

Traduttori a pila  
traduzioni deterministiche  
discendenti e ascendenti  
traduttori con procedure ricorsive

*Prof. A. Morzenti*  
*aa 2008-2009*

## AMBIGUITÀ DELLA GRAMMATICA SORGENTE

Le grammatiche o schemi di traduzione di interesse pratico sono quelli che definiscono una funzione a un solo valore.

Se la grammatica sorgente è ambigua  $\rightarrow$  più alberi sintattici,  
alberi sintattici possono diversi,  
diverse immagini.

Esempio: parentesizzazione ridondante

Gramm sorgente  $G_2$

$E \rightarrow \text{add } T E$

$E \rightarrow T$

$T \rightarrow \text{mult } F T$

$T \rightarrow F$

$F \rightarrow E$

$F \rightarrow i'$

Gramm pozzo  $G_1$

$E \rightarrow T + E$

$E \rightarrow T$

$T \rightarrow F \times T$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow i$

$G_2$  è ambigua perché ammette  
la derivazione circolare

$E \Rightarrow T \Rightarrow F \Rightarrow E$

$E \Rightarrow T \Rightarrow F \Rightarrow i',$

$E \Rightarrow T \Rightarrow F \Rightarrow E \Rightarrow T \Rightarrow F \Rightarrow i', \dots$

traduzioni diverse:

$E \Rightarrow T \Rightarrow F \Rightarrow i',$

$E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (T) \Rightarrow (F) \Rightarrow (i'), \dots$

Anche se la grammatica sorgente non è ambigua, la traduzione può non essere univoca.

### ESEMPIO

$$S \rightarrow \frac{a}{b} S \mid \frac{a}{c} S \mid \frac{a}{d}$$

$$G_1 = \{S \rightarrow aS \mid a\}$$

$$\tau(aa) = \{bd, cd\}$$

grammatica di traduzione  
ambigua; due o più regole di  
traduzione mappate sulla  
stessa regola sorgente

### PROPRIETÀ

Sia  $T$  una grammatica di traduzione tale che:

- 1) la grammatica sorgente  $G_1$  dello schema non sia ambigua
- 2) ogni regola di  $G_1$  corrisponda a una sola regola di  $T$

Allora la traduzione definita da  $T$  è univoca (è una funzione di traduzione a un solo valore)

ESEMPIO: Grammatica di traduzione non ambigua, grammatica sorgente ambigua.

$$S \rightarrow \frac{\text{if } c \text{ then}}{\text{if } c \text{ then}} S \frac{\varepsilon}{\text{end\_if}} \mid \frac{\text{if } c \text{ then}}{\text{if } c \text{ then}} S \frac{\text{else}}{\text{else}} S \frac{\varepsilon}{\text{end\_if}} \mid a$$

Nel progetto di compilatori si eviterà l'uso di grammatiche di traduzione che possano causare perdita di univocità della traduzione.

## GRAMMATICHE DI TRADUZIONE E TRASDUTTORE A PILA

Per eseguire le traduzioni definite da una grammatica di traduzione, occorre dotare l'automa trasduttore di una memoria illimitata, organizzata a pila. Un trasduttore o IO-automa a pila non è altro che un automa a pila arricchito della capacità di emettere una stringa di caratteri a ogni transizione.

## DEFINIZIONE – Trasduttore a pila

Otto entità:  
Q, insieme degli stati;  
 $\Sigma$ , alfabeto sorgente;  
 $\Gamma$ , alfabeto di pila;  
 $\Delta$ , alfabeto pozzo;  
 $\delta$ , funzione di transizione e d'uscita;  
 $q_0$ , stato iniziale;  
 $Z_0$ , simbolo iniziale della pila;  
F, insieme degli stati finali.

Dominio di  $\delta$ :  $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$

Codominio di  $\delta$ :  $Q \times \Gamma^* \times \Delta^*$

Significato: se  $(q'', \gamma, y) = \delta(q', a, Z)$ . Il trasduttore, dallo stato presente  $q'$ , avendo letto  $a$  dal nastro d'ingresso e  $Z$  dalla pila, si porta nello stato  $q''$ , scrive  $\gamma$  in pila e  $y$  in uscita.

