Homework 5

• Other kinds of recursion, e.g.
  – Relative clauses
    • the cheese that the rat ate
    • the cheese that the rat that the cat saw ate
    • the cheese that the rat that the cat that the dog chased saw ate

these are not complete sentences, but just noun phrases (NPs)
Homework 5

• Example:
  – the cheese that the rat ate

?- np(Parse, [the, cheese, that, the, rat, ate], []). 
Parse = np(np(d(the), nn(cheese)), sbar(c(that), s(np(d(the), nn(rat)), vp(v(ate), np(0))))));
false.

?-

1. np(np(np(D,N),SBAR)) --> det(D), nn(N), sbar(SBAR).
2. nn(nn(rat)) --> [rat].
3. nn(nn(cheese)) --> [cheese].
4. v(v(ate)) --> [ate].
5. np(np(0)) --> [].

Alternatively:
5. vp(vp(V)) --> v(V).
Homework 5

• Example:
  – the cheese that the rat that the cat saw ate

?- np(Parse,[the,cheese,that,that,cat,saw,ate],[[]]).
Parse = np(np(d(the), nn(cheese)), sbar(c(that), s(np(d(the), nn(rat)), sbar(c(that), s(np(d(the), n.n(cat)), vp(v(saw), np(0)))), vp(v(ate), np(0))));
false.

?- 

1. nn(nn(cat)) --> [cat].
2. v(v(saw)) --> [saw].
Homework 5

• Example:
  – the cheese that the rat that the cat saw ate

?- np(Parse, [the, cheese, that, the, rat, that, the, cat, saw, ate], []).
Parse = np(np(d(the), nn(cheese)), sbar(c(that), s(np(np(d(the), nn(rat)), sbar(c(that), s(np(d(the), n_n(cat)), vp(v(saw), np(0)))))), vp(v(ate), np(0))))
false.

?-

\[
\text{vp(vp(V))} \rightarrow v(V).
\]

version
Homework 5 review

• Example:
  – the cheese that the rat that the cat that the dog chased saw ate

?- np(Parse, [the, cheese, that, the, rat, that, the, cat, that, np(d(the), nn(cheese)), sbar(c(that), s(np(n, nn(cat))), sbar(c(that), s(np(d(the), nn(dog))), vp(v(c e), np(0))))));
false.

?- |

1. nn(nn(dog)) --> [dog].
2. v(v(chased)) --> [chased].
Subject Relative Clauses

- **Subject relative clauses (not center-embedded)**
  - the cat that saw the rat that saw the cheese that ...
  - \[NP \text{the cat} [SBAR \text{that} [S \text{saw} [NP \text{the rat} [SBAR \text{that} [S \text{saw} [NP \text{the cheese that} ... ]]]]]]]
  - the rat that Ø saw the cheese
  - the cat that Ø saw the rat that Ø saw the cheese
  - the dog that Ø saw the cat that Ø saw the rat that Ø saw the cheese
Subject Relative Clauses

- Subject relative clauses (not center-embedded)
  
  – the rat that Ø saw the cheese

Advantage of the empty category rule
np(np(0)) → [].
over the
vp(vp(V)) → v(V).
version
(However, there are disadvantages too...)
Subject Relative Clauses

- Subject relative clauses (not center-embedded)
  - the cat that Ø saw the rat that Ø saw the cheese

?- np(Parse,[the,cat,that,saw,the,rat,that,saw,cheese],[]).
Parse = np(np(d(the), np(cat)), sbar(c(that), s(np(Ø), vp(v(saw), np(np(d(the), nn(rat)), sbar(c(that), s(np(Ø), vp(v(saw), np(d(the), nn(cheese)))))))))).

false.

?-
Subject Relative Clauses

- **Subject relative clauses (not center-embedded)**
  - the dog that Ø saw the cat that Ø saw the rat that Ø saw the cheese

```prolog
?- np(Parse, [the, dog, that, saw, the, cat, that, saw, the, rat, that, saw, the, cheese], []).
Parse = np(np(d(the), nn(dog)), sbar(c(that), s(np(Ø), vp(v(saw)), np(np(d(the), nn(cat)), sbar(c(that), s(np(Ø), vp(v(saw)), np(np(d(the), nn(rat)), sbar(c(that), s(np(Ø), vp(v(saw)), np(np(d(the), nn(cheese)))))_))))))));
false.
?- 
```
Subject Relative Clauses

- Subject relative clauses (not center-embedded)
  - the dog that Ø saw the cat that Ø saw the rat that Ø saw the cheese
The empty category rule

A disadvantage of the grammar rule \(np(np(\emptyset)) \rightarrow \emptyset\) :

- Example:
  - *the rat that saw

\[
?- np(Parse, [the, rat, that, saw], []). \\
Parse = np(np(d(the), nn(rat)), sbar(c(that), s(np(\emptyset), vp(v(saw), np(\emptyset)))))); \\
false.
\]

freely permits the simultaneous omission of the subject and the object

*the rat that \(\emptyset\) saw \(\emptyset\)
The empty category rule

• Rule is too permissive, let’s verify that it permits ungrammatical sentences such as:
  – John saw Mary
  – Ø saw Mary
  – John saw Ø
  – Ø saw Ø
Complexity and Comprehension

