

CS 603: Programming Languages

Lecture 29

Spring 2004

Department of Computer Science

University of Alabama

Joel Jones

Overview

- More about Definite Clause Grammars
 - Extra Goals
 - Building parse trees
- Using Horn Clauses as a denotational semantics

Even More on Definite Clause Grammars

- Taken from:
 - <http://www.coli.uni-sb.de/~kris/prolog-course/html/node69.html>
- Using Peano numbers seems a little extreme, can't we use numbers?
- Yes, remember $N1$ is $N + 1$?
- Since DCGs are really syntactic sugar for ordinary Prolog rules, why not have syntactic sugar in DCGs for ordinary rules?

Sugar for DCGs

```
s(Count) --> [ablock(Count),  
              bblock(Count),  
              cblock(Count)].
```

```
ablock(0) --> [ ].
```

```
ablock(NewCount) --> [a], ablock(Count),  
                      {NewCount is Count + 1}.
```

```
bblock(0) --> [ ].
```

```
bblock(NewCount) --> [b], bblock(Count),  
                      {NewCount is Count + 1}.
```

```
cblock(0) --> [ ].
```

```
cblock(NewCount) --> [c], cblock(Count),  
                      {NewCount is Count + 1}.
```

Translation:

```
ablock(NewCount, A, B) :-  
    'C'(A, a, C),  
    ablock(Count, C, B),  
    NewCount is Count + 1.
```

Using Ordinary Rules in DCGs for Attributes

- Last time, we used arguments to carry information about the role the pronoun was playing in a particular rule:
 - `s --> np(subject), vp.`
`np(_) --> det, n.`
`np(X) --> pro(X).`
`vp --> v, np(object).`
`pro(subject) --> [he].`
- But what if we need pronouns to have more than one attribute?

Separate Rules and Lexicons

- What? Eliminate all mention of individual words in DCGs and record all information about words separately

Original:

```
np --> [ det , n .  
vp --> [ v , np .  
vp --> [ v .  
det --> [ the ] .  
det --> [ a ] .  
n --> [ woman ] .  
n --> [ man ] .  
v --> [ shoots ] .
```

Becomes:

```
det --> [ Word ] , { lex ( Word , det ) } .  
n --> [ Word ] , { lex ( Word , n ) } .  
v --> [ Word ] , { lex ( Word , v ) } .
```

Lexicon:

```
lex ( the , det ) .  
lex ( a , det ) .  
lex ( woman , n ) .  
lex ( man , n ) .  
lex ( shoots , v ) .
```

Changed rules:

Using Horn Logic as a Semantic System

- This discussion is taken from:
 - <http://www.cs.nmsu.edu/~gupta/logden/lnai.ps>
- Why another semantics specification technique?
 - More program-like, executable
 - Ease parallel translation

Logical Denotations

- Denotational Semantics consist of:
 - syntax: context free grammar
 - semantic algebra: basic domains along with operations over those domains
 - valuation functions: mappings from patterns of parse trees to values in the domains of the semantic algebra