Langages formels : grammaires

Stéphane Le Roux leroux@lsv.fr

ENS Paris-Saclay

2023-2024

Hiérarchie de Chomsky

4 types de grammaires de plus en plus restrictifs.

- Type 0, dit des grammaires générales : correspond aux machines de Turing.
- Type 1, grammaires contextuelles : $NSPACE(n)$.
- Type 2, grammaires hors-contexte (ou algébriques) : automates à pile.
- Type 3, grammaires rationnelles : automates finis.

Type 0, grammaires générales

Définition : grammaire de type 0

$G = (\Sigma, V, \to, S)$ où

- Σ est l'alphabet terminal (généralement lettres miniscules)
- V est l'alphabet non terminal, dit des variables (généralement lettres majuscules). $V \cap \Sigma = \emptyset$.
- S est la variable initiale.
- $\to \subseteq (\Sigma \cup V)^* \times (\Sigma \cup V)^*$ est un ensemble fini de règles de production.

Définition : dérivation terminale et langage engendré

- Soient $u, v, v', w \in (\Sigma \cup V)^*$ tels que $v \to v'$. On note alors aussi $uvw \to uv'w$. Clôture de $\to$ par passage au contexte.
- Le langage engendré par G est $L_G := \{u \in \Sigma^* \mid S \to^+ u\}$.
  ($\to^+$ est la clôture transitive de $\to$.)
- Un langage engendré par une grammaire de type 0 est dit de type 0.

Exemple

$$1 : \quad S \to DXaF \qquad 3 : \quad XF \to YF \qquad 5 : \quad DY \to DX \qquad 7 : \quad aZ \to Za$$
$$2 : \quad Xa \to aaX \qquad 4 : \quad aY \to Ya \qquad 6 : \quad XF \to Z \qquad 8 : \quad DZ \to \epsilon$$

1. $Xa^n \to^* a^{2n}X$ (règle 2 et récurrence sur $n > 0$)
2. $a^nY \to^* Ya^n$ (règle 4 et récurrence sur $n > 0$)
3. $DXa^nF \to^* DXa^{2n}F$ (règle 3, lemmes ci-dessus, et règle 5)
4. $S \to DXa^{2^n}F$ pour tout $n \in \mathbb{N}$. (règle 1 et récurrence sur n par le lemme ci-dessus)
5. $a^nZ \to^* Za^n$ (règle 7 et récurrence sur n)
6. $DXa^nF \to^* a^{2n}$ (règle 6, deux lemmes ci-dessus, et règle 8)
7. $S \to^* a^{2^n}$ pour tout $n > 0$ (deux lemmes ci-dessus)
8. Montrons (semi-formellement) que toute dérivation est de ce type :
   - La règle 8 est la seule qui ne produit pas de symboles non terminaux.
   - Le Z est produit par la règle 6, et la grammaire devient déterministe.
   - D et F sont en bout de mot, sauf quand ils sont effacés.
   - Une décision est prise seulement pour Da^nXF, et n est puissance de 2 (par récurrence).

Caractérisation du type 0

Théorème (Chomsky, 1959)

Un langage (alphabet fini) est de type 0 ssi il est récursivement énumérable.

Par double implication. Soit $G = (\Sigma, V, \to, S)$ une grammaire. Mq $L_G := \{u \in \Sigma^* \mid S \to^+ u\}$ est r.e. en définissant une MT non-déterministe à deux bandes qui l'accepte. La bande de travail est initialisé à S. Puis on simule une dérivation de G sur cette bande.

1. On sélectionne une règle de production de manière non-déterministe
2. On sélectionne un endroit dans le contenu de la bande de travail.
3. Si la zone du contenu égale la partie gauche de la règle, on met le contenu à jour.
4. Si le contenu égale l'entrée, on accepte. Sinon, on répète.

Les mots acceptés sont donc exactement les mots dérivables.

Caractérisation du type 0 (II)

Réciproquement, soit $L \subseteq \Sigma^*$ accepté par une MTD $(Q, q_0, \Sigma, \delta, B, \$)$. On définit une grammaire d'alphabet terminal Σ. La grammaire "devine" un mot, simule l'effet de la MTD sur ce mot, puis si la MTD accepte, termine avec le mot initial. Pour cela, il faut en garder une copie.

