# Compiladores: Alguns pontos-chave na Análise Sintática

Christiano Braga

Set. 2020

# Gramáticas de parsing

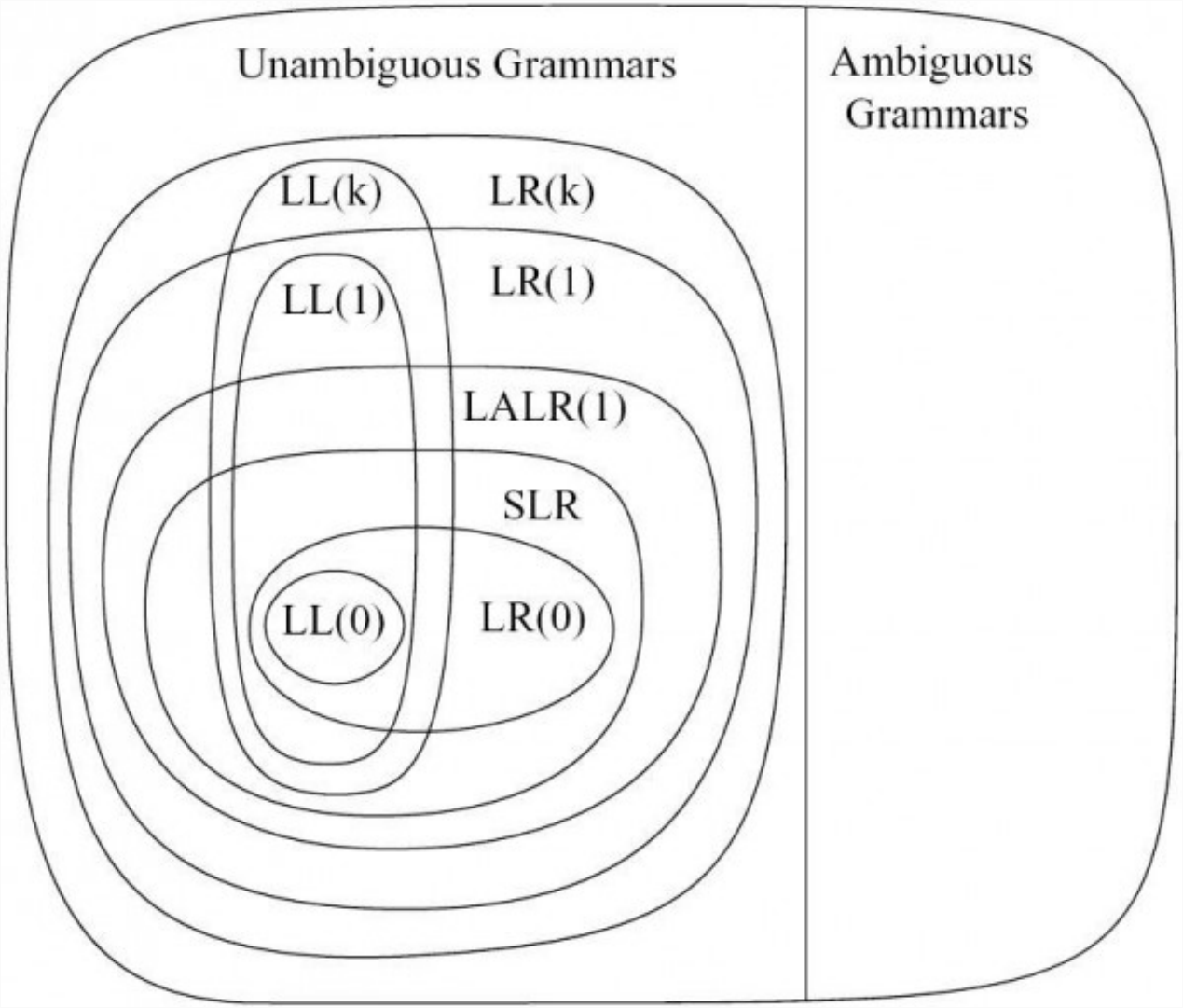


**Figure 1:** Gramáticas de parsing

# Alguns pontos importantes

- Eficiência: Earley parser $\times$ bottom-up (ou top-down) parsing.
- Derivação à esquerda e à direita.
- Gramáticas ambiguas.
- 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.
- FIRST e FOLLOW: o quê são e para que servem.

