**BILET2**

**1)**

1. **Ce este analiza sintactica?**

R: Analizã sintacticã pt. cuvantul w succesiunea de derivãri directe:

S => a1 => a2 => …. => w altfel spus: reprezintã o derivare pentru cuvântul w

1. **Reguli de construire a tabelului SLR**
2. **Colectia canonica** : [A->alfa.beta, u], u = FOLLOW1(A), |u|=1

Se tine cont de predicitie doar [pentru reducere

1. **Tabelul de analiza**:

Linii: elementele colectiei canonice (Starile, ex: I0, I1…)

Coloane: N U Sigma U {$}

Celula: sstare, rnr\_regula\_productie, acc

1. **Analizor** : tranzitii conform tabelului de analiza

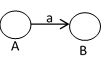
* Banda de intrare
* Stiva de lucru
* Banda de iesire

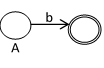
**Configuratie initiala** : ($0,w$, eps)

**Configuratie finala**: ($0S Iacc , $, banda\_de\_iesire)

1. **Descrie algoritmul care construieste un Automat Finit din Gramatica Regulara**

La fiecare neterminal i se asociază o stare + 1 stare finală.

A->aB 

A->b 

S -> eps 

**2)**

**a) Se da gramatica G: ({S,A,B},{0,1},P,S)**

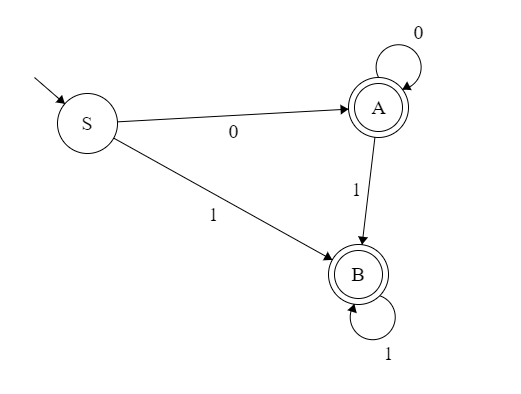
**P: S->0A | 1B**

**A->0A | 1B | 0**

**B->1 | 1B**

**Se cere Automatul finit si expresia regulara.**

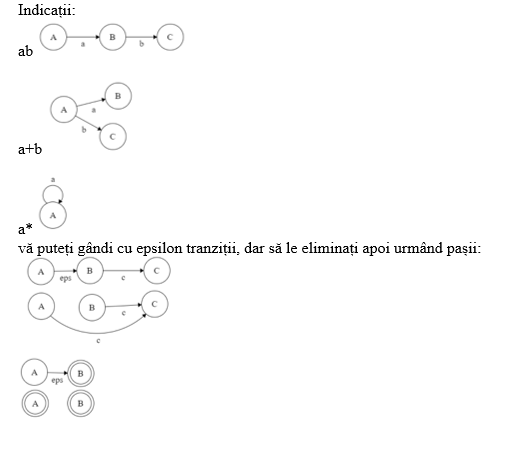
A si B sunt stari finale pentru ca opresc recursivitatea.



V1. (00\*) + (00\*11\*) + (11\*)

V2. (00\*(11\*)\*) + (11\*)

*Expresii regulare: + = “sau”, adica REUNIUNE de multimi*



**b) Se da gramatica G: ({S,A,B,C,D},{a,b,c,d,e},P,S)**

**P: S-> Aa** 1

**A-> BC** 2

**B-> b | Dd** 3

**C-> epsilon | cA** 4

**D-> e** 5

**Se cere functia First**

**FOLLOW:   
Cautam NETERMINALUL in dreapta regulilor de productie(MD).**

1. **Daca dupa el nu urmeaza nimic, Follow-ul lui va fi egal cu Follow-ul MS**
2. **Daca avem dupa el un terminal, Follow-ul lui va fi acel terminal (ex: S-> Ba => Follow(B) = a)**
3. **Daca avem dupa el un neterminal, Follow-ul lui va fi First-ul neterminalului de dupa el (ex: S-> BA, Follow(B) = First(A))**

**-> In cazul in care First(A) contine si eps, “inlocuim” pe A cu eps, si daca dupa B nu mai urmeaza nimic, atunci Follow(B) = Follow(S). (adica o sa avem ceva de genul S -> B => regula 1)**

**FIRST:**

**O sa ne uitam la primul terminal din dreapta regulilor de productie**

**Ex: S->aB (First(S) = a)**

**Daca avem un caz de genul S->Ab, A->c atunci First(S) = First(A) = c**

S-> BAC

Follow(B) = First(A)

Daca avem First(A) = eps, atunci Follow(B) = First(C)

Follow(A) = First(C)

Follow(C) = Follow(S)

Explicatii indian iubitu lu Patricia:

<https://www.youtube.com/watch?v=_uSlP91jmTM>

| **Simbol** | **First1** | **Follow1** |
| --- | --- | --- |
| **S** | **b, e** | **$** |
| **A** | **b, e** | **a** |
| **B** | **b, e** | **in pr2, pentru ca First(c) = eps: c, a, Follow(C)** |
| **C** | **eps, c** | **a** |
| **D** | **e** | **d** |

In regula 2, avem B urmat de C. Follow(B)= First(C)

In regula 4, pentru ca nu avem nimic dupa A, Follow(A) = Follow(C)

**3)**

1. **Daca L1 si L2 sunt limbaje independente de context , demonstrati ca L1L2 e limbaj independent de context**

L1: L1->a

L2: L2->b

L1L2:

S->L1L2

L1->a

L2->b

—-----------------------------------------------------------------------------------------------------------------------------

BILET 3(INFO ROMANA)

1.

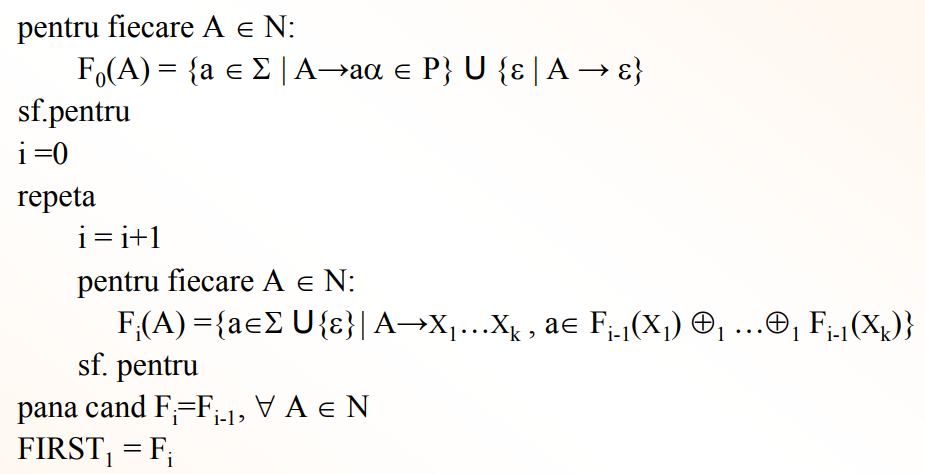
1. **enumerati modurile de reprezentare a TS**

* arbore binar de cautare echilibrat
* tabel lexicografic
* tabela de dispersie

1. **definiti functia goto LR(0)**

* goto(I,X) = Closure((A->aX.b, u) cu proprietatea ca (A->a.Xb,u) apartine I)

