The Fundamental Relations of Syntax and Conceptual Structure

(a working set of ideas)

Richard C. DeArmond

1 Introduction.

Our intention here is to create a set of terms for minimal syntax and their conceptual counterparts. We will start with a prime, which cannot be defined, and build up the definitions from these primes. The following box contains the corresponding units of conceptual structure and their correspondent units of syntactic structure.

Table 1:

Conceptual	Grammatical (Syntactic)
predicate	head (part of speech (N, V, A, P))
implicator	pointer
predicator	minimal head
(semantic) argument	(syntactic) argument
minimal eventuality	minimal phrase (unmodified), X1
complete eventuality	clause, CP (or NegP?)
(semantic) operator	(syntactic) operator
(semantic) modifier	(syntactic) modifier

2 First world

Let us imagine a world with one ‘thing’, It is immaterial what this ‘thing’ is; we could call it a predicate. However, there isn’t much we can say about it as it is an isolate predicate.

3 Second world.

Let us imagine a world with two ‘things’ or predicates. This is a slightly more interesting world. Two possible scenarios arise: either the two predicates exist independent of each, or there is some kind of relation between them. The first scenario is also rather uninteresting--what could be so interesting about two predicate that have no connection what so ever? That they share the same world is not very interesting since it is already known.

Let us look into the second scenario: there is a relation between the two predicates. Let us suppose that one of the predicates assumes the existence of the other. Now we have a much more interesting world. We will label one of the predicates A and the other B for the sake of identity:

(1)

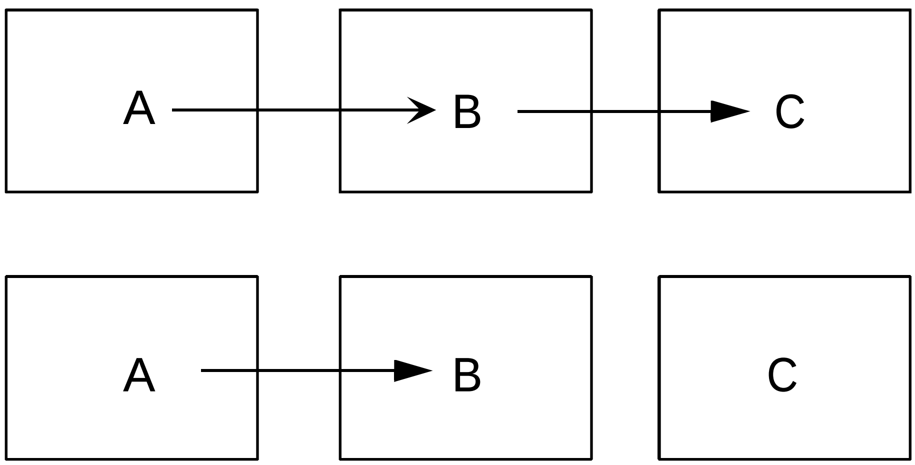
Note that the ‘thing’ imply was not given originally. Without this, we have two ‘thing’ scenario, which is uninteresting. We have to formally introduce into this world anew ‘thing’ which we will call an implicator. We could go on to say that B might also imply A--a distinct possibility. However, we are not sure that such can be the case. Let us impose a constraint on this world that two or more predicates cannot imply each other.

Now let us continue in this world noting the all predicates have the property of implication, but whether it is activated is a unique property of the predicate. It is obvious that this property is activated or has a positive value for A. What about B? B cannot imply A since A implies B. As long as these are the only two predicates in this world, the implicate value for B is minus since there is no other predicate.