- On devine un mot :
  $1 : S \to q_0 D \qquad 2 : D \to (a, a)D \quad \forall a \in \Sigma \qquad 3 : D \to E$
- On devine un espace suffisant sur la bande pour simuler la MTD.
  $4 : E \to (B, B)E \qquad 5 : E \to \epsilon.$
- On simule le calcul de la MTD sur la deuxième composante des couples.
  - Pour tout $\delta(q, X) = (q', Y, \mathit{droite})$ et $a \in \Sigma$, on pose $q(a, X) \to (a, Y)q'$.
  - Pour tout $\delta(q, X) = (q', Y, \mathit{gauche})$ et $a, b \in \Sigma$ et Z, on pose $(b, Z)q(a, X) \to q'(b, Z)(a, Y)$.
- Si la MTD accepte, on projette sur la première composante. Pour tout $q \in F$ et $a \in \Sigma$, on pose
  $(a, X)q \to qaq \qquad q(a, X) \to qaq \qquad q \to \epsilon$

Type 1, grammaires contextuelles

Définition : grammaire de type 1

Une grammaire $(\Sigma, V, \to, S)$ est dite contextuelle si toute règle $u \to v$ vérifie $|u| \leq |v|$, dite règle croissante. Un langage engendré par une grammaire de type 1 est dit de type 1 ou contextuel.

Remarque

Un langage de type 1 ne contient pas le mot vide. Certain.e.s auteur.trice.s exigent la croissance seulement pour les règles dont le membre gauche n'est pas S (et S n'apparaît jamais à droite).

Rappel de la grammaire de type 0 produisant $\{a^{2^n} \mid n > 0\}$.

1 :	$S \to DXaF$	3 :	$XF \to YF$	5 :	$DY \to DX$	7 :	$aZ \to Za$
2 :	$Xa \to aaX$	4 :	$aY \to Ya$	6 :	$XF \to Z$	8 :	$DZ \to \epsilon$

Les règles 6 et 8 ne sont pas croissantes.

Exemple

Proposition

Le langage $\{a^{2^n} \mid n > 0\}$ est contextuel.

1	$S \to aa$	3	$T \to aa$	5	$Xaa \to aaXa$	7	$XaaF \to aaaa$
2	$S \to XTF$	4	$T \to XT$	6	$XaF \to aaF$		

1. Le mot aa est produit par la règle 1.
2. Un mot $X^{n+1}aaF$ est produit par 2,4,3.
3. $Xa^{n+1}F \to^+ a^{2(n+1)}F$ (récurrence sur $n \in \mathbb{N}$, règles 5 et 6)
4. $Xa^{n+2}F \to^+ a^{2(n+2)}$ (récurrence sur $n \in \mathbb{N}$, règles 5 et 7)
5. $X^k a^{n+1}F \to^* a^{2^k(n+1)}F$ (récurrence sur $k \in \mathbb{N}$, ne pas fixer n, lemmes ci-dessus)
6. $X^k a^{n+2}F \to^* a^{2^k(n+2)}$ (récurrence sur $k \in \mathbb{N}$, lemmes ci-dessus)
7. $S \to^+ a^{2^{k+1}}$ (règle 2,4,3, puis lemme ci-dessus avec $n = 0$)

Grammaires en forme normale contextuelle

Définition : forme normale

Une grammaire $(\Sigma, V, \rightarrow, S)$ est en forme normale contextuelle si ses règles sont de la forme $uXw \rightarrow uvw$ avec $X \in V$ et $v \neq \epsilon$.

Remarques

- Une grammaire en forme normale contextuelle est donc contextuelle
- Les règles 5 et 7 de la grammaire ci-dessous ne sont pas en forme normale contextuelle.

1	$S \rightarrow aa$	3	$T \rightarrow aa$	5	$Xaa \rightarrow aaXa$	7	$XaaF \rightarrow aaaa$
2	$S \rightarrow XTF$	4	$T \rightarrow XT$	6	$XaF \rightarrow aaF$		

Théorème

Tout langage sans le mot vide est contextuel ssi il est engendré par une grammaire en forme normale contextuelle

$\Leftarrow$ par une remarque ci-dessus.

