

Compte Rendu de compilation TP2

Théo CHAPON, Alexandre AUDINOT

INSA de Rennes
4INFO, groupe A

October 23, 2014

TP2 : Conception de la classe chaine et surcharge

1 Nouvelle grammaire

```
<Super> -> <Expr> eof
<Expr> -> <Termb> <Expr2>
<Expr2> -> "ou" <Termb> <Expr2> | ε
<Termb> -> <Facteurb> <Termb2>
<Termb2> -> "et" <Facteurb> <Termb2> | ε
<Facteurb> -> <Relation> | parouv <Expr> parferm | si <Expr> alors
<Expr> sinon <Expr> fsi
<Relation> -> ident <Op> ident,
<Op> -> "=" | "<" | ">" | ">=" | "<="
```

2 Equation sur les règles

$$\begin{aligned} premier(Super) &= premier(Expr\ eof) \\ &= premier(Expr) \\ &= \{ "(", "si", "ident" \} \end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Expr}) &= \text{premier}(\text{Termb Expr2}) \\
&= \text{premier}(\text{Termb}) \\
&= \{ "(", "si", "ident" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Expr2}) &= \text{premier}(\text{"ou" Termb Expr2}) \cup \{ \epsilon \} \\
&= \{ "ou" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Termb}) &= \text{premier}(\text{Facteurb Termb2}) \\
&= \text{premier}(\text{Facteurb}) \\
&= \{ "(", "si", "ident" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Termb2}) &= \text{premier}(\text{"et" Facteurb Termb2}) \cup \{ \epsilon \} \\
&= \{ "et" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Facteurb}) &= \text{premier}(\text{Relation}) \cup \text{premier}(\text{"parouv" Expr "parferm"}) \\
&\cup \text{premier}(\text{"si" Expr "alors" Expr "sinon" Expr "fsi"}) \\
&= \text{premier}(\text{Relation}) \cup \{ "parouv" \} \cup \{ "si" \} \\
&= \{ "(", "si", "ident" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Relation}) &= \text{premier}(\text{"ident" Op "ident"}) \\
&= \{ "ident" \}
\end{aligned}$$

$$\begin{aligned}
\text{premier}(\text{Op}) &= \text{premier}(\text{" = "}) \cup \text{premier}(\text{" <> "}) \cup \text{premier}(\text{" < "}) \cup \text{premier}(\text{" > "}) \\
&\cup \text{premier}(\text{" >= "}) \cup \text{premier}(\text{" <= "}) \\
&= \{ "=", "<>", "<", ">", ">=", "<=" \}
\end{aligned}$$

$$\text{suivant}(\text{super}) = \emptyset$$

$$\begin{aligned} \text{suivant}(\text{Expr}) &= \{ \text{"eof"} \} \cup \{ \text{"par ferm"} \} \cup \{ \text{"alors"} \} \cup \{ \text{"sinon"} \} \cup \{ \text{"fsi"} \} \\ &= \{ \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\begin{aligned} \text{suivant}(\text{Expr2}) &= \text{suivant}(\text{Expr}) \cup \text{suivant}(\text{Expr2}) \cup \{ \epsilon \} \\ &= \text{suivant}(\text{Expr}) \\ &= \{ \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\begin{aligned} \text{suivant}(\text{Termb}) &= \text{premier}(\text{Expr2}) \cup \text{suivant}(\text{Expr2}) \cup \{ \epsilon \} \\ &= \text{premier}(\text{Expr2}) \cup \text{suivant}(\text{Expr2}) \\ &= \{ \text{"ou"}, \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\begin{aligned} \text{suivant}(\text{Termb2}) &= \text{premier}(\text{Termb}) \cup \text{suivant}(\text{Termb2}) \cup \{ \epsilon \} \\ &= \text{premier}(\text{Termb}) \\ &= \{ \text{"ou"}, \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\begin{aligned} \text{suivant}(\text{Facteurb}) &= \text{premier}(\text{Termb2}) \cup \text{suivant}(\text{Termb2}) \cup \{ \epsilon \} \\ &= \text{premier}(\text{Expr2}) \cup \text{suivant}(\text{Expr2}) \\ &= \{ \text{"et"}, \text{"ou"}, \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\begin{aligned} \text{suivant}(\text{Relation}) &= \text{suivant}(\text{Facteurb}) \\ &= \{ \text{"et"}, \text{"ou"}, \text{eof, "}, \text{"alors"}, \text{"sinon"}, \text{"fsi"} \} \end{aligned}$$

$$\text{suivant}(\text{Op}) = \{ \text{"ident"} \}$$

$null(Expr) = false$
 $null(Expr) = false$
 $null(Expr2) = true$
 $null(Termb) = false$
 $null(Termb2) = true$
 $null(Facteurb) = false$
 $null(Relation) = false$
 $null(Op) = false$

3 Vérification grammaire LL1

Super:

Pas d'intersection
 $null(Super) = false$

Expr:

Pas d'intersection
 $null(Expr) = false$

Expr2:

$premier("ou") \cap premier(\emptyset) = \emptyset$
 $null(Expr2) = true$
 $premier(Expr2) \cap suivant(Expr2) = \emptyset$

Termb:

Pas d'intersection
 $null(Super) = false$

Termb2:

$premier("et") \cap premier(\emptyset) = \emptyset$
 $null(Termb2) = true$
 $premier(Termb2) \cap suivant(Termb2) = \emptyset$

Facteurb:

$premier(Relation) \cap premier("parouv") \cap premier("si") = \emptyset$
 $null(Facteurb) = false$

Relation:

Pas d'intersection
 $null(Relation) = false$

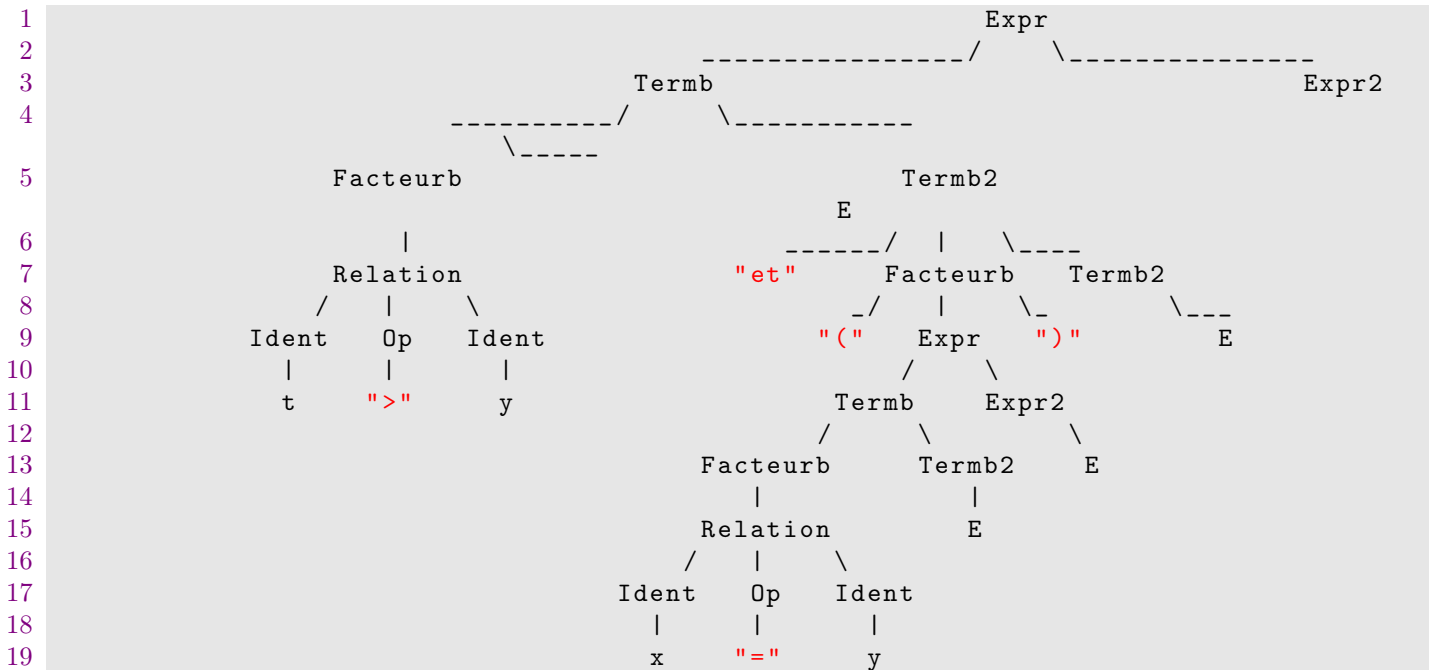
Op:

$\text{premier("=")} \cap \text{premier("<>")} \cap \text{premier("<")} \cap \text{premier(">")} \cap$
 $\text{premier(">=")} \cap \text{premier("<=")} = \emptyset$
 $\text{null(Op)} = \text{false}$

Toutes les règles respectent bien les vérifications. C'est une grammaire LL1.

4 Arbre concret

Listing 1: Arbre concret sur l'exemple "t > y et x



5 Jeux de tests

5.1 Non-terminaux inaccessibles

Test (sans Relation -> ident Op ident) : non-terminaux inaccessibles.

Dans ce cas, nous avons mis la règle "Relation" à vide pour empêcher d'atteindre le terminaux de la règle "Op".

Listing 2: Non-terminaux inaccessibles

```

1 ({parouv, parferm, egal, diff, inf, sup, et, ou, si, alors, sinon, fsi, supeg,
   infeg, ident}, {Super, Expr, Expr2, Termb, Termb2, Facteurb, Relation, Op},
   Super,
2 {

```

```

3   Super  -> Expr ,
4   Expr   -> Termb Expr2 ,
5   Expr2  -> ou Termb Expr2 ,
6   Expr2  -> ,
7   Termb  -> Facteurb Termb2 ,
8   Termb2 -> et Facteurb Termb2 ,
9   Termb2 -> ,
10  Facteurb -> Relation ,
11  Facteurb -> parouv Expr parferm ,
12  Facteurb -> si Expr alors Expr sinon Expr fsi ,
13  Relation -> ,
14  Op -> egal ,
15  Op -> diff ,
16  Op -> inf ,
17  Op -> sup ,
18  Op -> supeg ,
19  Op -> infeg
20  } )
21
22 Non terminal Op is not accessible

```

5.2 Non-terminaux non productifs

Test (sans Expr2 ->) : non-terminaux qui ne sont pas productifs.

Dans ce cas, nous avons retiré la partie vide de la règle "Expr2" afin de boucler sur la clause "ou" de la grammaire et ainsi obliger celle-ci à être non-terminal.

Listing 3: Non-terminaux non productifs

```

1  ({parouv, parferm, egal, diff, inf, sup, et, ou, si, alors, sinon, fsi, supeg,
   infeg, ident}, {Super, Expr, Expr2, Termb, Termb2, Facteurb, Relation, Op},
   Super,
2   {
3   Super  -> Expr ,
4   Expr   -> Termb Expr2 ,
5   Expr2  -> ou Termb Expr2 ,
6   Termb  -> Facteurb Termb2 ,
7   Termb2 -> et Facteurb Termb2 ,
8   Termb2 -> ,
9   Facteurb -> Relation ,
10  Facteurb -> parouv Expr parferm ,
11  Facteurb -> si Expr alors Expr sinon Expr fsi ,
12  Relation -> ident Op ident ,
13  Op -> egal ,
14  Op -> diff ,
15  Op -> inf ,
16  Op -> sup ,

```

```

17 Op -> supeg,
18 Op -> infeg
19 } )
20
21
22 Non terminal Expr is not generative
23 Non terminal Expr2 is not generative
24 Non terminal Super is not generative

```

5.3 Grammaire non LL1

Test (Expr ancienne grammaire) : grammaire non LL1.

Dans ce cas, nous avons réutiliser la première grammaire sur la règle "Expr" afin de placer la dérivation à gauche et non à droite comme convenue pour qu'une grammaire soit LL1.

Listing 4: Grammaire non LL1

```

1 ({parouv, parferm, egal, diff, inf, sup, et, ou, si, alors, sinon, fsi, supeg,
  infeg, ident}, {Super, Expr, Termb, Termb2, Facteurb, Relation, Op}, Super,
2   {
3   Super -> Expr,
4   Expr -> Expr ou Termb,
5   Expr -> Termb,
6   Termb -> Facteurb Termb2,
7   Termb2 -> et Facteurb Termb2,
8   Termb2 ->,
9   Facteurb -> Relation,
10  Facteurb -> parouv Expr parferm,
11  Facteurb -> si Expr alors Expr sinon Expr fsi,
12  Relation -> ident Op ident,
13  Op -> egal,
14  Op -> diff,
15  Op -> inf,
16  Op -> sup,
17  Op -> supeg,
18  Op -> infeg
19  } )
20
21 Grammar is well-formed
22
23 Null Expr = false
24 Null Facteurb = false
25 Null Op = false
26 Null Relation = false
27 Null Super = false
28 Null Termb = false
29 Null Termb2 = true

```

```

30
31 First Expr = { ident parouv si }
32 First Facteurb = { ident parouv si }
33 First Op = { diff egal inf infeg sup supeg }
34 First Relation = { ident }
35 First Super = { ident parouv si }
36 First Termb = { ident parouv si }
37 First Termb2 = { et }
38
39 Follow Expr = { # alors fsi ou parferm sinon }
40 Follow Facteurb = { # alors et fsi ou parferm sinon }
41 Follow Op = { ident }
42 Follow Relation = { # alors et fsi ou parferm sinon }
43 Follow Super = { # }
44 Follow Termb = { # alors fsi ou parferm sinon }
45 Follow Termb2 = { # alors fsi ou parferm sinon }
46
47 The grammar is NOT LL1

```

5.4 Grammaire LL1

Cas final du TP.

Listing 5: Grammaire LL1

```

1 ({parouv, parferm, egal, diff, inf, sup, et, ou, si, alors, sinon, fsi, supeg,
   infeg, ident}, {Super, Expr, Expr2, Termb, Termb2, Facteurb, Relation, Op},
   Super,
2   {
3     Super -> Expr,
4     Expr -> Termb Expr2,
5     Expr2 -> ou Termb Expr2,
6     Expr2 ->,
7     Termb -> Facteurb Termb2,
8     Termb2 -> et Facteurb Termb2,
9     Termb2 ->,
10    Facteurb -> Relation,
11    Facteurb -> parouv Expr parferm,
12    Facteurb -> si Expr alors Expr sinon Expr fsi,
13    Relation -> ident Op ident,
14    Op -> egal,
15    Op -> diff,
16    Op -> inf,
17    Op -> sup,
18    Op -> supeg,
19    Op -> infeg
20  } )
21
22

```



```

23 Grammar is well-formed
24
25 Null Expr = false
26 Null Expr2 = true
27 Null Facteurb = false
28 Null Op = false
29 Null Relation = false
30 Null Super = false
31 Null Termb = false
32 Null Termb2 = true
33
34 First Expr = { ident parouv si }
35 First Expr2 = { ou }
36 First Facteurb = { ident parouv si }
37 First Op = { diff egal inf infeg sup supeg }
38 First Relation = { ident }
39 First Super = { ident parouv si }
40 First Termb = { ident parouv si }
41 First Termb2 = { et }
42
43 Follow Expr = { # alors fsi parferm sinon }
44 Follow Expr2 = { # alors fsi parferm sinon }
45 Follow Facteurb = { # alors et fsi ou parferm sinon }
46 Follow Op = { ident }
47 Follow Relation = { # alors et fsi ou parferm sinon }
48 Follow Super = { # }
49 Follow Termb = { # alors fsi ou parferm sinon }
50 Follow Termb2 = { # alors fsi ou parferm sinon }
51
52 The grammar is LL1

```

5.5 Test de mots reconnus

Listing 6: $a = b$

```

1 Super---Expr-+-Termb-+-Facteurb---Relation-+-UL_IDENTa
2           |           |                               |-Op---UL_EGAL
3           |           |                               '-UL_IDENTb
4           |           '-Termb2
5           '-Expr2

```

Listing 7: $a = b$ et $a <> c$

```

1 Super---Expr-+-Termb-+-Facteurb---Relation-+-UL_IDENTa
2           |           |                               |-Op---UL_EGAL
3           |           |                               '-UL_IDENTb
4           |           '-Termb2-+-UL_ET
5           |                               |-Facteurb---Relation-+-UL_IDENTa

```

```

6      |                               |                               | -Op---UL_DIFF
7      |                               |                               ' -UL_IDENTc
8      |                               ' -Termb2
9      ' -Expr2

```

Listing 8: si $a = b$ alors $a = b$ sinon $a = c$ fsi

```

1 Super --- Expr +- Termb +- Facteurb +- UL_SI
2      |           |           | -Expr +- Termb +- Facteurb --- Relation +- UL_IDENTa
3      |           |           |
4      |           | -Op---UL_EGAL
5      |           |           |           |           ' -UL_IDENTb
6      |           |           |           |           ' -Termb2
7      |           |           |           ' -Expr2
8      |           |           | -UL_ALORS
9      |           |           | -Expr +- Termb +- Facteurb --- Relation +- UL_IDENTa
10     |           |           |           |           |
11     |           |           |           |           |           ' -UL_IDENTb
12     |           |           |           |           |           ' -Termb2
13     |           |           |           |           ' -Expr2
14     |           |           | -UL_SINON
15     |           |           | -Expr +- Termb +- Facteurb --- Relation +- UL_IDENTa
16     |           |           |           |           |
17     |           |           |           |           |           ' -UL_IDENTc
18     |           |           |           |           |           ' -Termb2
19     |           |           |           |           ' -Expr2
20     |           |           | -UL_FSI
21     |           ' -Termb2
22     ' -Expr2

```

5.6 Test de mots non reconnus

a et b

Failure("No derivation")

a(b) = c

Failure("No derivation")