4 Third World

In this world let us assume that there are three predicates. A with the same property as above, B and C. This world is a little more interesting that the Second World. It is now possible for B to has a positive or a negative value of implication. If Cis negative for implicative feature, then there are two possibilities:

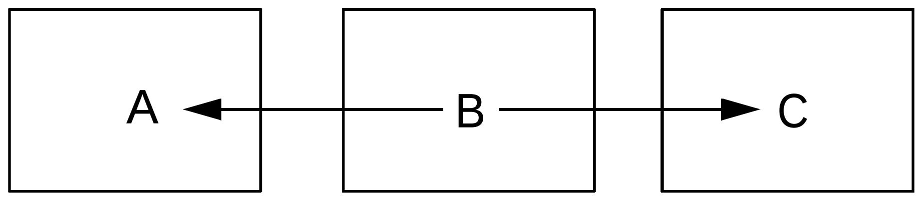
(2)



Actually the possibilities are numerous depending on the implicature feature of each predicate. We won't cover the remaining ones here. Note in (2) if both set of relations exist simultaneously, then the implicative feature either optional or marked "±". However, this may not be such a good idea because of future problems. "±" is not binary, and marking something as optional leads to problems semantic inference.

So far we have assumed that a predicate may or may not contain one implicative feature. Can it have it more? Yes, as long as we don't constrain this world. Suppose B has two such features, with restriction that it cannot implicate the same predicate twice:

(3)



Can A imply C or C, A? This depends on the constraints built into this world. We prefer to impose a constraint that two or more predicates cannot both imply the same predicate. This constrains the grammar, such that if B implies both A and C, then no other possibilities exist. Note that we have not mentioned order, which h will be another predicate--an operator. There is no ordering relationship in any of the above representations. The arrow points to the argument, and that is all.

Of course, the reader will have figured out by now that A and C in (3) are arguments of the head B, and that they are heads with no arguments, and that the arrows are implicators.

Of course there could exist other worlds with more and more predicates, and perhaps variations in the constraints. Natural languages contain a larger number of predicates with no fixed limit. And they are constrained. There are a set constraints that are universal in that they apply to all languages, and there others that are specific to one more languages, but not all of them.

The above discussion is about the fundamental basis of predicate logic. An indefinite number of systems are possible in that form. Natural language is one of them.

5 The Primes of natural Language

The term predicate is nearly impossible to define without containing circular definitions. The minimal idea is that a predicate is conceptually the heart of an eventuality--an event or a state. The best way to treat this problem is to consider a predicate a prime form--a form which cannot be defined in terms of smaller units. However, a predicate consists of a bunch of semantic-conceptual features. Thus, a predicate is not a prime. At this time no one knows what the set of conceptual primary features are.

We can think of a predicate as a bundle of features, but then this term needs to be defined. A appropriate bundle of features determines the meaning of a form. This is hard to illustrate, as the features are less than clear and easy to access. We can give a partial example. Consider the following:

(4) a. The ice melted.
    b. The sun melted the ice.

Melt in (1b) has a causative meaning associated with melt in (4a) (in most meanings of the verb). Suppose we extract the feature CAUSE from melt in (4b). Now we have a feature plus melt, which contains a bundle of features. CAUSE is probably a prime, although there is no direct evidence to support this hypothesis. We will tentatively assume that CAUSE is a prime. If evidence shows up that suggests that CAUSE can be broken into smaller plausible features, then we define CAUSE in those terms.

The core idea in melt is change; i.e. a substance undergoes a change from a solid state to a liquid state. Although there is more in the meaning of change, we can extract a feature CHANGE, which also a probably a prime. Of course, the initial state and the resultant are also implied in MELT and similar verb stems. Another prime would be STATE. CHANGE here implies an object going from one STATE to another. The object is an argument of both features--this is discussed immediately below.

The next term is argument. This term is nearly as difficult to define. This term has often been equated with participant (of an event); other terms are syntactic (theta role, thematic role). Let us tentatively suppose that argument is a prime, and let us assume that there is some kind of relationship between the two.

6 Predicates and Arguments

Predicates and arguments are usually easy to illustrate. Consider the following sentences:

(5) a. The wall is blue.
b. The ice melted.

Conceptually, blue is considered to be a predicate. The sentence is based on the predicate blue. Something is blue; in this case is the wall. The wall is an argument of the predicate; it is the only argument. BLUE contains the feature STATE, which could well be a prime. We won't attempt to cover the features that make up the colour BLUE. Similarly, ICE is an argument of MELT.

We can represent these relations in the following way:

(6) a. BLUE <WALL>
b. MELT <ICE>

Predicates are written in CAPS, and arguments are written in CAPS enclosed in angled brackets. The modifiers of the predicates and operators (tense and definiteness will be discussed below).Note that MELT implies a change of state--from a solid state to a liquid state. MELT is semantically restricted to certain solid objects such as ice and ice-cream, but not to other solids such as steel. which at high temperatures can be a liquid.

7 Arguments

Suppose we assume that there is a relation between BLUE and WALL. How do we know which is which? The answer to this is not easy. Let us first assume that all predicates assign at least one argument (perhaps they assign only one argument, but this is topic for more advanced discussion). In our discussion of BLUE and WALL, if one is a predicate and one is an argument, then only one of these terms assigns an argument. Which one? We may say that the predicate implies an argument. That is in the above case, BLUE implies that something must be blue--the wall in this case, but the term WALL does not imply its colour. We can talk about a wall without ever discussing the colour assuming that all walls have colour.

A wall is not defined as something that has a colour--it may be true, but not by definition. Similarly MELT implies that something undergoes the process of melting; in this case it is the ice (ICE).

In essence we are talking in circles--now we have the notion of imply to define. It appears that we need at least one more prime. Suppose we have the prime IMPLY. Although a prime, it is a feature, a prime that must be part of a predicate, it must be satisfied by pointing to an argument. Let us write features with a single initial CAP enclosed in square brackets, and X, which denotes the argument implied: [Imply X]:

(7) BLUE MELT

[Imply X] [Imply X]

We may write (7) as:

(8) [BLUE, [Imply X]] and [MELT, [Imply X]].

The outer square brackets indicates that the feature is part of the predicate. In binary terms, we could mark the feature with '+': [+Imply X], and the lack of the feature as [-Imply]. We consider these to be notational variants; we will continue using [Imply], and where it is not used, it implies [-Imply X]. (I prefer using binary features when writing a rule.) When X is WALL or ICE, we can rewrite (4) as:

(9) [BLUE, [Imply WALL]] and [MELT, [Imply ICE]].

Given the prime feature [Imply X], we can eliminate the argument as a prime. 'X' is a predicate.

(10) Given the predicator and its implicator [Y, [Imply X]]

