

Concept, Indexical, and Name as Software Mechanisms

Roland Hausser

Universität Erlangen-Nürnberg (em.)

©Roland Hausser, January 7, 2021

Abstract

The semantics of agent-based DBS is ‘grounded’ in that the Content kinds *concept*, *indexical*, and *name* have their foundation in the agent’s recognition and action, whereby each Content kind has its own computational Mechanism. For a concept it is pattern *matching* between the type provided by memory and raw data provided by the agent’s interface component. For an indexical it is *pointing* at a STAR value of the agent’s on-board orientation system (OBOS). For a name it is the address of the ‘named referent’ which is inserted as the core value into a lexical name proplet in an act of *baptism*.

Orthogonal to the Content kinds and their computational Mechanisms are the Semantic kinds *referent*, *property*, and *relation* with their associated Syntactic kind *noun*, *adj* and *intransitive verb*, and *transitive verb*. It is shown that the Semantic kind of *referent* is restricted to the Syntactic kind of noun, but utilizes the computational Mechanisms of matching, pointing, and baptism. Furthermore, figurative use is restricted to the computational Mechanism of matching, but uses the Semantic kinds referent, property, and relation.

keywords: Syntactic kinds of noun, adj, verb; Semantic kinds of referent, property, relation; Content kinds of concept, indexical, name; computational Mechanisms of matching, pointing, baptism; recognition vs. action; type vs. token; speak mode vs. hear mode; language vs. nonlanguage content

1 Apparent Terminological Redundancy

The notions noun, verb, and adjective from linguistics (philology) have counterparts in analytic philosophy, namely referent, relation, and property, and in symbolic logic, namely argument, functor, and modifier:

1.1 THREE TIMES THREE RELATED NOTIONS

(a) <i>linguistics</i>	(b) <i>philosophy</i>	(c) <i>symbolic logic</i>
1. noun	referent (object)	argument
2. verb	relation	functor
3. adj	property	modifier

We take it that these variants are not merely different terms for the same things, but different terms for different aspects of the same things. In particular, the linguistic terminology may be viewed as representing the syntactic aspect, the philosophical terminology as representing the associated semantic aspect, and the logical terminology as a preparatory step towards a computational implementation.

In DBS, the distinctions are related as follows:

1.2 1ST CORRELATION: SEMANTIC AND SYNTACTIC KIND

<i>Semantic kind</i>	<i>Syntactic kind</i>
1. referent	noun
2. property	adn, adv, adnv, intransitive verb
3. relation	transitive verb

The Semantic kinds *referent*, *property*, and *relation* correspond to argument, 1-place functor, and 2- or 3-place functor, respectively, in Symbolic Logic. Syntactically, *property* splits up into adn, adv, prepnoun, and 1-place verb. *Relation* splits up into 2- and 3-place verbs.

The distinction between (i) Semantic and (iii) Syntactic kinds is complemented by a second, orthogonal pair of triple distinctions, namely (ii) Content kinds and associated (iv) computational Mechanisms:

1.3 2ND CORRELATION: CONTENT KIND AND COMPUT. MECHANISM

<i>Content kind</i>	<i>computational Mechanism</i>
a. concept	matching
b. indexical	pointing
c. name	baptism

The terms of the three Content kinds and the correlated Mechanisms have had informal use in the literature,¹ but without an agent-based ontology. The essential points of the Mechanisms in DBS are their obvious computational realizations (Sects. 4–6), which have not been utilized until now.

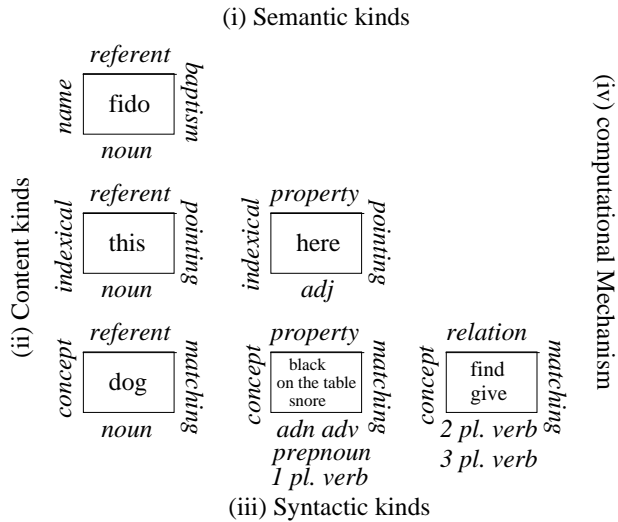
The dichotomies 1.2 and 1.3 provide 12 (2×2×3) basic notions. Empirically, they combine into six classes of proplets which constitute the semantic building blocks of DBS cognition in general and natural language communication in particular. The six classes form what we call the *cognitive square*²:

1.4 COGNITIVE SQUARE OF DBS

		(i) Semantic kinds					
		<i>referent property relation</i>					
(ii) Content kinds	<i>name</i>	Fido			<i>baptism</i>	(iv) computational Mechanism	
	<i>indexical</i>	this	here				<i>pointing</i>
	<i>concept</i>	dog	black on the table snore	find give			<i>matching</i>
		<i>noun</i>	<i>adn adv</i>	<i>2. pl. verb</i>			
		<i>prepnoun</i>		<i>3. pl. verb</i>			
		<i>1 pl. verb</i>					
		(iii) Syntactic kinds					

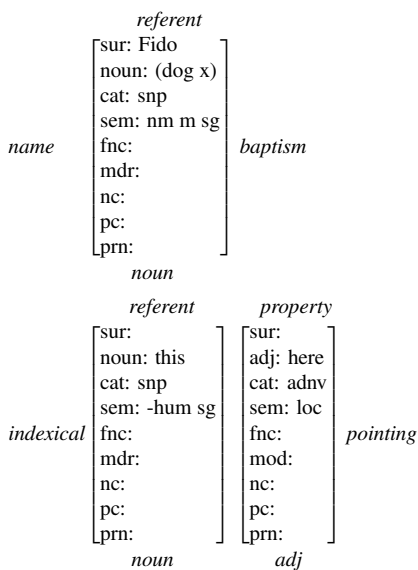
The twelve basic notions of this NLC 2.6.9 extension are distributed over six basic proplets kinds such that no two are characterized the same:

1.5 CLOSER VIEW OF THE COGNITIVE SQUARE



The surfaces inside the rectangles have the following proplet definitions:

1.6 PROPLETS INSTANTIATING THE COGNITIVE SQUARE OF DBS



¹Examples of precomputational uses are (i) matching for concepts but without the type-token relation and its computational implementation based on content and pattern proplets, (ii) pointing for indexicals but without an on-board orientation system (OBOS), and (iii) baptism but without the named referent as the core value for use in the speak and the hear mode.

²'Triangle' would be appropriate as well, but the term "cognitive triangle" is already used by the cognitive behavioral therapists (CBT). Earlier it was used by Ogden&Richards (1923) for their "Semiotic Triangle." The term 'square' is well suited to express the orthogonal relation between the Syntactic_kinds-Semantic_kinds and the Content_kinds-computational_Mechanisms.

	<i>referent</i>	<i>property</i>		<i>relation</i>			
<i>concept</i>	[sur: noun: dog cat: snp sem: def sg fnc: mdr: nc: pc: prn:]	[sur: adj: black cat: adv sem: pad fnc: mod: nc: pc: prn:]	[sur: noun: table cat: adv snp sem: on def sg fnc: mdr: nc: pc: prn:]	[sur: verb: snore cat: n-s3' v sem: pres arg: mdr: nc: pc: prn:]	[sur: verb: find cat: n-s3' a' v sem: pres arg: mdr: nc: pc: prn:]	[sur: verb: give cat: n-s3' d' a' v sem: pres arg: mdr: nc: pc: prn:]	<i>matching</i>
	<i>noun</i>	<i>adj</i>	<i>prepnoun</i>	<i>verb</i> 1-pl	<i>verb</i> 2-pl	<i>verb</i> 3-pl	

In a proplet, the Semantic kind *referent* is limited to the core attribute *noun*, *property* is limited to the core values *adn*, *adv*, *adnv* and to verbs characterized by their *cat* value as intransitive, and *relation* is limited to verbs characterized by their *cat* value as transitive.

The Content kind *name*, *indexical*, and *concept* is specified by the core value of a proplet. The corresponding computational Mechanisms *baptism*, *pointing*, and *matching* are implemented by inserting a ‘named referent’ as core value into *names*, by indexicals pointing at a STAR value of the onboard orientation system, (OBOS) and by computational type-token matching in the case of *concepts*.