Type 1 et formes normales

Théorème

Tout langage sans le mot vide est contextuel ssi il est engendré par une grammaire en forme normale contextuelle.

Soit $(\Sigma, V, \to, S)$ une grammaire contextuelle/croissante n'engendrant pas le mot vide. On définit une grammaire $(\Sigma, V \uplus X_\Sigma, \rightsquigarrow, S)$ en forme normale qui engendre le même langage, où $X_\Sigma := \{X_a \mid a \in \Sigma\}$.

1. Pour tout $a \in \Sigma$, on pose $X_a \rightsquigarrow a$.
2. Pour tout $u \to v$, on pose $f(u) \to_0 f(v)$, où dans $f(u)$, chaque $a \in \Sigma$ de u a été remplacé par X_a.
3. Pour tout $A_1 \dots A_n \to_0 B_1 \dots B_{n+k}$, on pose $A_1 \dots A_n \rightsquigarrow B_1 A_2 \dots A_n \rightsquigarrow B_1 B_2 A_3 \dots A_n \rightsquigarrow \dots \rightsquigarrow B_1 \dots B_{n-1} A_n \rightsquigarrow B_1 \dots B_{n+k}$

Caractérisation du type 1

Théorème

Un langage sans le mot vide est de type 1 ssi il est dans $NSPACE(n)$.

Par double implication. Soit $G = (\Sigma, V, \rightarrow, S)$ une grammaire croissante. On modifie la MT non-déterministe de la caractérisation "type 0 ssi r.e." en se restreignant à un espace de travail de même longueur que l'entrée. Cette restriction ne change pas le langage accepté, car la grammaire est croissante. Réciproquement, soit $L \subseteq \Sigma^*$ accepté par une MT non-déterministe $(Q, q_0, \Sigma, \delta, B, \$)$ en espace identique à l'entrée. On rappelle d'abord la grammaire de "type 0 ssi r.e." :

1. $S \rightarrow q_0 D \qquad D \rightarrow (a, a)D \quad \forall a \in \Sigma \qquad D \rightarrow E$
2. $E \rightarrow (B, B)E \qquad E \rightarrow \epsilon.$
3. $q(a, X) \rightarrow (a, Y)q'$ pour tout $\delta(q, X) = (q', Y, droite)$ et $a \in \Sigma$.
4. $(b, Z)q(a, X) \rightarrow q'(b, Z)(a, Y)$ pour tout $\delta(q, X) = (q', Y, gauche)$ et $a \in \Sigma$.
5. $(a, X)q \rightarrow qaq \qquad q(a, X) \rightarrow qaq \qquad q \rightarrow \epsilon$ pour tout $q \in F$ et $a \in \Sigma$.

Caractérisation du type 1 (II)

1. $S \to q_0 D \qquad D \to (a,a)D \quad \forall a \in \Sigma \qquad D \to E$
2. $E \to (B,B)E \qquad E \to \epsilon.$
3. $q(a,X) \to (a,Y)q'$ pour tout $\delta(q,X) = (q', Y, \mathit{droite})$ et $a \in \Sigma$.
4. $(b,Z)q(a,X) \to q'(b,Z)(a,Y)$ pour tout $\delta(q,X) = (q', Y, \mathit{gauche})$.
5. $(a,X)q \to qaq \qquad q(a,X) \to qaq \qquad q \to \epsilon$ pour tout $q \in F$.

On suppose pour simplifier que la MT n'utilise que la bande d'entrée.
Grammaire modifiée (de symboles terminaux Σ) :

1. $S \to (a, q_0 a)D \quad S \to (a, q_0 a F) \quad D \to (a,a)D \quad D \to (a, aF)$
2. Pour tout $\delta(q,X) = (q', Y, \mathit{droite})$ et $a \in \Sigma$, on pose $(a, qX)(b,Z) \to (a,Y)(b, q'Z)$.
3. Pour tout $\delta(q,X) = (q', Y, \mathit{gauche})$ et $a \in \Sigma$, on pose $(b,Z)(a,qX) \to (b,q'Z)(a,Y)$ et $(b,Z)(a,qXF) \to (b,q'Z)(a,YF)$
4. Pour tout $q \in F$ et $a, b \in \Sigma$ et X, Y, on pose $(a,qX) \to a$ et $(a,qX)(b,Y) \to (a,qX)(b,qY)$ et $(b,Y)(a,qX) \to (b,qY)(a,qX)$

Type 1 et décidabilité

Théorème

Le problème suivant est indécidable :

- Entrée : une grammaire de type 1.
- Sortie : le langage engendré est-il vide ?

