

Formal Languages and Compilers

More on Yacc

Lists in productions

- Sometime in our productions we find, or we would like to have, list of certain things...
- Actually, lists are a milestone for every program. Without them we could not specify even a program with a couple of line of instruction (not easily at least)
- How do obtain a list of arbitrary length using production?

Lists in productions

- Sometime in our productions we find, or we would like to have, list of certain things...
- We start by declaring the item we want to have..
- Item : body1
 | body2
 | body3
 |
 ;

■ Item_list: ?????
 what is the correct thing to write here?
 We want to obtain a list of item.. So...? Any guess?

Lists in productions

- Case 1)
Item_list:
 item
 | item_list item;
- OR
- Case 2)
Item_list:
 item
 | item item_list;
- Do we obtain a list of item? In both cases?
What will you buy?

Lists in productions – parsing the two cases

- Case 1) \rightarrow is a left recursion
- Case 2) \rightarrow is a right recursion
- Right recursion looks more interesting and readable to human
- What kind of action does the parser undertake up on reading an input sequence like the following

item1 item2 item1 item3 item4 item5 item1 item9

- Any difference now?

Right is wrong

- Right recursion is wrong..
- If we have a recursion on a big list and each items occupies a certain amount of memory we will end up filling the memory space before reducing any of the item to the list production..
- We have 2 simple examples in the directory 06..
- What if we place on the stack.. Something more wide... than a simple integer/pointer?

Error tracking in Yacc

- We want to have a user friendly front end.
- If we meet an error we want to be able to help the user in spotting it
- How can we proceed in this way?
- Collect some ideas and let's discuss about it

Error tracking in Yacc

- We want to have
- Line numbers
- Column number
- Expected correct syntax

Error tracking in Yacc

- Let's see another example
- We use **yylloc** and in the lexer upon reading stuff we increment values such as last column and last line...
- Bison is a bit far advanced than Yacc, and has a default action for location (addressed by @)
- As for semantic values, there is a default action for locations that is run each time a rule is matched. It sets the beginning of @\$ to the beginning of the first symbol, and the end of @\$ to the end of the last symbol.
- In this way we can use localized message for prompting warning and errors to the user.

Bibliography

- <http://www.tldp.org/HOWTO/Lex-YACC-HOWTO-6.html>
- Bison manual