The cognitive square of DBS is empirically important because (i) figurative use is restricted to concepts, i.e. the bottom row in 1.4–1.6, and (ii) reference is restricted to nouns, i.e. the left-most column. Thus only concept nouns may be used both figuratively and as referents, while indexical properties like *here* and *now* may not be used as either, and names only as referents.

2 Restriction of Figurative Use to Concepts

To show the restriction of figurative use to the Content kind *concept* let us go systematically through the three Semantic kinds:

2.1 THREE CONTENT KINDS FOR THE SEMANTIC KIND *REFERENT*

<i>concept</i>	<i>indexical</i>	<i>name</i>
[sur: noun: animal cat: sn ...]	[sur: noun: pro2 cat: sp2 ...]	[sur: tom noun: [person x] cat: snp ...]

The three Content kinds of the Semantic kind *referent* all have literal use, but only the concepts allow figurative use.

Next consider the Semantic kind *property*, which occurs as the Content kinds (i) *concept* and (ii) *indexical*, but not as *name*. Property proplets of the content kind *concept* may have the core attributes *adj*, *noun*, or *verb* if it is 1-place (1.6). If they have the core value *adnv*, they may be (a) elementary (*fast*) or phrasal (*in the park*), and (b) adnominal (*tree in the park*) or adverbial (*walk in the park*).³

³Phrasal modifiers, called prepnouns in DBS, are derived from a referent by means of a preposi-

2.2 TWO CONTENT KINDS FOR THE SEMANTIC KIND *PROPERTY*

<i>concept</i>	<i>indexical</i>	<i>name</i>
[sur: adj: great cat: adn ...]	[sur: adj: now cat: adnv ...]	
[sur: adj: enough cat: adnv ...]	[sur: adj: here cat: adnv ...]	
[sur: noun: table cat: snp ...]	[sur: noun: pro2 cat: sp2 ...]	
[sur: verb: melt cat: n-s3' v ...]		

Of the Semantic kind *property*, only the concepts may be used nonliterally.⁴

The Grammatical kind *transitive verb* with its single Semantic kind *relation* exists as the Content kind *concept*, but not as *indexical* or *name*:

2.3 ONE CONTENT KIND FOR THE SEMANTIC KIND *RELATION*

<i>concept</i>	<i>indexical</i>	<i>name</i>
[sur: verb: steal cat: n-s3' a' v ...]		
[sur: verb: give cat: n-s3' d' a' v ...]		

Being concepts, *transitive verbs* have literal and nonliteral use.

3 Additional Constraint on Figurative Use

The restriction of figurative use to concepts is constrained further by the condition that the literal term and its figurative counterpart must be grammatically equivalent:

tion or an affix, depending on the language. Therefore, a prenoun like in the park refers by means of park, in contradistinction to the other modifiers, e.g. elementary fast or intransitive snore, which do not refer.

⁴For a nonliteral use of *on the table* see CC 9.2.4 and of *melt* CC 9.5.1.

3.1 INVARIANCE CONSTRAINT

A figurative use and its literal counterpart must be of the same Syntactic and Semantic kind.

Thus, one cannot use a 1-place verb like *bark* to refer figuratively to a *dog* unless *bark* is nominalized, as in *the little barker* (i.e. by turning the property of intransitive *bark* into the referent *barker*, *sleep* into *sleeper*, *stink* into *stinker*, etc.). Similarly for the adj *fat*, which for figurative use must be nominalized, as in *the old fatso*. Functionally, the constraint helps the hearer to find the literal counterpart of a figurative use by reducing the search space.

The systematic examples in CC Chap. 9 all satisfy the invariance constraint:

3.2 SYNTACTIC-SEMANTIC INVARIANCE OF FIGURATIVE USE

Semantic kind	Syntactic kind	nonliteral use	literal counterpart	in CC
noun	referent	animal	dog	9.1.2
property	prepnoun	on the table	on the orange crate	9.2.1
property	adn	great	greater than average	9.6.3
property	adv	enough	more than enough	9.6.6
property	intransitive verb	melt	disappear	9.5.1
relation	transitive verb	steal	take over	9.4.2

The Semantic kind *property* has several Syntactic kinds, while each Syntactic kind, e.g. *prepnoun*, has only of one Semantic kind, i.e. *property*, regardless of whether it is used literally or figuratively. The other two Semantic kinds, i.e. *referent* and *relation*, each have only a single syntactic counterpart.

As an example of using all three Semantic kinds figuratively consider the following description of a dog contorting itself catching a frisbee in mid air:

3.3 EXAMPLE USING ALL THREE SEMANTIC KINDS FIGURATIVELY

The animal flew acrobatically towards the disc.

The content obeys the invariance constraint: literal *dog* and figurative *animal* are both singular nouns, literal *jumped* and figurative *flew* are both finite verbs in the indicative past, literal *in a spectacular gymnastic feat* and figurative *acrobatically* are both adverbials (one phrasal, the other elementary), and literal *frisbee* and figurative *disc* are both singular nouns. For successful communication, the hearer-reader must relate figurative *animal* to literal *dog* and figurative *disc* to literal *frisbee*. The relation *flew* and the property *acrobatically*, in contrast, do not refer directly, but only indirectly via their function as relation or modifier.

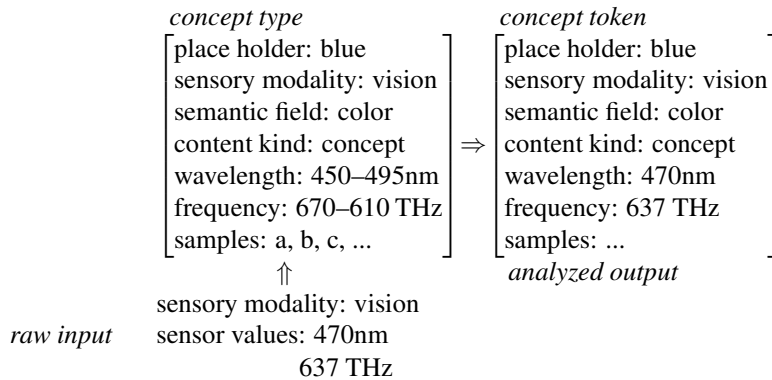
4 Declarative Specification of Concepts for Recognition

Concepts are the only Semantic kind which interacts directly with the agent's cognition-external environment. The interaction consists of matching between (i)

raw data provided by sensors and activators of the agent’s interface component and (ii) concept types provided by the agent’s memory.⁵

Consider the rule for the recognition of a color:

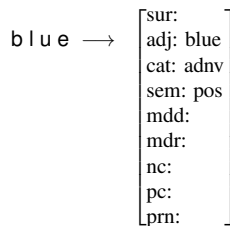
4.1 DECLARATIVE SPECIFICATION FOR RECOGNIZING THE COLOR blue



The raw input data 470nm and 637 THz are provided by the agent’s interface component and recognized as the color **blue** because they fall into the type’s wavelength interval of 450–495nm and the frequency interval of 670–610 THz. The analyzed output token results from replacing the wavelength and frequency intervals of the type with the raw data measurements of the input.

The place holder value of the recognized token, i.e. the letter sequence **b l u e**, is used for lookup of the lexical proplet which contains the place holder as its core value (CC 1.6.3):

4.2 PLACE HOLDER VALUE OF CONCEPT USED FOR LEXICAL LOOKUP



Like the concept type, the proplet is retrieved from the artificial agent’s memory (on-board database). Computationally, the lookup is based on string search (Knuth et al. 1977) in combination with a trie structure (Briandais 1959).

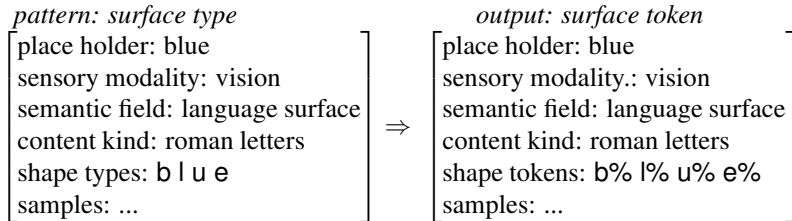
The language counterpart to the recognition of nonlanguage concepts is the interpretation of language-dependent surfaces. As an example, consider the DBS robot’s recognition of the letter sequence **blue** by matching raw visual input data with letter patterns as shape types, resulting in a surface token:

⁵From a theory of science point of view, computational pattern matching based on the type-token relation constitutes a fruitful interaction between the humanities and the sciences.

