L-Attributed Grammars

L-Attributed Grammars is **a class** of Attributed Grammars (or SDD) that has **Depth-First** as a **Topological Sort** of the **Dependency Graph** of the **Parse-Tree attributes** of the grammar.

Let $G^A \equiv \{\sum, V, s, P^A, \{a_i\}\}$ be an attribute grammar.
Let $p \equiv B := B_1 \ldots B_n \{\alpha\} \in P^A$.
G^A is L-attributed if and only if:
$\forall X_i.a_{ij} = e_{ij} \in \{\alpha\}$ for $X_i \in Sym(B_1 \ldots B_n)$:
if $X_k.a_{ik} \in Var(e_{ij})$ then:
- either $X_i \equiv B_{hi}$, $X_k \equiv B_{hk}$ and $1 \le h_k \le h_i \le n$
- or $X_k \equiv B$ and $a_{ik} \in$ A-Inh(B)

- **S-attributed Grammars** are containing only synthesized attributes
- S-attributed are L-attributed.

Theorem. If G has Top-Down/Bottom-up Parser and G^A is L-attribued then **G^A has Top-Down/Bottom-up oblivious evaluator**

2

Bottom-Up Evaluator for S-attributed How do it by extending LR Parsers

Extend the values of the push-down automata, LR control stack:

- Associate to each grammatical symbol B:
  - the syntesized attributes or none (if it has no attribute)
  - the transtion state of LR analysis

- At each reduction with handle A::=B1…Bn {α} compute all the actions in {α}.
  - Let $A.a_i=e_i$ be one of them.
    If e_i contains occurrences of attributes of the grammatical B_i then:
    - access (n-i)-th position, below the top of the stack, and
    - select the value $I_i\ B_i\ [v_i]$ (where $[v_i]\equiv v_{i1}\ldots v_{in}$) and find the correct v_{ij}
  - Let $[v]\equiv v_1\ldots v_m$ be the values resulting for the attributes $a_1\ldots a_m$ of A.
- Reduce and insert $I_j\ A\ [v]$, where I_j is the transition state of LR analysis.

