Unbounded Dependency Constructions (UDCs) in HPSG

Introduction to HPSG
26. Mai 2009
Kordula De Kuthy

From local to non-local dependencies

- A head generally realizes its arguments locally within its head domain.
- Certain kind of constructions resist this generalization, such as, for example, the wh-questions discussed below.
- How can the non-local relation between a head and such arguments be licensed? How can the properties be captured?

A first example: Wh-elements

Wh-elements can have different functions:

(1) a. Who did Hobbs see _? Object of verb
b. Who do you think _ saw the man? Subject of verb
c. Who did Hobbs give the book to _? Object of prep
d. Who did Hobbs consider _ to be a fool? Object of obj-control verb

Wh-elements can also occur in subordinate clauses:

(2) a. I asked who the man saw _.
b. I asked who the man considered _ to be a fool .
c. I asked who Hobbs gave the book to _.
d. I asked who you thought _ saw Hobbs.

Different categories can be extracted:

(3) a. Which man did you talk to _? NP
b. [To [which man]] did you talk _? PP
c. [How ill] has the man been _? AdjP
d. [How frequently] did you see the man _? AdvP

This sometimes provides multiple options for a constituent:

(4) a. Who does he rely [on _]? NP
b. [On whom] does he rely _? PP

Unboundedness:

(5) a. Who do you think Hobbs saw _?
b. Who do you think Hobbs said he saw _?
c. Who do you think Hobbs said he imagined that he saw _?
Unbounded dependency constructions

An unbounded dependency construction
- involves constituents with different functions
- involves constituents of different categories
- is in principle unbounded

Two kind of unbounded dependency constructions (UDCs)
- Strong UDCs
- Weak UDCs

Weak UDCs

No overt constituent in a non-argument position:

Purpose infinitive (for-to clauses):
(11) I bought it, for Sandy to eat .

Tough movement:
(12) Sandy is hard to love .

Relative clause without overt relative pronoun:
(13) This is [the politician], [Sandy loves ].

It-clefts without overt relative pronoun:
(14) It is Kim, [Sandy loves ].

Strong UDCs

An overt constituent occurs in a non-argument position:

Topicalization:
(6) Kim, Sandy loves .

Wh-questions:
(7) I wonder [who, Sandy loves ].

Wh-relative clauses:
(8) This is the politician [who, Sandy loves ].

It-clefts:
(9) It is Kim [who, Sandy loves ].

Pseudoclefts:
(10) [What, Sandy loves ] is Kim.

Some properties of UDC constructions

Link between filler and gap with category information needed:

(15) a. Kim, Sandy trusts .
b. [On Kim], Sandy depends .
(16) a. * [On Kim], Sandy trusts .
b. * Kim, Sandy depends .
And this link has to be established for an unbounded length:

(17) a. Kim, Chris knows Sandy trusts 
   b. [On Kim], Chris knows Sandy depends 

(18) a. *[On Kim], Chris knows Sandy trusts 
   b. * Kim, Chris knows Sandy depends 

(19) a. Kim, Dana believes Chris knows Sandy trusts 
   b. *[On Kim], Dana believes Chris knows Sandy depends 

(20) a. *[On Kim], Dana believes Chris knows Sandy trusts 
   b. * Kim, Dana believes Chris knows Sandy depends 

Using the feature slash

To account for UDCs, we will use the feature slash (so-named because it comes from notation like S/NP to mean an S missing an NP)

- This is a non-local feature which originates with a trace,
- gets passed up the tree,
- and is finally bound by a filler

An example for a strong UDC

The bottom of a UDC: Traces

- phonologically null, but structure-shares local and slash values
- we’ll talk about to-bind later
Traces

Because the local value of a trace is structure-shared with the slash value, constraints on the trace will be constraints on the filler.

- For example, *hates* specifies that its object be accusative, and this case information is local
- So, the trace has [SYNSEM|LOCAL|CAT|HEAD|CASE acc] as part of its entry, and thus the filler will also have to be accusative

(21) *He/Him, John likes _

The middle of a UDC: The Nonlocal Feature Principle (NFP)

For each nonlocal feature, the inherited value on the mother is the union of the inherited values on the daughter minus the to-bind value on the head daughter.

- In other words, the slash information (which is part of inherited) percolates “up” the tree
- This allows the all the local information of a trace to “move up” to the filler

The top of a UDC: Filler-head structures

Filler-head schema

\[
\begin{align*}
\text{phrase} & \quad \text{DTRS} \quad \text{head-filler-struc} \\
\text{HEAD-DTR|SYNSEM} & \quad \text{LOC|CAT} \\
\text{NONLOC} & \quad \text{HEAD-DTR|SYNSEM|LOCAL} \\
\text{FILLER-DTR|SYNSEM} & \quad \text{VFORM fin} \\
\text{VERB} & \quad \text{FIN} \\
\text{SUBCAT} & \quad \text{element} \\
\text{INHERITED|SLASH} & \quad \text{to-bind|slash} \\
\end{align*}
\]

- Filler and trace are identified as the exact same thing (as far as their local structure is concerned)
- The trace is “bound” by the to-bind feature; this prevents the slash value from going any higher in the tree
- Only saturated finite verbs (i.e., sentences) license such structures
The analysis of weak UDCs

(22) a. Kim is easy (for John) to please.
   b. Kim is easy to prove that Mary asked Paul to bribe.

