CLOSURE

repeat

for each [X → α•Yβ, a] in Items

for each production Y → γ

for each b ∈ First(βa)

add [Y → •γ, b] to Items

until Items is unchanged

TRANSITION

Transition(State, y)

Items ← ∅

for each [X → α • yβ, b] ∈ State

add [X → αy • β, b] to Items

return Closure(Items)

**Es delle slide:**

Data la seguente grammatica:

S → L = R | R

L → \*R | id

R → L

Costruire l'automa LR(1).

1) Aggiungo S' → S, quindi lo stato di partenza contiene l'Item [S' → •S, $].

2) Trovo tutti gli item dello stato di partenza, facendo la Closure del primo item

3) Faccio le transizioni in uscita dallo stato iniziale e trovo tutti gli stati di destinazione

**Closure( { [[S' → •S, $]] } ): (prende in input un insieme di items)**

-------------

Considero l'item [S' → •S, $],

espando la variabile S,

Ba = $,

First(Ba) = $

Produzioni(S): S → L = R | R

Aggiungo item [S → •L = R, $] e [S → •R, $]

-------------------  
  
Considero l'item [S → •L = R, $],

espando L,

Produzioni(L): L → \*R | id

Ba = '=R'

First(Ba) = '='

aggiungo item [L → •\*R, =] e [L → •id, =]

-----------------------  
  
Considero item [L → •\*R, =]

Non si può fare niente, non c'è una variabile dopo il pallino

--------------------

Considero [L → •id, =]

Non si può fare niente, non c'è una variabile dopo il pallino

--------------------------------------  
  
Considero item [S → •R, $]

espando R

produzioni(R): R → L

Ba = $

First(Ba) = $

aggiungo item [R → •L, $]

------------------------  
  
Considero [R → •L, $]

espando L

produzioni(L): L → \*R | id

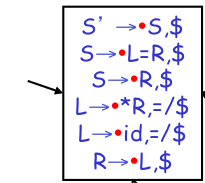
Ba = $

First(Ba) = $

aggiungo [L → •\*R, $] e [L → •id, $]

---------------------------------  
  
A questo punto bisogna controllare solo gli ultimi 2 item aggiunti, solo che non generano nuovi Item perché dopo il pallino non c'è una variabile.

Quindi, lo stato iniziale dell'automa LR(1) è:



**Transition(State, y):**

Per lo stato iniziale sono possibili 5 transizioni (basta guardare, per ogni item, il simbolo che segue il pallino):

S, L, R, \*, id

Per ogni simbolo, si chiama l'algoritmo che calcola tutti gli item che avrà lo stato di destinazione.

S0 è lo stato iniziale.

----------------------

Transition( S0, S ):

Il pallino prima di S si trova solo nell'item [S' → •S, $]

Quindi il nuovo item è [S' → S•, $]

Per trovare tutti gli item dello stato di destinazione si fa la Closure sull'insieme dei nuovi item.

CLOSURE( { [S' → S•, $] } )

Praticamente è solo questo item, dopo non c'è nessuna variabile da espandere.

Allora lo stato S1 contiene solo l'item { [S' → S•, $] }

---------------------  
  
Transition( S0, L ):

old\_item: { [R → •L, $], [S → •L = R, $] }

new\_item: { [R → L•, $], [S → L• = R, $] }

CLOSURE (new\_Item):

Considero [R → L•, $], niente

Considero [S → L• = R, $], niente

Allora lo stato S2 contiene gli item { [R → L•, $], [S → L• = R, $] }

-----------------------------------

Transition( S0, R ):

old\_item: { [S → •R, $] }

new\_item: { [S → R•, $] }

CLOSURE (new\_Item):

Considero [S → R•, $], niente

Allora lo stato S3 contiene gli item { [S → R•, $] }

---------------------  
  
Transition( S0, \* ):

old\_item: { [L → •\*R, $], [L → •\*R, =] }

new\_item: { [L → \*•R, $], [L → \*•R, =] }

CLOSURE (new\_Item):

* Considero [L → \*•R, $]

espando R

produzioni(R): R → L

Ba = $

First(Ba) = $

aggiungo item [R → •L, $]  
  
La stessa cosa vale anche per l' =,   
  
quindi aggiungo anche [R → •L, =]

* Considero [R → •L, $]  
    
  espando L  
    
  produzioni(L): L → \*R | id  
    
  Ba = $  
    
  First(Ba) = $  
    
  aggiungo [L → •\*R, $] e [L → •id, $]  
    
  per l'= è la stessa cosa,   
    
  quindi aggiungo anche [L → •\*R, =] e [L → •id, =]

Allora lo stato S4 contiene gli item

{ [L → •\*R, =/$], [L → •id, =/$], [R → •L, =], [L → \*•R, =/$] }

-------------------  
  
Transition( S0, id )

old\_item: { [L → •id, =], [L → •id, $] }

new\_item: { [L → id•, =], [L → id•, $] }

La Closure produce gli stessi item perché non c'è nessuna variabile da espandere.

Allora lo stato S5 contiene gli item

{ [L → id•, =/$] }

-------------------------  
  
Quindi le transizioni in uscita dallo stato iniziale sono quelle mostrate in figura:

