Parsing
Unger's parser: Example

Laura Kallmeyer

Winter 2016/17

Consider the following CFG: $S \to Tc, S \to AC, T \to AB, C \to Bc, A \to a, B \to b$

This CFG generates only the word *abc* but with two different analyses.

We assume that we have a CFG without ϵ-productions and without loops (is the case for this grammar). Furthermore, we assume that when partitioning the input into spans of rhs symbols, terminals receive a span of length 1.

Trace of the Unger parse of *abc*:

calls	results (in chart)	productions (parse forest)
`unger(`S, ${}_1abc_3$`)`		
$S \to Tc$?		
`unger(`T, ${}_1ab_2$`)`		
$T \to AB$?		
`unger(`A, ${}_1a_1$`)`		
$A \to a$?		
`unger(`a, ${}_1a_1$`)` $\to$ `t`	$\langle a, {}_1a_1, t\rangle$	${}_1A_1 \to {}_1a_1$
$\to$ `t`	$\langle A, {}_1a_1, t\rangle$	
`unger(`B, ${}_2b_2$`)`		
$B \to b$?		
`unger(`b, ${}_2b_2$`)` $\to$ `t`	$\langle b, {}_2b_2, t\rangle$	${}_2B_2 \to {}_2b_2$
$\to$ `t`	$\langle B, {}_2b_2, t\rangle$	${}_1T_2 \to {}_1A_1\, {}_2B_2$
$\to$ `t`	$\langle T, {}_1ab_2, t\rangle$	
`unger(`c, ${}_3c_3$`)` $\to$ `t`	$\langle c, {}_3c_3, t\rangle$	${}_1S_3 \to {}_1T_2\, {}_3c_3$
$S \to AC$?		
`unger(`A, ${}_1a_1$`)` $\to$ `t`		
`unger(`C, ${}_2bc_3$`)`		
$C \to Bc$?		
`unger(`B, ${}_2b_2$`)` $\to$ `t`		
`unger(`c, ${}_3c_3$`)` $\to$ `t`		${}_2C_3 \to {}_2B_2\, {}_3c_3$
$\to$ `t`	$\langle C, {}_2bc_3, t\rangle$	${}_1S_3 \to {}_1A_1\, {}_2C_3$
`unger(`A, ${}_1ab_2$`)`		
$A \to a$? $\to$ no partition		
$\to$ `f`	$\langle A, {}_1ab_2, f\rangle$	
$\to$ `t`	$\langle S, {}_1abc_3, t\rangle$	