Théorème (Kuroda 1964 pour les grammaires déterministe)

Le problème suivant est PSPACE-complet :

- Entrée : une grammaire de type 1 et un mot.
- Sortie : ce mot est-il engendré par la grammaire ?

Type 2, grammaires hors contexte

Définition

Une grammaire $(\Sigma, V, \to, S)$ est dite de type 2 ou hors contexte ou algébrique si $u \to v$ implique $u \in V$, i.e. chaque membre gauche est une variable, i.e. $\to \subseteq V \times (V \uplus \Sigma)^*$.
Un langage est de type 2 ou hors context ou algébrique s'il est engendré par une grammaire de type 2.

Exemples

- Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ est algébrique. $S \to \epsilon$ et $S \to aSb$
- La language des expressions bien parenthésées est algébrique. Cf TD.

Les grammaires de type 2 sont presque toutes de type 1, mais certaines de leurs règles ne sont pas croissantes.

Grammaires hors contexte, lemme fondamental

Lemme

Soit une grammaire algébrique $(\Sigma, V, \to, S)$ et u, v, w et $n \in \mathbb{N}$ tels que $uv \to^n w$. Alors il existe $w_1, w_2 \in \Sigma^*$ et $n_1, n_2 \in \mathbb{N}$ tels que $w = w_1 w_2$ et $n = n_1 + n_2$ et $u \to^{n_1} w_1$ et $v \to^{n_2} w_2$.

Preuve

Par récurrence sur n.

- Pour $n = 0$, les (seuls) témoins sont $w_1 := u$ et $w_2 := v$ et $n_1 = n_2 = 0$.
- Cas inductif : Supposons $uv \to^{n+1} w$. Soit y tel que $uv \to y$ et $y \to^n w$. 1er cas, $u = u_1 X u_2$ et $y = u_1 u_X u_2 v$ avec $X \to u_X$. Par HR, soient w_1, w_2, n_1, n_2 tels que $w = w_1 w_2$ et $u_1 u_X u_2 \to^{n_1} w_1$ et $v \to^{n_2} w_2$. On a $u \to u_1 u_X u_2 \to^{n_1} w_1$ donc $u \to^{n_1+1} w_1$, donc $w_1, w_2, n_1 + 1, n_2$ sont des témoins. Le 2ème cas est similaire.

Type 3, grammaires linéaires

Définition

Une grammaire $(\Sigma, V, \rightarrow, S)$ est linéaire si elle est hors contexte et que dans chaque membre droit des règles apparaît au plus une variable, i.e. $\rightarrow \subseteq V \times (\Sigma^* \cup \Sigma^* V \Sigma^*)$

- Elle est linéaire gauche si $\rightarrow \subseteq V \times (\Sigma^* \cup V\Sigma^*)$
- Elle est linéaire droite si $\rightarrow \subseteq V \times (\Sigma^* \cup \Sigma^* V)$

Exemples

- Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ est linéaire. $S \rightarrow \epsilon$ et $S \rightarrow aSb$
- Le langage $\{a^n b^n c^p \mid n, p \in \mathbb{N}\}$ est linéaire.
  $S \rightarrow Sc | T$ (raccourci de $S \rightarrow Sc \quad S \rightarrow T$) et $T \rightarrow \epsilon | aTb$

- Les grammaires de type 3 sont de type 2.
- Les grammaires de type 3 sont presque toutes de type 1, mais certaines de leurs règles ne sont pas croissantes.

Grammaires linéaires droites

Rappel : une grammaire $(\Sigma, V, \rightarrow, S)$ est linéaire droite si $\rightarrow \subseteq V \times (\Sigma^* \cup \Sigma^* V)$.