Gli stati finali coincidono con Q se, come spesso avviene, il riconoscimento avviene a pila vuota.

## DALLA GRAMMATICA DI TRADUZIONE AL TRASDUTTORE A PILA

Gli schemi sintattici di traduzione e i trasduttori a pila sono due modi di descrivere una relazione di traduzione: il primo è di tipo generativo, il secondo di tipo procedurale. Dall'uno di può passare all'altro grazie alla seguente proprietà:

PROPRIETA': Una relazione di traduzione è definita da una grammatica di traduzione (o schema sintattico) se, e solo se, essa è calcolata da un trasduttore a pila.

## COSTRUZIONE DEL TRASDUTTORE A PILA PREDITTIVO INDETERMINISTICO

Formalizzazione della corrispondenza tra le regole e le mosse del trasduttore.

Normalizzazione delle regole di traduzione:

per semplificare descrizione e senza

perdere in generalità

assumiamo che le coppie presenti nella grammatica di traduzione abbiano la forma

$$\frac{\varepsilon}{v}, v \in \Delta^+ \quad \text{e} \quad \frac{b}{w}, b \in \Sigma, w \in \Delta^*$$

NB: qui assumiamo unica memoria stia nella pila: come per i riconoscitori, si possono aggiungere stati all'unità di controllo...

Chiamiamo C l'insieme delle coppie presenti nella grammatica di traduzione

	Regola	Mossa	Commento
1	$A \rightarrow \frac{\varepsilon}{v} B A_1 \dots A_n$ $n \geq 0, v \in \Delta^+, B \in V,$ $A_i \in (C \cup V)$	if cima=A then write(v); pop; push( $A_n \dots A_1 B$ )	Emetti la stringa pozzo v e impila la predizione $B A_1 \dots A_n$
2	$A \rightarrow \frac{b}{w} A_1 \dots A_n$ $n \geq 0, b \in \Sigma, w \in \Delta^*,$ $A_i \in (C \cup V)$	if (car_corr=b $\wedge$ cima=A) then write(w); pop; push( $A_n \dots A_1$ ); avanza testina lettura;	b era il carattere atteso ed è stato letto; emetti la stringa pozzo w; impila la predizione $A_1 \dots A_n$
3	$A \rightarrow B A_1 \dots A_n$ $n \geq 0, B \in V,$ $A_i \in (C \cup V)$	if (cima=A) then pop; push( $A_n \dots A_1 B$ );	impila la predizione $A_1 \dots A_n$
4	$A \rightarrow \frac{\varepsilon}{v} \quad v \in \Delta^+$	if (cima=A) then write (v); pop;	emetti la stringa pozzo v

Chiamiamo C l'insieme delle coppie presenti nella grammatica di traduzione

	Regola	Mossa	Commento
5	$A \rightarrow \varepsilon$	if cima=A then pop;	
6	per ogni coppia $\frac{\varepsilon}{v} \in C$	if (cima= $\varepsilon/v$ ) then write(v); pop;	la passata previsione $\varepsilon / v$ si attua scrivendo v
7	per ogni coppia $\frac{b}{w} \in C$	if (car_corr=b $\wedge$ cima=b/w ) then write(w); pop; avanza testina lettura;	la passata previsione b / w si attua leggendo b e scrivendo w.
8	---	if (car_corr=-  $\wedge$ pila è vuota) then accetta; alt;	la stringa sorgente è stata scandita per intero e non restano obiettivi in agenda.



