Td n° 15

Grammaires non contextuelles (2)

Hugo Salou MPI*

Dernière mise à jour le 14 mars 2023

1 Forme normale conjonctive

1. On pose la grammaire $\mathcal{G}$ de symbole initial U et ayant pour règles de production

$$
\begin{aligned}
\mathrm{U} &\to \mathrm{S} \mid \varepsilon, \\
\mathrm{S} &\to \mathrm{S}\&\mathrm{X} \mid \mathrm{X}, \\
\mathrm{X} &\to \texttt{-}\mathrm{V} \mid \mathrm{V} \mid \mathrm{V}\texttt{|}\mathrm{X} \mid \texttt{-}\mathrm{V}\texttt{|}\mathrm{X}, \\
\mathrm{V} &\to \mathtt{p} \mid \mathtt{q} \mid \cdots .
\end{aligned}
$$

2.

```
'p' ⇝ la variable p
('p', true) ⇝ le littéral p
('p', false) ⇝ le littéral ¬p
[ ('p', true); ('q', true) ] ⇝ la clause p ∨ q
[[ ('p', true); ('q', true) ]] ⇝ la formule p ∨ q
[[('p', true)]; [('q', true)]]] ⇝ la formule p ∧ q
[[('p', true)]; [('q', true); ('p', false)]]] ⇝ la formule p ∧ (q ∨ ¬p)
```

Code 1 – Expressions OCaml

3.

```
let rec parse_f (s: string) (start: int) (len: int): fnc =
  if String.length s = 0 then []
  else parse_d s start len
and parse_d (s: string) (start: int) (len: int): fnc =
  let i = ref start in
  while s.[!i] != '&' && !i < len do incr i; done;
  if !i = len then [parse_c s start len]
  else (parse_c s start (!i - 1)) :: (parse_d s (!i+1) len)
and parse_c (s: string) (start: int) (len: int): cls =
  let i = ref start in
  while s.[!i] != '|' && !i < len do incr i; done;
  if !i = len then [parse_l s start len]
  else (parse_l s start (!i - 1)) :: (parse_c s (!i+1) len)
and parse_l (s: string) (start: int) (len: int): lit =
  if s.[start] = '-' then (parse_v s (start + 1) len, false)
  else (parse_v s start len, true)
and parse_v (s: string) (start: int) (len: int): char =
  let x = s.[start] in
  assert(Char.code x >= 97 && Char.code x <= 122);
  assert(len - start = 1);
  x
```

Code 2 – Parsing des fonctions OCaml

2 Réduction de grammaire et systèmes de conséquences

2.1 Digression OCaml

3 Grammaires propres

4 Un lemme d'itération

5 Les langages réguliers sont non contextuels

5.1 Avec des automates

1. On pose P l'ensemble des règles de productions définies comme

$$\{X_q \to \ell X_{q'} \mid (q, \ell, q') \in \delta\} \cup \{X_q \to \varepsilon \mid q \in F\}.$$

Montrons par récurrence $\mathscr{P}(n)$:

$$« \forall q \in \mathbb{Q},\ \mathscr{L}_n(\mathscr{A}_q) = \{w \in \mathscr{L} \mid |w| = n\} = \{w \in \Sigma^n \mid X_q \overset{\star}{\Rightarrow} w\} = G_n(q). »$$

— Pour $n = 0$, soit $q \in Q$ et soit $w \in \Sigma^\star$. Si $w \in \mathscr{L}_n(\mathscr{A}_q)$, alors $w = \varepsilon$ et $q \in F$, d'où $(X_q \to \varepsilon) \in P$ donc $X_q \overset{\star}{\Rightarrow} w$ donc $w \in G_n(q)$. Réciproquement, si $w \in G_n(q)$, alors $X_q \overset{\star}{\Rightarrow} \varepsilon$ car il n'y a pas d'ε-transitions, donc $q \in F$ et donc $w = \varepsilon \in \mathscr{L}_n(\mathscr{A}_q)$.

— Soit $n \in \mathbb{N}$. On suppose $\mathscr{P}(n)$ vraie. Soit $q \in Q$ et soit $w \in \Sigma^{n+1}$. Si $w \in \mathscr{L}_{n+1}(\mathscr{A}_q)$, alors il existe une exécution acceptante

$$q \xrightarrow{w_1} q_1 \to \cdots \to q_n \xrightarrow{w_{n+1}} q_{n+1} \in F.$$

D'où, $w_2 \dots w_{n+1} \in \mathscr{L}_n(\mathscr{A}_{q_1}) = G_n(q_1)$ par hypothèse donc $X_{q_1} \overset{\star}{\Rightarrow} w_2 \dots w_{n+1}$. Or, $(q, w_1, q_1) \in \delta$ donc $(X_q \to w_n X_{q_1}) \in P$ et donc $X_q \Rightarrow w_1 X_{q_1} \overset{\star}{\Rightarrow} w$. D'où, $X_q \overset{\star}{\Rightarrow} w$ et donc $w \in G_{n+1}(q)$.

Réciproquement, si $w \in G_{n+1}(q)$ alors $X_q \overset{\star}{\Rightarrow} w$. Soit $w' \in Q$ tel que $X_q \Rightarrow w_1 X_{q'}$. Alors, $(q, w_1, q') \in \delta$. De plus, $|w_2 \dots w_n| = n$ et $X_{q'} \overset{\star}{\Rightarrow} w_2 \dots w_{n+1}$ donc $w_2 \dots w_{n+1} \in G_n(q') = \mathscr{L}_n(\mathscr{A}_{q'})$. Il existe donc une exécution acceptante

$$q' \xrightarrow{w_2} q_2 \to \cdots \to q_n \xrightarrow{w_{n+1}} q_{n+1} \in F.$$

Or, $(q, w_1, q') \in \delta$ d'où

$$q \xrightarrow{w_1} q' \xrightarrow{w_2} q_2 \to \cdots \to q_n \xrightarrow{w_{n+1}} q_{n+1} \in F$$

est une exécution acceptante de $\mathscr{A}_q$. On en déduit que $w \in \mathscr{L}_{n+1}(\mathscr{A}_q)$.

2. On en conclut que tout langage régulier est représentable par une grammaire non-contextuelle.

5.2 Avec des expressions régulières

3. On pose P l'ensemble de règles de productions défini comme

$$\begin{aligned} P = \quad & \{X_r \to X_r X_{r'} \mid \varepsilon \text{ tel que } r = (r')^\star\} \\ & \cup \{X_r \to X_{r_1} \mid X_{r_2} \text{ tel que } r = r_1 \mid r_2\} \\ & \cup \{X_r \to X_{r_1} X_{r_2} \text{ tel que } r = r_1 \cdot r_2\} \\ & \cup \{X_r \to r \text{ tel que } r \in \Sigma\} \\ & \cup \{X_r \to \varepsilon \text{ tel que } r = \varepsilon\} \end{aligned}$$