

GRAMMATICHE AD  
ATTRIBUTI

DEFINIZ.  
GUADATA  
DA  
SINTASSI

SCHEMA  
DI  
TRASLIZIONE

→ ATTRIBUTI  
REGOLE CALCOLO ATTRIBUTI

ATI SINTETIZZANTI

$\underline{A}$   $\rightarrow x_1 x_2 x_3$

DIPENDONO DAI FIGLI

ATI EREDITATI (DA SINISTRA)

$A \rightarrow x_1 \underline{x_2} x_3$

DIPENDONO DA PADRELLI ( $A x_1$ )  
DADADAD

## DIVIDERE GRAMMATICHE IN VARIE CLASSE

- S-ATTRIBUTE SOLO ATTR. SINTETIZZATI
- L-ATTRIBUTE ATTR. SINT + EREDITARI DA SX
- NON S-ATTR. NON L-ATTR. SENZA CICLI
- NON CALCOLABILI CON DIPENDENZE CICLICHE

### GLI ATTRIBUTI

→ SI CALCOLANO, SU UN INPUT, VISITANDO L'ALBERO DI DERIVAZ. SULL'INPUT IN MODO BOTTOM-UP

### GLI ATTRIBUTI

→ SI CALCOLANO ...

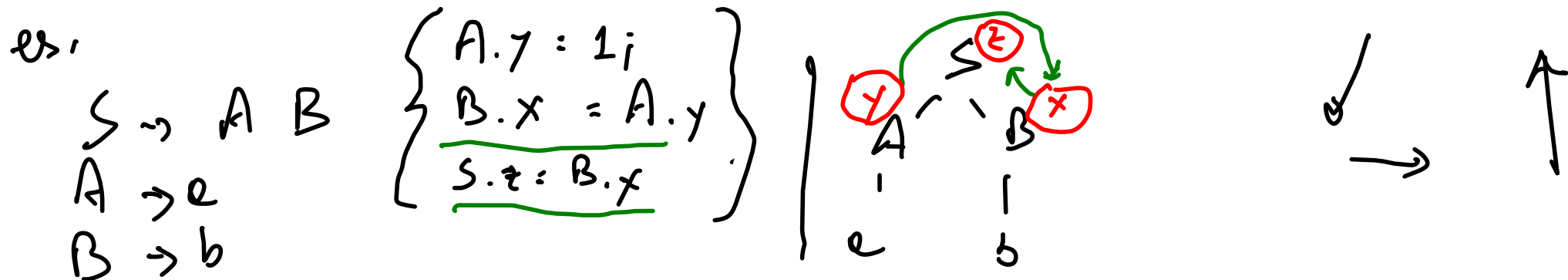
... IN MODO DEPTH-FIRST DA SX → DX

→ IL CALCOLO SI PUÒ FARE DURANTE (E NON DOPO)  
BOTTOM-UP

→ IL CALCOLO SI PUÒ FARE DURANTE TOP-DOWN

REGOLA GENERALE DI CALCOLO DI ATTRIBUTI  
DI UNA GRAMMATICA AD ATTRIBUTI SU UN DATO  
INPUT (ESAGERATO, NEL CASO DI S-ML-ATTRIBUTE)

1. COSTRUIRE ALB. DERIV. SU INPUT
2. ANNOTARE TERMIN. E NON TERM. NELL'ALBERO  
CON I PAIRI ATTRIBUTI
3. USANDO LE REGOLE DI CIASCUNA PRODUZIONE  
DELLA GRAMMATICA, CREARE GRAFO DI DIPENDENZE
4. SE NON CI SONO CICLI, SEGUIRE ORDINE TOPICOLOG.  
PER IL CALCOLO DEGLI ATTRIBUTI



PER IL PROGETTO UTILIZZEREMO

JAVACUP  $\rightarrow$  PARSING BOTTOM-UP (LAIR(1))  $\rightarrow$

COSTRUZIONE BOTTOM-UP DELL'ALBERO DI DERIVAZIONE

SU UN INPUT  $\rightarrow$  GRAMMATICHE S-ATTRIBUITE

SCHEMI DI TRADUZIONE

GRAMMATICA POSTFISSE

es.

$$E \rightarrow E_1 + E_2 \quad \left\{ E.val = E_1.val + E_2.val \right\}$$

JavaCup trasforma produzioni non postfisse in produzioni postfisse aggiungendo produzioni vuote

$$\begin{cases} S \rightarrow A \{val_1\} B \\ S \rightarrow A X B \\ X \rightarrow \epsilon \{val_1\} \end{cases}$$