1. **un algoritm de calcul pt FIRST**

****

**Analiza descendenta cu reveniri**

Configuratie: (s, i, alfa, beta)

s = stare

- q = stare normala

- r = revenire

- t = terminare

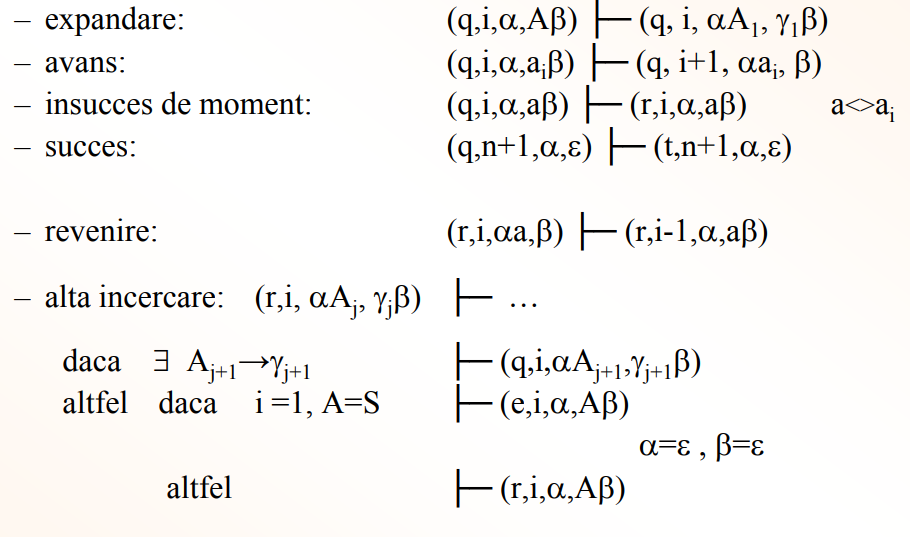
- e = eroare

i = pozitia urmatoare in secventa de intrare ( beta )

alfa = stiva de lucru

beta = stiva de intrare ( de verificat)

**Configuratia INITIALA: (q, 1, eps, S)**



Exemplu:

S → aSbS S1

S → aS S2

S → c S3

Se cere:

acbc L(G) (?)

(q, 1, eps, S) |- (expandare) (q, 1, S1, aSbS)

(q, 1, S1, aSbS) |- (avans) (q, 2, S1a, SbS)

(q, 2, S1a, SbS) |- (expandare) (q, 2, S1a**S1**, **aSbS**bS)

(q, **2**, S1aS1, **a**SbSbS) |- (insucces de moment) (**r**, 2, S1aS1, aSbSbS)

(**r**, 2, S1aS1, **aSbS**bS) |- (alta incercare) (q, 2, S1aS2, **aS**bs)

(q, 2, S1aS2, aSbs) |- (insucces de moment) (**r**, 2, S1aS2, aSbS)

(**r**, 2, S1aS2, aSbS) |- (alta incercare) (q, 2, S1aS3, **c**bS)

(q, 2, S1aS3, cbS) |- (avans) (q, **3**, S1aS3**c**, bS)

(q, 3, S1aS3c, bS) |- (avans) (q, 4, S1aS3cb, S)

(q, 4, S1aS3cb, S) |- (expandare) (q, 4, S1aS3cbS1, aSbS)

(q, 4, S1aS3cbS1, aSbS) |- (insucces de moment) (**r**, 4, S1aS3cbS1, aSbS)

(**r**, 4, S1aS3cbS**1**, **aSbS**) |- (alta incercare) (q, 4, S1aS3cbS2, **aS)**

(q, 4, S1aS3cbS2, aS) |- (insucces de moment) (r, 4, S1aS3cbS2, aS)

(r, 4, S1aS3cbS2, **aS**) |- (alta incercare) (q, 4, S1aS3cbS3, **c**)

(q, 4, S1aS3cbS3, c) |- (avans) (q, 5, S1aS3cbS3c, eps)

(q, 5, S1aS3cbS3c, eps) |- (succes) (**t**, 5, S1aS3cbS3c, eps)

ab

(q,1,eps,S) |- expandare (q,1,S1,aSbS)

(q,1,S1,aSbS) |- avans (q,2,S1a,SbS)

(q,2,S1a,**S**bS) |- expandare (q,2,S1a**S1**, **aSbS**bS)

