Reasoning in Natural Language in Using Combinatory Logic and Topology
An Example with Aspect and Temporal Relations

Jean-Pierre Desclés
LaLIC, University of Paris-Sorbonne
Jean-Pierre.descles@paris-sorbonne.fr

Abstract
We are studying how Curry’s Combinatory Logic can be used for giving an adequate analysis of different grammatical problems such as diatheses, tenses and aspects, and lexical analyses by formal representations of meanings of verbal predicates and prepositions. The paper intends to show how Combinatory Logic can solve on the one hand, the formal representations of tenses and aspects in natural languages with the help of the topology of intervals and, on the other hand, the problem of the synthesis of a lexical predicate from a formal description of its meaning by means of a semantic cognitive scheme (SCS). We want to explain, in following an example, how can be explained the “natural” inference between two utterances like John took the Mary’s pen -> Now, John has got the pen.

1. Introduction
Combinatory Logic (CL) of Curry (1958, 1972) is used for giving an adequate analysis of different grammatical problems such as diatheses, tenses and aspects, lexical analyses by formal representations of meanings of verbal predicates and grammatical operators (Desclés, 1990, 2004, 2005). Conceptions of aspects and tenses are given by several logicians an philosophers (H. Reichenbach (1947) or Z. Vendler (1967) are criticisable works quoted and used by a lot of linguists) but, according to us, these analyses do not give an operational way yielding to an automatic processing. This paper intends to show how CL is an useful tool for solving on the one hand, the formal representations of temporal relations and aspects in natural languages and, on the other hand, the synthesis of a lexical predicate from a formal description of its meaning by a scheme defined with the help of semantic and cognitive primitives.

An example of a “natural inference” between two utterances like John took the Mary’s pen -> Now, John has got the pen will be used to explain the different steps of a formal processing. As far as we are concerned, the formal analyses we use are anchored onto the grammatical conceptions developed by linguists as E. Benveniste (1974), B. Comrie (1976), J.A. Mourelatos (1981), Culioli (1999), Desclés (1989), Desclés & Guentchéva (1995, 2008), with semantic notions such that event, state, unaccomplished process, accomplished (or completed) process, resultative state
\[1\]...We have formalized (Desclés, 2004, 2005) these notions by means of aspectual and temporal operators in using on the one hand, topological intervals of instants, on the other hand, the formal framework of the CL. For illustrating our formal approach, we represent the meaning of a lexical predicate take \((x,y)\) with a scheme defined as an applicative cognitive representation of cognitive primitives; we show how this predicate can be obtained by a synthesis processing in using combinator of CL; we show also how aspectual and temporal operators can be represented in the same applicative formalism to explain an inferential reasoning by a formal calculus. The analysis of meaning and the formal processing of aspects and temporal relations are general and are not specific to the given example. We give here a formal explanation of the semantic relations between a “past event” (John took the Mary’s pen) and its “resultative state” (Now, John has got the pen), involving changes in the lexical forms. Other examples can be analyzed by the same device, for instance Yesterday, the hunter killed the deer -> Now, the deer is dead.

We cannot present, in this short paper, neither all the steps of the formal calculus, nor the types of different operators and the types schemes of combinators.

\[1\] Note that the terms of ‘activity’, ‘process’, ‘accomplishment’ and ‘achievement’, used in this paper, should not be taken in Z. Vendler's sense (1967).
2. Combinatory Logic and Combinators

