Analisi sintattica – LR(k)

Prof. A. Morzenti

aa 2008-2009

ANALISI SINTATTICA ASCENDENTE DETERMINISTICA

Il metodo LL(1) non applicabile se da uno stato di una macchina escono due frecce con insiemi guida sovrapposti.

Tattica efficace: rinviare decisione e proseguire il calcolo fino a quando incontrati simboli che consentono di decidere.

Nell'intervallo tra la comparsa dell'incertezza e la risoluzione, occorre preservare le informazioni necessarie per decidere

Analisi sintattica ascendente LR(k)
scansione del testo da sinistra a destra, R = derivazione calcolata a destra,
k = lunghezza della prospezione

Vedremo prima il metodo LR(0) poi il metodo LR(1) che consente di trattare tutti i linguaggi deterministici.

ANALISI A SPOSTAMENTO E RIDUZIONE

Gli analizzatori ascendenti considerati sono una speciale classe di automi a pila deterministici detti *a spostamento e riduzione.*

L'automa
- scandisce i caratteri del testo da sinistra a destra impilandoli (*spostamento)*,
-quando in cima alla pila parte destra di una regola (*parte riducibile*);
-la parte riducibile disimpilata: al suo posto parte sinistra (*riduzione*)

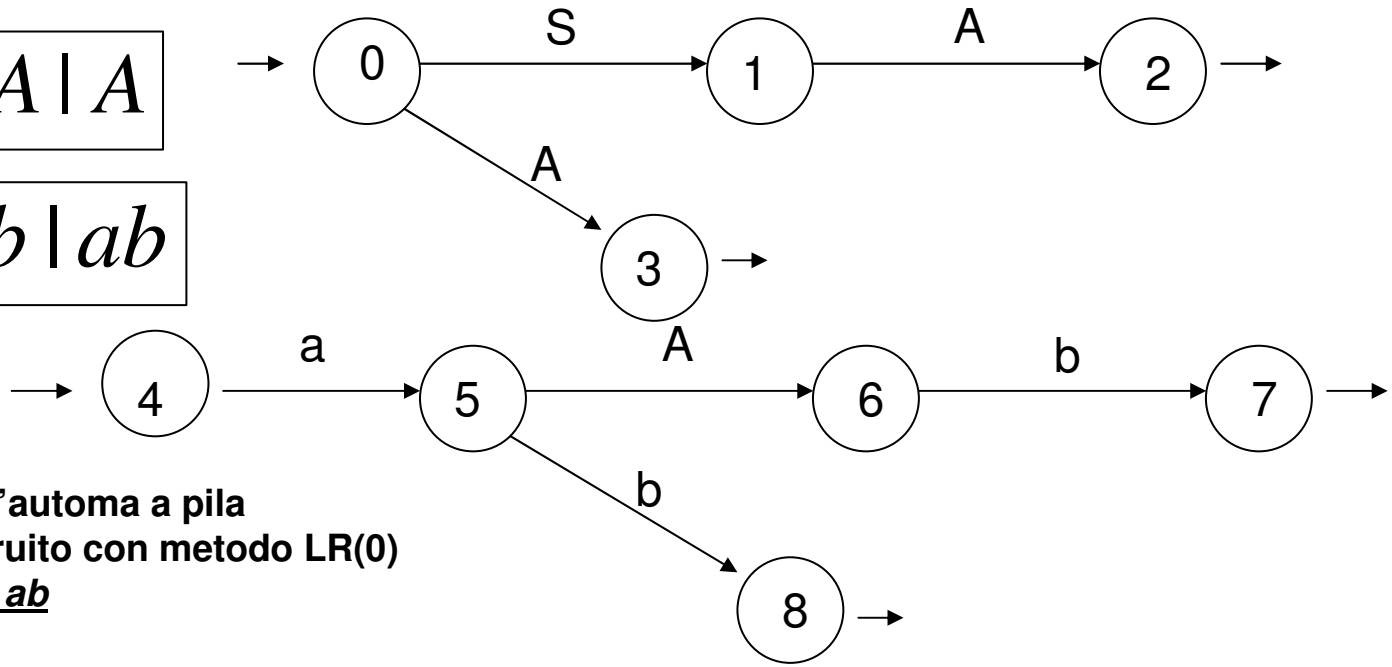
Il processo continua con altre eventuali azioni di spostamento, fino ad ottenere la prossima parte riducibile ed effettuare la riduzione e così via ...

Terminazione:
- impossibile procedere **testo rifiutato,** o
- scandito tutto il testo è stato, sulla pila solo assioma: **testo accettato**

ESEMPIO – Linguaggio: lista non vuota di costrutti $a^n b^n$, $n \geq 1$. La condizione LL(k) fallisce in 0 (la grammatica è ricorsiva a sinistra e sia A che S iniziano per *a*

$$S \rightarrow SA \mid A$$

$$A \rightarrow aAb \mid ab$$



Funzionamento dell'automa a pila deterministico costruito con metodo LR(0)
<u>analisi della stringa *ab*</u>

- stato iniziale I_0={0,4} in pila
- δ(0,a)Uδ(4,a)=ØU{5}={5}
- da 5 si può invocare A quindi si aggiunge lo stato iniziale 4 → I_2={4,5} in pila
- δ(4,b)Uδ(5,b)=ØU{8}={8} → I_3={8} in pila
- stato 8 finale, applicata la regola A→ab riducendo la stringa *ab* nel nonterminale A
- riduzione cancella dalla pila I_2I_3 (un numero di elementi pari a lunghezza della parte destra, */ab/=2,* della regola riconosciuta nello stato 8)
- ora pila contiene I_0={0,4} compiendo la mossa A (perchè A e' stata riconosciuta) impila I_4
  dove I_4={3} = δ(4,A)Uδ(0,A) =ØU{3}={3} ….

Essendo 3 uno stato finale della macchina dell'assioma S, la stringa è accettata….

ANALISI SINTATTICA LR(0) - lunghezza della prospezione = 0: la scelta dell'arco da percorrere è unicamente determinata dallo stato in cui si trova l'automa a pila.

Si costruisce un automa finito: la MACCHINA PILOTA DELL'ANALISI LR(0) che piloterà il parsificatore consentendo il riconoscimento dei prefissi ascendenti validi.

Si considera la rete di macchine anche contemporaneamente attive a causa delle indecisioni incontrate. Tutte le macchine attive contemporaneamente usano la stessa pila

L'automa pilota ha MACROSTATI, insiemi di stati delle macchine di partenza

1) Uno stato finale appartenente a un macrostato è detto STATO DI RIDUZIONE.
2) Uno stato non finale in un macrostato STATO DI SPOSTAMENTO
3) Un macrostato è detto RIDUZIONE se contiene ***soltanto*** stati di riduzione.
4) Un macrostato è detto SPOSTAMENTO se contiene ***soltanto*** stati di spostamento.
5) Un macrostato è detto MISTO, altrimenti.

CONDIZIONE LR(0)
Una grammatica soddisfa la condizione LR(0) se ogni macrostato della macchina verifica le seguenti condizioni:

1) non è misto

e

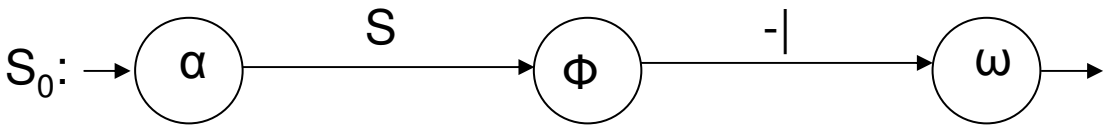
2) non contiene due o più stati di riduzione.

LINGUAGGI LR(0): generati da grammatiche LR(0)

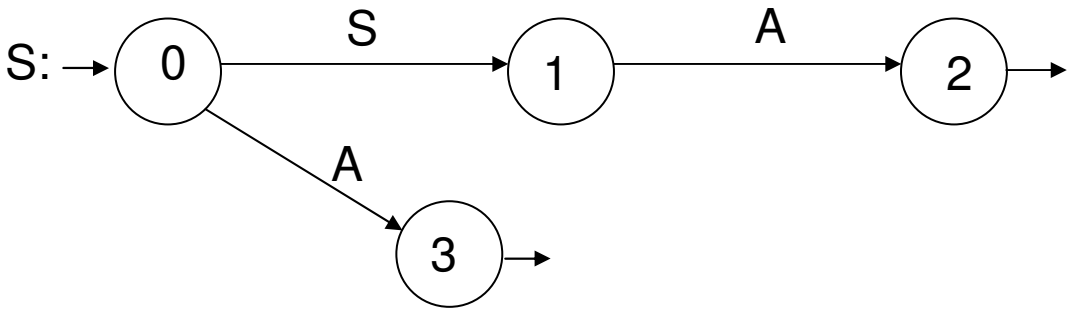
Se una grammatica G è LR(0), dalla macchina pilota si ottiene subito l'automa a pila deterministico che riconosce le frasi di L(G) e ne costruisce gli alberi sintattici.

ESEMPIO – Costruzione della macchina pilota – riconoscitore dei prefissi ascendenti validi – Il linguaggio è quello delle liste non vuote di costrutti $a^n b^n$, $n \geq 1$
NB: Introdotto "nuovo" assioma S_0, S degradato a comune N.T.

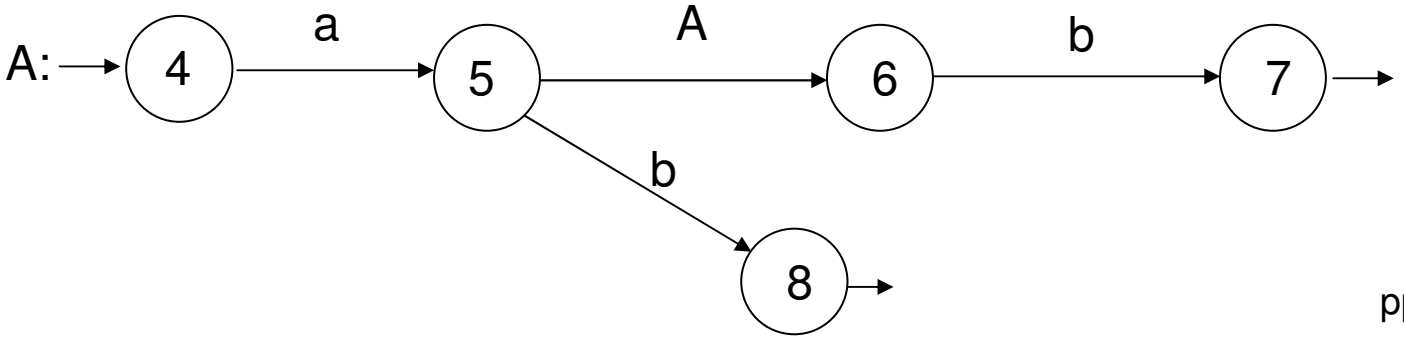
$$S_0 \rightarrow S\text{-|}$$



$$S \rightarrow SA \mid A$$



$$A \rightarrow aAb \mid ab$$



Costruzione dell'automa pilota: descrizione (quasi) formale

Definizione preliminare:
Chiusura LR(0) chius(q) di uno stato q

C := {q};
repeat
  C := C ∪ {$q_{A,0}$ | ∃p∈ C, ∃A∈ V per i quali è definita la mossa δ(p, A)) }
until nessun nuovo elemento trovato (punto fisso)
chius(q):=C

Per un insieme di stati R:
chius(R) è l'insieme delle chiusure di degli stati di R $= \bigcup_{r \in R} chius(r)$

Auoma pilota N=(R, Σ∪V, θ, I_0, R)
Calcolo dell'insieme degli stati R={I_0,…} e della funzione di transizione q

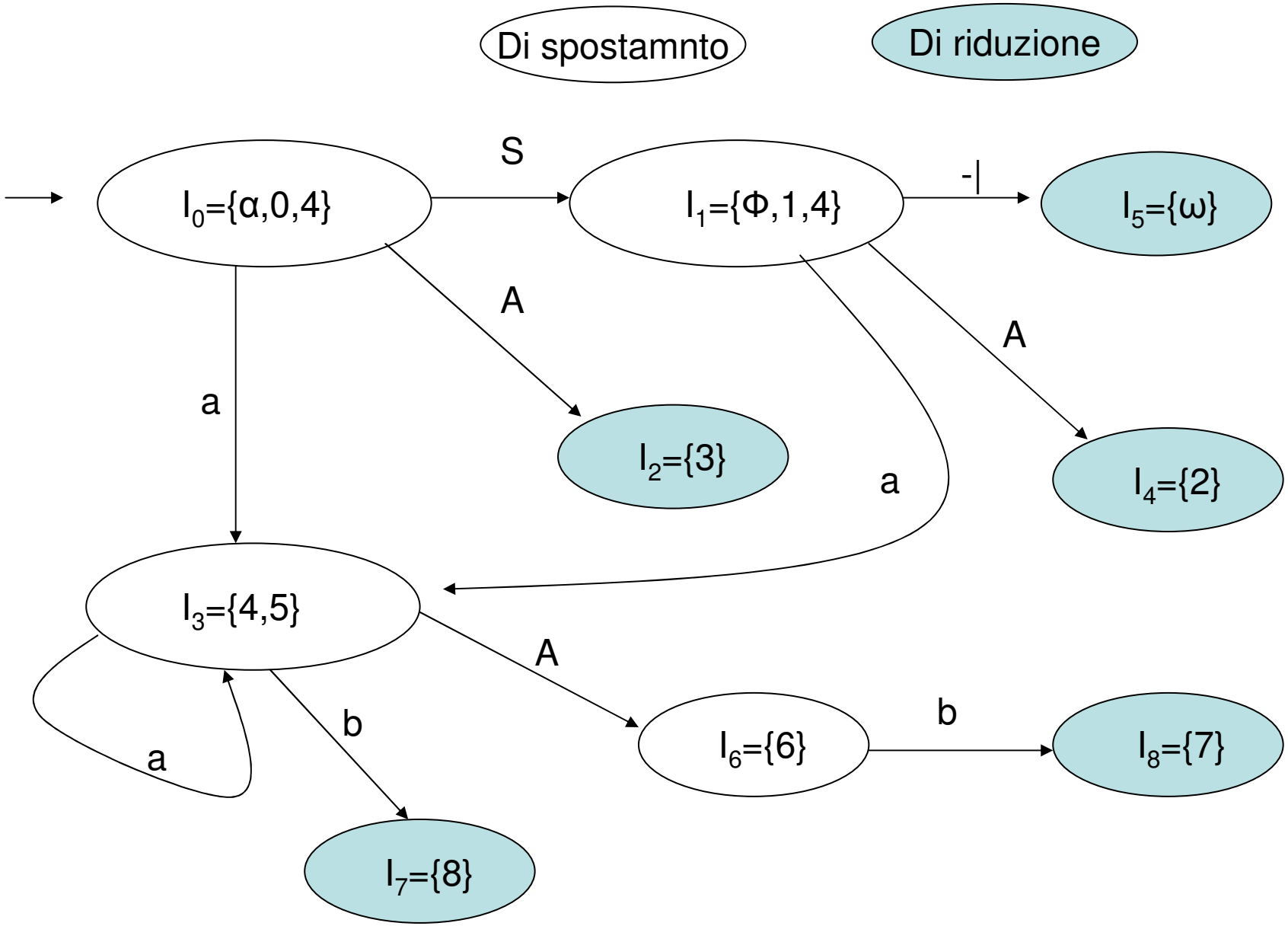
I_0 := chius(α);
R := I_0;
repeat per ogni I∈ R e per ogni X∈ Σ∪V
  θ(I, X) := $\bigcup_{q \in I} chius(\delta(q, X))$
  R := R ∪ θ(I, X);
until raggiunto un punto fisso

Calcolo dei macrostati e delle mosse

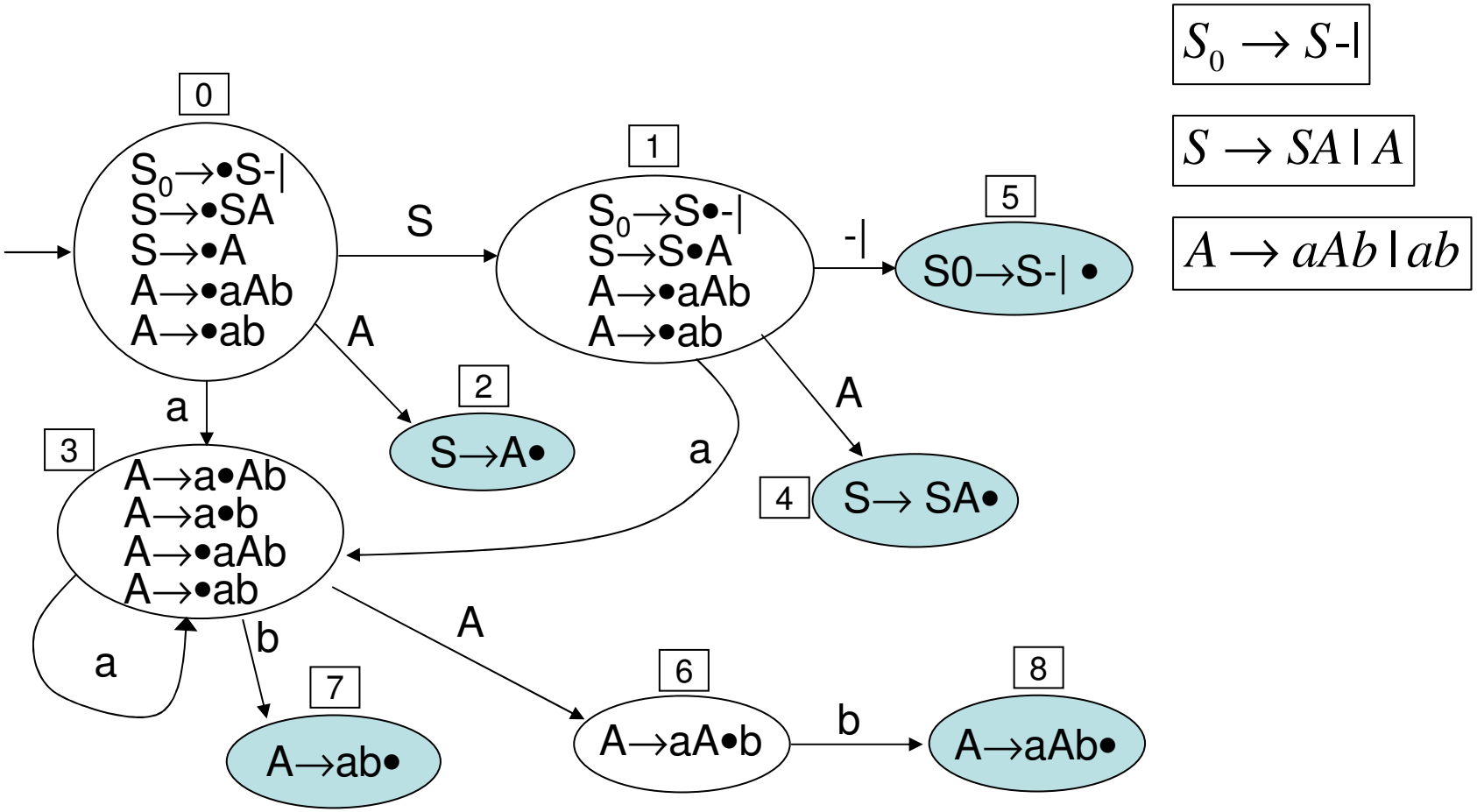
Insieme dei macrostati	Macrostato creato	Mossa creata
0. $\varnothing$	$\text{chius}(\alpha)=\{\alpha,0,4\}=I_0$	
1. $I_0=\{\alpha,0,4\}$	$\text{chius}(\delta(\alpha,S)\cup\delta(0,S)\cup\delta(4,S))=$ $\text{chius}(\{\phi,1\})=\{\phi,1,4\}=I_1$	$\vartheta(I_0,S)=I_1$
2. $I_0=\{\alpha,0,4\},I_1$	$\text{chius}(\delta(\alpha,A)\cup\delta(0,A)\cup\delta(4,A))=$ $\text{chius}(3)=\{3\}=I_2$	$\vartheta(I_0,A)=I_2$
3. $I_0=\{\alpha,0,4\},I_1,I_2$	$\text{chius}(\delta(\alpha,a)\cup\delta(0,a)\cup\delta(4,a))=$ $\text{chius}(5)=\{4,5\}=I_3$	$\vartheta(I_0,a)=I_3$
4. $I_0,I_1=\{\phi,1,4\},I_2,I_3$	$\text{chius}(\delta(\phi,A)\cup\delta(1,A)\cup\delta(4,A))=$ $\text{chius}(2)=\{2\}=I_4$	$\vartheta(I_1,A)=I_4$
5. $I_0,I_1=\{\phi,1,4\},I_2,I_3,I_4$	$\text{chius}(\delta(\phi,a)\cup\delta(1,a)\cup\delta(4,a))=$ $\text{chius}(5)=\{4,5\}=I_3$	$\vartheta(I_1,a)=I_3$

Calcolo dei macrostati e delle mosse (continua)

Insieme dei macrostati	Macrostato creato	Mossa creata
6. $I_0, I_1 = \{\phi, 1, 4\}, I_2, I_3, I_4$	$\text{chius}(\delta(\phi, \text{-l}) \cup \delta(1, \text{-l}) \cup \delta(4, \text{-l})) =$ $\text{chius}(\omega) = \{\omega\} = I_5$	$\vartheta(I_1, \text{-l}) = I_5$
7. $I_0, I_1, I_2, I_3 = \{4,5\}, I_4, I_5$	$\text{chius}(\delta(4, A) \cup \delta(5, A)) =$ $\text{chius}(6) = \{6\} = I_6$	$\vartheta(I_3, A) = I_6$
8. $I_0, I_1, I_2, I_3 = \{4,5\}, I_4, I_5, I_6$	$\text{chius}(\delta(4, a) \cup \delta(5, a)) =$ $\text{chius}(5) = \{4,5\} = I_3$	$\vartheta(I_3, a) = I_3$
9. $I_0, I_1, I_2, I_3 = \{4,5\}, I_4, I_5, I_6$	$\text{chius}(\delta(4, b) \cup \delta(5, b)) =$ $\text{chius}(8) = \{8\} = I_7$	$\vartheta(I_3, b) = I_8$
10. $I_0, I_1, I_2, I_3, I_4, I_5, I_6 = \{6\}, I_7$	$\text{chius}(\delta(6, b)) =$ $\text{chius}(7) = \{7\} = I_8$	$\vartheta(I_6, b) = I_8$
11. $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8$		alt


Di spostamnto
Di riduzione
S
-|
I_0={α,0,4}
I_1={Φ,1,4}
I_5={ω}
A
A
a
I_2={3}
a
I_4={2}
I_3={4,5}
A
b
b
a
I_6={6}
I_8={7}
I_7={8}

Utile presentazione alternativa dell'automa pilota:
macrostati contengono regole marcate:
rappresentazione meno compatta ma leggibile e autocontenuta



PROPRIETÀ DELLA MACCHINA PILOTA:

1. Tutte le frecce entranti in un macrostato portano la stessa etichetta.
2. Ogni macrostato di riduzione è privo di successori.
3. Ogni macrostato di spostamento ha almeno un successore.

Qual è il linguaggio riconosciuto dalla macchina pilota (supponendo che ogni macrostato sia finale)? Nell'esempio le stringhe riconosciute sono:

S, A, a, S-|,SA, aA, aa, ab,a Ab, aaa, aab, aaAb ,...

prefissi delle forme di frasi generate dalla grammatica G, mediante derivazione destra

$$S_0 \Rightarrow S\text{-}| \Rightarrow A\text{-}| \Rightarrow aAb\text{-}| \Rightarrow aaAbb\text{-}| \Rightarrow aaabbb\text{-}|$$

presenta i prefissi accettati dal pilota: *S, S-|, A, A-|,aA,aAb,aaAb,aaab*.
Ma non tutti i prefissi delle forme destre sono accettati dal pilota: per esempio *aaabb* non lo è. La stringa riconosciuta non puo` contenere sottostringhe ***interne*** che siano parti destre di una produzione (queste vengono subito ridotte alla parte sinistra)

ANALIZZATORE A SPOSTAMENTO E RIDUZIONE

Se una grammatica G è LR(0), dalla macchina pilota si ottiene facilmente l'automa a pila deterministico che riconosce le frasi di L(G) e ne costruisce gli alberi sintattici.

L'analizzatore opera nel seguente modo:

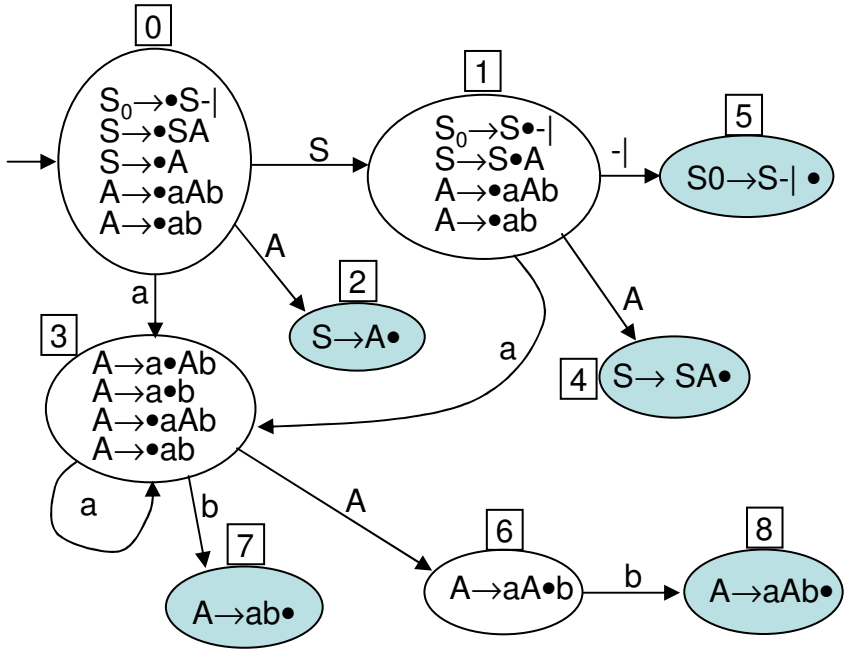
1) I simboli della pila sono i macrostati, ai quali si possono affiancare per migliorare la leggibilità anche i simboli della grammatica scanditi per raggiungerli
2) Esaminando il primo carattere, l'automa avendo in cima alla pila il macrostato I, esegue la mossa prescritta dall'automa pilota e impila il prossimo macrostato (operazione di spostamento)
3) Se il nuovo stato ha associata una riduzione (che per l'ipotesi LR(0) è unica) esegue una serie di operazioni note come riduzione. Esse simulano l'omonimo passo nella derivazione di una frase, prima cancellando poi inserendo i simboli sulla pila (operazione di riduzione)
4) Al termine della scansione, l'automa accetta la stringa sorgente se la pila è vuota o contiene il solo macrostato iniziale (operazione di accettazione/rifiuto)

