

Software Specifications Grammars and Recursive Descent

Cain Susko

Queen's University
School of Computing

February 28, 2022

Recursive Descent Parsing

A context free grammar allows recursive descent parsing iff for any two productions having the same variable on the left side:

$$N \rightarrow \alpha \mid \beta$$

the following condition holds:

$$\text{RD1 } FIRST(\alpha) \cap FIRST(\beta) = \emptyset$$

RD2 if $\beta \Rightarrow^* \epsilon$ then:

$$FIRST(\alpha) \cap FOLLOW(N) = \emptyset$$

Example

given:

$$S \rightarrow cAa \mid aAb \mid bB$$

$$A \rightarrow dAb \mid cB \mid \epsilon$$

$$B \rightarrow bB \mid cBa \mid \epsilon$$

to show that Recursive Descent is possible on the above grammar we must show that it satisfies the above requirements, observe the following. Note, we can use the derivations of the above grammar in order to help find the *FOLLOW* sets.

RD1 productions for each variable begin with distinct terminals or are ϵ .
thus the grammar holds for RD1

RD2 $FOLLOW(S) = \{EOS\}$

$$FOLLOW(A) = \{a, b\}$$

$$FOLLOW(B) = \{EOS, a, b\}$$

thus, this grammar does not satisfy RD2 as the production $B \rightarrow bB$ which makes the following

$$FIRST(bB) \cap FOLLOW(B) \neq \emptyset$$

therefore, RD2 does not hold.