

Syntactic Parsing: Assignment 1

Shifei Chen

The code took around 736 seconds to run on the lab computer and my final F1 score was 0.6.

Attempt to Speed Up

The original code took me over 4000 seconds (about 1 hour and 10 minutes) to finish, which is too slow to debug on the actual dev set. Then from last year student Johannes Gontrum's GitHub repo¹ I got some inspiration including multithreading, integer comparing and RHS intersection.

In my experiment setup, I parsed the dev set by using a CNF grammar from the dev set itself. The original non-optimized code finished the job in 40 seconds.

The multithreading part was the first I tried to speed things up as it is the easiest modification to my existing code. And surprisingly it was also very effective, shortened the running time to only 12 seconds by splitting the parsing task on 4 threads. I tried 8 threads as well but that didn't shorten the running time further, probably because both the lab computer and my own laptop only had 4 physical cores.

The other two approaches didn't work as expected. Integer comparison slightly prolonged the running time by 4 seconds due to the extra cost in backtracing, since we suffered from converting between the index integers and the actual terminal/non-terminals. RHS intersection needed 138 seconds to parse instead of 12 seconds. This suggests that for the grammar generated from the `dev.dat`, the time saved from looping through 1690 rules was not enough to pay the extra cost in set creation and intersection calculation. Johannes's code, on the other hand, succeeded because of his sublime data structure representation for the CKY algorithm.

After all, the running time of CKY algorithm depends on the length of the grammar, as we need to loop through all of the grammar rules everytime we move the working window (or the `start` and `end` cursor). The longer the grammar is, the longer CKY algorithm needs to parse.[1]

References

- [1] Aaron Dunlop and Brian Roark. *Optimizing CYK parsing on modern processors*.

¹<https://github.com/jgontrum/cky-parser-optimization>