# Ambiguidade

`stmt → if exp then stmt | if exp then stmt else stmt`

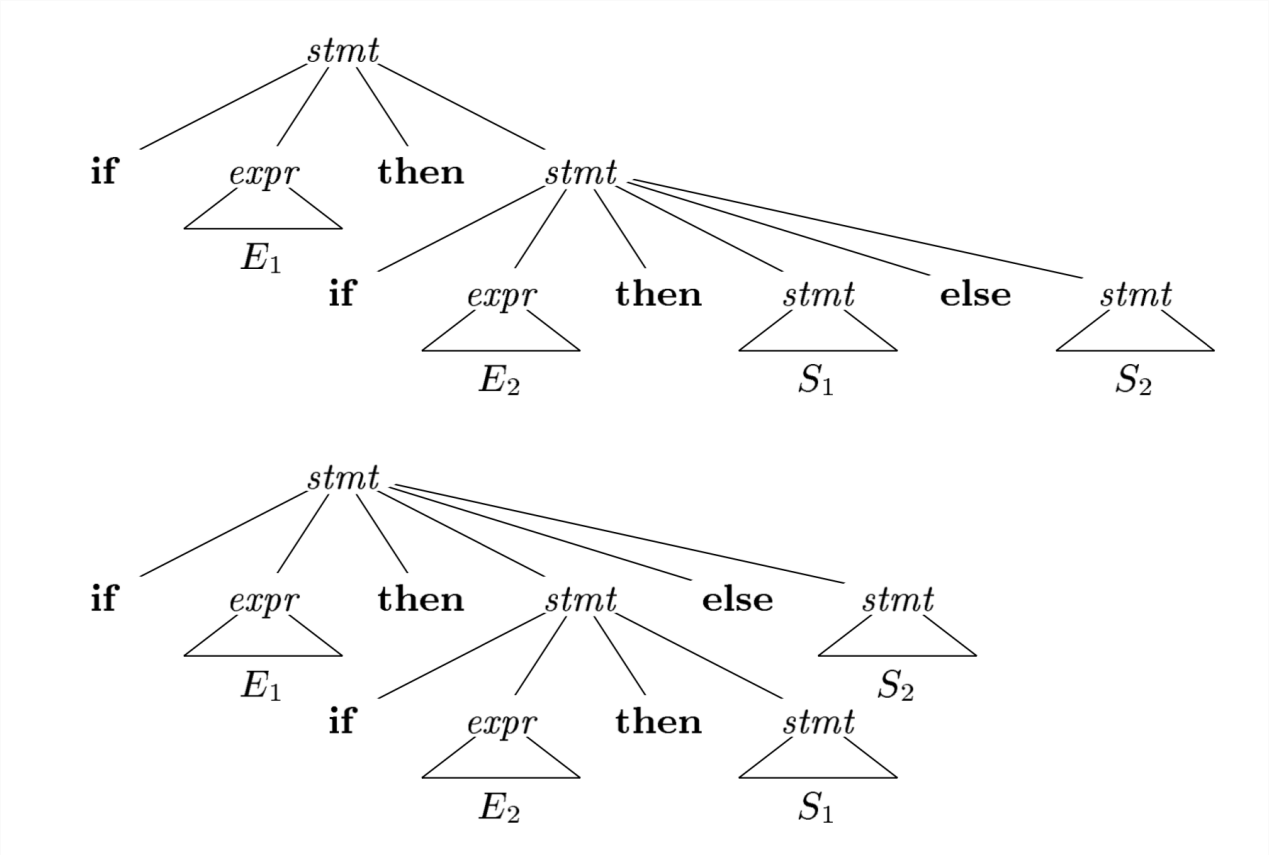


**Figure 2:** Duas árvores sintáticas para `if E1 then if E2 then S1 else S2`

# Solução para o “dangling else”

*Match each `else` with closest unmatched `then`*

```
stmt → matchedStmt |
    openStmt
matchedStmt → if expr then matchedStmt else matchedStmt |
    stmt
openStmt → if expr then stmt |
    if expr then matchedStmt else openStmt
```