Combinatory Logic (CL), developed principally by H.B. Curry (1958, 1972), Hindley & Seldin (1986) - see also Quine (1960) - is a logic of operators founded on the binary operation called application: an operator ‘X’ is applied, by means of the operation ‘@’, to an operand ‘Y’; the result is an applicative expression, noted ‘X @ Y’ or ‘XY’. CL is a relatively adequate formalism for giving formal representations of linguistic utterances and, more generally, to analyse different problems in Linguistics, Philosophy, Artificial Intelligence, Cognitive Sciences, and Nanosciences. CL is very similar to the Church’s λ-calculus; but not completely equivalent for the following reasons: (i) CL uses abstract operators, called combinator s, allowing to combine intrinsically other operators, independently from meanings of combined operators; (ii) CL does not use bound variables as λ-calculus does; thus, the implementation is easier because there is no side effect; (iii) CL is equivalent to the λ-calculus in extension but not in intension (Hindley 1986). CL verifies the Church-Rosser’s property (the property of confluence) according to which a reduced form (without combinator s), a “normal form”, is unique, if it exists. CL allows inside of a same computational architecture, a formal articulation between different representation levels during a processing; this situation is usual in Artificial Intelligence and computer science in a compiling process. Thanks to these properties about reduction and changing of representation levels, CL is used to express, by means of a formal calculus, a synthesis of a lexical (or grammatical) predicate from a formal representation of its meaning. CL and functional types of Church are adequate to study natural languages when linguistic units are considered as operators of different types, like in Categorial Grammar (CG) and its extensions (Biskri & Desclés, 2005), the theory of operators of Z. Harris (1982), Applicative Grammar of S.K. Shaumyan (1977, 1987) and Applicative and Cognitive Grammar (ACG) of J-P. Desclés (1990, 2004).

The action of each combinator of CL is defined by a β reduction rule. The actions of elementary combinator s are as follows:

- **I** (identity):
  \[ I \ X \ \overset{\beta}{=} \ X \]

- **B** (functional composition):
  \[ B \ X \ Y \ Z \ \overset{\beta}{=} X \ (Y \ Z) \]

- **C** (conversion):
  \[ C \ X \ Y \ Z \ \overset{\beta}{=} X \ Z \ Y \]

- **C* (type raising):**
  \[ C^* \ X \ Y \ \overset{\beta}{=} Y \ X \]

- **W** (diagonalization):
  \[ W \ X \ Y \ \overset{\beta}{=} X \ Y \ Y \]

- **K** (cancelling an argument):
  \[ K \ X \ Y \ \overset{\beta}{=} X \]

- **S** (composition and duplication):
  \[ S \ X \ Y \ Z \ \overset{\beta}{=} X \ Z \ (Y \ Z) \]

- **Φ** (application in parallel):
  \[ Φ \ X \ Y \ Z \ U \ \overset{\beta}{=} X \ (Y \ U) \ (Z \ U) \]

- **Ψ** (distribution):
  \[ Ψ \ X \ Y \ Z \ U \ \overset{\beta}{=} X \ (Y \ Z) \ (Y \ U) \]

The action of a combinator can be presented in the Gentzen’s style, by an introduction and by an elimination rules (Fitch 1974). From a combinatorical ‘X’, we derive the iterate combinator ‘X^n’, which is the functional composition, by B, in n steps of the combinatorical ‘X’ with itself and the combinatorical ‘X^n’, acting at distance above n terms (Curry 1958 ) ; Desclés, 1990 ).

3. Formal processing of an aspectual and temporal analysis

How to explain the intuitive inference between the event (John took the Mary’s pen) and its resultative state (John has got the pen) by means of a formal calculus? This semantic inference is carried out at several steps; we present them briefly, in a bottom up analysis, without giving, here, all the formal steps. We start with the utterance (1):

(1) **John took the Mary’s pen**

Applicative and Combinatory Categorial Grammar (Biskri & Desclés, 2005) is an operational process that builds an applicative expression with operators applied to operands (written with a prefixed notation, as all other applicative expressions in this paper):

(2) (took (the (‘s (Mary) pen))) John

The operational role of linguistic units is indicated by a syntactic functional type that takes into account the concatenations of operator positions with regard to operands.

To analyze of the finite form took of the verb to take, we define operators of English including the past verb suffix for regular verbs; to simplify notations, we use the following identifications:

- \[ <T^2 := the \ Mary’s \ pen> \]
- \[ <T^1 := John> \]
- \[ <P2 := to \ take> \]

\[ (V_{conjugate-past} \ T^2) \ T^1 \]

i.e. predicative relation containing the past conjugated verb ;

\[ (P_{past-suffix} \ T^2) \ T^1 \]

i.e. predicative relation of the infinitive with the past verb suffix ;

\[ (even_{past} P \ T^2) \ T^1 \]

i.e. the aspectual operator directly associated to the morphological operator past.

