

10-608: Assignment 1

Igor Labutov, Bishan Yang

January 2018

Semantic Parsing I

In class we have seen that arithmetic expressions can be ambiguous when expressed directly in natural language. For example:

- “*three times two plus four*”

can be interpreted as both $(\times 3 (+ 2 4))$ and $(+ (\times 3 2) 4)$. In speech we cannot use parantheses to disambiguate order of operations. But we as humans recognize the possibility of such ambiguities in speech, and often choose to phrase our utterances deliberately in a way that minimizes ambiguity. For example, the following two statements unambiguously parse to their logical forms:

- “*add two and four then multiply that by three*” $\rightarrow (\times 3 (+ 2 4))$
- “*multiply three and two then add four to that*” $\rightarrow (+ (\times 3 2) 4)$

In this assignment, your task is to implement a grammar for a semantic parser that can parse such rich natural language expressions.

Your assignment

Your assignment is to use SEMPRES to implement a grammar to be able to parse the following natural language inputs into a numerical answer. Submit your grammar file, and the output candidate logical forms for each statement.

1. “*One plus two*”
2. “*Add one to nine*”
3. “*Ten minus three*”
4. “*Ten take away three*”
5. “*Subtract three from five*”
6. “*Add two times three and four times five*”
7. “*Subtract four times three from nine times five*”
8. “*Add three to four and multiply that by two*”
9. “*Subtract four from seven and multiply the result by two times three*”
10. “*Add three and five and add nine to that*”
11. “*Add three and five and add nine to the result*”