# Árvores sintática com `matchedStament`

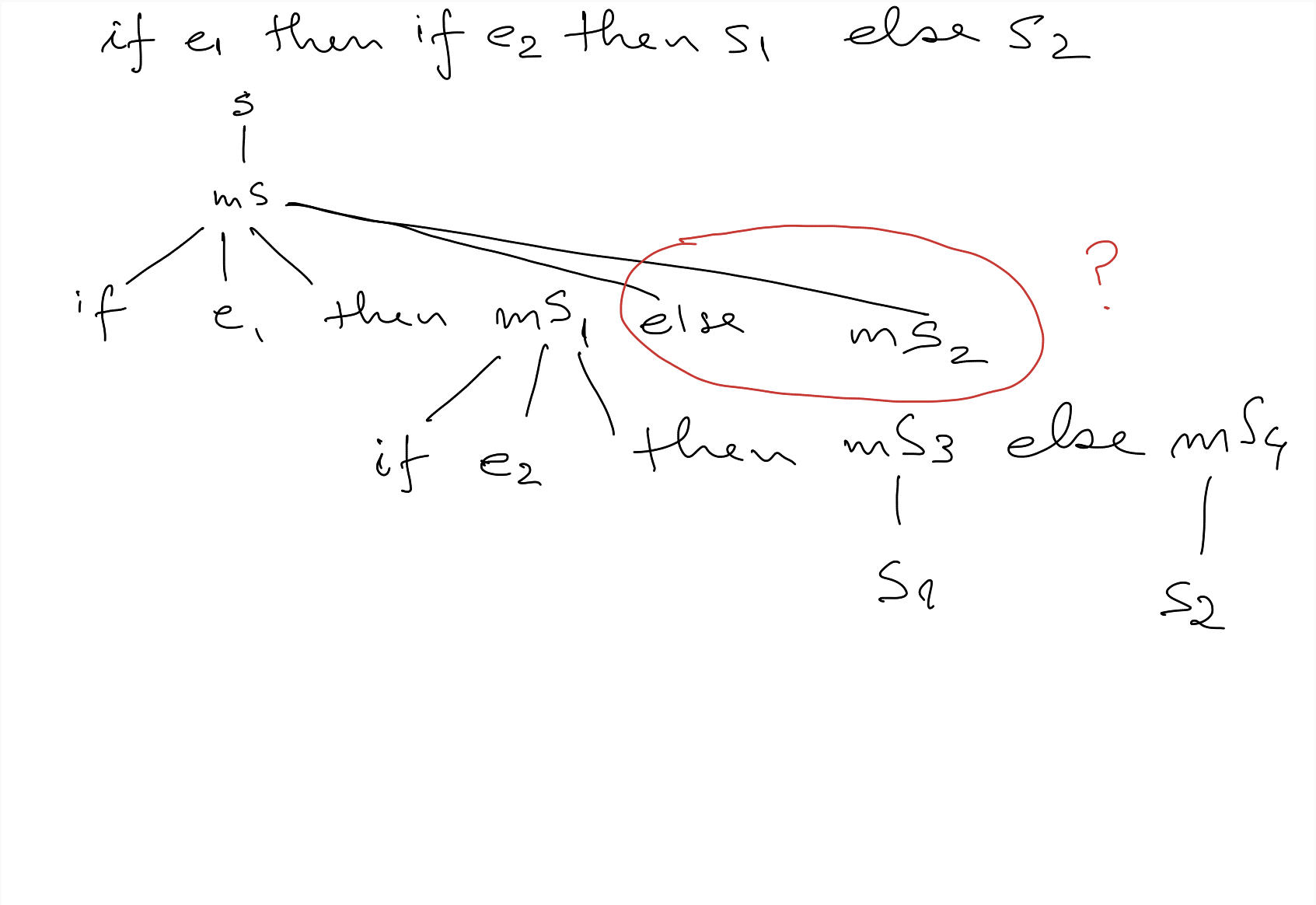


**Figure 3:** Usando `machedStmt`