---

2 The verb to take being irregular, the metalinguistic operator past suffix is interpreted here as a vowel interchange of the stem.
4. Aspectual operators

From a general viewpoint, *Aspect* is analysed as an operator ‘ASP,’ applied to a predicative relation ‘P₂T² T¹,’ hence an *aspectual predicative relation ‘ASP₁* (P₂T² T¹), realized onto a topological interval ‘I’ of instants (that is: true at each instant of this interval ‘I’). Different aspectual values of ‘ASP’ are concerned and defined in (Descles, 1989, 2005; Descles & Guentchéva, 1995, 2008): *state (STATE)* or *process (PROC)* or *event (EVEN).* For instance, in (1), the value of the aspect is an event. According to the values of the aspectual operator ‘ASP,’ the topological type of the interval ‘I’ is specified: it is closed in an event, open in a state and closed to the left and open to the right.

How to represent together, on one hand, the meaning of the aspectual operator (with its aspectual value) applied of the predicative relation and, on the other hand, how to localize the aspectual predicative relation with respect the speaking act? Indeed, a speaking act is not a punctual event but an *unaccomplished process* realized on the interval ‘J₀’ of instants, where the left boundary is closed and the right boundary ‘T’ of ‘I’ (I’ is an open (Descles, 1989, 2005). The *speaking operator ‘ENONC J₀* (or enunciative operator ‘I-am-saying’) is the result of the functional composition of the process ‘PROC J₀,’ with the speech-act operator ‘(SAY (…) (EGO))’ - where ‘EGO’ is an abstract symbol for representing any speaker.

4.1. In the example (1), chosen for illustrating the formal analysis, the predicative relation is an operand of an operator ‘even_past’ which means an ‘event-in-the-past’ and implies a more abstract grammatical aspectual operator ‘EVEN,’ hence the expression (3): (3) \[ \text{EVEN} (P₂ T² T¹) \]

This aspectual operator ‘EVEN’ specifies the zone of validation of the predicative event realized on some closed interval ‘F.’ The meaning of the operator ‘EVEN’ is an applicative combination of: (i) the speaking operator ‘ENONC J₀,’ (ii) an event operator ‘EVENₚ,’ where ‘F’ is an interval where the predicative is realized, and (iii) a temporal constraint establishing a relation between the intervals ‘F’ and ‘J₀.’ ‘F’ is before ‘J₀,’ \( (\delta(F) < \gamma(J₀)) \). The applicative combination is expressed by means of a non elementary combinator ‘X’ which express a functional program to compose the operators ‘ENONC J₀’ and ‘EVENₚ’ with a temporal constraint. The grammatical and abstract aspectual operator ‘EVEN’ is defined by introduction of an existential quantification, yielding to (4) (where the temporal relation \( \delta(F) < \gamma(J₀) \)) is expressed by means of an infixed notation): (4) \[ \text{EVEN} =_{df} \exists \text{intervals (closed)} F \text{ and } J₀ ; \text{X} \& (\delta(F) < \gamma(J₀)) \text{ ENONC J₀ EVENₚ} \]

4.2. In the framework of CL, the more elementary combinator - the components of ‘X’ - are successively applied, hence (5), deduced from (3) and (4): (5) \[ \text{PROC J₀} \text{ (SAY} \& (\text{ENONC J₀} (P₂ T² T¹)) (\delta(F) \gamma(J₀)) \text{ EGO} \]

The applicative expression (5) express that the predicative event ‘EVENₚ (P₂ T² T¹)’ is embedded into the speaking process realized on the interval ‘J₀’; it express also that this event is realized on the closed interval ‘F’ with implicit temporal constraints: the interval ‘F’; it is limited by the two open intervals ‘O₁’ and ‘O₂’ where the right boundary ‘δ(O₂)’ of ‘O₂’ is identified with the left boundary ‘δ(O₁)’ of ‘O₁’ (the beginning of the event) and the right boundary ‘δ(F)’ of ‘F’ is identified with the left boundary ‘δ(O₂)’ of ‘O₂’ (the end of the event); ‘O₁’ and ‘O₂’ are the zones of validation of the states denoting respectively that the event has not yet taken place and that the event has already taken place.

