CS 241 — Lecture 15

Bartosz Antczak *Instructor: Kevin Lanctot* *February 28, 2017*

15.1 Bottom-up Parsing

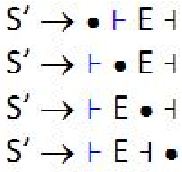
Our old strategy was using a leftmost derivation, called "LL" parsing. Now we will look at *LR Parsing*, which is more powerful than LL parsing. Basically, LR parsing does what LL parsing does, except backwards.

Working with the Stack in LR Parsing

To start, keep on shifting input onto the stack until you have a match with the right hand side of some production rule. If you match with some production rule, then you reduce your stack by changing the right hand side of some production rule to the left hand side.
From this, a natural question arises:

How do you know when to shift and when to reduce?

We introduce a place holder "*" to help us keep track of where we are in the RHS of a production rule, for instance: This allows us to keep track of the place holder.



$$S' \to \bullet \vdash E \dashv$$
$$S' \to \vdash \bullet E \dashv$$
$$S' \to \vdash E \bullet \dashv$$
$$S' \to \vdash E \dashv \bullet$$

Figure 15.1: Here, the place holder is depicted as a black dot. Courtesy of Prof. Lanctot's slides.