

Compiladores: Alguns pontos-chave na Análise Sintática

Christiano Braga

Set. 2020

Gramáticas de parsing

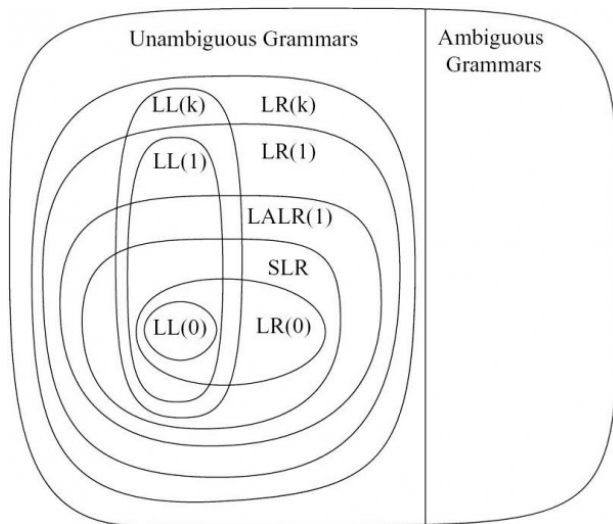


Figure 1: Gramáticas de parsing

Alguns pontos importantes

- ▶ Eficiência: Earley parser \times bottom-up (ou top-down) parsing.
- ▶ Derivação à esquerda e à direita.
- ▶ Gramáticas ambíguas.
- ▶ Recursão à esquerda, e eliminação de recursão à esquerda.
- ▶ Muita coisa não é livre de contexto:
 - ▶ Teste se uma variável foi declarada antes do seu uso não é livre de contexto.
 - ▶ Teste se número de parâmetros atuais coincide com número de parâmetros formais não é livre de contexto.

Ambiguidade

`stmt` \rightarrow `if exp then stmt` | `if exp then stmt else stmt`

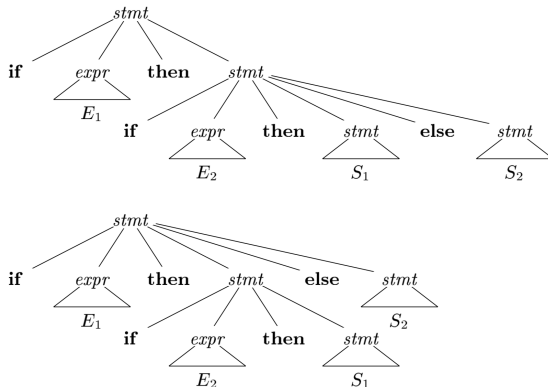


Figure 2: Duas árvores sintáticas para `if E1 then if E2 then S1 else S2`

Solução para o “dangling else”

Match each else with closest unmatched then

stmt \rightarrow matchedStmt |
 openStmt

matchedStmt \rightarrow if expr then matchedStmt else matchedStmt |
 stmt

openStmt \rightarrow if expr then stmt |
 if expr then matchedStmt else openStmt

Árvores sintática com matchedStatement

if e_1 then if e_2 then S_1 else S_2

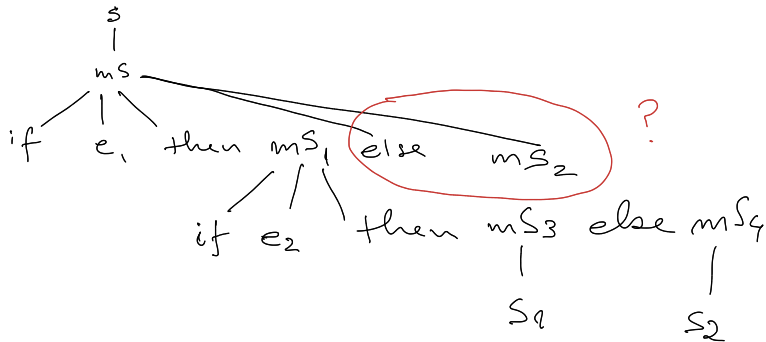


Figure 3: Usando matchedStmnt

Árvores sintática com openStament

if e_1 then if e_2 then s_1 else s_2

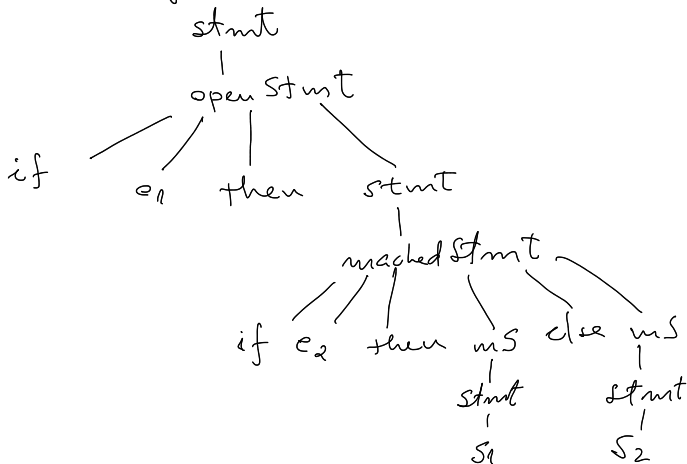


Figure 4: Usando openStmnt