4.3 SENSOR INTERACTING WITH LANG. CONCEPT TYPE (SURFACE)

sensory modality: vision
input: raw data

↓

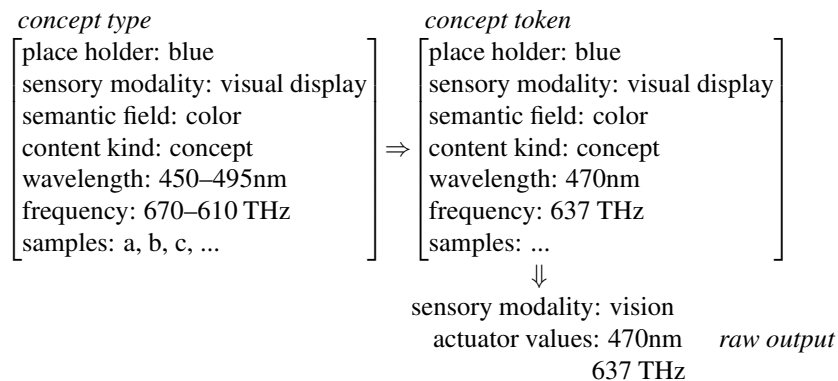


Raw data input is matched by the shape types of the letters b l u e. The output replaces the matching shape types with the shape tokens b% l% u% e%⁶ to record such accidental properties as the font, size, color, etc. in the sensory medium of print, and pronunciation, pitch, speed, loudness, etc. in the sensory medium of speech. The shape types are used (i) for matching the raw data and (ii) for look-up of the lexical definition. For developing the linguistic side of automatic word form recognition, the type-token matching of raw data in different media may be cut short temporarily by typing letters directly into a standard computer.

5 Declarative Specification of Concepts for Action

The action counterpart to the recognition of nonlanguage concepts is their cognition-external realization as raw data. It consists in adapting a type to the agent's purpose as a token which is passed to the appropriate actuator. As an example, consider a cuttlefish (*Metasepia pfefferi*) turning on the color blue:

5.1 RULE FOR PRODUCING THE COLOR blue

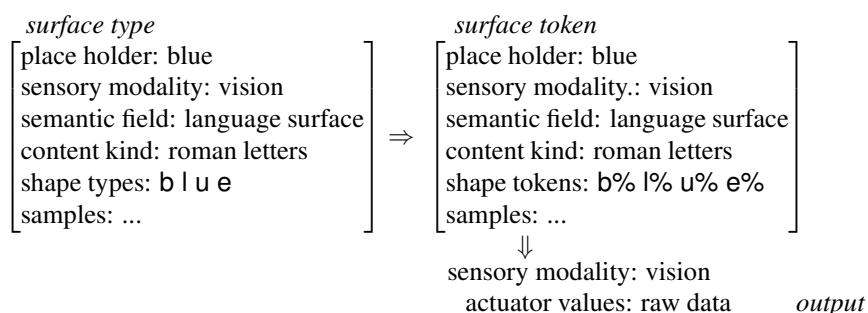


The type is adapted into a token by replacing the wavelength interval of 450–495nm and frequency interval of 670–610 THz with the agent-selected values of 470nm and 637 THz. In the cuttlefish, these values are realized by natural actuators

for color control (chromatophores) as raw data.

The language counterpart to a nonlanguage action is the realization of a language-dependent surface in a medium of choice. As an example, consider the DBS robot's production of the surface **blue** as raw data in vision:

5.2 REALIZING LETTER TOKENS AS RAW DATA IN VISION MEDIUM



The input to the actuator consists of a sequence the shape tokens representing roman letters. The output replaces the shape tokens, here **b% l% u% e%**, with matching raw data, for example on a computer screen.

6 Indirect Grounding of Indexicals and Names

In DBS, the second computational Mechanism of *indexicals* is pointing at STAR values of the agent's on-board orientation system (OBOS) and is as such cognition-internal. However, because the STAR values originate as concept recognitions, past or present (CC Chaps. 7, 8), indexicals rely indirectly on the Mechanism of computational pattern matching. More specifically, the indexical **pro1** points at the **A** value of the STAR, **pro2** at the **R** value, **pro3** at 3rd value, **here** at the **S** value, and **now** at the **T** value (with **S** and **T** values nominalized).