a. Y is a predicator,

b. Imply X is an implicator,

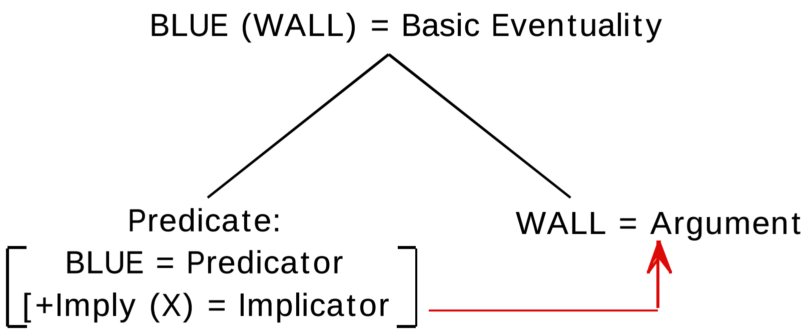
c. [Y, [Imply X]] is a predicate,

d. X is a predicate, and

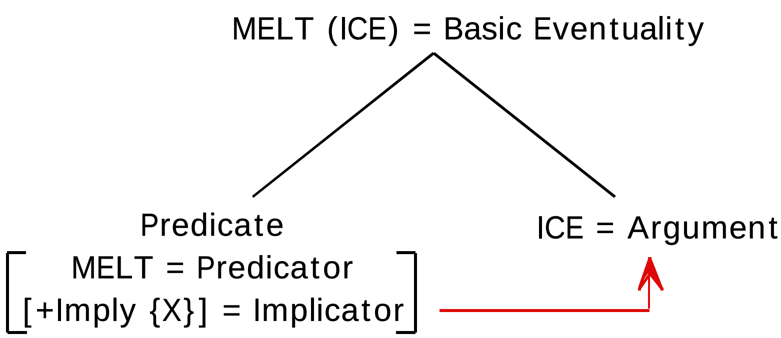
e. X is an argument of Y.

If a predicate is implied in the relation Imply-X, then we can define that predicate as an argument. A predicate such as BLUE is a prime. The set 'BLUE, [Imply X],' we will call a predicate. We propose the syntactic term "minimal head" to correspond with predicator. Let us call the feature [Imply X] an implicator. In the syntax we will call the corresponding form a "pointer" in order to keep these terms distinct. A predicator plus an implicator is a predicate (semantically), a minimal head plus a pointer is a head (syntactically).

(11)

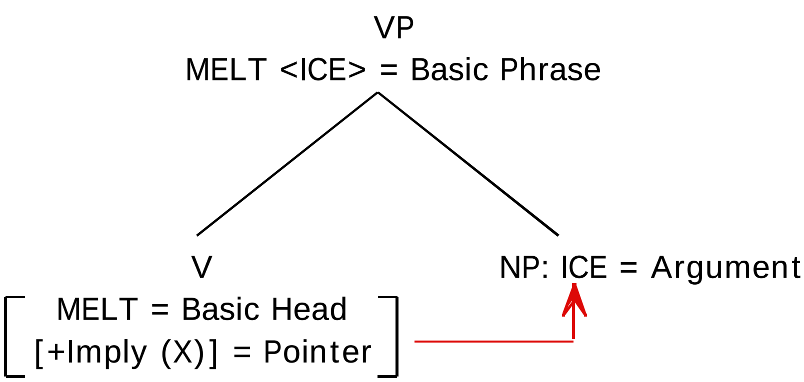


(12)

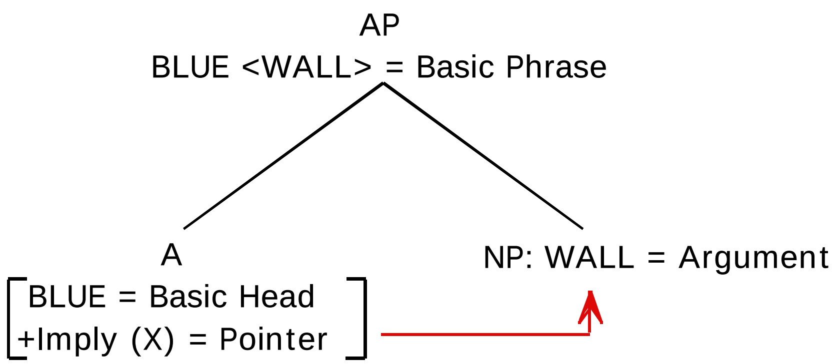


In the syntax the forms in (6) have the following configuration:

(13)



(14)



Since a minimal eventuality is defined here as a predicator and its implicator plus the argument of the implicator, it is logically true that a minimal eventuality must c-command its argument in the syntax.

In the syntax we will call an implicator a “pointer”, to maintain the difference in terminology between semantics and syntax. The lexical items BLUE and MELT each contain a pointer which points (establishing a link) to its argument. In the above case, WALL and ICE, respectively, are each the argument their respective head.

