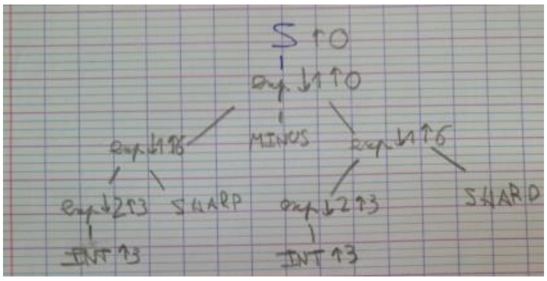
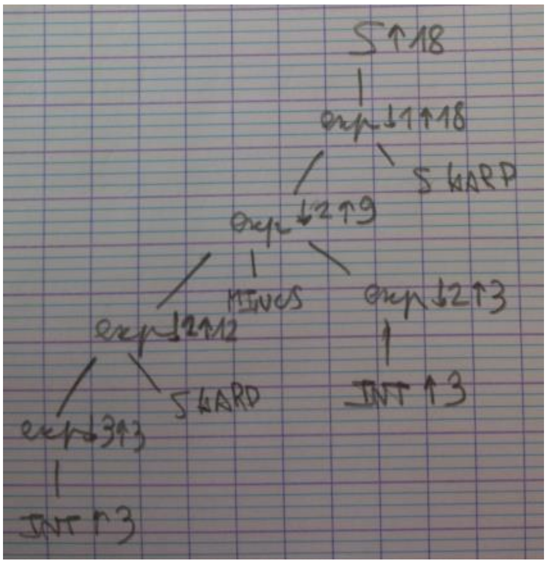
Exercice 1

vendredi 3 mars 2017 15:40

Question 1

Pour obtenir 3 # - 3 #, on peut faire :
(3 #) - (3 #)
(3 # - 3) #





Question 2

Prenons un exemple.

$S \rightarrow a|Ac|X$
$A \rightarrow bAa|c$
$X \rightarrow aY$
$Y \rightarrow bX$
$Z \rightarrow c$

Z n'est pas atteignable (On ne peut pas l'obtenir en partant de l'axiome) et X et Y ne sont pas productifs (Ils bouclent) donc on les supprime. Cherchons les directeurs :
$\text{Dir}(S \rightarrow a) = \{a\}$
$\text{Dir}(S \rightarrow Ac) = \text{Prem}(Ac) = \text{Prem}(A) \cup \{c\} = \{b, c\}$
$\text{Dir}(A \rightarrow bAa) = \{b\}$
$\text{Dir}(A \rightarrow \varepsilon) = \text{Suiv}(A)$

Calcul des directeurs :
$\text{Dir}(A \rightarrow w) = \text{Prem}(w) \cup \varepsilon(w).\text{Suiv}(A)$
$\varepsilon(w) = \exists w \rightarrow^* \varepsilon$
$\text{Prem}\left(X \rightarrow \underbrace{e_1 | \dots}_{e} \middle| e_n\right)$
$\text{Prem}(X) = \text{Prem}(e)$
$\text{Prem}(\varepsilon) = \emptyset$
$\text{Prem}(a) = \{a\}$
$\text{Prem}(w_1.w_2) = \text{Prem}(w_1) \cup \varepsilon(w_1)\text{Prem}(w_2)$
$\text{Prem}(\alpha|\beta) = \text{Prem}(\alpha) \cup \text{Prem}(\beta)$

$\text{Suiv}(X) = \{\$\}$ si X axiome, sinon :
$\text{Suiv}(X) = \cup\ (\text{Prem}(\beta) \cup \varepsilon(\beta).\text{Suiv}(A))$ avec $A \rightarrow \alpha X \beta$

$\text{Prem}(A) = b \rightarrow \ \text{Suiv}(A) = \{a, c\}$

$1 : S \rightarrow a$
$2 : S \rightarrow Ac$
$3 : A \rightarrow bAa$
$4 : A \rightarrow \varepsilon$

Voici les règles à appliquer selon l'état courant et le caractère lu :

	a	b	c
S	1	2	2
A	4	3	4

Question 2

$\text{Dir}(S \rightarrow \text{exp}) = \text{Prem}(\text{exp})$
$\text{Dir}(\text{exp} \rightarrow \text{INT}) = \{\text{INT}\}$
$\text{Dir}(\text{exp} \rightarrow \text{OPAR exp CPAR}) = \{\text{OPAR}\}$
$\text{Dir}(\text{exp} \rightarrow \text{exp SHARP}) = \text{Prem}(\text{exp})$
$\text{Dir}(\text{exp} \rightarrow \text{exp MINUS exp}) = \text{Prem}(\text{exp})$
$\text{Prem}(\text{exp}) = \{\text{INT}, \text{OPAR}\} \cup \text{PREM}(\text{exp}) = \{\text{INT}, \text{OPAR}\}$

La grammaire n'est pas LL(1) car les deux derniers directeurs ne sont pas disjoints

Exercice 2

vendredi 3 mars 2017 16:29

Question 1

1)

$\text{Dir}(S \rightarrow BbB) = \text{Prem}(BbB) = \text{Prem}(B) \cup \text{Prem}(bB) = \{b\}$

$\text{Dir}(B \rightarrow bS) = \{b\}$

$\text{Dir}(B \rightarrow \varepsilon) = \text{Suiv}(B)$

$\text{Suiv}(B) = \text{Prem}(bB) \cup \text{Suiv}(S) = \{b, \$\}$

$\text{Suiv}(S) = \{\$\} \cup \text{Suiv}(B) = \{b, \$\}$

Cette grammaire n'est pas LL1 car les directeurs ne sont pas disjoints.
Le langage de cette grammaire est l'ensemble des suites de longueur impaires du caractère b.

2)

$\text{Dir}(S \rightarrow BaABcB) = ?$

$\text{Dir}(A \rightarrow a) = \{a\}$

$\text{Dir}(A \rightarrow \varepsilon) = \text{Prem}(\varepsilon) \cup \text{Suiv}(A) = \text{Suiv}(A)$

$\text{Dir}(B \rightarrow bS) = \text{Prem}(bS) = \{b\}$

$\text{Dir}(B \rightarrow \varepsilon) = \text{Suiv}(B)$

$\text{Suiv}(A) = \text{Prem}(BcB) = \text{Prem}(B) \cup \{c\} = \{b, c\}$

$\text{Suiv}(B) = \{a, c\} \cup \text{Suiv}(S) = \{a, c, \$\}$

$\text{Suiv}(S) = \text{Suiv}(B) \cup \{\$\} = \{a, c, \$\}$

Cette grammaire est LL1 car pour toute règle de la forme $A \rightarrow X$, les directeurs sont disjoints

Question 2

Une grammaire est LL1 si, à chaque caractère rencontré, on sait quelle règle appliquer.

1) $L_1 = \{a^n b^m | 0 \leq n \leq m\}$
Soit mots de la forme $a^n b^n b^p$
$S \rightarrow AB$
$A \rightarrow aAb \mid \varepsilon$
$B \rightarrow bB \mid \varepsilon$

Exercice 3

vendredi 10 mars 2017 15:45

S ::= aXY
X ::= Sb | ε
Y ::= cYa | ε

1)
$Dir(S \to aXY) = ?$
$Dir(X \to Sb) = Prem(Sb) = Prem(S) = \{a\}$
$Dir(X \to \varepsilon) = Suiv(X) = \{b, c, \$\}$
$Dir(Y \to cYa) = \{c\}$
$Dir(Y \to \varepsilon) = Suiv(Y) = \{a, b, \$\}$

