

Detectando ambiguidade

- Infelizmente, não existe um algoritmo para detectar se uma gramática qualquer é ambígua ou não
- Mas existem *heurísticas*, a principal delas é verificar se existe uma regra misturando *recursão à esquerda* e *recursão à direita*
 - É o caso da gramática de expressões
 - Às vezes isso é bem sutil: ambiguidade do if-else

$S \rightarrow C$

$C \rightarrow \text{if exp then } C$

$C \rightarrow \text{if exp then } C \text{ else } C$

$C \rightarrow \text{outros}$

Removendo ambiguidade

$m + n * m$
 $m + n + m$

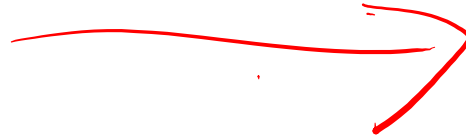
- Do mesmo modo, não há um algoritmo para remover ambiguidade
- Se a ambiguidade está na gramática, e não na própria linguagem, o jeito é encontrar a *fonte* da ambiguidade e reescrever a gramática para eliminá-la
- No caso de ambiguidade em gramáticas de expressões e operadores, a ambiguidade vem da gramática não estar levando em conta as regras de associatividade e precedência dos operadores
- Em uma gramática de expressões, cada nível de precedência tem que ganhar seu próprio não-terminal
- Operadores que devem ser associativos à esquerda precisam usar recursão à esquerda, e associativos à direita precisam de recursão à direita

Expressões simples, sem ambiguidade

- Assumindo que $*$ tem precedência sobre $+$, e ambos são associativos à esquerda (ou seja, $\text{num} + \text{num} + \text{num}$ deve ser interpretado como $(\text{num} + \text{num}) + \text{num}$)

$S \rightarrow E$
 $E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow (E)$
 $E \rightarrow \text{num}$

$* > +$



$S \rightarrow E$
 $E \rightarrow E + T$
 $E \rightarrow T$
 $T \rightarrow T * F$
 $T \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{num}$

menor precedência

Expressões simples, sem ambiguidade

- Assumindo que \wedge tem precedência sobre $*$ que tem precedência sobre $+$, \wedge é associativo à direita, $*$ e $+$ são associativos à esquerda (ou seja, $\text{num} + \text{num} + \text{num}$ deve ser interpretado como $(\text{num} + \text{num}) + \text{num}$)

$S \rightarrow E$
 $E \rightarrow E + T$
 $E \rightarrow E - T$
 $E \rightarrow T$
 $T \rightarrow T * P$
 $T \rightarrow P$
 $P \rightarrow F \wedge P$
 $P \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{num}$

If-else sem ambiguidade

- Uma solução adotada por diversas linguagens é acrescentar um delimitador que fecha o if:

$S \rightarrow C$

$C \rightarrow \text{if exp then } C \text{ end}$

$C \rightarrow \text{if exp then } C \text{ else } C \text{ end}$

$C \rightarrow \text{outros}$

- Uma desvantagem é que agora é necessário ter uma construção “else-if” para ter ifs em cascata sem uma multiplicação de ends
- E claro, estamos mudando a linguagem!

*if exp then if exp then outros else outros end end
if exp then if exp then outros end else outros end*

If-else sem ambiguidade, com a “cara de C”

- Uma solução adotada por diversas linguagens é acrescentar um delimitador que fecha o if:

$S \rightarrow C$

$C \rightarrow \text{if exp } \{ C \}$

$C \rightarrow \text{if exp } \{ C \} \text{ else } \{ C \}$

$C \rightarrow \text{outros}$

*if exp {
...
} else { if exp {
...
}
}*

- Uma desvantagem é que agora é necessário ter uma construção “else-if” para ter ifs em cascata sem uma multiplicação de ends
- E claro, estamos mudando a linguagem!

*if exp { if exp { outros } else { outros } }
if exp { if exp { outros } } else { outros }*

If-else sem mudar a linguagem

- Outra solução é separar os ifs em dois tipos, com não-terminais diferentes:

$S \rightarrow C$

$C \rightarrow \text{if exp then } C$

$C \rightarrow \text{if exp then CE else } C$

$C \rightarrow \text{outros}$

$CE \rightarrow \text{if exp then CE else CE}$

$CE \rightarrow \text{outros}$

- Notem a semelhança com a gramática não ambígua de expressões

Quiz

- Qual a versão não ambígua da gramática:

$S \rightarrow S S$

$S \rightarrow a$

$S \rightarrow b$

~~$S \rightarrow S a$~~

~~$S \rightarrow S b$~~

~~$S \rightarrow$~~

~~$S \rightarrow S$~~

~~$S \rightarrow S'$~~

~~$S' \rightarrow a$~~

~~$S' \rightarrow b$~~

~~$S \rightarrow S S'$~~

~~$S' \rightarrow a$~~

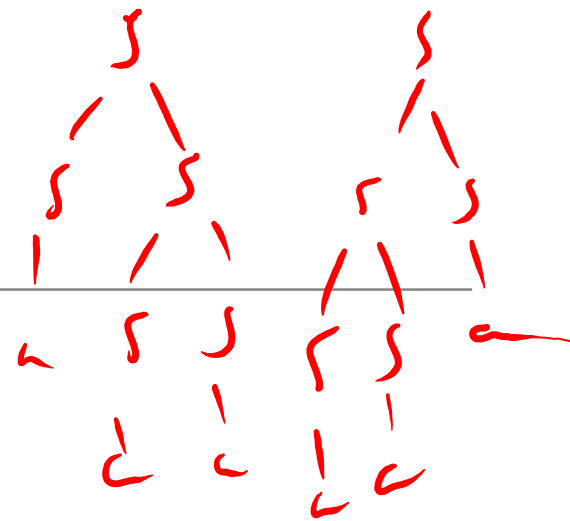
~~$S' \rightarrow b$~~

$S \rightarrow S a$

$S \rightarrow S b$

$S \rightarrow a$

$S \rightarrow b$



Quiz

- Qual a versão não ambígua da gramática:

$S \rightarrow S S$

$S \rightarrow a$

$S \rightarrow b$

$S \rightarrow S a$

$S \rightarrow S b$

$S \rightarrow$

$S \rightarrow S S'$

$S' \rightarrow a$

$S' \rightarrow b$

$S \rightarrow S$

$S \rightarrow S'$

$S' \rightarrow a$

$S' \rightarrow b$

* $S \rightarrow S a$

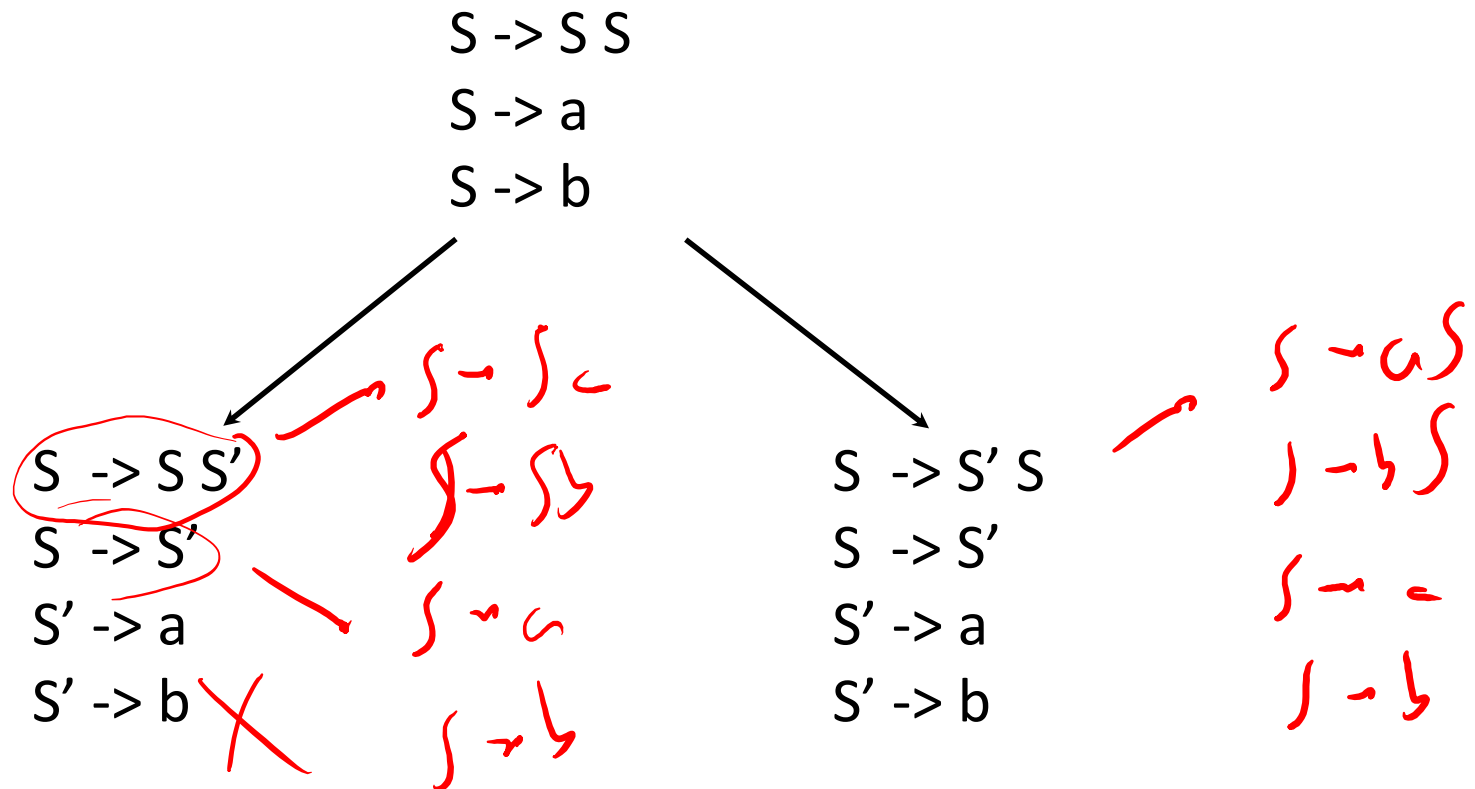
$S \rightarrow S b$

$S \rightarrow a$

$S \rightarrow b$

Quiz

- Qual a versão não ambígua da gramática:



Contornando ambiguidade

- Na prática, um uso judicioso de ambiguidade pode simplificar a gramática, e deixar ela mais natural
- Tanto a versão ambígua da gramática de expressões simples quanto a gramática do if-else são mais simples que suas versões não ambíguas!
- Podemos eliminar a ambiguidade não na gramática, mas na *implementação do analisador sintático*
- As ferramentas de geração de analisadores possuem regras de eliminação de ambiguidade, e diretivas de precedência e associatividade que permitem controlar como essa eliminação é feita