9. More Grammars

\[ s(Z) :- \text{np}(X), \; \text{vp}(Y), \; \text{append}(X, Y, Z). \]
\[ \text{np}(Z) :- \text{det}(X), \; \text{n}(Y), \; \text{append}(X, Y, Z). \]
\[ \text{vp}(Z) :- \text{v}(X), \; \text{np}(Y), \; \text{append}(X, Y, Z). \]
\[ \text{vp}(Z) :- \text{v}(Z). \]
\[ \text{det}([\text{the}]). \]
\[ \text{det}([\text{a}]). \]
\[ \text{n}([\text{woman}]). \]
\[ \text{n}([\text{man}]). \]
\[ \text{v}([\text{shoots}]). \]

The type of grammar that we saw last week is fine for simple problems, i.e. simple and non-recursive phrases. Using `append` repeatedly on a large number of combinations is computationally expensive or might not even produce a solution in a *reasonable* amount of time. In addition, given a query, such as \( s([\text{a,man,shoots}]) \), the program doesn’t use the input sentence to guide the search, but instantiates individual parts first and then sees whether they yield the given sentence. So depending on the size of the knowledge base and the order of nouns, verbs etc. with respect to query, this could take longer!

This week, we’re going to see *difference* grammars based of difference lists encoding the same rules, but doing the search more efficiently and thus allowing for an easier use of recursive rules, such as prepositional phrases within noun phrases.

### Difference Lists

Difference lists in this context are used as follows: a sentence is now represented as the difference between two lists:

\[ [a, \text{woman}, \text{shoots}, a, \text{man}] [] . \]

The list on the left-hand side is the one to be consumed, leaving behind the empty list as output. If not everything on the left can be mapped to a rule or fact, it would not be a *correct sentence in our grammar*. The following is a grammar just like the one above, but using difference lists instead of appending:

\[ s(X, Z) :- \text{np}(X, Y), \; \text{vp}(Y, Z). \]
\[ \text{np}(X, Z) :- \text{det}(X, Y), \; \text{n}(Y, Z). \]
\[ \text{vp}(X, Z) :- \text{v}(X, Y), \; \text{np}(Y, Z). \]
\[ \text{vp}(X, Z) :- \text{v}(X, Z). \]
\[ \text{det}([\text{the}|W], W). \]
\[ \text{det}([\text{a}|W], W). \]
\[ \text{n}([\text{woman}|W], W). \]
\[ \text{n}([\text{man}|W], W). \]
\[ \text{v}([\text{shoots}|W], W). \]

### Exercise

As a first exercise, convert your grammar from last week into one using difference lists. Just as last week, it should be able to handle *simple past* and prepositional phrases. In addition, if you haven’t done so already, add a rule to deal

---

^1 This could be because we tried to query for a genuinely ungrammatical sentence, such as *‘The the man shoots the woman.’* OR because we are querying sentences that are not covered by our grammar, such as *‘I think the man shot the woman.’*: it’s important to differentiate between the two!
with both transitive and intransitive verbs (you might have to encode either type differently for cases where only one form is grammatical, compare: ‘She slept *the ball.’ vs. ‘The man shoot (the woman).’). Also add a rule to include sentences, such as ‘I think the man shot the woman.’ Again, try to avoid redundancy in your predicates whenever possible and try to make clever use of Prolog’s conjunctions to reduce your total number of predicates in the grammar. Also, for the recursive rules, remember to put the base cases first.

Submission

Submit your .pl file including explanations about the intuitions behind your predicates in comments by 4th of April, 2016: 6pm by email (klaussnc@tcd.ie).

Sources

- [www.scss.tcd.ie/~moreaue/prolog](http://www.scss.tcd.ie/~moreaue/prolog)