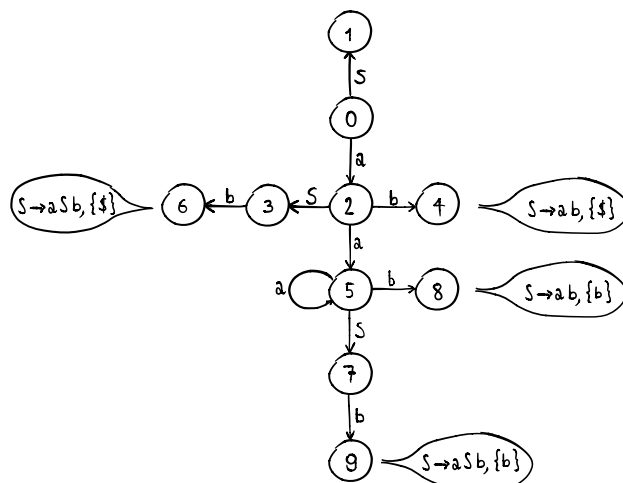


Vediamo la *tabella di parsing canonico LR(1)* per l'automa visto nella lezione precedente, per la grammatica $S \rightarrow aSb \mid ab$:

	a	b	\$	S
0	shift 2			1
1			accept	
2	shift 5	shift 4		3
3		shift 6		
4			reduce $S \rightarrow ab$	
5	shift 5	shift 8		7
6			reduce $S \rightarrow aSb$	
7		shift 9		
8		reduce $S \rightarrow ab$		
9		reduce $S \rightarrow aSb$		



Il parsing **shift-reduce** è una forma di parsing bottom-up.

Algorithm 1: Shift-Reduce Algorithm

input : bottom-up parsing table T for \mathcal{G} ; w
output : the rightmost derivation of w in reverse order if $w \in \mathcal{L}(\mathcal{G})$,
error() if $w \notin \mathcal{L}(\mathcal{G})$
data structures: *input_buffer* (init: $w\$$);
state_stack (init: initial state of the CA underlying T);
symbol_stack (init: empty)

```

VARIABLE  $b \leftarrow \text{get\_next\_char}(\text{input\_buffer});$ 
while true do
  if  $T[\text{top}(\text{state\_stack}), b] = \text{shift } P$  then
    push  $b$  onto the symbol_stack;
    push  $P$  onto the state_stack;
     $b \leftarrow \text{get\_next\_char}(\text{input\_buffer});$ 
  else
    if  $T[\text{top}(\text{state\_stack}), b] = \text{reduce } A \rightarrow \beta$  then
      pop  $|\beta|$  times from the symbol_stack;
      push  $A$  onto the symbol_stack;
      pop  $|\beta|$  times from the state_stack;
      push  $T[\text{top}(\text{state\_stack}), A]$  onto the state_stack;
      print(" $A \rightarrow \beta$ ");
    else
      if  $T[\text{top}(\text{state\_stack}), b] = \text{accept}$  then
        break;
      else
        break invoking error();

```

Esercizio 1

INPUT: $w = a_1 a_2 a_3 b_3 b_2 b_1$, precedente tabella di parsing T

DATA STRUCTURES: *input_buffer* = $w\$ = a_1 a_2 a_3 b_3 b_2 b_1 \$$, *state_stack* = $[0]$, *symbol_stack* = $[\]$

					a_2, a_3 diventa S		a_2, S, b_2 diventa S		a_1, S, b_1 diventa S	
					$5, 8$ diventa 7		$5, 7, 9$ diventa 3		$2, 3, 6$ diventa 1	
					" $S \rightarrow ab$ "		" $S \rightarrow aSb$ "		" $S \rightarrow aSb$ "	
VARIABLE	a_1	a_2	a_3	b_3	b_2	b_1	b_1	$\$$	$\$$	$T[1, \$] =$
symbol_stack		a_1	a_1, a_2	a_1, a_2, a_3	a_1, a_2, a_3, b_3	a_1, a_2, S	a_1, a_2, S, b_2	a_1, S	a_1, S, b_1	$ACCEPT$
state_stack	0	0,2	0,2,5	0,2,5,5	0,2,5,5,8	0,2,5,7	0,2,5,7,9	0,2,3	0,2,3,6	0,1

ACCEPT

Esercizio 2

INPUT: $w = a_1 a_1 a_2 b_2$, tabella T

DATA STRUCTURES: $input_buffer = w \$ = a_1 a_1 a_2 b_2 \$$, $state_stack = [0]$, $symbol_stack = []$

VARIABLE	a_1	b_1	a_2	
$symbol_stack$	a_1	a_1, b_1		$T[4, b] = \text{ERROR}$
$state_stack$	0	0, 2	0, 2, 4	

Anticipazione lezioni successive

Per creare la tabella di parsing LR(1) ci sono due passaggi:

- Determinare l'automa caratteristico.
- Determinare i lookahead set per le riduzioni.

Definizione di **LR(1) item**:

$A \rightarrow \alpha \bullet \beta$, Δ ($\Delta \subseteq T \cup \{\$ \}$)

Esempi di possibili LR(1) item:

$S \rightarrow \bullet aS$, $\{\$ \}$

$S \rightarrow \bullet aSb$, $\{\$ \}$

$S \rightarrow \bullet ab$, $\{\$ \}$

Chi sono gli item per $A \rightarrow \varepsilon$?

$A \rightarrow \bullet$