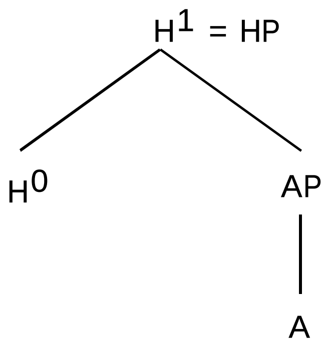
In the syntax, the term argument is used in the same way as in conceptual structure. Hence, we may say that a conceptual argument maps directly to a syntactic argument. However, there is no term in syntax that directly corresponds to predicate in conceptual structure. The parts of speech, verb, adjective, and preposition correspond to predicator, as the term verb does not imply no internal arguments. The predicator maps directly to a part of speech. The feature [+Imply Pred] has not been used in syntax. But we may introduce the term pointer defined here as a syntactic implicator.

The minimal eventuality is an unmodified or minimal phrase in the syntax. We will use the term phrase as is done now as a minimal phrase plus one or more modifiers.

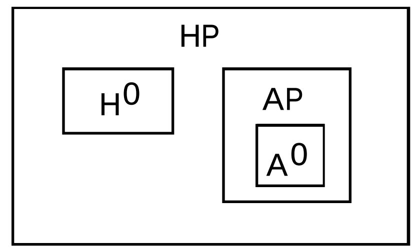
8 Government

Given the phrase HP, HP contains H if H is a head, and A if it is its an argument of H. A minimal or level-1 phrase is one where there is no phrase X such that HP contains X, and X contains H. We can represent a minimal phrase in a tree-structure format, in a bracketed format, or in a container format:

(15)



$[_{HP}\ [_{H^0}\ ]\ [_{AP}\ A^0]]$



The three structures are notational variants of one another and they have absolutely no theoretical value. Other notational variants are possible; perhaps there are an infi9nite number of possible notational variants. We elect to use the conventional tree diagrams for most represenetation. We can now define head-government:

(16) 10. Head-Government

X governs YP and Y iff:

a. if XP is a phrase such that it minimally contains X, the head of XP, and YP, and Y is the head of YP,

b. There is no intervening governor such that XP governs WP and WP governs YP,

c. X is a head-governor.

Be definition, only heads can be head-governors.

9 Governor

No constituent can be an h-governor except a head, which is by default an h-governor. Feature government is similar to head government:

(17) f-government:

A feature X^1 governs another feature X^2 iff:

a. X is the same feature where the value of X is irrelevant,

b. The constituent containing X^1 governs $X^{2.}$

Antecedent government is similar to head government:

(18) Antecedent Government:

A phrase XP governs YP iff:

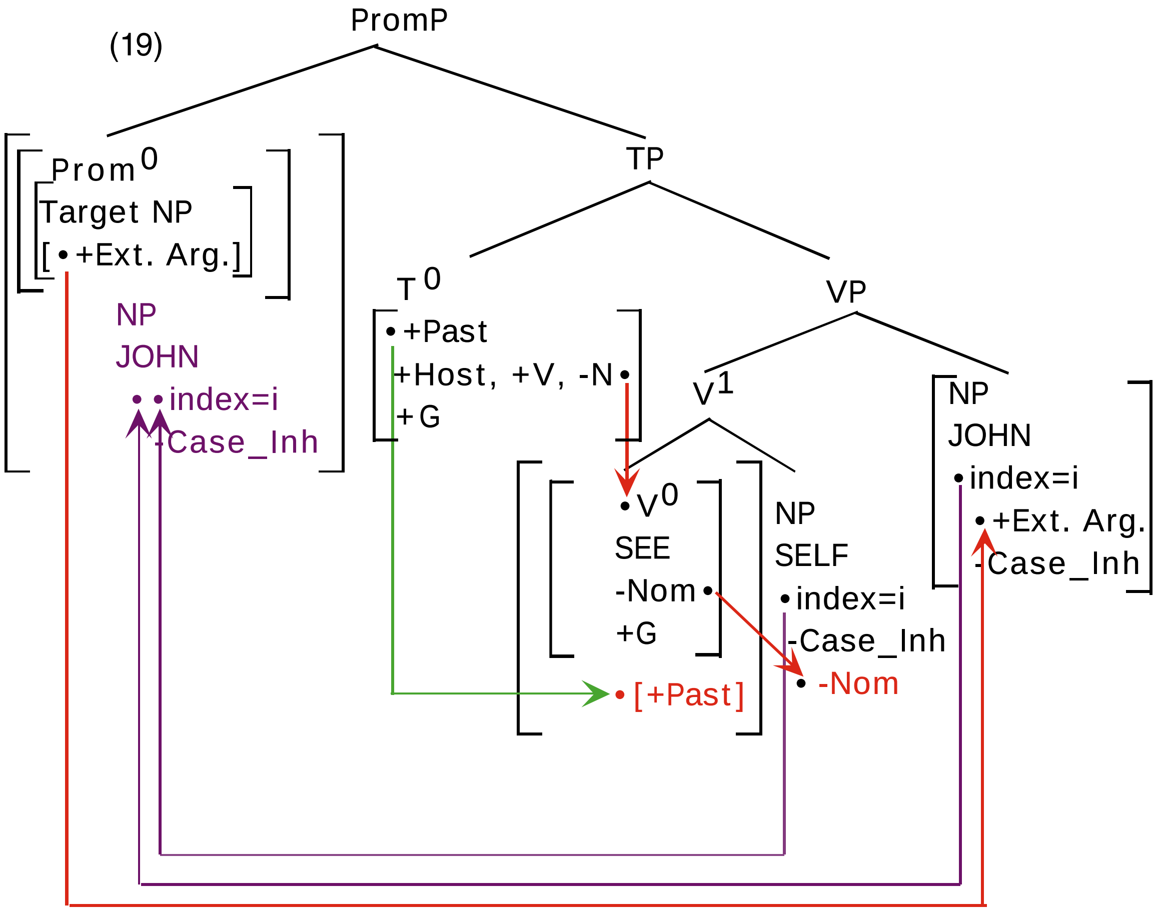
a. XP contains YP,

b. There is no phrase WP such that XP A-governs WP an d WP A-governs YP

c. XP and YP are coindexed.

A-government accounts for such conventional syntactic derivations as Binding and Raising. 12c) implies in some sense that all features of XP are linked to the exact same features in YP. The only features that cannot be linked are the Case features since these features a property of the position rather than the referent:

(19)



In the above diagram not all brackets are included, and the null verbal operators are omitted. Note the [+Ext. Arg] in Prom c-commands and hence f-governs the same in the external argument of the verb. This sets up the derivation in which the features of NP are copied back and adjoined to Prom. We propose the feature "index=x" to indicate the referential properties of the phrase of which it is a part. When this feature is linked to the same in another phrase, all the features of tail node are linked to the head node. The only feature not copied back is the structural feature of the position. That is, Prom cannot contain the feature [+Ext. Arg.], but it can be linked to it. The head of the arrow in a=linking points to the antecedent. There are two examples of A-government. The first is that Prom governs the external argument, and the other is that Prom governs the internal argument which has the feature index=i. This government cannot be established until the features of the external argument are copied to Prom. Note that given the currently theory of government, there would be no way to coindex the internal and external argument if one of them were not copied to the prom position.

In all of these operations the principle of Agreement hold; Agreement is bound by Government. There are three strategies:

a. Copy the value of a feature to a governed feature whose value has not been determined.

b. Target a host.

c. Insert a dummy.

We will not cover the last strategy here. In the first case, the value of the feature is copied to the governee feature. If a wrong value is copied, the result will be mismatched features. Agreement means that the value of both the governor and the governee are the same (plus, plus; or, minus, minus).

The next relationship that we find in conceptual structure is the modifier: modifiers

This page last updated 17 MR 2002