$Suiv(S) = \{b, \$\}$
$Suiv(X) = Prem(Y) \cup Suiv(S) = \{b, c, \$\}$
$Suiv(Y) = Suiv(S) \cup \{a\} = \{a, b, \$\}$

C'est une grammaire LL1

3)
```
int parse_S(int h, int *r)
{
    int r1, r2;
    parse_token(a);
    parse_X(h, &r1);
    parse_Y(h, &r2);
    *r = max(r1, r2);
}
```

```c
int parse_X(int h, int *r)
{ //On utilise les directeurs
    if (current == b) {
        parse_S(h + 1, r);
        parse_token(b);
    } else {
        *r = 3*h;
    }
}

int parse_Y(int h, int *r)
{
    if (current == c) {
        parse_token(c);
        parse_Y(h + 1, r);
        parse_token(a);
    } else {
        *r = power2(h);
    }
}

void parse() {
    int r;
    parse_S(0, &r);
    parse_token(END);
}
```

Exercice 4

vendredi 10 mars 2017 16:47

1)
exp ::= idf = exp | num
inst ::= idf instX | num;
inst X ::= :inst | = exp;

2)
On factorise "list ::== inst | inst list" par list. Cette règle devient alors :
list ::== inst listX
listX :== ε | list

On garde la règle inst ::== exp ;

On factorise "exp ::== num | lvalue | lvalue = exp | lvalue ++" par lvalue :
exp ::== num | lvalue expX
expX ::== ε | = exp | ++

On fait une élimination de la récursivité dans
"lvalue ::== lvalue . idf | lvalue [exp] | idf" par lvalue :
lvalue :== idf | lvalueX
lvalueX :== . idf lvalueX | [exp] lvalueX | ε |

3.1) Il y a ambiguïté dans l'expression :
if (e_1) if (e_2) e_3; else e_4;

Elimination de la récursivité gauche
On a la règle :
X ::== Xα_1 |...|Xα_n|β_1|...|β_m
Celle-ci devient :
X ::== β_1X'|...|β_mX'
X' ::== α_1X'|...|α_nX'|ε

Factorisation :
On a la règle :
X ::== $\alpha\beta_1$|... |$\alpha\beta_n$|γ_1|. . |γ_m
Elle devient :
X ::== αX'|γ_1|...|γ_m
X' ::== β_1|...|β_n

Exercice 5

vendredi 31 mars 2017 15:45

$\text{inst} \uparrow I \downarrow U ::= \varepsilon \quad I = \emptyset$

$\quad | \ \text{exp} \uparrow I \downarrow U$

$\quad | \ \text{inst} \uparrow I_1 \downarrow U \ \text{inst} \uparrow I_2 \downarrow (U \cup I_1) \quad I = I_1 \cup I_2$

$\quad | \ \text{if } (\text{exp} \uparrow I_1 \downarrow U) \ \{\text{inst} \uparrow I_2 \downarrow (U \cup I_1)\} \text{ else } \{\text{inst} \uparrow I_3 \downarrow (U \cup I_1)\} \quad I = I_1 \cup (I_2 \cap I_3)$

$\quad | \ \text{while } (\text{exp} \uparrow I_1) \ \{\text{inst} \uparrow I_2 \downarrow (U \cup I_1)\} \quad I = I_1$

$\text{exp} \uparrow I ::= \text{idf} \uparrow x \quad I = \emptyset, \text{assert}(x \in U)$

$\quad | \ \text{idf} \uparrow x = \text{exp} \uparrow I_1 \downarrow U \quad I = \{x\} \cup I_1$

$\quad | \ \text{num} \quad I = \emptyset$

$\quad | \ \text{exp} \uparrow I_1 \downarrow U + \text{exp} \uparrow I_2 \downarrow (U \cup I_1) \quad I = I_1 \cup I_2$

$\quad | \ \text{exp} \uparrow I_1 \downarrow U \leq \text{exp} \uparrow I_2 \downarrow (U \cup I_1) \quad I = I_1 \cup I_2$