The third computational Mechanism of *names* relies on an act of baptism, which inserts the 'named referent' as the core value into the lexical name proplet (CTGR). Because the named referent originates as concept recognition, names – like indexicals – rely indirectly on the first computational Mechanism of *concepts*, i.e. computational pattern matching.

After working out the basic functioning of the computational Mechanism for the recognition and action of certain concepts in a codesigned but real environment, more concepts of the same kind (semantic field) may be added routinely, as shown by the following example:

⁶For ease of illustration, the letter shapes are represented by the letters themselves, e.g. **e** (type) and **e%** (token).

6.1 SIMILARITY AND DIFFERENCE BETWEEN COLOR CONCEPT TYPES

place holder: red sensory modality: vision semantic field: color content kind: concept wavelength: 700-635 nm frequency: 430-480 THz samples: a, b, c, ...	place holder: green sensory modality: vision semantic field: color content kind: concept wavelength: 495-570 nm frequency: 526-606 THz samples: a', b', c', ...	place holder: blue sensory modality: vision semantic field: color content kind: concept wavelength: 490-450 nm frequency: 610-670 THz samples: a'', b'', c'', ...
--	---	---

Once the recognition and action side of these concepts is working as intended, more colors may be easily added as an efficient, transparent upscaling.⁷

Similarly for geometric forms: once the concepts of square (CC 1.3.2) and rectangle work as intended, more two-dimensional forms, such as triangle, heptagon, hexagon, and rhombus, may be added routinely. After implementing the concept pick including the associated hand-eye coordination and the semantic relation of object\predicate (CC 2.5.1, 2), the robot should be able to execute language-based requests like Pick the blue square or Pick the green rectangle correctly from a set of items in its task environment.

Conclusion

In data-driven agent-based DBS, recognition and action must be *grounded* in the form of a computational interaction between raw data and a robot's digital cognition. For practical reasons, grounding may be temporarily suspended by the shortcut of typing place holder values into a standard computer's key board and displaying output on the screen. This allows systematic upscaling of an artificial cognition even today, yet prepares for integrating operational core values for referents, properties, and relations when they become available in robotics.

Computational upscaling has two basic aspects: the *declarative specification* and the *procedural implementation*. A declarative specification must be both, (i) easily readable by humans and (ii) easily translatable into a general purpose programming language like Lisp, C, Java, or Perl. From a humanities point of view, a declarative specification must represent the necessary properties of a software solution by omitting the accidental properties which distinguish the individual programming languages and make them difficult to read. Methodologically, a procedural implementation complements a declarative specification with automatic verification, which supports systematic incremental upscaling.

⁷Set-theoretically, the colors red, green, and blue are (i) disjunct and (ii) subsets of color. This structure is inherent in the color concepts, but regardless of being true, it is neither the only nor the predominant aspect of their meaning: knowing that red and green are disjunct, for example, is not sufficient for naming these colors correctly.

Bibliography

- Briandais, R. de la (1959) "File searching using variable length keys," *Proc. Western Joint Computer Conf. 15*, 295–298
- CC = Hausser, R. (2019) *Computational Cognition; Integrated DBS Software Design for Data-Driven Cognitive Processing*, lagrammar.net
- CTGR = Hausser, R. (2017a) "A computational treatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26.
- Knuth, D.E., J.H. Morris, and V.R. Pratt (1977) "Fast Pattern Matching in Strings," *SIAM Journal of Computing* Vol. 6.2:323–350
- NLC = Hausser R. (2006) *A Computational Model of Natural Language Communication; Interpretation, Inferencing, and Production in Database Semantics*, Springer, pp. 360; 2nd ed. 2017, pp. 363, lagrammar.net
- Ogden, C.K., and I.A. Richards (1923) *The Meaning of Meaning*, London: Routledge and Kegan Paul
- Quillian, M. (1968) "Semantic memory," in M. Minsky (ed.), *Semantic Information Processing*, p. 227–270, Cambridge, MA: MIT Press
- Quine, W.v.O. (1960) *Word and Object*, Cambridge, Mass.: MIT Press
- TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis; DBS Software Design for the Hear and the Speak mode of a Talking Robot*, pp. 318. DOI: 10.13140/RG.2.2.13035.39200, lagrammar.net