

## Algo naïf. intro parsing

ed. top naïf ( $\alpha, u$ )

$\alpha$ : préfixe  
 $u$ : mot à analyser

if ( $\alpha == u$ ): return True

if  $|\alpha|_x > |u|$ : return False

Une condition d'arrêt  
parmi d'autres\*

$\alpha = wA\gamma$  où  $w \in X^*$   
 $\gamma \in (X \cup V)^*$   
 $A \in V$

$|\alpha|_x$ : nb de terminaux

for  $r$  in gram:

si  $r = A \rightarrow \beta$ ,

si ed. top naïf ( $w\beta\gamma, u$ ): return True

return False.

\* longueur de la préfixe (incertain)

\* nb de non-terminaux (sûr, mais coûteux)

\* préfixes qui ne matchent pas (cf algs suivants).

## Algo descendant std, version avec match & stripping.

ed. standard ( $\alpha, u$ )

if ( $\alpha, u$ ) == ( $\epsilon, \epsilon$ ): return True

if  $\alpha == u$ : return True

if  $\alpha == \epsilon$ : return False

$u = wA\alpha$   $w \in X^*$

si  $w = \epsilon$ :

Pour toute règle  $A \rightarrow \beta$ :

si ed. standard ( $\beta\alpha, u$ ): return True

sinon ( $\exists$  préfixe terminal: apparicement)

si  $u = w\eta$ :

return ed. standard ( $A\alpha, \eta$ )

sinon

return False.