Analisi della stringa ‘a a b b a b’

$$S_0 \rightarrow S\text{-}|$$

$$S \rightarrow SA \mid A$$

$$A \rightarrow aAb \mid ab$$



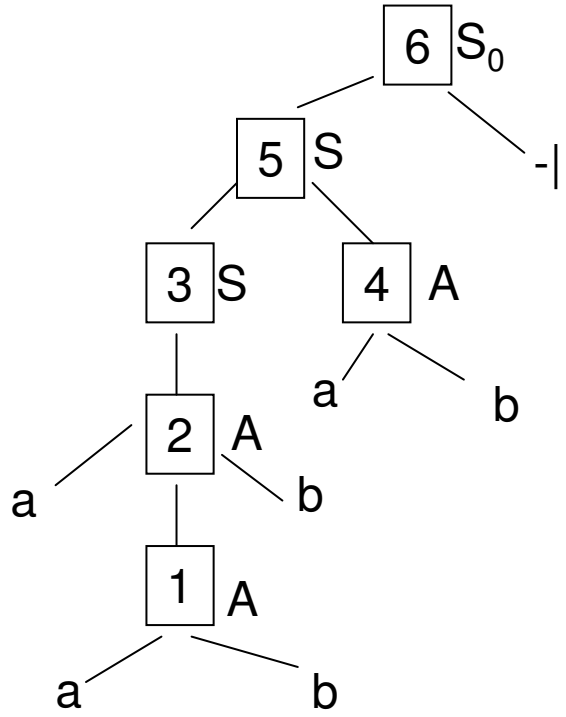
Pila	x	Commento
I_0	a a b b a b -|	sposta
$I_0\ aI_3$	a b b a b -|	sposta
$I_0\ aI_3\ aI_3$	b b a b -|	sposta
$I_0\ aI_3\ aI_3\ bI_7$	b a b -|	riduci con $A \rightarrow ab$
$I_0\ aI_3\ AI_6$	b a b -|	sposta
$I_0\ aI_3\ AI_6\ bI_8$	a b -|	riduci con $A \rightarrow aAb$
$I_0\ AI_2$	a b -|	riduci con $S \rightarrow A$
$I_0\ SI_1$	a b -|	sposta
$I_0\ SI_1\ aI_3$	b -|	sposta
$I_0\ SI_1\ aI_3\ bI_7$	-|	riduci con $A \rightarrow ab$
$I_0\ SI_1\ AI_4$	-|	riduci con S $\rightarrow SA$
$I_0\ SI_1$	-|	sposta
$I_0\ SI_1\ \text{-}\|I_5$		riduci con $S_0 \rightarrow S$-|
$I_0\ S_0$		accetta

$A \to ab,\ A \to aAb,\ S \to A,\ A \to ab,\ S \to SA,\ S_0 \to S\text{-}|$ — mosse di riduzione applicate

$aabbab\text{-}| \Leftarrow aAbab\text{-}| \Leftarrow Aab\text{-}| \Leftarrow Sab\text{-}| \Leftarrow SA\text{-}| \Leftarrow S\text{-}| \Leftarrow S_0$ — derivazione corrispondente (destra, invertita)

Si tratta di un parsificatore ascendente. Invertendo specularmente l'ordine della riduzione, si ottiene la derivazione destra:

$$S_0 \overset{+}{\Rightarrow} aabbab\text{-}|$$

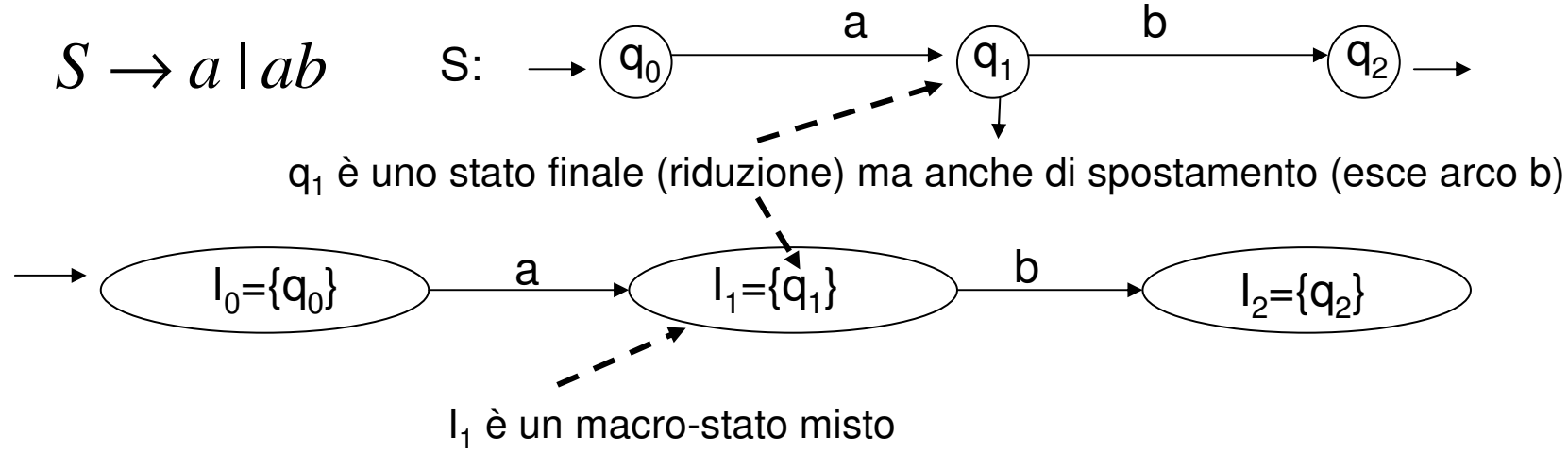


Limitazione SERIA dei linguaggi LR(0): se una stringa appartiene a linguaggio non può appartenervi nessun suo prefisso

Cioè: i linguaggi LR(0) sono ***privi di prefissi***

ESEMPIO: L = {a,ab}

$S \rightarrow a \mid ab$

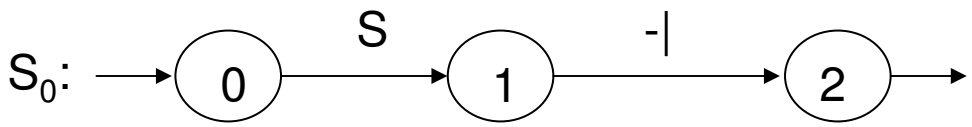


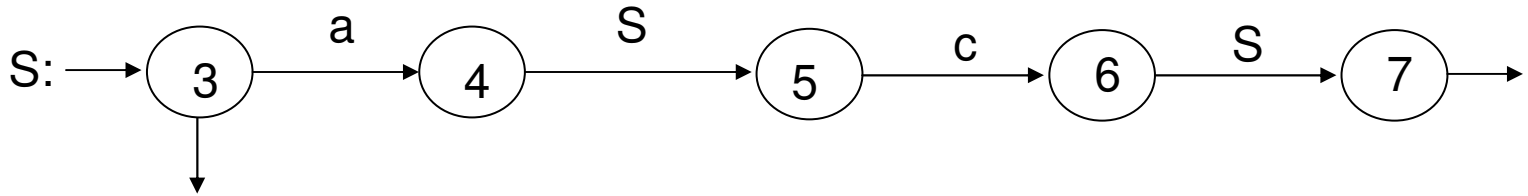
<u>Una grammatica contenente regole vuote viola la condizione LR(0).</u> Infatti una regola A→ ε | β fa sì che lo stato iniziale $q_{A,0}$ della macchina A sia anche finale. Ma da tale stato esce anche l'arco associato alla parte destra (non vuota) β. Di conseguenza, il macrostato contenente lo stato $q_{A,0}$ viola la condizione LR(0)

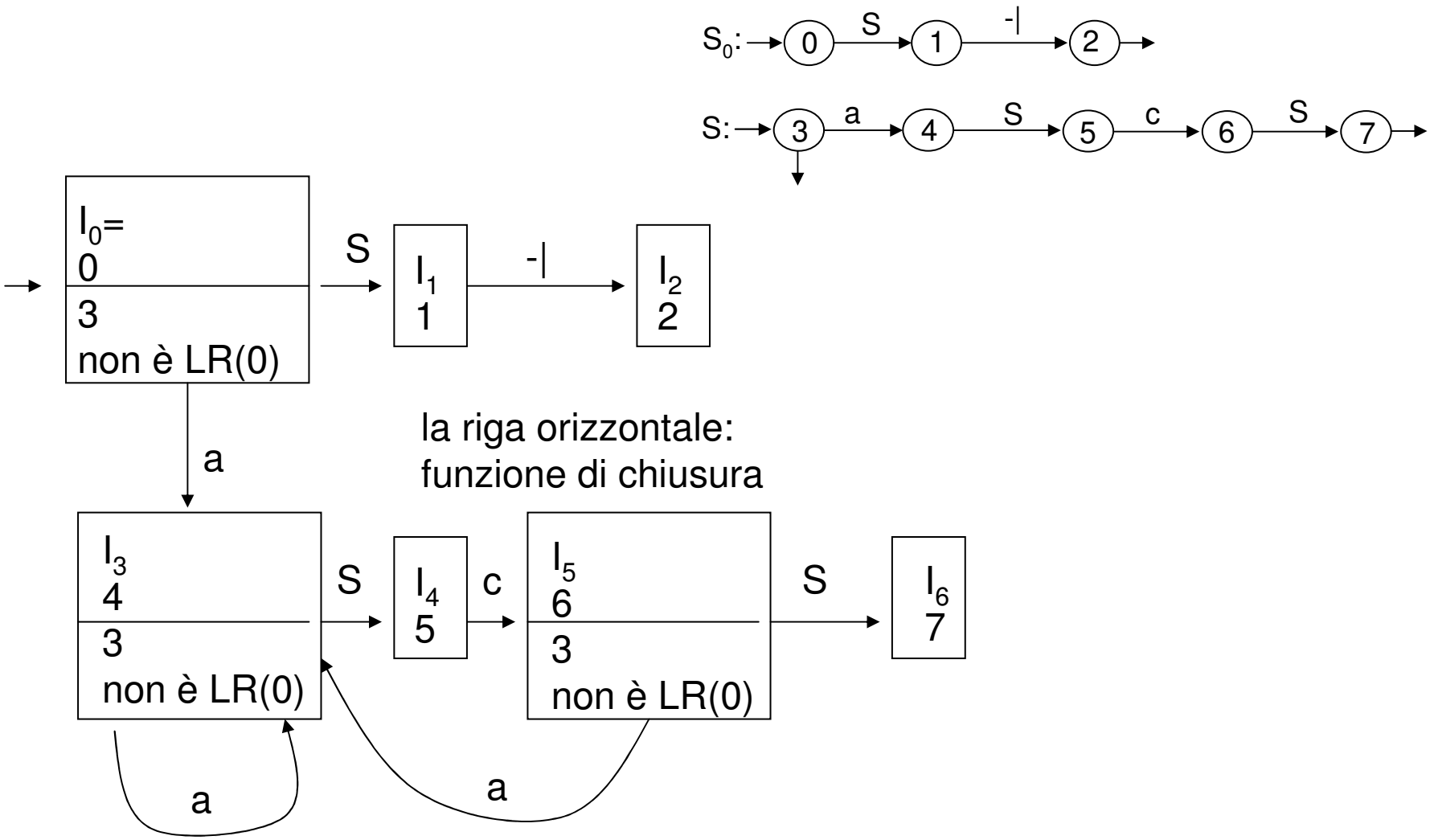
ESEMPIO - La grammatica del linguaggio di Dyck non è LR(0)

$S_0 \rightarrow S\text{-|}$

$S \rightarrow aScS \mid \varepsilon$







Tutti i macrostati contenenti lo stato 3 sono misti e inadeguati per LR(0)
Quando è in questi macrostati l'analizzatore non sa se spostare o ridurre

Aggiunta della prospezione alla macchina pilota

Per potenziare il metodo LR(0) trasformandolo nel metodo LR(1), si aggiunge ad ogni stato l'informazione sulla prospezione.

Poiché ogni macrostato contiene in generale più stati, è necessario calcolare l'insieme dei caratteri terminali di prospezione per ciascuno stato

Intuitivamente, un macrostato risulta adeguato per l'analisi LR(1) se la prossima mossa dell'automa a pila è univocamente determinata dal macrostato posto in cima alla pila e dal confronto del carattere corrente con gli insiemi di prospezione

La macchina pilota LR(1) è costruita in modo simile al caso LR(0) ma ogni macrostato I della macchina pilota LR(1) è un insieme di coppie, dette *candidate*, di forma: $<q,a> \in Q \times (\Sigma \cup \{\dashv\})$ dove Q è l'insieme degli stati della rete di macchine.

Più candidate aventi in comune lo stesso stato saranno per brevità raggruppate.

$$<q, \{a_1,a_2, \ldots a_k\}> \equiv \{<q,a_1>,<q,a_2>, \ldots <q,a_k>\}$$

L'insieme $\{a_1,a_2, \ldots a_k\}$ è detto insieme di prospezione dello stato *q*

domanda cruciale:

qual è il significato di un carattere di prospezione?

NB: il carattere di prospezione è associato a uno stato (in un macrostato)

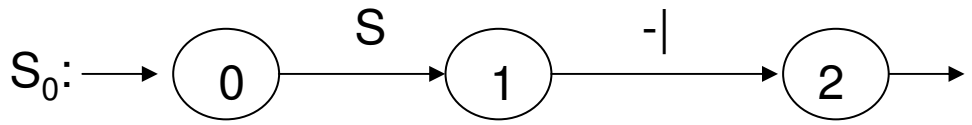
Uno stato corrisponde a una parte destra di regola “marcata” con il pallino

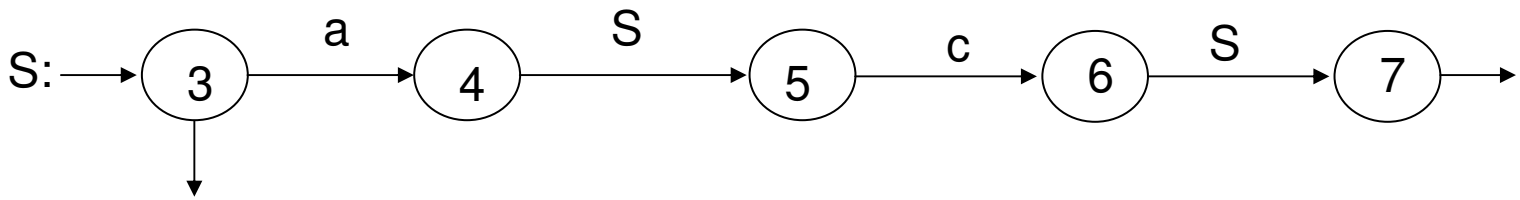
Un carattere di prospezione indica il simbolo che ci sarà in ingresso
quando verrà eseguita la riduzione della parte destra della regola nel n.t. parte sinistra
cioè quando il “pallino” che marca la parte destra della regola sarà arrivato alla fine

ESEMPIO - La grammatica del linguaggio di Dyck è LR(1) – continua

$S_0 \rightarrow S\text{-|}$

$S \rightarrow aScS \mid \varepsilon$





Costruzione dell'automa pilota LR(1): descrizione (quasi) formale (1/2)

Definizione preliminare:
Chiusura LR(1) $chius_1(c)$ di una ***candidata*** $c=<q, \pi>$

C := {<q, π>};
repeat
  C := C ∪ {$<q_{A,0}$, ρ>}
    dove ∃<p,π>∈ C, ∃A∈ V per i quali è definita la mossa δ(p, A)=p'
    e la prospezione ρ è calcolata come
      - ρ = Ini(L(p')) se L(p') non è annullabile, oppure
      - ρ = Ini(L(p')) ∪ π altrimenti

until nessun nuovo elemento trovato (punto fisso);
$chius_1(c)$:= C;

Per un insieme di candidate I:
$chius_1(I)$ è l'insieme delle chiusure delle candidate di $I = \bigcup_{c \in I} chius_1(c)$

Costruzione dell'automa pilota LR(1): descrizione (quasi) formale (2/2)

Automa pilota N=(R, $\Sigma \cup V$, θ, I_0, R)

NB: come in LR(0) ma ora macrostati composti da candidate

Calcolo dell'insieme degli stati R={I_0,…} e della funzione di transizione θ

I_0 := $chius_1(<q_0, \dashv|>)$;

R := I_0;

repeat per ogni $I \in R$ e per ogni $X \in \Sigma \cup V$

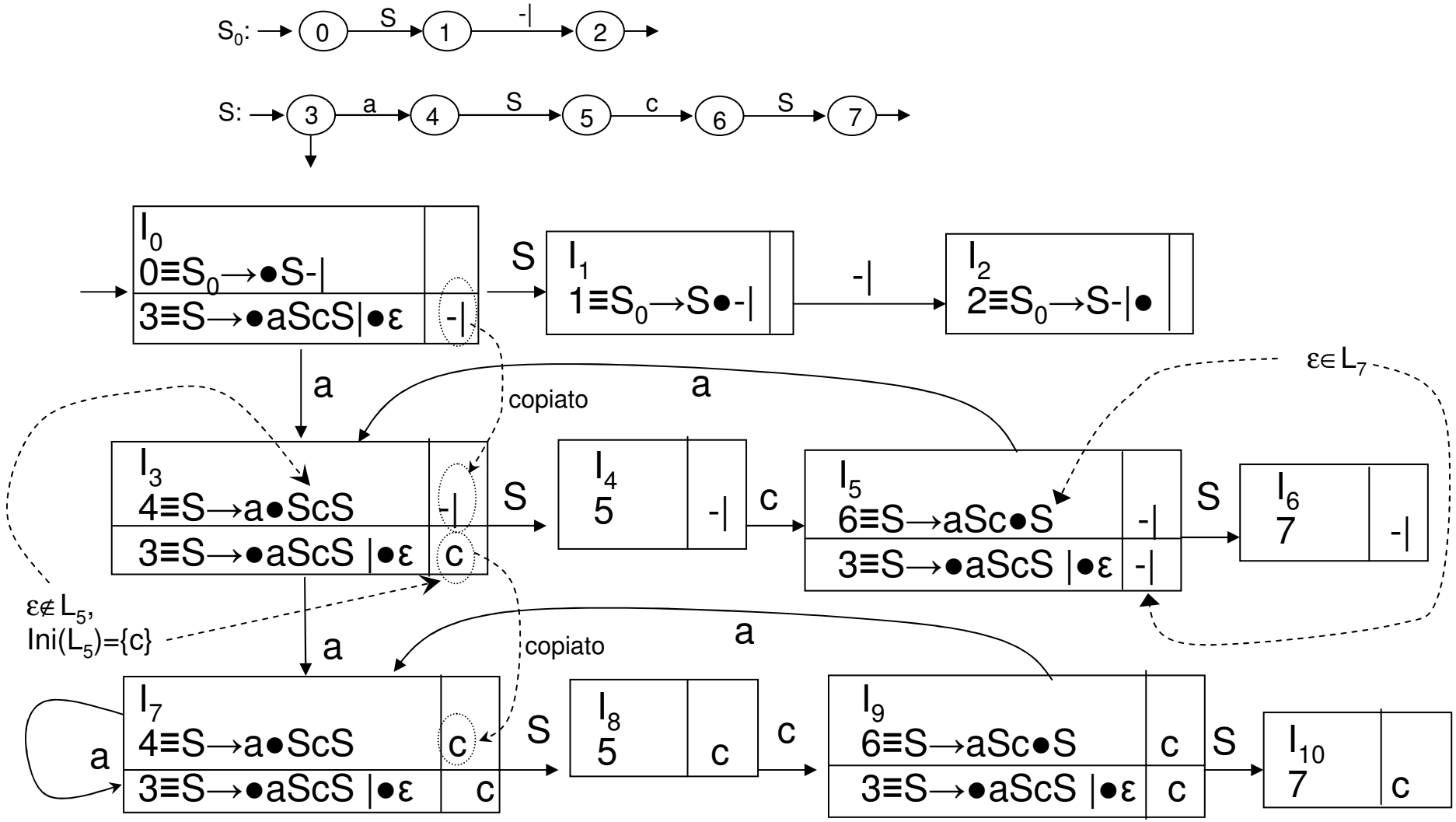
$\theta(I, X) := \bigcup_{<q,\rho>\in I} chius_1(<\delta(q, X), \rho>)$;

R := R $\cup$ θ(I, X);

until raggiunto un punto fisso

Esiste nella macchina l'arco uscente da q, etichettato con X

NB: l'applicazione della δ "si porta dietro" la stessa prospezione
Eventuali nuovi elementi della prospezione aggiunti, ad altre candidate, da $chius_1$



NB: caso particolare, candidate nei macrostati (I_0, I_1 e I_2) della regola $S_0 \rightarrow$S-| non hanno prospezione: è una regola convenzionale, S_0 non compare in alcun'altra regola, non ha senso parlare di prospezione

Nel seguito spesso lasciamo implicita questa regola, uguale per ogni grammatica,

CONDIZIONE LR(1)

Vale ancora la classificazione dei macrostati del caso k=0:

1) Una candidata è detta di RIDUZIONE se lo stato è finale in una macchina della rete (regola grammaticale completata, “pallino” in fondo)
2) Una candidata è detta di spostamento se dallo stato ad esso associato, in una macchina esce un arco etichettato con un simbolo terminale o non (regola non completata, “pallino” interno).

Ma la condizione LR(1) è meno drastica di quella LR(0):

CONDIZIONE LR(1) –per ogni macrostato della macchina pilota valgono entrambe le seguenti condizioni:
1) assenza di conflitto riduzione-spostamento: ogni candidata di riduzione ha un insieme di prospezione disgiunto dall’insieme dei terminali che etichettano gli archi uscenti dal macrostato
e
2) assenza di conflitto riduzione-riduzione: se vi sono due candidate di riduzione, i loro insiemi di prospezione sono disgiunti.

NB: la prospezione usata solo per la riduzione

ESEMPIO PRECEDENTE – Verifica della condizione LR(1)

I macrostati I_1, I_2, I_4,I_6,I_8,I_{10} hanno una sola candidata quindi soddisfano banalmente la condizione

Gli altri macrostati posseggono la candidata di riduzione S→●ε (equivalente a S →ε● dato che la parte destra è vuota), la candidata di spostamento S →●aScS – entrambe associate allo stato 3 – e un'altra candidata di spostamento.

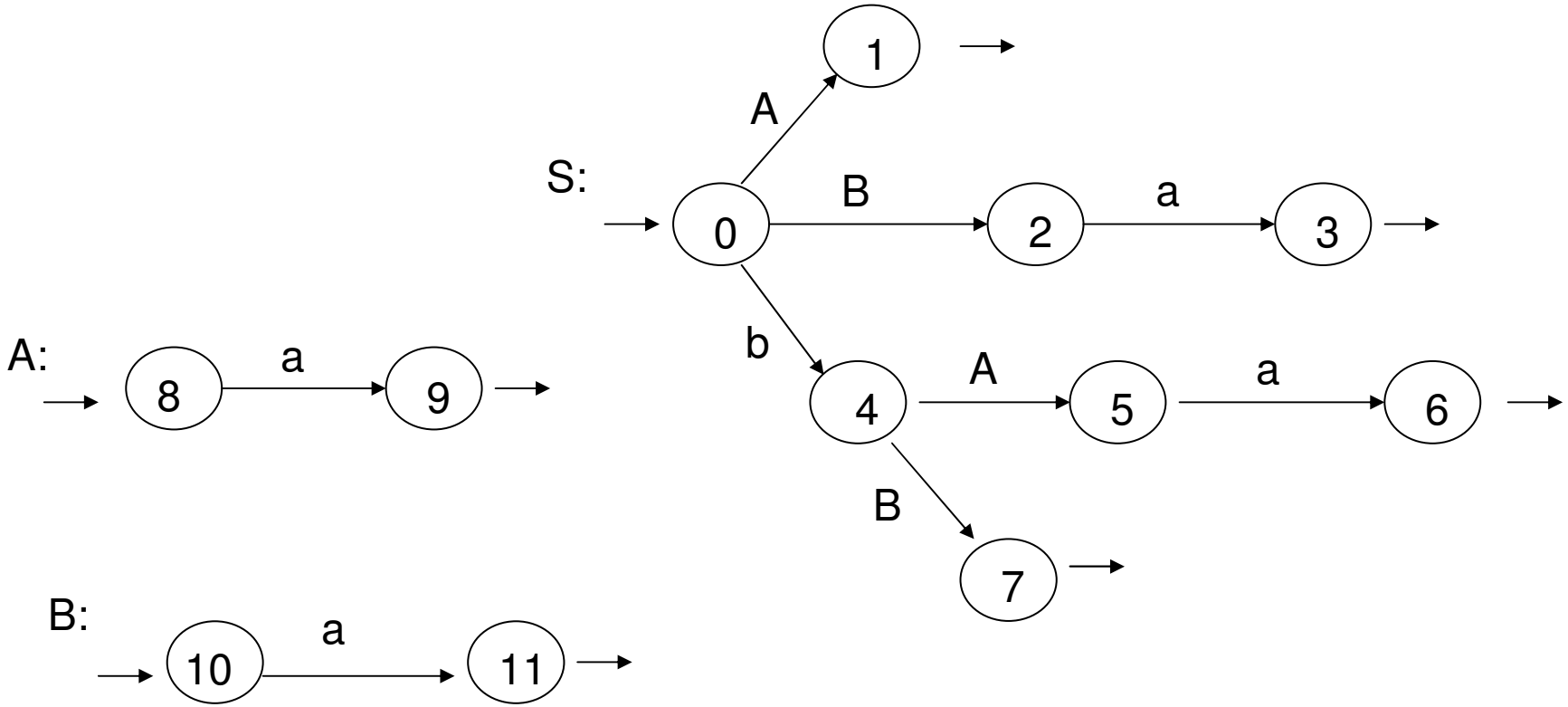
Verifica di disgiunzione:

macrostato	prospezione	etichette uscenti	condizione
I_0	-|	a	$\{\text{-|}\} \cap \{a\} = \varnothing$
I_3	c	a	$\{c\} \cap \{a\} = \varnothing$
I_5	-|	a	$\{\text{-|}\} \cap \{a\} = \varnothing$
I_7	c	a	$\{c\} \cap \{a\} = \varnothing$
I_9	c	a	$\{c\} \cap \{a\} = \varnothing$

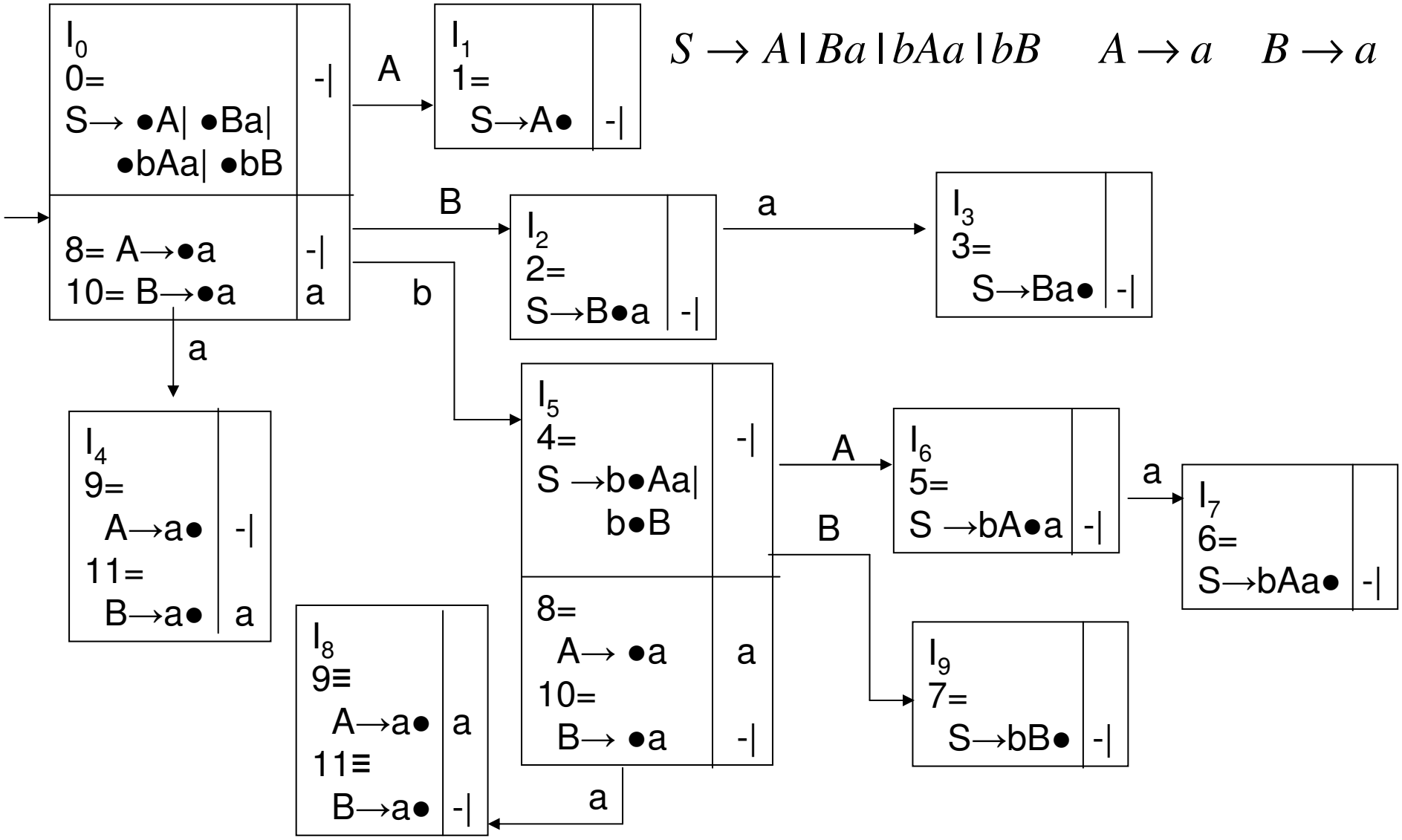
La grammatica risulta LR(1)

ESEMPIO - verifica di LR(1) in presenza di due riduzioni nel macrostato
(punto 2 della condizione LR(1))

$$S \to A \mid Ba \mid bAa \mid bB \qquad A \to a \qquad B \to a$$



Macchina pilota LR(1) con evidenziate le riduzioni associate ai macrostati



La grammatica dell'esempio è LR(1) ma non LR(0) perché in I_4 e I_8 vi sono due riduzioni, tra le quali solo la prospezione permette di scegliere.

I macrostati che potrebbero violare LR(1) sono I_4 e I_8. Tutti gli altri soddisfano già la condizione LR(0).

Da nessuno dei due escono archi, quindi la parte 1 della condizione è manifestamente soddisfatta.

I_4 e I_8 contengono due candidate di riduzione, ma gli insiemi di prospezione sono disgiunti, quindi punto (2) della condizione LR(1) è soddisfatta.

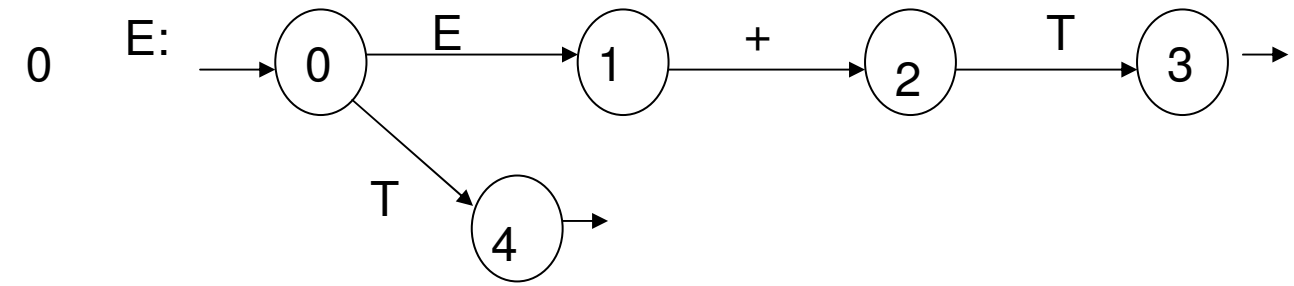
Intuitivamente, nello stato I_4:
- se letta una 'a' e poi c'è (prospezione) un '-|' allora la derivazione è $S \Rightarrow A \Rightarrow a$
- se invece prospezione è una 'a' allora la derivazione è $S \Rightarrow Ba \Rightarrow aa$

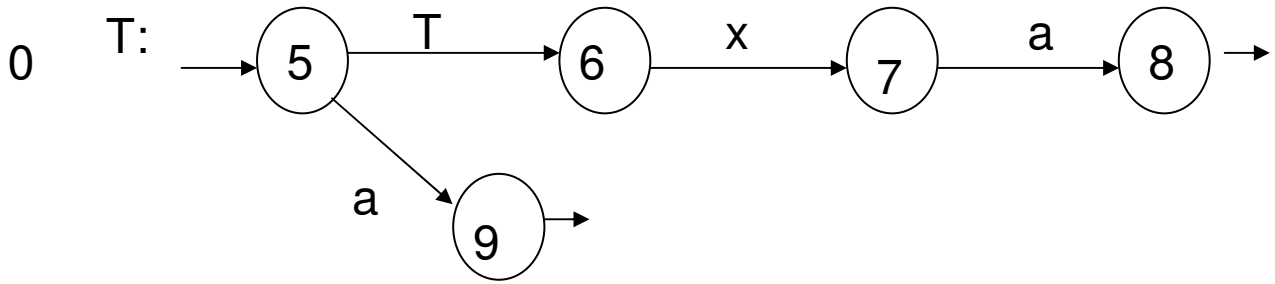
Similmente, nello stato I_8:
- se letta una 'b' e poi una 'a' e prospezione è 'a' allora derivazione $S \Rightarrow bAa \Rightarrow baa$
- se invece prospezione è '-|' allora la derivazione è $S \Rightarrow bB \Rightarrow ba$

ESEMPIO – Espressioni aritmetiche – condizione LR(1)
Ruolo degli insiemi prospezione

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T \times a \mid a$$





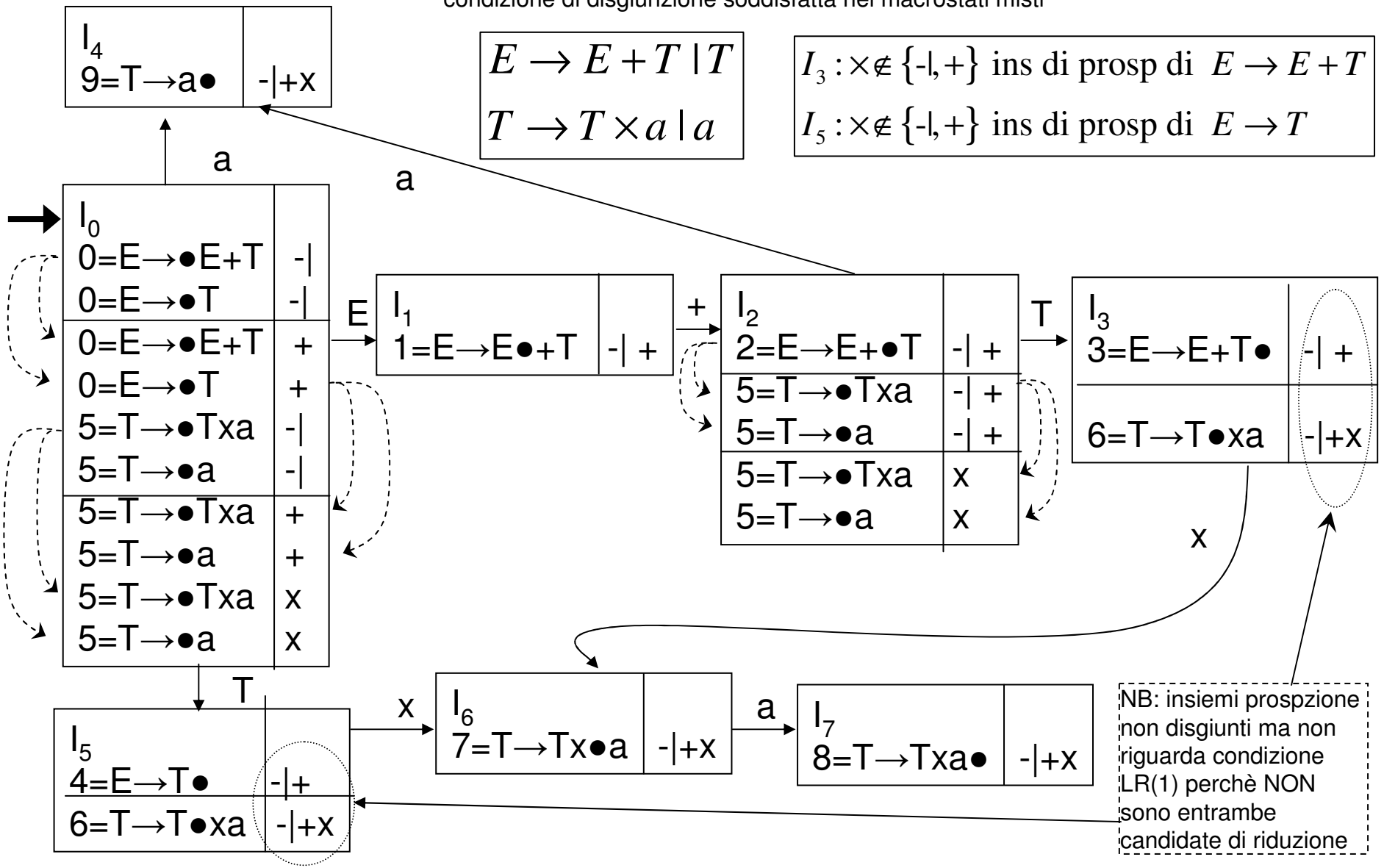
Macchina pilota LR(1)

grammatica LR(1).
nessun macrostato contiene due stati finali
condizione di disgiunzione soddisfatta nei macrostati misti

$$E \to E + T \mid T$$
$$T \to T \times a \mid a$$

$I_3 : \times \notin \{\text{-|},+\}$ ins di prosp di $E \to E + T$

$I_5 : \times \notin \{\text{-|},+\}$ ins di prosp di $E \to T$



Condizione LALR(1) – condizione di determinismo intermedia tra LR(0) e LR(1)

Si semplifica il pilota LR(1) fondendo insieme i macrostati che sono indistinguibili nel pilota LR(0), ossia quelli le cui candidate differiscono solo nella seconda componente (la prospezione). Si preservano però le informazioni sulla prospezione.

Il grafo della macchina LALR(1) è isomorfo a quello del pilota LR(0), esso può però contenere macrostati misti o condizioni di riduzione plurime, a condizione che valga la condizione seguente, identica a quella del caso LR(1).

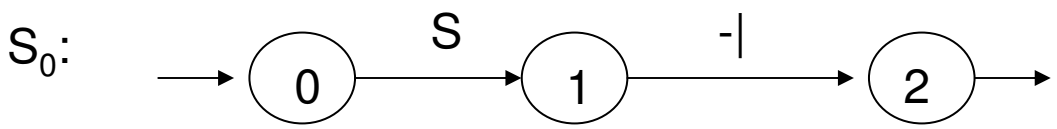
Una grammatica soddisfa la condizione LALR(1) se valgono entrambe le condizione:
1) ogni candidata di riduzione ha un insieme di prospezione disgiunto dall'insieme delle etichette terminali degli archi uscenti dal macrostato;
e
2) se vi sono due candidate di riduzione, i loro insiemi di prospezione sono disgiunti.

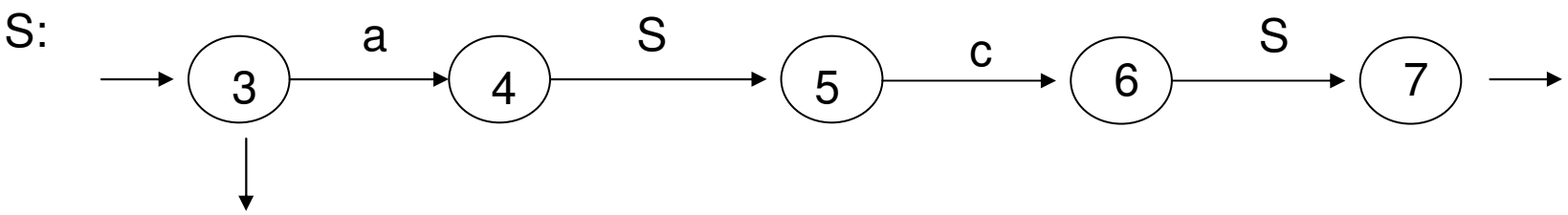
La famiglia delle grammatiche LALR(1) è inclusa in quella delle grammatiche LR(1) e include la famiglia LR(0).

ESEMPIO - Il linguaggio di Dyck come LALR(1)

$S_0 \rightarrow S\text{-|}$

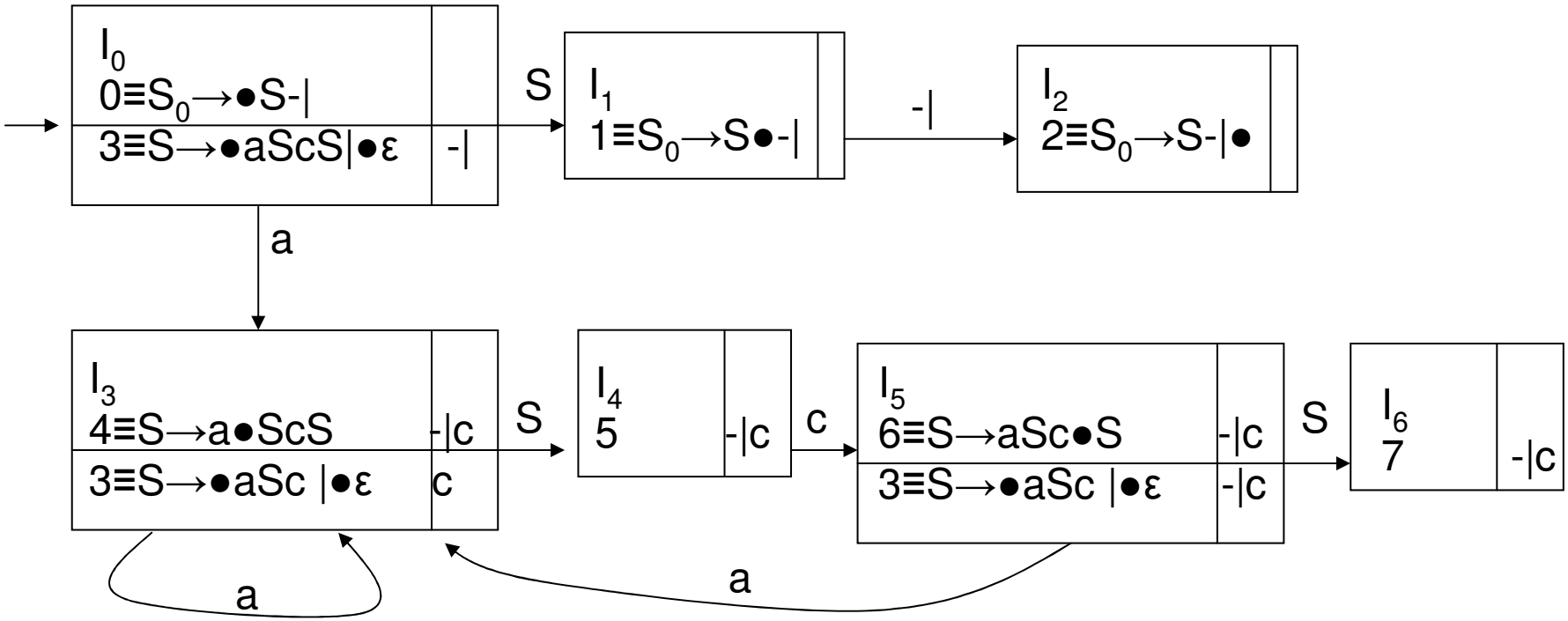
$S \rightarrow aScS \mid \varepsilon$





Automa pilota LALR(1) per linguaggio di Dyck

$$S_0 \rightarrow S\text{-|}$$
$$S \rightarrow aScS \mid \varepsilon$$



Nessun macrostato contiene più riduzioni. Nei macrostati misti, l'insieme di prospezione della riduzione S→ε è disgiunto dalle etichette degli archi uscenti. Questo pilota svolge, con un numero minore di stati, lo stesso compito del pilota LR(1)

Esempio slide n. **31-32** **(espr. aritmetiche)**: la macchina pilota LR(1) è già isomorfa a quella LR(0),poiché **NON** vi sono macrostati identici a meno degli insiemi di prospezione.La grammatica E→E+T|T T→Txa|a soddisfa la condizione LALR(1)

Esempio slide n. **28-29** – la grammatica è LR(1), non LALR(1). Fondendo i due macrostati I_4 e I_8, indistinguibili senza la prospezione, si ottiene il macrostato del pilota LALR(1):

I_4, I_8	
9=A→a●	-|a
11=B→a●	-|a

Poiché le due riduzioni hanno insiemi di prospezione identici, il macrostato non è LALR(1).

Nel progetto dei compilatori il metodo LALR(1) si è dimostrato spesso adeguato alla costruzione dei parsificatori, di solito al costo di piccole modifiche delle grammatiche di riferimento del linguaggio.

ESEMPIO – Traccia dell'analisi della stringa a+a.
Per espressioni aritmetiche con pilota LR(1) .... o LALR(1)

Pila	x	commento
I_0	$a + a$-|	sposta
$I_0\ aI_4$	$+ a$-|	riduci con $T \to a$
$I_0\ TI_5$	$+ a$-|	$+ \in$ prosp. di 4: riduci con $E \to T$
$I_0\ EI_1$	$+ a$-|	sposta
$I_0\ EI_1 + I_2$	a-|	sposta
$I_0\ EI_1 + I_2\ aI_4$	-|	riduci con $T \to a$
$I_0\ EI_1 + I_2\ TI_3$	-|	riduci con $E \to E + T$
$I_0\ E$	-|	accetta