Software Specifications
Predictive Recursive Descent Parsing Example

Cain Susko

Queen's University
School of Computing

March 1, 2022

Recall

given the set of balanced strings, the grammar for this set is:

$$< balanced > \rightarrow 0 < balanced > 1 \mid \epsilon$$

thus, the parser operations for the set are:

- if next token is 0, use $< balanced > \rightarrow 0 < balanced > 1$
- if next token is 1 or EOS, then use $< balanced > \rightarrow \epsilon$

This is the process commonly used in program compilers.

Limitations there are, however, limitations to what languages can use Recursive Descent Parsing.
take for example, the language of palendromes $L_{pal} = \{w \in \{0,1\} \mid w = w^r\}$ where if $w = 110$ then $w^r = 011$
the Grammar would thus be:

$$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$$

which outlines the problems of predictive parsing with L_{pal}:

- two productions for same variable begin with the same token

  $$S \rightarrow 0S0 \mid 0$$

- the variable S has productions that begin with 0 *and* 0 can occur directly after S **and** S has a erasing production:

  $$S \rightarrow 0S0 \quad S \rightarrow \epsilon$$

  the problem is there is no way for the parser to know which production to use and there is more than 1 option.