

# TP8 - Traitement automatique des langues

PAUL CHAIGNON - CLÉMENT GAUTRAIS

December 4, 2013

## 1 Question 1

Listing 1: talq1.pro

```
1 /**
2 TP 8 Traitement Automatique de la Langue (TAL) - Prolog
3 @author Paul CHAIGNON
4 @author Clement GAUTRAIS
5 @version Annee scolaire 2013/2014
6 */
7
8 article(le).
9 article(les).
10 article(la).
11 article(un).
12
13 nom_commun(chien).
14 nom_commun(enfants).
15 nom_commun(steack).
16 nom_commun(pull).
17 nom_commun(rue).
18 nom_commun(femme).
19
20 verbe(aboie).
21 verbe(jouent).
22 verbe(marche).
23 verbe(porte).
24 verbe(mange).
25 verbe(dort).
26
27 nom_propre(paul).
28
29 preposition(dans).
30
31 pronom(qui).
32
33 adjectif(noir).
34 adjectif(petit).
35
36 /*
37 =====
38 =====
39 DÃ©finition des prÃ©dicats
40 =====
```

```

41 */
42 /**
43  * Les propositions subordonnees relatives ont ete sorties du groupe
    nominal pour simplifier le code.
44 */
45 phrase_simple(List, Rest):-
46     gn(List, Rest1),
47     gv(Rest1, Rest).
48
49 gn(List, Rest):-
50     gn_simple(List, Rest).
51 gn(List, Rest):-
52     gn_simple(List, Rest1),
53     prop_relative(Rest1, Rest).
54
55 gn_simple([Elem|List], List):-
56     nom_propre(Elem).
57 gn_simple([First, Second|List], List):-
58     article(First),
59     nom_commun(Second).
60 gn_simple([First, Second, Third|List], List):-
61     article(First),
62     nom_commun(Second),
63     adjectif(Third).
64 gn_simple([First, Second, Third|List], List):-
65     article(First),
66     adjectif(Second),
67     nom_commun(Third).
68 gn_simple([First, Second, Third, Fourth|List], List):-
69     article(First),
70     adjectif(Second),
71     nom_commun(Third),
72     adjectif(Fourth).
73
74 gv([Elem|List], List):-
75     verbe(Elem).
76 gv([First|List], Rest):-
77     verbe(First),
78     gn(List, Rest).
79 gv([First|List], Rest):-
80     verbe(First),
81     gn(List, Rest1),
82     gp(Rest1, Rest).
83 gv([First|List], Rest):-
84     verbe(First),
85     gp(List, Rest).
86
87 gp([First|List], Rest):-
88     preposition(First),
89     gn(List, Rest).
90
91 prop_relative([First|List], Rest):-
92     pronom(First),
93     gv(List, Rest).
94
95 analyse(List):-
96     phrase_simple(List, []).
97

```

```

98 /*
99 =====
100 =====
101 Tests
102 =====
103 */
104
105 % Quelques phrases de test Ã copier coller pour vous faire gagner du
    temps, mais
106 % n'hÃ©sitez pas Ã en dÃ©finir d'autres
107
108 /*
109 analyse([le, chien, aboie]).
110 analyse([les, enfants, jouent]).
111 analyse([paul, marche, dans, la, rue]).
112 analyse([la, femme, qui, porte, un, pull, noir, mange, un, steack]).
113 analyse([les, chien, aboie]).
114 analyse([la, femme, qui, porte, un, pull, noir, mange, un, chien]).
115 analyse([la, petit, chien, noir, qui, dort, mange]).
116 analyse(Sentence).
117 */

```

## 2 Question 2

Listing 2: talq2.pro

```

1 /**
2 TP 8 Traitement Automatique de la Langue (TAL) - Prolog
3 @author Paul CHAIGNON
4 @author Clement GAUTRAIS
5 @version Annee scolaire 2013/2014
6 */
7
8 article(le).
9 article(les).
10 article(la).
11 article(un).
12
13 nom_commun(chien).
14 nom_commun(enfants).
15 nom_commun(steack).
16 nom_commun(pull).
17 nom_commun(rue).
18 nom_commun(femme).
19
20 verbe(aboie).
21 verbe(jouent).
22 verbe(marche).
23 verbe(porte).
24 verbe(mange).
25 verbe(dort).
26
27 nom_propre(paul).

```

```

28
29 preposition(dans).
30
31 pronom(qui).
32
33 adjectif(noir).
34 adjectif(petit).
35
36 /*
37 =====
38 =====
39 Définition des prédicats
40 =====
41 */
42 phrase_simple(List, Rest, phr(GnTree, GvTree)):-
43     gn(List, Rest1, GnTree),
44     gv(Rest1, Rest, GvTree).
45
46 gn(List, Rest, GnTree):-
47     gn_simple(List, Rest, GnTree).
48 gn(List, Rest, gn(GnTree, PropTree)):-
49     gn_simple(List, Rest1, GnTree),
50     prop_relative(Rest1, Rest, PropTree).
51
52 gn_simple([First|List], List, gn(nom_propre(First))):-
53     nom_propre(First).
54 gn_simple([First, Second|List], List, gn(art(First), nom_com(Second))):-
55     article(First),
56     nom_commun(Second).
57 gn_simple([First, Second, Third|List], List, gn(art(First),
58     nom_com(Second), adj(Third))):-
59     article(First),
60     nom_commun(Second),
61     adjectif(Third).
62 gn_simple([First, Second, Third|List], List, gn(art(First),
63     adj(Second), nom_com(Third))):-
64     article(First),
65     adjectif(Second),
66     nom_commun(Third).
67 gn_simple([First, Second, Third, Fourth|List], List, gn(art(First),
68     adj(Second), nom_com(Third), adj(Fourth))):-
69     article(First),
70     adjectif(Second),
71     nom_commun(Third),
72     adjectif(Fourth).
73
74 gv([First|List], List, gv(verbe(First))):-
75     verbe(First).
76 gv([First|List], Rest, gv(verbe(First), GnTree)):-
77     verbe(First),
78     gn(List, Rest1, GnTree).
79 gv([First|List], Rest, gv(verbe(First), GnTree, GpTree)):-
80     verbe(First),
81     gn(List, Rest1, GnTree),
82     gp(Rest1, Rest, GpTree).

```

```

83
84 gp([First|List], Rest, gp_prep(prepare(First), GnTree)):-
85     preposition(First),
86     gn(List, Rest, GnTree).
87
88 prop_relative([First|List], Rest, rel(pronom(First), GvTree)):-
89     pronom(First),
90     gv(List, Rest, GvTree).
91
92 analyse(List, Tree):-
93     phrase_simple(List, [], Tree).
94
95 /*
96 =====
97 =====
98 Tests
99 =====
100
101 Les arbres ne sont pas identiques a l'enonce.
102 Les groupes nominaux contenant des propositions subordonnees relatives
103     ont ete modifies afin de simplifier le code.
104 */
105
106 /*
107 analyse([le, chien, aboie], Tree).
108     Tree = phr(gn(art(le), nom_com(chien)), gv(verbe(aboie)))
109 analyse([paul, qui, porte, un, pull, noir, mange, un, steack], Tree).
110     Tree = phr(gn(nom_prop(paul), rel(pronom(qui), gv(verbe(porte),
111         gn(art(un), nom_com(pull), adj(noir)))))), gv(verbe(mange),
112         gn(art(un), nom_com(steack))))
113 analyse([la, petit, chien, noir, qui, dort, mange], Tree).
114     Tree = phr(gn(gn(art(la), adj(petit), nom_com(chien), adj(noir)),
115         rel(pronom(qui), gv(verbe(dort)))), gv(verbe(mange)))
116 */

```

### 3 Question 3

Listing 3: talq3.pro

```

1 /**
2 TP 8 Traitement Automatique de la Langue (TAL) - Prolog
3 @author Paul CHAIGNON
4 @author Clement GAUTRAIS
5 @version Annee scolaire 2013/2014
6 */
7
8 article(m, s, le).
9 article(m, s, un).
10 article(f, s, la).
11 article(_, p, les).
12
13 nom_commun(m, s, chien).

```

```

14 nom_commun(m, s, steack).
15 nom_commun(m, s, pull).
16 nom_commun(f, s, rue).
17 nom_commun(f, s, femme).
18 nom_commun(m, p, enfants).
19
20 verbe(s, aboie).
21 verbe(s, marche).
22 verbe(s, porte).
23 verbe(s, mange).
24 verbe(p, mangent).
25 verbe(s, dort).
26 verbe(p, jouent).
27 verbe(s, joue).
28 verbe(p, dorment).
29
30 nom_propre(m, s, paul).
31
32 preposition(dans).
33
34 pronom(qui).
35
36 adjectif(m, s, noir).
37 adjectif(m, s, petit).
38 adjectif(f, s, petite).
39 adjectif(m, p, petits).
40
41 /*
42 =====
43 =====
44 Définition des prédicats
45 =====
46 */
47 phrase_simple(List, Rest, phr(GnTree, GvTree)):-
48     gn(Number, List, Rest1, GnTree),
49     gv(Number, Rest1, Rest, GvTree).
50
51 gn(Number, List, Rest, GnTree):-
52     gn_simple(Number, List, Rest, GnTree).
53 gn(Number, List, Rest, gn(GnTree, PropTree)):-
54     gn_simple(Number, List, Rest1, GnTree),
55     prop_relative(Number, Rest1, Rest, PropTree).
56
57 gn_simple(Number, [First|List], List, gn(nom_propre(First))):-
58     nom_propre(_, Number, First).
59 gn_simple(Number, [First, Second|List], List, gn(art(First),
60     nom_com(Second))):-
61     article(Gender, Number, First),
62     nom_commun(Gender, Number, Second).
63 gn_simple(Number, [First, Second, Third|List], List, gn(art(First),
64     nom_com(Second), adj(Third))):-
65     article(Gender, Number, First),
66     nom_commun(Gender, Number, Second),
67     adjectif(Gender, Number, Third).
68 gn_simple(Number, [First, Second, Third|List], List, gn(art(First),
69     adj(Second), nom_com(Third))):-
70     article(Gender, Number, First),
71     adjectif(Gender, Number, Second),

```

```

69     nom_commun(Gender, Number, Third).
70 gn_simple(Number, [First, Second, Third, Fourth|List], List,
71     gn(art(First), adj(Second), nom_com(Third), adj(Fourth))):-
72     article(Gender, Number, First),
73     adjectif(Gender, Number, Second),
74     nom_commun(Gender, Number, Third),
75     adjectif(Gender, Number, Fourth).
76 gv(Number, [First|List], List, gv(verbe(First))):-
77     verbe(Number, First).
78 gv(Number, [First|List], Rest, gv(verbe(First), GnTree)):-
79     verbe(Number, First),
80     gn(_, List, Rest, GnTree).
81 gv(Number, [First|List], Rest, gv(verbe(First), GnTree, GpTree)):-
82     verbe(Number, First),
83     gn(_, List, Rest1, GnTree),
84     gp(Rest1, Rest, GpTree).
85 gv(Number, [First|List], Rest, gv(verbe(First), GpTree)):-
86     verbe(Number, First),
87     gp(List, Rest, GpTree).
88
89 gp([First|List], Rest, gp_prep(preposition(First), GnTree)):-
90     preposition(First),
91     gn(_, List, Rest, GnTree).
92
93 prop_relative(Number, [First|List], Rest, rel(pronom(First), GvTree)):-
94     pronom(First),
95     gv(Number, List, Rest, GvTree).
96
97 analyse(List, Tree):-
98     phrase_simple(List, [], Tree).
99
100 /*
101 =====
102 =====
103 Tests
104 =====
105 */
106
107 /*
108 analyse([les, chien, aboie], Tree).
109     No
110 analyse([paul, qui, jouent, mange, un, steack], Tree).
111     No
112 analyse([la, petit, chien, noir, qui, dort, mange], Tree).
113     No
114 analyse([les, enfants, qui, dorment, mangent], Tree).
115     Yes
116 */

```