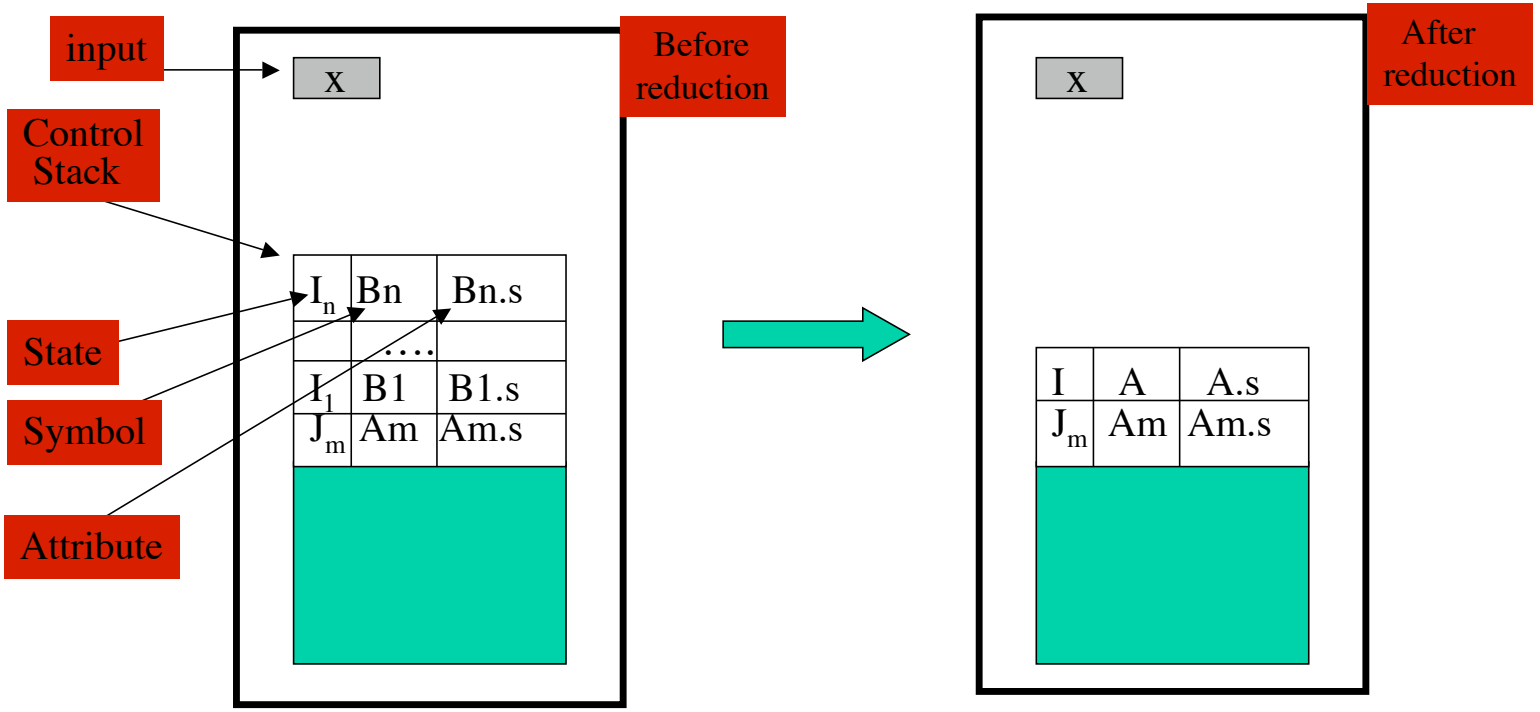
How do it: LR Control Stack

production

(k) A::=B1…Bn {α}

LR Table

Action(I_n,x)=R/k Goto(J_m,A)=I



Each B*i* and its attributes B*i*.s are computed by the previvious reductions (sons - depth first)

A.s=α has been just computed: It can only depend from A-Syn(B*i*) (A's sons) that are in the stack

4

EXAMPLE. We apply the obtained approach in implementing an evaluator for an S-attributed (only synthesized attributes)

a) A (language of expressions over $\mathbb{N}$) grammar, G

G:
0: $E ::= E + N$
1: $E ::= N$

$Colle_G(1)$:

$I_0 = \{E' \to .E,\ E \to .E+N,\ E \to .N\}$
$GoTo(0,E) = \{E' \to E.,\ E \to E.+N\} = I_1$
$GoTo(0,N) = \{E \to N.\} = I_2$
$GoTo(1,+) = \{E \to E+.N\} = I_3$
$GoTo(3,N) = \{E \to E+N.\} = I_4$

SLR-Parsing Table

	+	N	$	E
0	-	S/2	-	1
1	S/3	-	Acc	-
2	R/1		R/1	-
3	-	S/4	-	-
4	R/0	-	R/0	-

b) An attribute grammar G^A that extends G with an INTERPRETER of the expressions

G:
$E_1 ::= E_2 + N$ {$E_1.V := E_2.V + N.VAL$;}
$E ::= N$ {$E.V := N.VAL$;}

- where, N.VAL is the integer that is associated to token N in the Symbol Table (lexical may provide for the conversion of the lexeme in a machine representation of int.

c) The PARSER of G during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$

b) An attribute g^A that extends G with an INTERPRETER of the expressions

G:

$E_1 ::= E_2 + N \quad \{E_1.V := E_2.V + N.VAL;\}$

$E ::= N \quad \{E.V := N.VAL;\}$

SLR-Parsing Table

	+	N	$	E
0	-	S2	-	1
1	S3	-	Acc	-
2	R1		R1	-
3	-	S4	-	-
4	R0	-	R0	-

c) The PARSER of G during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$

$N_1 + N_2$ $	+ N_2 $	+ N_2 $	N_2 $	$	$
0	2	1	3	4	1
$	N_1	E	+	N_2	E
	0	0	1	3	0
	$	$	E	+	$
			0	1	
			$	E	
				0	
				$	

Stop: ACCEPT

d) The EVALUATOR during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$

b) An attribute g^A that extends G with an INTERPRETER of the expressions

G:

$E_1 ::= E_2 + N \quad \{E_1.V := E_2.V + N.VAL;\}$

$E ::= N \quad \{E.V := N.VAL;\}$

	+	N	$	E
0	-	s/2	-	1
1	s/3	-	Acc	-
2	R/1		R/1	-
3	-	s/4	-	-
4	R/0	-	R/0	-

d) The EVALUATOR during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$

N_1	+	N_2	$
0	$	-	

	+	N_2	$
2	N_1	3	
0	$	-	

	+	N_2	$
1	E	3	
0	$	-	

	N_2	$	
3	+	-	
1	E	3	
0	$	-	

		$	
4	N_2	7	
3	+	-	
1	E	3	
0	$	-	

	$	
1	E	10
0	$	-

Stop: ACCEPT

Top-Down Evaluators for L-Attributed From L-Attributed to Translation Schemes

Translation Schemes = Grammars with Productions where actions and grammatical symbols are mixed

$$A ::= \{\beta 1\} B1 \ldots \{\beta k\} Bk \{\alpha\}$$

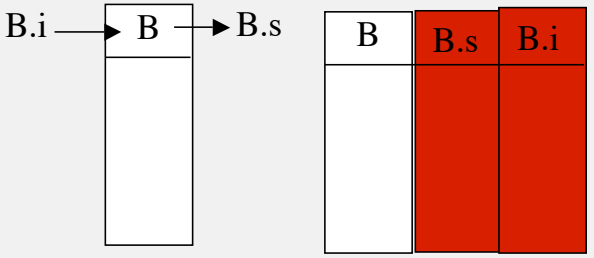
in a way that:

- A-Inh(Bi) are defined only in actions {βi} that precede Bi (for ach i)
- A-Syn(A) are defined in {α}

If G is L-attributed, its TS has actions that can use only, attributes of symbols that precede the actions.

Top-Down Evaluator for L-attributed How do it by extending LL Parsers

- **Transform** L-attributed **in Translation Scheme**
- **Pair** the LL control stack, C, with
  - one data **stack for synthesized** values, **S**,
  - one data **stack for inherited** values, **I**.
- **Extend C to contain actions**:
  - **At each derivation** with **A::={β1}B1...{βk}Bk{α},**
    - {β1}B1...{βk}Bk{α}
  - (Let B0≡A and $\beta_{k+1}\equiv\alpha$)