ESEMPIO – Si estende il riconoscitore del linguaggio  $L = \{a^*a^mb^m \mid m > 0\}$   
per definire la traduzione  $\tau(a^ka^mb^m) = d^mc^k$

	Regola	Mossa
1 indet	$S \rightarrow \frac{a}{\epsilon} S \frac{\epsilon}{c}$	if (car_corr = a $\wedge$ cima = S) then pop; push( $\epsilon$ / cS); avanza testina lettura;
2	$S \rightarrow A$	if (cima = S) then pop; push(A);
3 indet	$A \rightarrow \frac{a}{d} A \frac{b}{\epsilon}$	if (car_corr = a $\wedge$ cima = A) then pop; write (d); push(b/ $\epsilon$ A); avanza testina lettura
4	$A \rightarrow \frac{a}{d} \frac{b}{\epsilon}$	if car_corr = a $\wedge$ cima = A then pop; write (d); push(b / $\epsilon$ ); avanza testina lettura
5	---	if (cima = $\epsilon$ / c) then pop; write(c);
6	---	if (car_corr = b $\wedge$ cima = b / $\epsilon$ ) then pop; avanza testina lettura
7	---	if (car_corr = -   $\wedge$ pila è vuota then accetta; alt;

Non tutte le relazioni definite da una grammatica di traduzione libera sono calcolabili deterministicamente

ESEMPIO – Una traduzione libera non deterministica  
(esempio classico di ciò che non si può fare)

La funzione di traduzione non può essere calcolata da un trasduttore a pila deterministico.

$$\tau(u-|)=u^R u, \quad u \in \{a,b\}^*$$

La macchina deve subito emettere la copia riflessa della stringa sorgente. Per calcolare la riflessione, si deve impilare la stringa e, dopo la lettura del terminatore, disimpilarla sul nastro d'uscita. Dopo tale azione, la pila sarà vuota la macchina avrà perso ogni informazione sulla stringa sorgente e non potrà scriverla in uscita, come richiesto.

In pratica, per fortuna, basta studiare il caso deterministico  
Vediamo ora diversi algoritmi di traduzione a seconda del tipo di analisi sintattica

## ANALISI SINTATTICA CON TRADUZIONE IN LINEA

Costruzione diretta ed efficiente della traduzione. Trasformazione di un analizzatore sintattico deterministico del linguaggio sorgente in algoritmo di traduzione.

Data una grammatica di traduzione, si suppone che la grammatica sorgente permetta la costruzione di un parsificatore deterministico. Per calcolare la traduzione, si esegue l'analisi sintattica e, via via che si costruisce un sottoalbero, se ne emette la traduzione.

**ANALIZZATORI DISCENDENTI:** non occorre porre limiti sul calcolo della traduzione

**ANALIZZATORI ASCENDENTI:** è necessario introdurre una condizione restrittiva sulla forma delle regole della grammatica di traduzione.

## TRADUZIONI DETERMINISTICHE DISCENDENTI

Se la grammatica sorgente è LL(k), il parsificatore deterministico del linguaggio sorgente, completato con le azioni di scrittura, può eseguire la traduzione.

La costruzione del trasduttore a pila è una semplice modifica di quella dell'analizzatore sintattico. Il trasduttore può essere realizzato anche a discesa ricorsiva e il caso di una grammatica di traduzione estesa può essere trattato in modo analogo.

ESEMPIO. Traduzione di una stringa nella sua riflessa dalla grammatica di traduzione

La grammatica sorgente è LL(1) con insiemi guida rispettivi {a} {b} {-|}.

Le mosse dell'automa sono:

$$S \rightarrow \frac{a}{\varepsilon} S \frac{\varepsilon}{a} \mid \frac{b}{\varepsilon} S \frac{\varepsilon}{b} \mid \varepsilon$$