# Árvores sintática com `openStament`

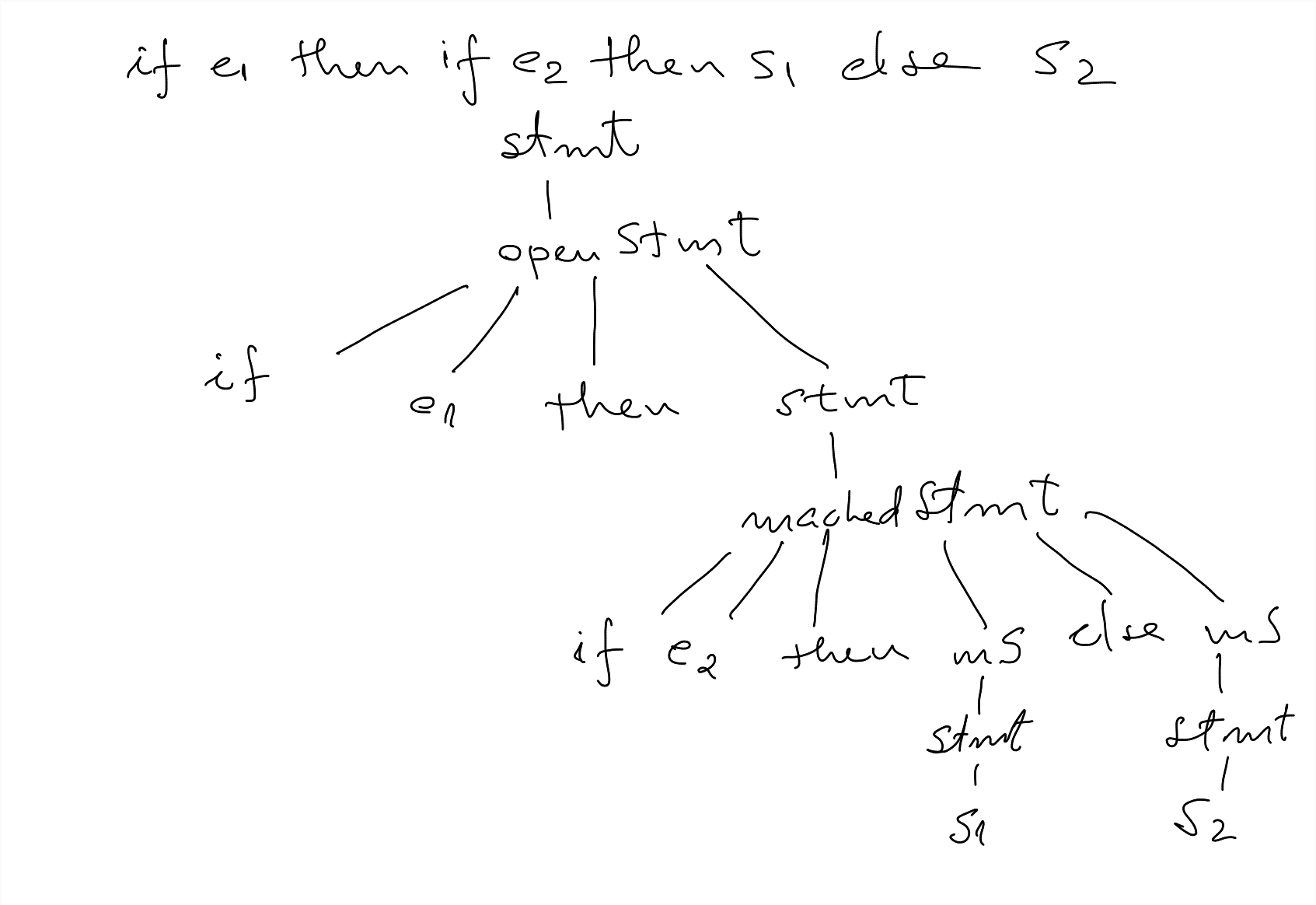


**Figure 4:** Usando `openStmt`