(23) a. It is easy to please him. * he
   b. I am easy to please.

⇒ No true (non-argument) filler, only coindexed items serving as arguments

Subject is role assigned:

(24) a. I believe there to be a unicorn in the garden.
   b. * There is easy to believe a unicorn in the garden.

(25) a. [This sonata] is easy to play on that violin.
   b. [This violin] is easy to play this sonata [on].

⇒ Lexical entry selects for infinitive complement missing an NP, which is coindexed with the subject
A weak UDC analysis

Multiple traces with strong and weak UDCs

(26) This is a problem which\textsubscript{1} John\textsubscript{2} is easy to talk to \textsubscript{2} about \textsubscript{1}.

This is analyzed by allowing SLASH to be a set value and binding each trace off in turn

• easy binds trace\textsubscript{2}
• The head-filler construction at the top of the tree binds trace\textsubscript{1}

Multiple traces example

Limiting the occurrence of traces

The that-trace effect, one of the island effects:

(27) Who\textsubscript{i} did he claim that she kissed \textsubscript{2}.

(28) * Who\textsubscript{i} did he claim that \textsubscript{2} kissed her.

The trace principle (for English)

Every trace must be strictly subcategorized by a substantive head, i.e., its SYNSEM value must be a non-initial member of a substantive head’s SUBCAT list.

→ That is, traces should be arguments (subcategorized) and they should not be subjects (strictly) ... Other languages may just require subcategorization
Subject extraction

But subjects can be extracted out of embedded clauses

(29) * Who, did he claim that \hspace{1em} kissed her.
(30) Who, did he claim \hspace{1em} kissed her.

⇒ So, we treat subject extraction in a special, traceless manner, allowing it to happen only in embedded clauses

Subject extraction lexical rule (SELR):

\[
\begin{align*}
\text{word} & \quad \text{synsem} | \text{local} | \text{cat} | \text{subcat} | \text{rest} \\
\text{subj} & \Rightarrow \text{synsem} \left( \text{local} | \text{cat} | \text{subcat} | \text{rest} \right) \\
\end{align*}
\]

With the modifications in chapter 9, this will of course be reformulated as a rule applying to SUBJ, not SUBCAT

A subject extraction analysis

Island constraints

With the definition of strict subcategorization in the Trace Principle, certain island constraints are correctly predicted.

(31) a. How tall do you think the building is _?
   b. *How do you think the building is _ tall? (modifier)

(32) a. That is the building [whose design our architects rejected _].
   b. *That is the building [whose our architects rejected _ design]. (not strict)

(33) a. The books that I like, Leslie donated _ to the library.
   b. *The books, Leslie donated _ that I like to the library. (head)
Multiple unbounded dependencies

(34) a. It will be easy to play even the most difficult sonata on a violin this well crafted.
   b. [A violin this well crafted], even [the most difficult sonata] will be easy to play on.

(35) a. It is easy to talk to John about this topic.
   b. This is the topic which John is easy to talk to about.

- Traces are bound at different points in the tree (as we saw earlier)
- There are also cases, on the other hand, where traces are bound in unison

Multiple unbounded dependencies (cont.)

(36) That was the rebel leader who rivals of shot.

An example like this is handled rather smoothly because slash is a set-valued feature, and so the two traces can be identified as the same.

What distinguishes the following, however?

(37) a. *That was the rebel leader who rivals of shot the British consul.
   b. That was the rebel leader who agents of foreign powers shot.

Parasitic gaps

Extraction out of objects is possible in English:

(38) Who did John assassinate rivals of?

Extraction out of subjects, however, is only possible in the presence of a second gap:

(39) Who did rivals of assassinate?
(40) a. *Who did rivals of assassinate the President?
   b. Who did rivals of the president assassinate?

The first trace is thus parasitic on the second one

Capturing parasitic gaps

The subject condition

The initial element of a lexical head’s subcat list may be slashed only if that list contains another slashed element.

- All parasitic gaps are contained within subjects.
- Any subject gap must be licensed by a gap from the same subcat list.
Summary of HPSG trace theory

1. Nonlocal Feature Principle: For each nonlocal feature, the INHERITED value on the mother is the union of the INHERITED values on the daughter minus the TO-BIND value on the head daughter.

2. Trace Principle (English): Every trace must be strictly subcategorized by a substantive head, i.e., its SYNSEM value must be a non-initial member of a substantive head's SUBCAT list.

3. Subject Condition (English): The initial element of a lexical head's SUBCAT list may be slashed only if that list contains another slashed element.