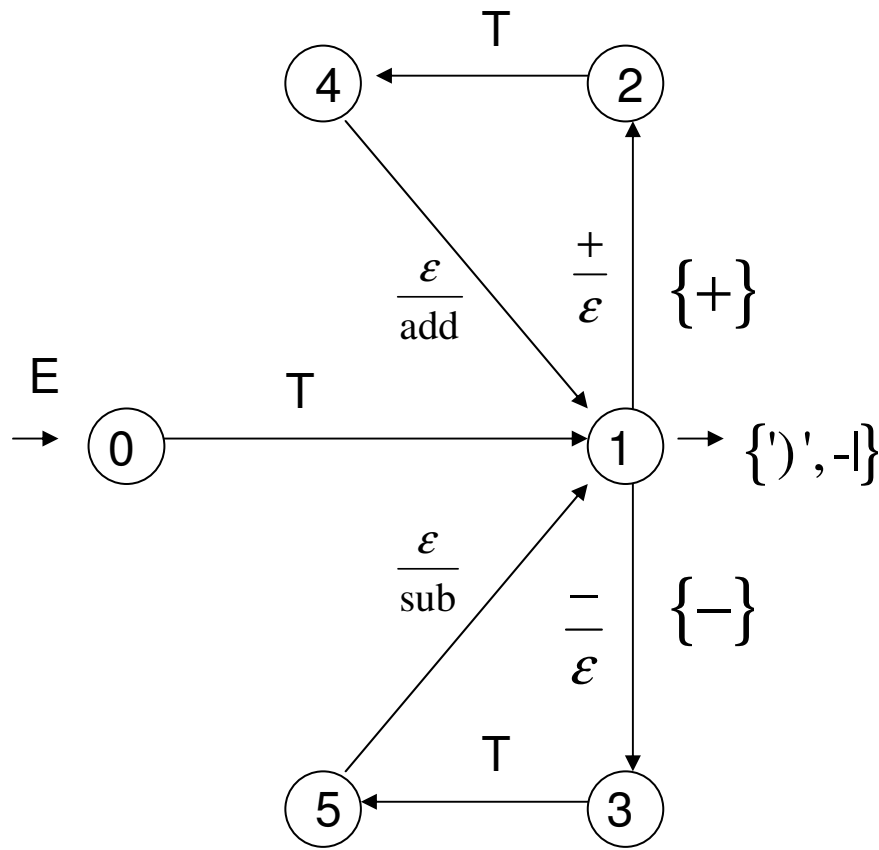
pila	cc = a	cc = b	cc = -	$\varepsilon$
S	pop; push( $\varepsilon/a$ S);	pop; push( $\varepsilon/b$ S);	pop;	
$\varepsilon / a$				write(a);
$\varepsilon / b$				write(b);

## ESEMPIO – Traduzione in postfisso – procedure ricorsive

Linguaggio sorgente: espressioni aritmetiche con operatori a due livelli di precedenza e con parentesi. La traduzione converte le espressioni nella scrittura postfissa. Es:  $v \times (v + v)$   $v \ v \ v \ \text{add} \ \text{mult}$

Grammatica di traduzione BNF estesa:

$$\begin{aligned} E &\rightarrow T \left( \frac{+}{\varepsilon} T \frac{\varepsilon}{\text{add}} \mid \frac{-}{\varepsilon} T \frac{\varepsilon}{\text{sub}} \right)^* \\ T &\rightarrow F \left( \frac{\times}{\varepsilon} F \frac{\varepsilon}{\text{mult}} \mid \frac{\div}{\varepsilon} F \frac{\varepsilon}{\text{div}} \right)^* \\ F &\rightarrow \frac{v}{v} \mid \frac{('}{\varepsilon} E \frac{')}{\varepsilon} \\ \Sigma &= \{+, \times, -, \div, (, ), v\}, \quad \Delta = \{\text{add}, \text{sub}, \text{mult}, \text{div}, v\} \end{aligned}$$



```

procedure E
call T;
while cc ∈ {+,-} do
  case
    cc = '+': begin
      cc := Prossimo; call T;
      write ('add');
    end
    cc = '-': begin
      cc := Prossimo; call T;
      write ('sub');
    end
    otherwise Errore
  end case
end do
end E
  
```

# TRADUZIONI DETERMINISTICHE ASCENDENTI

## ESEMPIO – Traduzione di espressioni in postfisso

$$E \rightarrow E \overset{+}{\mid} T \overset{\varepsilon}{\mid} T \text{ add}$$

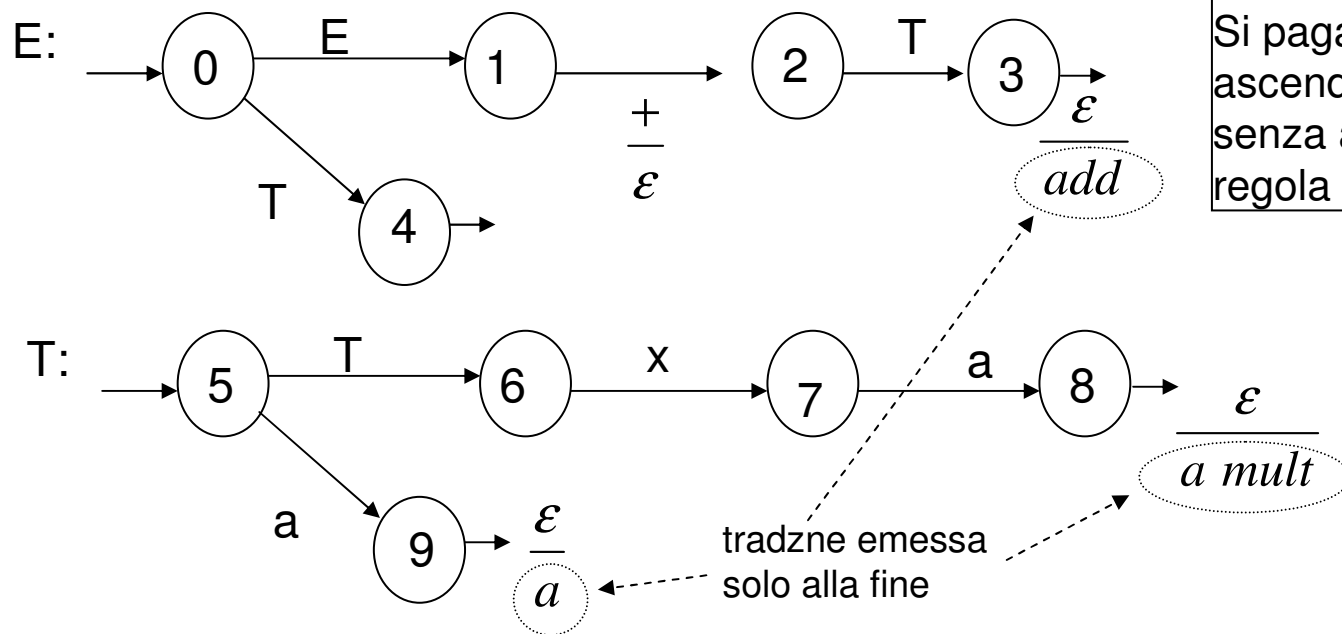
$$T \rightarrow T \overset{\times a}{\mid} T \overset{\varepsilon}{\mid} T \text{ mult} \quad \varepsilon a$$

NB: azioni di stampa associate alle sole riduzioni

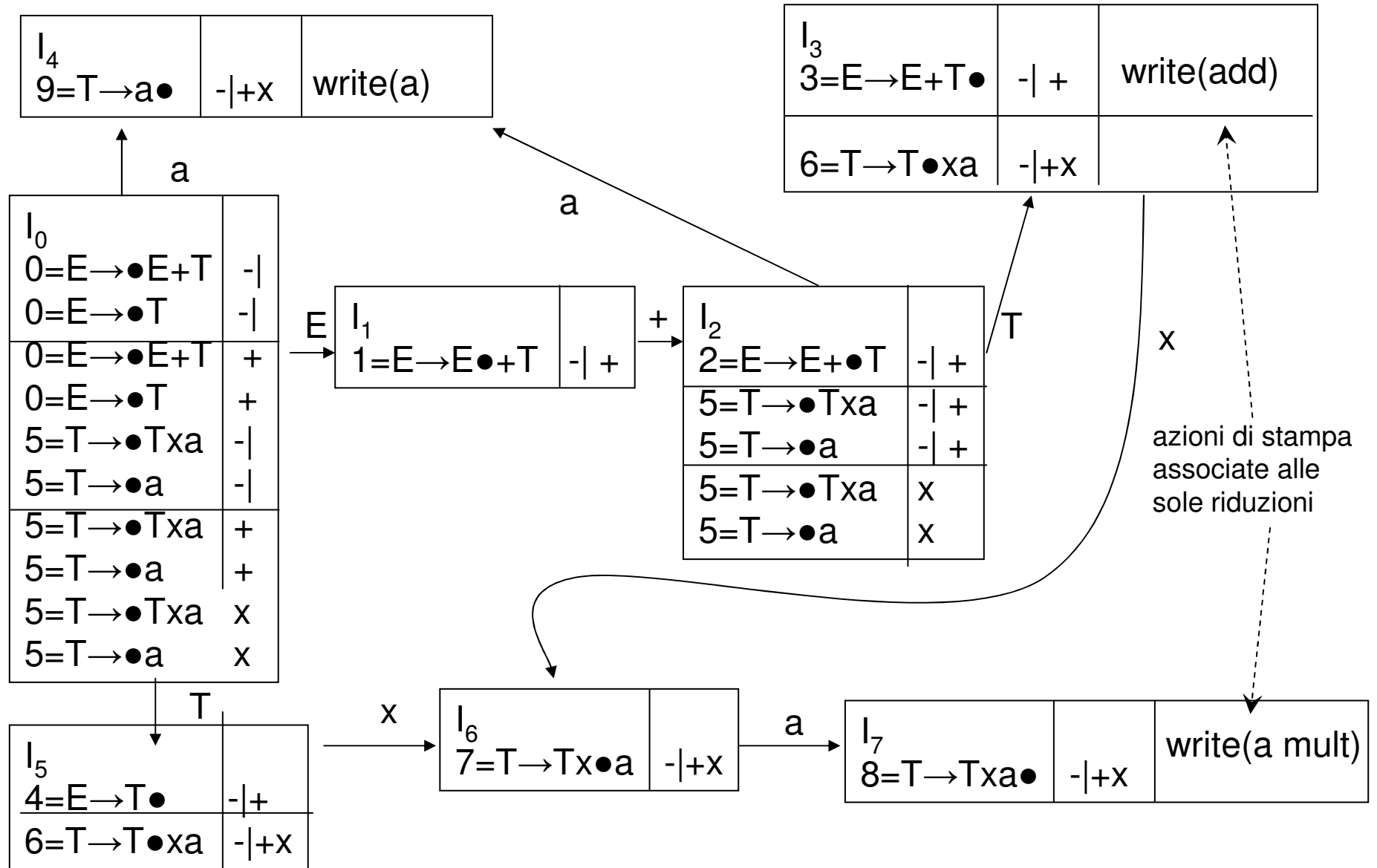
Il parsificatore, arricchito con le azioni di scrittura delle stringhe pozzo, può calcolare la traduzione ma non in tutti i casi.

Solo una mossa di riduzione può scegliere a colpo sicuro la stringa da emettere. Le mosse di spostamento impilano più regole marcate associate di norma a diverse stringhe da emettere per la traduzione

Si paga il fatto, tipico della tecnica ascendente, di scandire il testo senza aver ancora deciso quale regola applicare



## Macchina pilota del traduttore





## FORMA NORMALE POSTFISSA

Si definisce una forma normale delle grammatiche di traduzione che permette di calcolare la traduzione con azioni di scrittura nelle sole mosse di riduzione

### GRAMMATICA DI TRADUZIONE POSTFISSA:

Una grammatica di traduzione è postfissa se le regole della grammatica possono essere del tipo

$$A \rightarrow \gamma\omega \text{ dove } \gamma \in V^* \text{ e } \omega \in \Delta^*$$

Cioè non ci sono stringhe possono essere all'interno della regola, ma solo alla fine

NB: questa forma non è restrittiva

- Si possono definire le stesse traduzioni (ma in forma meno leggibile...)
- Si può trasformare una grammatica in forma postfissa