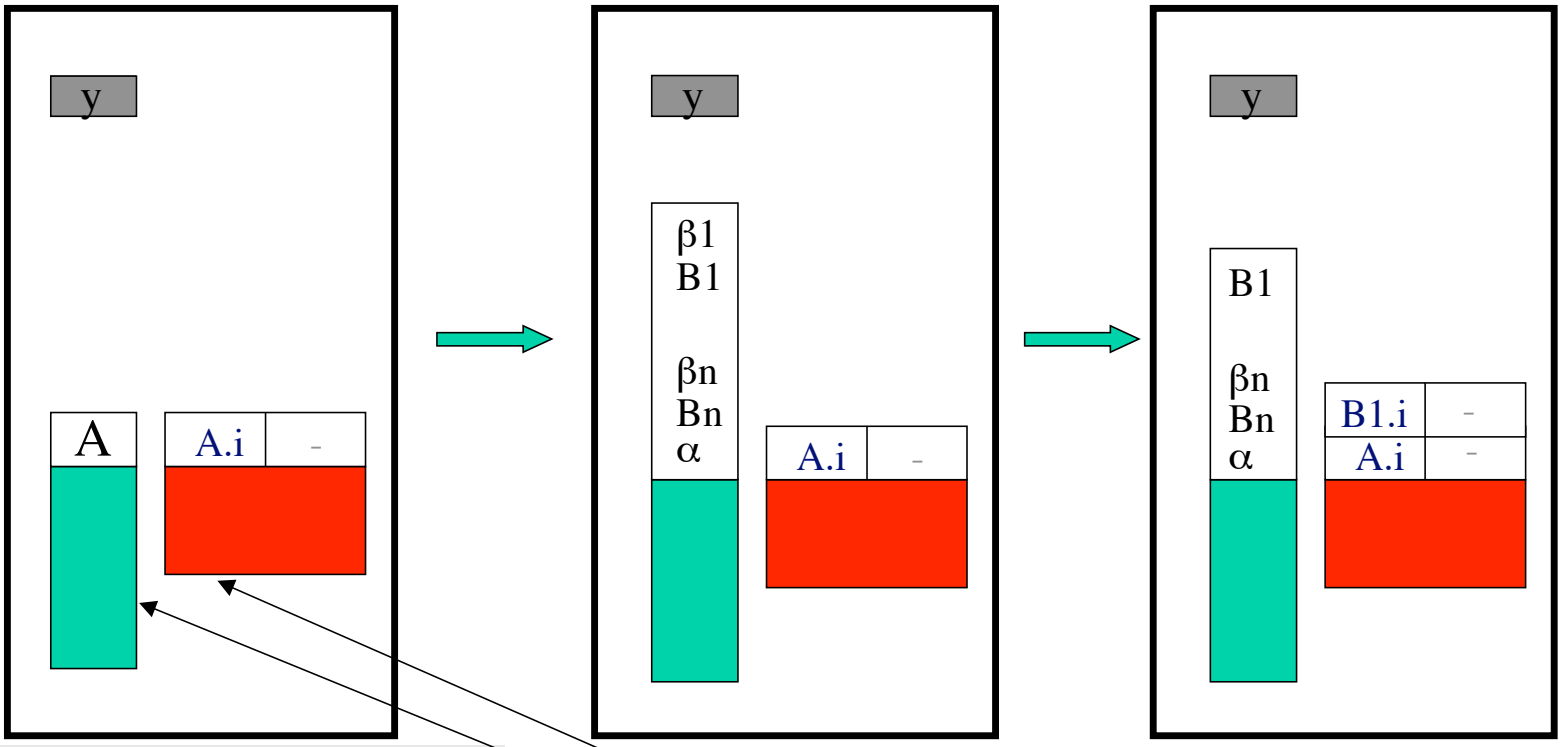


When an action βi (1≤i≤k+1) is selected from the top of C
- Action **is evaluated:**
  - **by using the evaluator** of Meta, and
  - **by replacing attributes** of:
    - **Bj** (j<i) with the values extracted, from I or S, at the (i-j-1)-th position from top
    - **A -** as above, by letting: B0≡A and $\beta_{k+1}\equiv\alpha$
  - **by putting its result on**:
    - the **top of I**, if action is βi
    - **k-th position below** top of S, if action is α

How do it: LL Control Stack - 1

(k) A::={β1}B1…{βn} Bn {α}

M(A,y)=k



A.i comes in from the previous derivation that involved its brothers at left

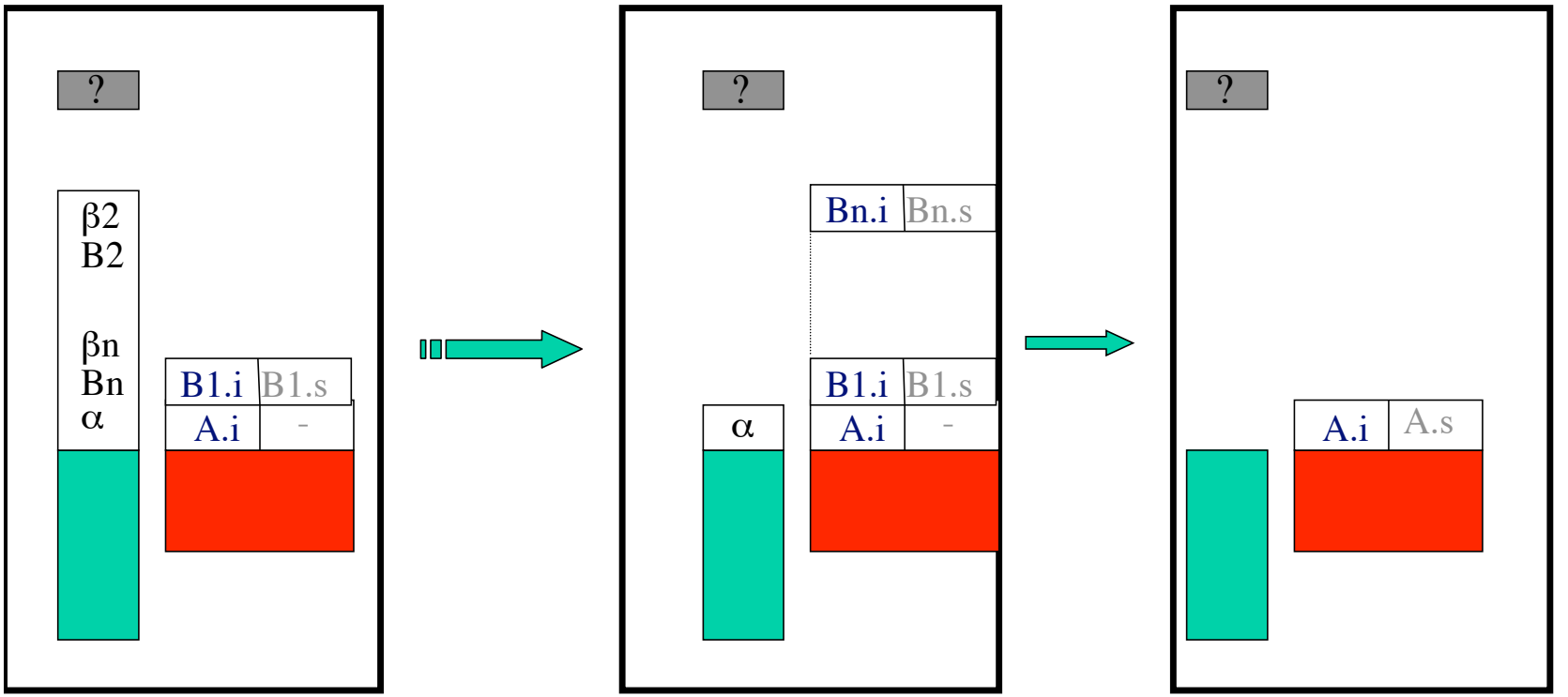
Stack C

Stack I/S

17

B1.i=β1 can contain only inherithed of A

How do it: LL Control Stack - 2



All the attributes that α can use

Top of Data Stacks just after the derivation from A completes

EXAMPLE. We apply the oblivious approach in implementing a Top-down evaluation for an L-attributed (only synthesized attributes)

