## Expression régulière:

{a, b}² = aa + ab + ba + bb

L = aL + b = a\* b (theorem arden)

λn = λ

a . (a + b) = aa + ab



Exercice 1

1)

L0 = a . (a + b)\*

= a L1

L1 = (a + b)\*

= (a + b) L1 + λ

= a L1 + b L1 + λ

2)

L0 = a b\* + (a + b) c\*

= a b\* + a c\* + b c\*

= a (b\* + c\*) + b c\*

= a L1 + b L2

L1 = b\* + c\*

= b b\* + λ + c c\* + λ

= b b\* + λ + c L2 + λ

= b L3 + λ + c L2 + λ

L2 = c\*

= c L2 + λ

L3 = b\*

= b L3 + λ 

3)

L0 = (p + m + λ) D D\*

= p D D\* + m D D\* + D D\*

L1 = D D\*

= D L2

L2 = D\*

= D L2 + λ

## 

## Analyse Syntaxique:

1)

G1 = S’ → S $k (0)

S → a S b S (1)

S → λ (2)

voir si G1 est LL(0) ? si LL(0) alors il n’a que 1 chemin 1 seul mot valide donc G1 n’est pas LL(0). S a deux règles de prod donc non.

voir si G1 est LL(1) ?

= 1 - Lookahead (S’ → S $)

= first1(S $ ~~follow~~~~1~~~~(S’)~~) (follow inutile car obligatoirement un caractère avant donc le first prendra toujours le character avant le follow)

= first1(a S b S $ ~~follow~~~~1~~~~(S’)~~ ) **∪** first1(λ $ ~~follow~~~~1~~~~(S’)~~ )

= {a, $}

= 1 - lookahead(S → aSbS)

= first¹(aSbS)

= {a}

= 1 - lookahead(S → λ)

= first¹(λ follow¹(S))

= follow¹(S)

= first¹($ follow ¹(S')) ∪ first ¹(bS follow ¹(S)) ∪ first ¹(follow ¹(S))

= {$} ∪ {b} ∪ follow ¹(S)

= {$, b}

I¹ ∩ I² = ø

G¹ est LL(1)

|  | a | b | $ |
| --- | --- | --- | --- |
| S' | S$ (0) | Err | S$ (0) |
| S | aSbS (1) | λ (2) | λ (2) |

Préparé pour la prochaine fois le mardi g de abab$ et le LL(1) de la question 2)

| mot | pile | action |
| --- | --- | --- |
| abab$ | S’ | (0) |
| abab$ | S$ | (1) |
| abab$ | aSbS$ | pop(a) |
| bab$ | SbS$ | (2) |
| bab$ | bS$ | pop(b) |
| ab$ | S$ | (1) |
| ab$ | aSbS$ | pop(a) |
| b$ | SbS$ | (2) |
| b$ | bS$ | pop(b) |
| $ | S$ | (2) |
| $ | $ | Accepted |

2)

S → S $k (0)

S → b R S (1)

S → R c S a (2)

S → λ (3)

R → a c R (4)

R → b (5)

* LL(0) ? Non car R et S ce dérive de plusieurs façons.
* LL(1) ? Non car (1) fini sur b et que (2) peut aussi finir sur b.

1 - lookahead(S → bRS)

= {b} = I1

1 - lookahead(S → RcSa)

= first1(RcSa follow1(S))

= first1(acRcSa) ∪ first1(bcSa)

= {a, b} = I2

I1 ∩ I2 = {b} ≠ Ø

donc pas LL(1)

* LL(2) ?

2 - lookahead(S → bRS)

= first2(bRS follow2(S))

= b . first1(RS follow2(S))

= {ba, bb} = I1

2 - lookahead(S → RcSa)

= first2(RcSa follow2(S))

= first2(acRcSa…) ∪ first2(bcSa…)

= {ac, bc} = I2

2 - lookahead(S → λ )

= follow2(S)

= first2($$...) ∪ ~~first~~~~2~~~~(follow~~~~2~~~~(S))~~ ∪ first2(a follow2(S))

= {$$} ∪ a . follow1(S)

= {$$, a$, aa} = I3

I1 ∩I2 = ø

I2 ∩I3 = ø

I3 ∩I1 = ø

2 - lookahead(S’ → S$2)

= first2(S$$ follow2(S’))

= first2(bRs$$) ∪ first2(RcSa$$) ∪ first2($$)

= b . first1(Rs$$) ∪ first2(acRcSa$$) ∪ first2(bcSa$$) ∪ {$$}

= b. (first1(acR…) ∪ first1(b….)) ∪ {ac} ∪ {bc} ∪ {$$}

= {ba, bb, ac, bc, $$}

2 - lookahead(R → acR)