NON POSTFISSE

POSTFISSE

# CONTARE IL 1 IN UNA STRINGA BINARIA

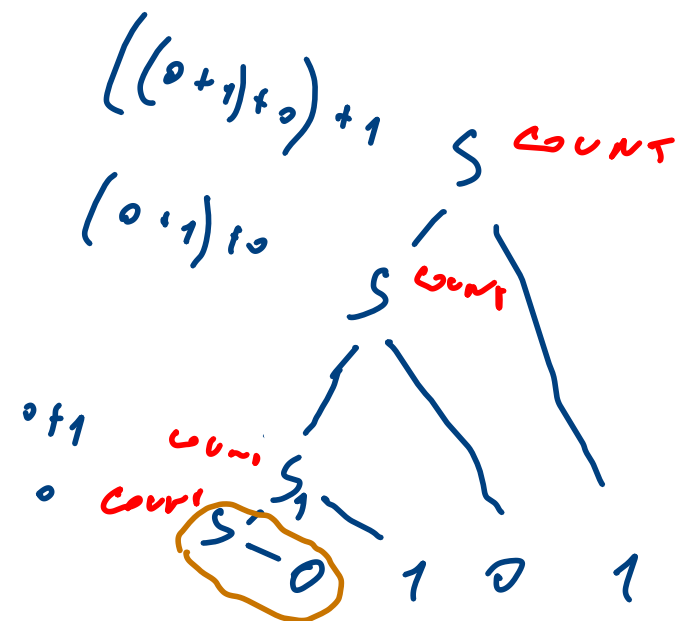
$S' \rightarrow S \quad \{ \text{STAMPA}(S.\text{COUNT}) \}$

$S \rightarrow S_1 0 \quad \{ S.\text{COUNT} = S_1.\text{COUNT} + 0 \}$

$S \rightarrow S_1 1 \quad \{ S.\text{COUNT} = S_1.\text{COUNT} + 1 \}$

$S \rightarrow 0 \quad \{ S.\text{COUNT} = 0 \}$

$S \rightarrow 1 \quad \{ S.\text{COUNT} = 1 \}$



## 0. CREARE GRAMMATICA

1. CREO INPUT SIGNIFICATIVO (CHE USI ACENDO TUTTE LE PRODUB.)

2. VI CREO ALBERO DI DERIVAZIONE

3. ANNOTARE CON ATTRIBUTI PER (OGNI) SIMBOLO E

PENSARE REGOLE "CORRE" CHE RISPETTI IL TIPO DI VISITA RICHIESTO DAL TIPO DI GRAMM AD ATTRIBUTI CHE SI VUOL CREARE

POSTFISSA  $\rightarrow$  BOTTOM-UP

PREFISSA  $\rightarrow$  D.F DA SX A DX

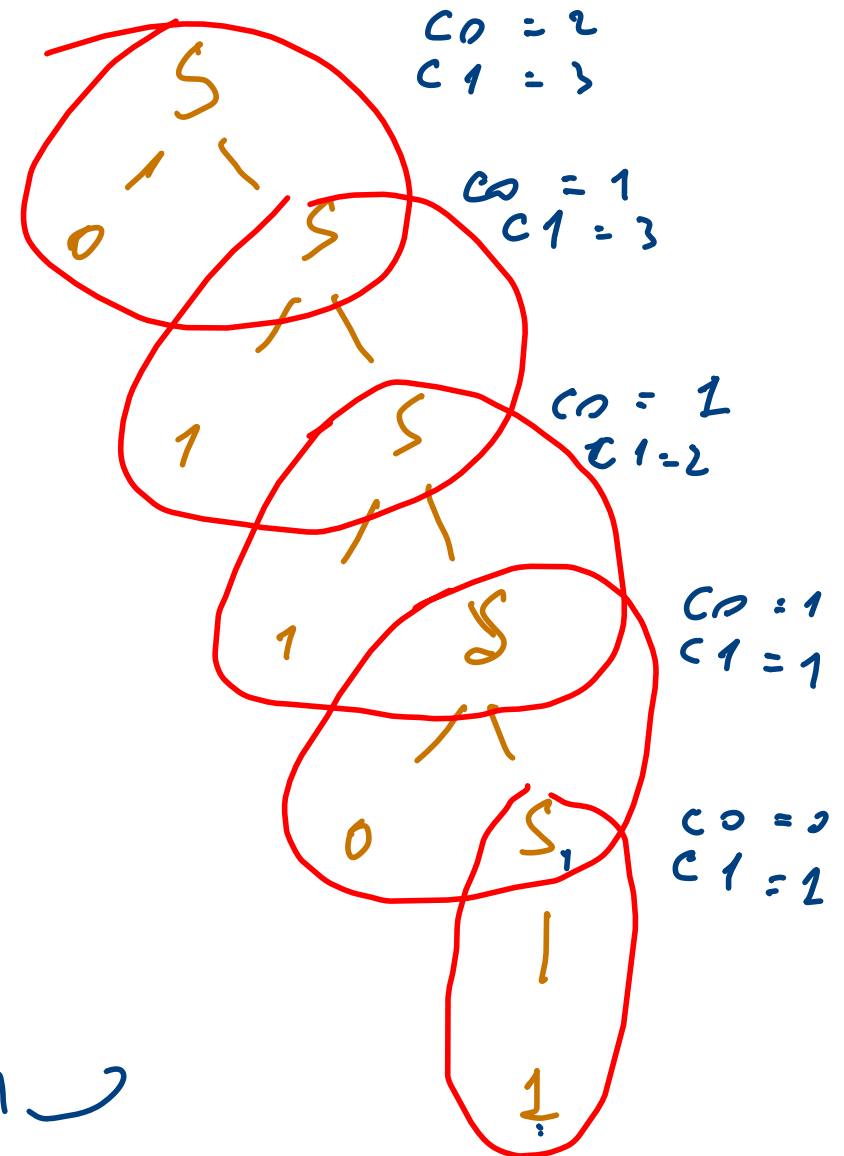
CONTARE 0 e 1

$S' \rightarrow S$   
 $S \rightarrow 0 S_1 \left\{ \begin{array}{l} S.w = S_1.w + 1 \\ S.c_1 = S_1.c_1 \end{array} \right\}$   
 $S \rightarrow 1 S_1 \left\{ \begin{array}{l} S.w = S_1.w \\ S.c_1 = S_1.c_1 + 1 \end{array} \right\}$   
 $S \rightarrow 0 \left\{ \begin{array}{l} S.w = 1 \\ S.c_1 = 0 \end{array} \right\}$   
 $S \rightarrow 1 \left\{ \begin{array}{l} S.w = 0 \\ S.c_1 = 1 \end{array} \right\}$

POSTFISSA

CREARE REGOLE

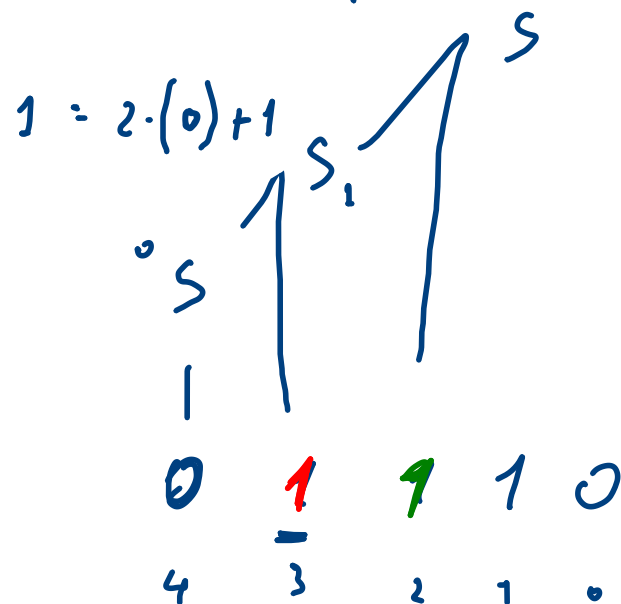
SU  
 VISITA DELL'ALBERO



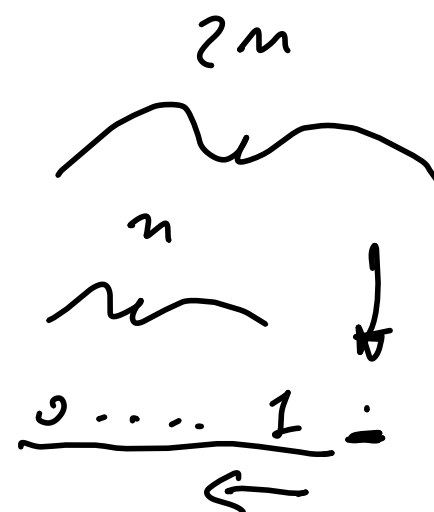
$$S \rightarrow S_1 \quad \{ S_{\text{val}} = 2 S_{1\text{val}} + 1 \}$$

$$S \rightarrow S_0 \quad \{ S_{\text{val}} = 2 S_{1\text{val}} + 0 \}$$

$$= 2(2 \cdot 0 + 1) + 1 \cdot 2^0$$



$$1 \cdot 2^0 + 0 \cdot 2^1$$



$$\rightarrow 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4$$

$$10111$$