a) A (language of expressions over $\mathbb{N}$) grammar:

G:

$F ::= N * F$

$F ::= N$

LL(1) grammar

0 $F ::= N\ G$

1 $G ::= * F$

2 $G ::= \varepsilon$

G

L L(parsing) table

	*	N	$
F	-	0	-
G	1	-	2

b) An attribute grammar G^A that extends G with an INTERPRETER of the expressions

$F ::= N$ { G.inh := N.val; }

G { F.v := G.v; }

$G ::= *$

F { G.v := G.inh * F.v; }

$G ::= \varepsilon$ { G.v := G.inh; }

- where, N.VAL is the integer that is associated to token N in the symbol table (lexical may provide for the conversion of the lexeme in a machine representation of int.

c) The PARSER of G during the analysis of: 3+7+5, i.e: $N_1 + N_2 + N_3$

b) An attribute g^A that extends G with an INTERPRETER of the expressions

F ::= N {G.inv := N.val;}
      G {F.v := G.v;}

G ::= * F {G.v := G.inv * F.v;}

G ::= {G.v := G.inv;}

L LParsingTable

	*	N	$
F	-	0	-
G	1	-	2

c) The PARSER of G during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$

$N_1 * N_2 \$$	$N_1 * N_2 \$$	$* N_2 \$$	$* N_2 \$$	$N_2 \$$	$N_2 \$$	$\$$	$\$$
F	N	G	*	F	N	G	$
$	G	$	F	$	G	$	
	$		$		$		

ACCEPT

d) The EVALUATOR during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$ [in the data stack, only attributes for N, F, G]

b) An attribute g^A that extends G with an INTERPRETER of the expressions

$F ::= N \; \{G.inv := N.Value;\}^{\beta_1}$
$\quad\quad G \; \{F.V := G.V;\}^{\beta_2}$

$G ::= *$
$\quad\quad F \; \{G.V := G.inv * F.V;\}^{\beta_3}$

$G ::= \quad \{G.V := G.inv;\}^{\beta_4}$

L	L.Parsing Table		
	*	N	$
F	-	0	-
G	1	-	2

c) The EVALUATOR during the analysis of: 3+7+5 i.e. $N_1 + N_2 + N_3$ [in the data stack, only attributes for N, F, G]

	N_1	*	N_2	$
F				
$	-		-	

Init Sym

	N_1	*	N_2	$
N				
β_1				
G				
β_2	-		-	
$	-		-	

	*	N_2	$
β_1			
G			
β_2	-	3	
$	-		-

	*	N_2	$
G	3	-	
β_2	-	3	
$	-		-

	*	N_2	$
*			
F			
β_3	3	-	
β_2	-	3	
$	-		-

	N_2	$
F	-	-
β_3	3	-
β_2	-	3
$	-	-

	N_2	$
N		
β_1		
G	-	-
β_2	-	-
β_3	3	-
β_2	-	3
$	-	-

- continued -

b) An attribute G^A that extends G with an INTERPRETER of the expressions

F ::= N {β1 G.inv := N.val;} G {β2 F.v := G.v;}

G ::= * F {β3 G.v := G.inv * F.v;}

G ::= {β4 G.v := G.inv;}

LL Parsing Table

	*	N	$
F	-	0	-
G	1	-	2

$

β1		
G	-	7
β2	-	-
β3	3	-
β2	-	3
$	-	-

$

	7	-
G	-	7
β2	-	-
β3	3	-
β2	-	3
$	-	-

$

	7	-
G	-	7
β2	-	-
β3	3	-
β2	-	3
$	-	-

$

	7	-
β4	-	7
β2	-	-
β3	3	-
β2	-	3
$	-	-

$

	7	7
	-	7
β2	-	-
β3	3	-
β2	-	3
$	-	-

$

	-	7
β3	3	-
β2	-	3
$	-	-

$

	3	21
β2	-	3
$	-	-

$

$	-	21