5. Representation of the verbal Meaning

We continue with the previous analysis going to a new step of the calculus. The meaning of the lexical predicate P₂ in (3) (we recall that [P₂ = take]) is represented with a help of a *semantic-cognitive scheme (SCS)*, that is an applicative combination of primitives defined inside a theory of cognitive and semantic representations (Descles 1990, 2004), where each primitive is anchored onto the field of human perception and action. We give here some primitives used in our semantic analysis:

‘ACCS’ is a binary relator for locating an entity relative to another (a locator), defining an accessibility domain of located entities from the locator\(^4\).

‘MOV’T is a binary operator expressing a spatio temporal movement of an entity from one location to another location.

‘FAIT’ is a binary operator expressing an effectuation of an action by an agent (or an instrument) of a movement or a change.

‘CONTR’ is the expression of a control by an agent on a movement (or a change).

‘TELEO’ is teleonomy binary operator between an agent and a wanted situation.

‘TRANS’ is an operator which is the result of a functional composition, by ‘B’, of the primitives ‘CONTR’ and

\(^4\) The notion of “possession” is not a primitive. Possession becomes a particular case of “accessibility” in which an entity is accessible from another entity (a locator) (Ivanova 2009).
‘FAIT’; it means that an agent controls and carries out an action5.

5.1. The corresponding SCS associated to the meaning of the lexical predicate take is expressed by the applicative expression (6):

\[
(6) \quad (\text{TELEO} (\text{SIT}_{0}^{12} \gamma [y, x])) x
\]

and the temporal constraints:

\[
\text{SIT}_{0}^{12} [y, x] = \text{STATE}_{0}^{12} (\text{NOT} (\text{ACCS} x y))
\]

\[
\text{SIT}_{0}^{21} [y, x] = \text{STATE}_{0}^{21} (\text{ACCS} x y)
\]

5.2. The lexical predicate ‘P2’ (take in our example) is the result of a synthetic integrative process by means of combinators successively introduced (by introduction rules) from (6) (a SCS). The exact form of this integrative process is a combinator ‘Y’ which combines the kinematical and dynamical primitives ‘TRANSF ’12’, ‘CHANGF ’12’, the primitives ‘\text{STAT}_{0} \theta \text{0 NOT’, ‘STAT}_{0} \omega \text{1’, ‘TELEO 0 STATE}_{0} \omega 1’ where :

\[
(7) \quad [\text{STAT}_{0} \omega 2 \text{ NOT} =_{\text{def}} \text{B STATE}_{0} \omega 2 \text{ NOT} ]
\]

\[
[\text{TELEO 0 STAT}_{0} \omega 1 =_{\text{def}} \text{B TELEO STATE}_{0} \omega 1]
\]

with the static primitive ‘ACCES’ and the arguments ‘y’ and ‘x’. After a λ-abstraction of places of arguments of the lexical predicate, we obtain the relation between the definiendum ‘P2’, and its definiens:

\[
(8) \quad [P2 =_{\text{def}} \lambda \gamma \lambda x.
\{ \quad (\text{Y} & \text{TRANS}_{F} ^{12} \text{CHANG}_{F} ^{12} \\
& (\text{STATE}_{0} \omega 2 \theta \text{ NOT}) (\text{STATE}_{0} \omega 1) \\
& (\text{TELEO} \theta \text{STATE}_{0} \omega 2) \text{ACCES}
\} y x
\}
\]

When ‘Y’ is applied to its different arguments, the “normal form” (6) is deduced, specifying the meaning of the predicate ‘P2’.

6. Inferential Reasoning

Let us introduce a rule describing a general property of events (Descles 2005, Ro 2008):

IF an event applied to any underlying predicative relation ‘A’, is realized on the closed interval ‘F’ which precedes the interval ‘J’ of speaking act

THEN:

(i) there are two open intervals ‘O1’ and ‘O2’ such as the event provides a transition between an initial state, realized on the interval ‘O1’ and a following state, realized on the interval ‘O2’, both states being adjacent to ‘F’, i.e. the two instants ‘γ(F1)’ and ‘δ(F1)’ are two continuous cuts (in the mathematical Dedekind’s sense) between the closed interval ‘F’ and two respective intervals ‘O1’ and ‘O2’;

(ii) the predicative relation ‘A’ is true at the final instant ‘δ(F1)’ of realization of the event;

(iii) the state following the transition is realized at each instant of the open interval ‘O2’ which is going on until the right boundary ‘δ(F1)’ of the unaccomplished speaking process.

The event ‘EVERN’(P2 ’F’T)’ is realized in the past, before the enunciative process. Thus, there is some closed interval ‘F’ where the transition implied by the event is realized between the two open intervals ‘O1’ and ‘O2’. From the above rule, we deduce, that a resultative state (expressed by the morphological form of “present perfect” in John has got the pen) of this event; the resultative state is realized at each instant of the intersection of the two intervals ‘O2’ and ‘J0’ where ‘δ(O2)’ = ‘δ(J0)’ = ‘T1’; it is deduced from the utterance John took the Mary’s pen denoting an occurrence of an event in the past.

6.1. After inserting the meaning of the predicate into the instanced scheme (5) (i.e. after the embedding of the meaning of the predicative relation into the speaking process), we get (9) from a unification [F12 := F] of ‘F12’ with ‘F’ (hence: [O12 := O1], [O21 := O2]); ‘F12’ is a closed interval relating to the meaning expressed by a SCS; ‘F’ is also a closed interval relating to the aspctual operator ‘EVERN’ of the predicative relation located inside the enunciative referential framework defined by the speaking act.

\[
(9) \quad [\text{EVERN}_{\gamma} (\& (\text{TRANS}_{F} ^{12} \text{CHANG}_{F} ^{12} \\
& (\text{STATE}_{0} \omega 1 (\text{F} [T^1, T^0])) (\text{STATE}_{0} \omega 2 (\text{F} [T^2, T^1])) T^{1}) \\
& (\text{TELEO} \theta (\text{STATE}_{0} \omega 2 (\text{F} [T^2, T^1))) T^{1}) \\
& \& (\& [O1 < F < O2])
\]

\& [\gamma(F) = \delta(O1) \& \delta(F) = \gamma(O2)]
\& [\delta(O2) = \delta(F0)]]
\]
6.2. The situation ‘SIT\(_{O2} [T2, T1]\)’ is realized on the interval ‘O\(^2\)’. Since the event ‘EVEN\(_E (P2, T2T1)\)’ is realized on ‘F’ it is inferred that this event has generated the resultative state

*John has got the pen*

whose meaning is represented by (10):

\[
\text{(10) } \text{PROC}\_\text{\(O\)} (\text{SAY} (& (\text{STATE}\_\text{\(O\)} \^2 (\text{ACCS} T\_1T\_2)) & [\text{\(\delta(O) = \delta(J^0)\)}]) ) \text{ EGO})
\]

which means literally that the enunciator ‘EGO’ is saying that the state of accessibility of ‘T\(_2\)’ (=: the pen) by ‘T\(_1\)’ (=John) is true during all the instants of the open interval ‘O\(^2\)’, the right boundary ‘\(\delta(O)\)’ being concomitant to that of the speech-act interval ‘\(\delta(J^0)\)’; it follows from it that the morphological form of “present perfect” encodes this meaning. By this formal calculus, we explain how *John has got the pen* is “naturally” deduced by any listener who understands *John took the Mary’s pen*.

7. Conclusion

This analysis shows that the CL with the topology (of intervals) is very useful to take into account: (i) representations of verbal meanings by SCS; (ii) aspectual values and temporal relations between topological intervals. It is possible to explain inferential relations between utterances inside the same formal applicative language (a metalanguage) used to give formal representations of utterances and to infer consequences from these formal representations. Such a calculus allows to show how it is possible, thanks to the combinators of CL, to develop progressively a ‘logic of natural language’ since the classical first-order logic is unable to give a deep analysis of aspectual representations and lexical meanings. The framework for our analysis is the Cognitive Applicative Grammar - CAG - (Desclés 1990, 2004) which provides an interplay between several cognitive representation levels and morphosyntactic configurations by means of both the CG techniques and composition of operators by combinators of the CL.

The formal analysis of aspects and SCC is general; it is applied to different languages: French, Bulgarian (Daynovska, 2008), Greek (Van den Haandel, 2008), Polish Gwiazdor, (2006), Russian (Ivanova 2009), Korean (Son, 2006). The implementation is in working in the functional programming languages HASKELL and CAML (Ro, 2008).

**Bibliographical References**