(q,2,S1aS1, aSbSbS) |- (insucces de moment) (r,2,S1aS1, aSbSbS)

(r,2,S1a**S1**, **aSbS**bS) |- (alta incercare) (q, 2, S1a**S2**, **aS**bS)

(q, 2, S1aS2, aSbS) |- (insucces de moment) (r,2,S1aS2, aSbS)

(r,2,S1a**S2**, **aS**bS) |- (alta incercare) (q, 2, S1a**S3**, **c**bS)

(q, 2, S1aS3, cbS) |- (insucces de moment) (r, 2, S1aS3, cbS)

(r, 2, S1aS3, cbS) |- (alta incercare altfel) (r, 2, S1a, SbS)

(r, 2, S1a, SbS) |- (revenire) (r, 1, S1, aSbS)

(r, 1, **S1**, **aSbS**) |- (alta incercare) (q, 1, **S2**, **aS)**

(q, 1, S2, aS) |- (avans) (q, 2, S2a, S)

(q, 2, S2a, S) |- expandare (q, 2, S2aS1, aSbS)

…

(q,2, S2aS3, c) |- (insucces de moment) (r, 2, S2aS3, c)

(r, 2, S2aS3, c) |- alta incercare altfel (r, 2, S2a, S)

(r, 2, S2a, S) |- revenire (r, 1, S2, aS)

(r, 1, S2, aS) |- (alta incercare) (q,1, S3, c)

(q,1 S3, c) |- insucces de moment (r,1,S3,c)

(r,1,S3,c) |- alta incercare (r,1,eps, S)

(r,1,eps, S) |- alta incercare (e,1,eps,S)

**Analiza sintactica descendenta LL(1)**

Avem nevoie de First si Follow

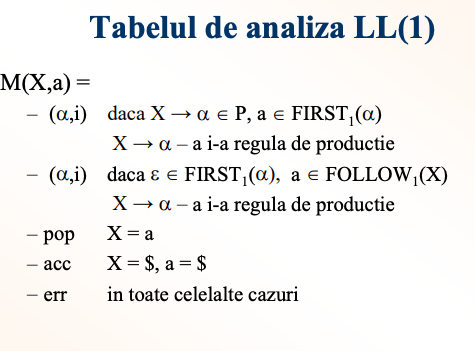
Tabel :

Linii : terminale + neterminale + $

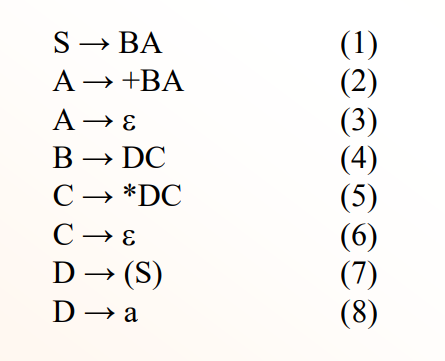
Coloane : terminale + $

Celule : (MD a regulii de prod, nr reg)

4 cazuri :

****

Ex:



| **Simbol** | **First** | **Follow** |
| --- | --- | --- |
| **S** | **(,a** | **$, )** |
| **A** | **+, eps** | **$, )** |
| **B** | **(,a** | **+, $, )** |
| **C** | **\*, eps** | **+, $, )** |
| **D** | **(, a** | **\*, +, $, )** |

S->BA (1) => FIRST(BA) = FIRST(B) = {a, ( }. eps nu apartine FIRST(B).

Pe linia lui S ( m.s) coloanele **a** si **((FIRST(MD) = FIRST(B))** se scrie (MD, nr\_reg) = (BA, 1)

|  | a | + | \* | ( | ) | $ |
| --- | --- | --- | --- | --- | --- | --- |
| S | (BA, 1) |  |  | (BA, 1) |  |  |
| A |  | (+BA, 2) |  |  | (eps, 3) | (eps, 3) |
| B | (DC, 4) |  |  | (DC, 4) |  |  |
| C |  | (eps, 6) | (\*DC, 5) |  | (eps, 6) | (eps, 6) |
| D | (a, 8) |  |  | ((S), 7) |  |  |
| a | pop |  |  |  |  |  |
| + |  | pop |  |  |  |  |
| \* |  |  | pop |  |  |  |
| ( |  |  |  | pop |  |  |
| ) |  |  |  |  | pop |  |
| $ |  |  |  |  |  | acc |

Nu este conflict=> E de tip LL1

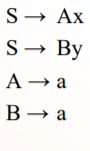
Ex: S-> BA ->DCA -> a\*a

| Stiva de lucru | Banda de intrare | Banda de iesire |
| --- | --- | --- |
| S$ | a\*a$ | eps |
| BA$ | a\*a$ | 1 |
| DCA$ | a\*a$ | 4 1 |
| aCA$ | a\*a$ | 8 4 1 |
| CA$ | \*a$ | 8 4 1 |
| \*DCA$ | \*a$ | 5 8 4 1 |
| DCA$ | a$ | 5 8 4 1 |
| aCA$ | a$ | 8 5 8 4 1 |
| CA$ | $ | 8 5 8 4 1 |
| A$ | $ | 6 8 5 8 4 1 |
| $ | $ | 3 6 8 5 8 4 1 |

Acceptare

**Analiza ascendenta LR(0)**

1. Imbogatirea gramaticii
2. Colectia canonica (acea mare figura)
3. Tabelul de analiza
4. Analiza propriu-zisa (testarea unei secvente)



1. Imbogatirea gramaticii - adaugam un nou simbol de stare **S’**.

**S’** -> S (0)

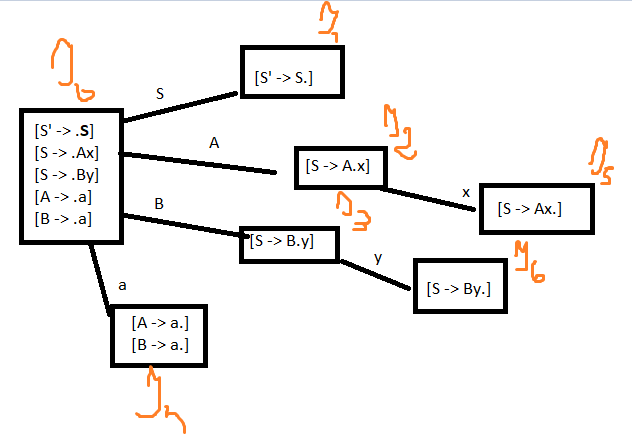
S -> Ax (1)

S -> By (2)

A -> a (3)

B -> a (4)

1. Colectia canonica



|  | Actiune | S | A | B | a | x | y |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I0 | s | I1 | I2 | I3 | I4 |  |  |
| I1 | acc |  |  |  |  |  |  |
| I2 | s |  |  |  |  | I5 |  |
| I3 | s |  |  |  |  |  | I6 |
| I4 | 3, 4 |  |  |  |  |  |  |
| I5 | 1 |  |  |  |  |  |  |
| I6 | 2 |  |  |  |  |  |  |

Avem conflict => gramatica nu este de tip LR(0)

—- w = acb

S -> aSb

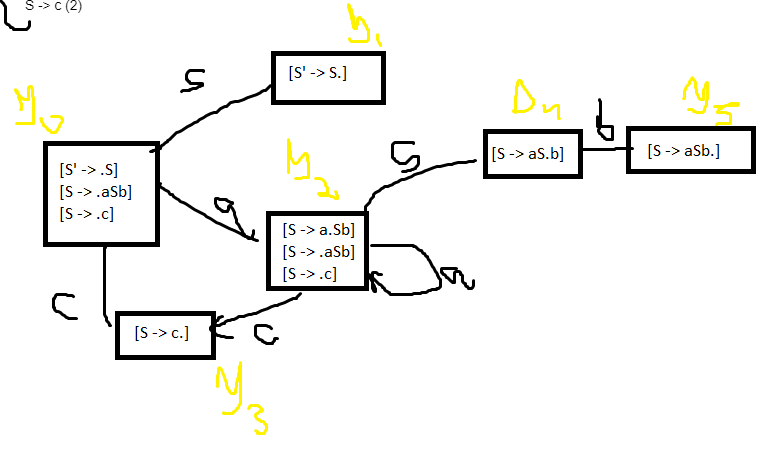
S -> c

1. Imbogatirea gramaticii

S’ -> S (0)