= {ac}

2 - lookahead(R → b)

= first2(b follow2(R))

= b . first1(follow1(R))

= b . ( first1(S follow1(S)) ∪ first1(cSa…) ∪ ~~first~~~~1~~~~(follow~~~~1~~~~(R)~~ )

= {ba, b$, bb, bc}

S → S $k (0)

S → b R S (1)

S → R c S a (2)

S → λ (3)

R → a c R (4)

R → b (5)

|  | aa | ac | a$ | ba | bb | bc | b$ | $$ |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S' | Err | S$$ (0) | Err | S$$ (0) | S$$ (0) | S$$ (0) | Err | S$$ (0) |
| S | (3) | (2) | (3) | (1) | (1) | (2) | Err | (3) |
| R | Err | (4) | Err | (5) | (5) | (5) | (5) | Err |

| mot | pile | action |
| --- | --- | --- |
| acb$$ | S’ | (0) |
| acb$$ | S$$ | (2) |
| acb$$ | RcSa$$ | (4) |
| acb$$ | acRcSa$$ | pop(ac) |
| b$$ | RcSa$$ | (5) |
| b$$ | bcSa$$ | pop(b) |
| $$ | cSa$$ | Err |

S → S $k (0)

S → b R S (1)

S → R c S a (2)

S → λ (3)

R → a c R (4)

R → b (5)

|  | aa | ac | a$ | ba | bb | bc | b$ | $$ |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S' | Err | S$$ (0) | Err | S$$ (0) | S$$ (0) | S$$ (0) | Err | S$$ (0) |
| S | (3) | (2) | (3) | (1) | (1) | (2) | Err | (3) |
| R | Err | (4) | Err | (5) | (5) | (5) | (5) | Err |

| mot | pile | action |
| --- | --- | --- |
| bcbcaa$$ | S’ | (0) |
| bcbcaa$$ | S$$ | (2) |
| bcbcaa$$ | RcSa$$ | (5) |
| bcbcaa$ | bcSa$$ | pop(bc) |
| bcaa$$ | Sa$$ | (2) |
| bcaa$$ | RcSaa$$ | (5) |
| bcaa$$ | bcSaa$$ | pop(bc) |
| aa$$ | Saa$$ | (3) |
| aa$$ | λaa$$ | pop(aa) |
| $$ | $$ | Accept |

4) Let the grammar G4 below :

S → S $k (0)

S → a A (1)

A → b B (2)

B → c (3)

Find k s.t. G4 is LL(k). Give the analysis table.

* LL(0) ? Oui car il n’existe que 1 chemin possible, un seul mot possible. une seul règle par non terminal

L(G4) = {abc}

|  | λ |
| --- | --- |
| S’ | (0) |
| S | (1) |
| A | (2) |
| B | (3) |

5) Let the partial grammar G5 below :

< program > → program < declList > begin < instList > end (0)

< instList > → < inst > (1)

→ < instList > ; < inst > (2)

< inst > → if < exp > then < instList > else < instList > endif (3)

→ while < exp > loop < instList > endloop (4)

→ repeat < instList > until < exp > endloop (5)

→ ID := < exp > (6)

< declList > → …

< exp > → …

Is G5 LL(k) ? Why do we need to transform G5 to make it LL(k) ? And how ?

G5 est récursif a gauche

A → B (1)

A → AcB (2)

L(A) = L(B) + L(A) c L(B)

= L(A) c L(B) + L(B)

= L(B) (c L(B))\*

= (L(B) c)\* L(B)

= L(B) c L(A) + L(B)

⇔ < instList > → < inst > (1)

→ < inst > ; < instList > (2)

G5 est maintenant récursif à droite plus à gauche.

mais < isnt > est de longueur non bornée donc toujours pas LL(k)

L(A) = L(B) (c L(A) + λ )

= L(B) L(C)

L(C) = (c L(A) + λ )

A → B C

C → c A

C → λ

⇔ < instList > → < inst > < instListOpt >

< instListOpt > → ; < instList >

→ λ

Let the augmented grammar G3 with axiom S

0 below :

S’ → S $k (0)

S → a R (1)

S → R b (2)

S → a b (3)

R → c R (4)

R → λ

2 - lookahead

{ab, ac, a$, b$, cb, cc}

{ac, a$}

{ab}

{cb, cc, c$}

{b$, $$}

|  | ab | ac | a$ | b$ | cb | cc | c$ | $$ |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S’ | (0) | (0) | (0) | (0) | (0) | (0) | err | err |
| S | (3) | (1) | (1) | (2) | (2) | (2) | err | err |
| R | err | err | err | (5) | (4) | (4) | (4) | (5) |

| mot | pile | action |
| --- | --- | --- |
| ccb$$ | S’ |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |