pragmatics in Natural Man-Machine Communication. Artificial Intelligence Symbolic Computation. Springer, Heidelberg (1989)

Hausser, R.: Complexity in left-associative grammar. Theor. Comput. Sci. **106**(2), 283–308 (1992)

Hausser, R.: Foundations of Computational Linguistics. Human-Computer Communication in Natural Language. Springer, Heidelberg (1999). (3rd edn, 2014)

Hausser, R.: Memory-based pattern completion in database semantics. Lang. Inf. **9**(1), 69–92 (2005)

Hausser, R.: A Computational Model of Natural Language Commmunication: Interpretation, Inference, and Production in DBS. Springer, Heidelberg (2006)

Hausser, R.: Computational Linguistics and Talking Robots: Processing Content in DBS. Springer, Heidelberg (2011)

Kass, R., Finin, T.: Modeling the user in natural language systems. Comput. Linguist. **14**, 5–22 (1988)

King, J.: Speaker intentions in context. Noûs **48**(2), 219–237 (2014)

Kripke, S.: Naming and necessity. In: Davidson, D., Harmann, G. (eds.) Semantics of Natural Language, pp. 253–355. D. Reidel, Dordrecht (1972)

McKeown, K.: Discourse strategies for generating natural-language text. Artif. Intell. **27**, 1–41 (1985)

Modrak, D.: Aristotle's Theory of Language and Meaning. Cambridge University Press, New York (2001)

Montague, R.: The proper treatment of quantification in ordinary English. In: Hintikka, J., Moravcsik, J., Suppes, P. (eds.) Approaches to Natural Language, pp. 221–242. D. Reidel, Dordrecht (1973)

Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs (1972)

Peirce, C.S.: Collected papers. In: Hartshorne, C., Weiss, P. (eds.). Harvard University Press, Cambridge (1931–1935)

Pehar, D.: Language and Diplomacy. Lambert, Saarbrücken (2001)

Pelczar, M., Rainsbury, J.: The indexical character of names. Synthese **114**, 293–317 (1998)

Piaget, J.: The Construction of Reality in the Child. Basic Books, New York (1954)

Quine, W.V.O.: Word, Object. MIT Press, Cambridge (1960)

Recanati, F.: Direct Reference: From Language to Thought. Blackwell, Oxford (1997)

Reimer, M., Michaelson, E.: Reference. In: Zalta, E. (ed.) The Stanford Encyclopedia of Philosophy, Winter 2014 edn. http://plato.stanford.edu/entries/reference/

Russell, B.: On denoting. Mind **14**, 479–493 (1905)

Steels, L.: The Talking Heads Experiment, limited Pre-edition for the Laboratorium Exhibition, Antwerpp (1999)

Weiss, M.A.: Data Structures and Problem Solving Using Java, 3rd edn. Pearson Addison-Wesley, Upper Saddle River (2005)

Wiener, N.: Cybernetics, or Control and Communication in Animal and Machine, 2nd edn. MIT press, Cambridge (1961)