S -> aSb (1)

S -> c (2)

1. Colectia canonica
2. Tabelul de analiza LR(0)

|  | Actiune | S | a | b | c |
| --- | --- | --- | --- | --- | --- |
| I0 | s | I1 | I2 |  | I3 |
| I1 | acc |  |  |  |  |
| I2 | s | I4 | I2 |  | I3 |
| I3 | 2 |  |  |  |  |
| I4 | s |  |  | I5 |  |
| I5 | 1 |  |  |  |  |

1. Analiza propriu-zisa

W = acb

S -> aSb (1)

S -> c (2)

| Stiva de lucru | Banda de intrare | Banda de iesire |
| --- | --- | --- |
| $0 | acb$ | Epsilon |
| $0a2 | cb$ | Epsilon |
| $0a2**c3** | b$ | Epsilon |
| $0a2S4 | b$ | 2 |
| $0**a2S4b5** | $ | 2 |
| $0S1 | $ | 1 2 |
| acc |  |  |

W = acb apartine LR(0) cu regulile de productie 1, 2

W = acb, format prin S -> (1) -> aSb -> (2) -> acb

Daca am fi pus 2, 1 ar fi venit S -> (2) -> c -> (1) -> nu se poate

**Analiza ascendenta SLR**

Element de analiza : [A-> alfa.beta, u], u=FOLLOW1(A), |u| = 1

Predicția se foloseste doar pentru reduceri

1. Constructia colectiei canonice
2. Constructia tabelului de analiza

Se tine cont de predictie pentru reduceri

**Linii**: elementele colectiei canonice (starile)

**Coloane**: N U Sigma U {$}

**Celule**: s\_stare (shift), r\_nr\_reg\_prod (reducere cu numarul regulii de productie), acc (acceptare)

1. Analizor (~LR(0))

Exemplu:

**S’** -> S (0)

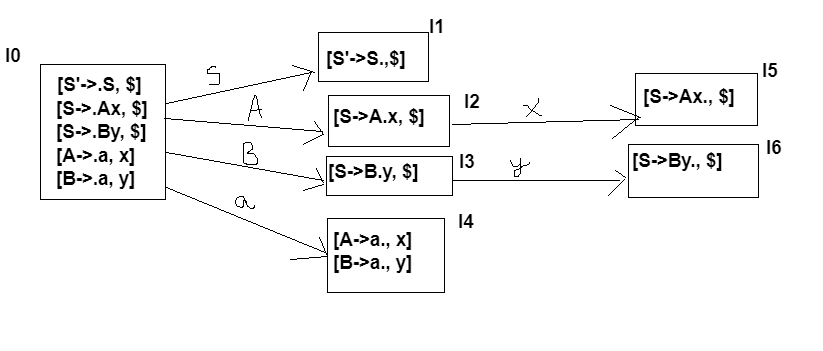
S -> Ax (1)

S -> By (2)

A -> a (3)

B -> a (4)

1. Colectia canonica



|  | S | A | B | a | x | y | $ |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I0 | s1 | s2 | s3 | s4 |  |  |  |
| I1 |  |  |  |  |  |  | acc |
| I2 |  |  |  |  | s5 |  |  |
| I3 |  |  |  |  |  | s6 |  |
| I4 |  |  |  |  | r3 | r4 |  |
| I5 |  |  |  |  |  |  | r1 |
| I6 |  |  |  |  |  |  | r2 |