Subject and Object Relative Clauses
• From easy to hard to comprehend for humans?
  1. Subject relative + subject relative
  2. Subject relative + object relative
  3. Object relative + subject relative
  4. Object relative + object relative

For a computer, no difference...
Subject and Object Relative Clauses

- **Subject relative + subject relative**
  - the cat that Ø chased the dog that Ø ate the cheese saw the rat
Subject and Object Relative Clauses

- Subject relative + object relative
  - the cat that Ø saw the rat that the dog chased Ø ate the cheese
Subject and Object Relative Clauses

- **Object relative + subject relative**
  - the cat that the dog that Ø saw the rat chased Ø ate the cheese
Subject and Object Relative Clauses

- **Object relative + object relative**
  - the cat that the dog that the rat saw Ø chased Ø ate the cheese
Class Exercise 1

Grammar rule:

np(np(Ø)) --> []

permits subjects and objects to be freely omitted

• Subject relative clauses
  – the rat that Ø saw the cheese
  – the cat that Ø saw the rat that Ø saw the cheese
  – the dog that Ø saw the cat that Ø saw the rat that Ø saw the cheese

• Object relative clauses
  – the cheese that the rat ate Ø
  – the cheese that the rat that the cat saw Ø ate Ø
  – the cheese that the rat that the cat that the dog chased Ø saw Ø ate Ø
Class Exercise 1

• Rule is too permissive, verify that it permits ungrammatical sentences:
  – John saw Mary
  – Ø saw Mary
  – John saw Ø
  – Ø saw Ø

• Thus it also permits:
  • the cheese that the rat ate Ø
  • the cheese that Ø ate Ø
  • the rat that Ø saw the cheese
  • the rat that Ø saw Ø
  • the cheese that the rat ate the cheese
  • the rat that the rat saw the cheese
Class Exercise 2

- Let’s modify (or restrict) the grammar so that subject relative clauses force an empty subject only
  - the rat that Ø saw the cheese
  - the rat that Ø saw Ø
  - the rat that the rat saw the cheese

\[ np(np(np(D,N), SBAR)) \rightarrow \text{det}(D), \text{n}(N), \text{sbar}(SBAR). \]

**idea:** make this sbar special to subject relative clauses, i.e. force subject NP to be empty and object NP to be overt
Class Exercise 2

• First, delete the overgenerating rule... i.e. delete the empty category rule from the grammar.

\[ \times^{10} \text{np(np(0))} \rightarrow \square. \]

\[ \text{np(np(D,N))} \rightarrow \text{det(D)}, \text{n(N)}. \]
\[ \text{np(np(N))} \rightarrow \text{proper\_noun(N)} \; ; \; \text{pronoun(N)}. \]

We are left with only rules for generating overt NPs
Class Exercise 2

- Grammar rules involved:

1. `s(s(NP,VP)) --> empty_np(NP), vp(VP).`
2. `np(np(∅)) --> [].
3. `empty_np(np(∅)) --> []`
4. `sbar(sbar(C,S)) --> c(C), s(S).`
5. `np(np(D,N),SBAR)) --> det(D), n(N), sbar(SBAR).`
6. `np(np(D,N)) --> det(D), n(N).`
Class Exercise 2

- Do copy and rename:

\[
\begin{align*}
\text{np(np(np(D,N),S,BAR)))} & \rightarrow \text{det(D), n(N), sbar(S,BAR).} \\
\text{sbar(sbar(C,S))} & \rightarrow \text{c(C), subjrel_s(S).} \\
\text{subjrel_s(s(NP,VP))} & \rightarrow \text{empty_np(NP), vp(VP).}
\end{align*}
\]

- Rename nonterminal (S)

\[
\begin{align*}
\text{empty_np(np(∅))} & \rightarrow \square.
\end{align*}
\]

\[
\begin{align*}
\text{np(np(D,N))} & \rightarrow \text{det(D), n(N).} \\
\text{np(np(N))} & \rightarrow \text{proper_noun(N); pronoun(N).}
\end{align*}
\]
Class Exercise 2

• Do copy and rename:

  Rename nonterminal (sbar)

```plaintext
np(np(np(D,N),SBAR)) --> det(D), n(N), subjrel_sbar(SBAR).
subjrel_sbar(sbar(C,S)) --> c(C), subjrel_s(s(S).
subjrel_s(s(NP,VP)) --> empty_np(NP), vp(VP).
empty_np(np(∅)) --> □.
np(np(D,N)) --> det(D), n(N).
np(np(N)) --> proper_noun(N); pronoun(N).
```
Class Exercise 2

• Verify subject relative clauses force an empty subject only
  • the rat that $\emptyset$ saw the cheese
  • the rat that $\emptyset$ saw $\emptyset$
  • the rat that the rat saw the cheese