

# Parsing

①

2 técnicas : top-down e bottom-up

constroi a árvore de cima para baixo,  
da raiz até às folhas

constroi a árvore de baixo para cima,  
das folhas até à raiz.

---

## Top-Down Parsing

- Recursivo descendente ( pode requerer backtracking )  
→ método mais geral mas ineficiente
- Parsing preditivo  
→ menos geral mas mais eficiente que o anterior.

# Parsing recursivo descendente

Ilustração através de um exemplo. (Ver mais detalhes no livro)

Ex:  $S \rightarrow c A d$  } Maiúscula  $\rightarrow$  não term.  
 $A \rightarrow ab | a$  } Minúscula  $\rightarrow$  terminal.

• input : c a d

(a)  
S

c a d  
↑

init.

(b)  
S  
/ | \  
c A d

c a d  
↑

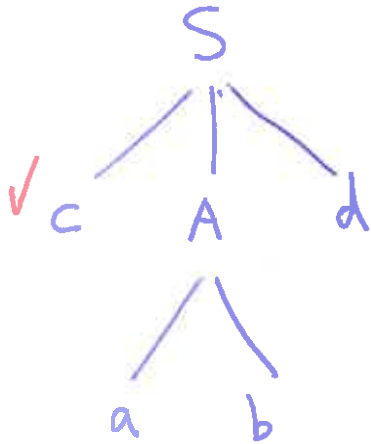
expand S  
 $S \Rightarrow c A d$

(c)  
S  
/ | \  
✓ c A d

c a d  
↑

match c  
avança o input

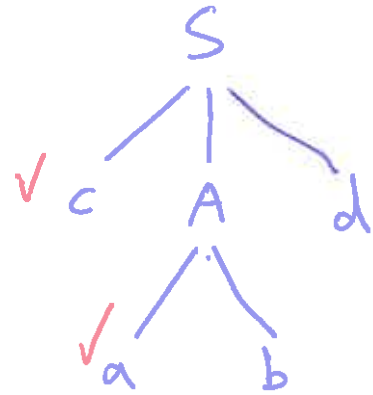
(d)



c a d  
          ↑

expand  $A \Rightarrow ab$

(e)



c a d  
          ↑

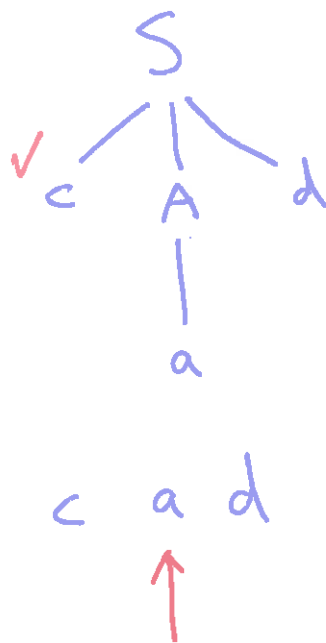
match a  
advance 0 input

(f)

④

d não faz match com b

→ volta para trás (backtracking) e tenta outra expansão de A (é necessário fazer reset ao input de modo a que o token corrente fique na posição em que estava quando chegamos a A (figura c))



Expand  $A \Rightarrow a$

g) match a

h) match d

Chega-se ao fim do input e obteve-se uma árvore de parsing. Reportamos "Parsing successful". Caso não conseguíssemos reportávamos "Parsing not successful".