Verificam ax

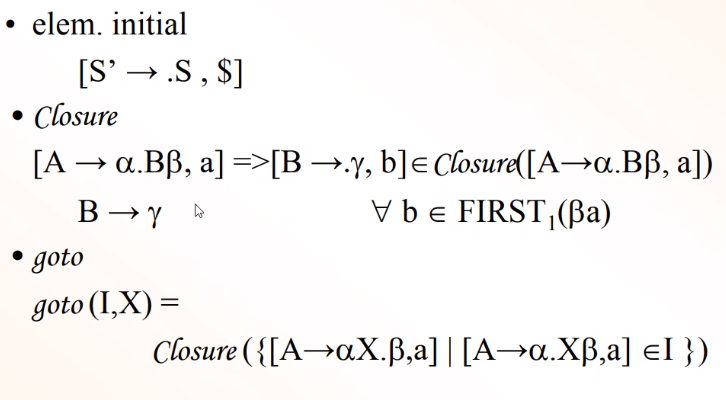
| Stiva de lucru | Banda de intrare | Banda de iesire |
| --- | --- | --- |
| $0 | ax$ | eps |
| $0a4 | x$ | eps |
| $0A2 | x$ | 3 |
| $0A2x5 | $ | 3 |
| $0S1 | $ | 1 3 |
| acceptare |  |  |

Secventa ax este acceptata cu regulile de productie 1 si 3.

**Analiza ascendenta LR(1)**

1. Imbogatirea gramaticii
2. Constructia colectiei canonice

[A->alfa.beta, u], |u|=1



1. Constructia tabelului de analiza (~ SLR)
2. Analiza

Exemplu:

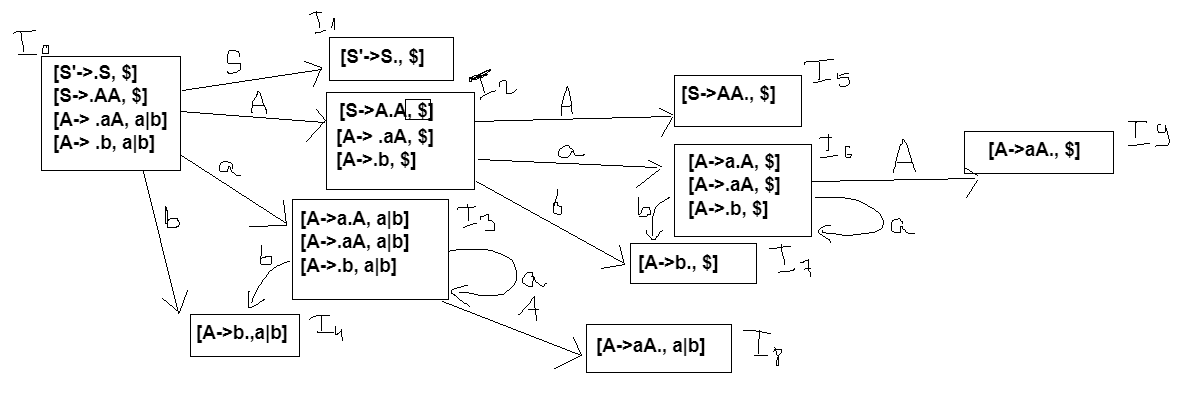
Imbogatim gramatica

S’ -> S (0)

S → AA (1)

A → aA (2)

A → b (3)



|  | S | A | a | b | $ |
| --- | --- | --- | --- | --- | --- |
| I0 | s1 | s2 | s3 | s4 |  |
| I1 |  |  |  |  | acc |
| I2 |  | s5 | s6 | s7 |  |
| I3 |  | s8 | s3 | s4 |  |
| I4 |  |  | r3 | r3 |  |
| I5 |  |  |  |  | r1 |
| I6 |  | s9 | s6 | s7 |  |
| I7 |  |  |  |  | r3 |
| I8 |  |  | r2 | r2 |  |
| I9 |  |  |  |  | r2 |

Stiva de lucru, banda de intrare, banda de iesire