Définition

Une grammaire $(\Sigma, V, \rightarrow, S)$ est linéaire droite stricte si $\rightarrow \subseteq V \times (\{\epsilon\} \cup \Sigma \cup \Sigma V)$.

Lemme

Pour toute grammaire linéaire droite, il existe une grammaire linéaire droite stricte produisant le même langage.

Preuve

- Si $A \rightarrow \epsilon$, on pose $A \rightarrow' \epsilon$
- Soit $R := (A, uB) \in \rightarrow$, et soit $n_R := |u|$. On pose $A \rightarrow' u_1 X_1^R$ et $X_1^R \rightarrow' u_2 X_3^R \ ... \ X_{n_R-2}^R \rightarrow' u_{n_R-1} X_{n_R-1}^R$ et $X_{n_R-1}^R \rightarrow' u_{n_R} B$.
- Soit $R := (A, u) \in \rightarrow$. Similaire à ci-dessus, en remplaçant $X_{n_R-1}^R \rightarrow' u_{n_R} B$ par $X_{n_R-1}^R \rightarrow' u_{n_R}$.

Linéaire droit vers rationnel

Lemme

Si un langage est linéaire droit (ou gauche), alors il est rationnel.

Preuve

Soit $L \subseteq \Sigma^*$ un langage linéaire droit. Par un lemme précédent, soit $(\Sigma, V, \to, S)$ une grammaire linéaire droite stricte le produisant. Soit $\mathcal{A} = (\Sigma, Q, \Delta, I, F)$ l'AFN tel que :

- $Q := V \sqcup \{q_f\}$
- $I := \{S\}$
- Si $S \to \epsilon$ est dans G, alors $F := \{q_f\} \sqcup \{S\}$, sinon $F := \{q_f\}$.
- $(A, a, B) \in \Delta$ si $A \to aB$ est dans G.
- $(A, a, q_f) \in \Delta$ si $A \to a$ est dans G.

On a $A \to aB$ ssi $A \xrightarrow{a}_{\mathcal{A}} B$. Donc $A \to^* uB$ ssi $A \xrightarrow{u}_{\mathcal{A}} B$, par récurrence sur u. Donc, pour $u \neq \epsilon$ on a $S \to^* u$ ssi $S \xrightarrow{u}_{\mathcal{A}} q_f$. Et $S \to^* \epsilon$ ssi $S \in F$. Ainsi u est produit par G ssi u est accepté par $\mathcal{A}$.

Rationnel vers linéaire droit

Lemme

Si un langage est rationel, il est linéaire droit (ou gauche)

Soit $L \subseteq \Sigma^*$ reconnu par un automate déterministe $(\Sigma, Q, \delta, i, F)$ (avec $\Sigma \cap Q = \emptyset$). On suppose d'abord que $i \notin F$. On définit une grammaire $G = (\Sigma, Q, \to, i)$:

- Pour tout $q \in Q$ et $a \in \Sigma$, on pose $q \to a\delta(q, a)$.
- Pour tout $q \in Q$ et $a \in \Sigma$ tel que $\delta(q, a) \in F$, on pose $q \to a$.

On montre que $i \to^* u\delta(i, u)$ pour tout $u \in \Sigma^*$, par récurrence sur u. Ainsi, si $\delta(i, u) \in F$ on a aussi $i \to^* u$. En effet, si $\delta(i, u) \in F$ alors $u \neq \epsilon$ par l'hypothèse $i \notin F$, donc il suffit de remplacer la dernière règle utilisée dans $i \to^* ua\delta(i, ua)$ par $\delta(i, u) \to a$. Réciproquement, si $q \to^* u$, alors la dernière règle utilisée est $\delta(i, u) \to a$ donc $\delta(i, u) \in F$, i.e. $u \in L$.
Si $i \in F$, on modifie un automate reconnaissant L en un automate reconnaissant $L \setminus \{\epsilon\}$ et on obtient une grammaire linéaire droite $G = (\Sigma, Q, \to, i)$ produisant $L \setminus \{\epsilon\}$. On définit enfin une grammaire $G' = (\Sigma, Q, \to, S)$ en ajoutant $S \to \epsilon$ et $S \to i$ aux règles de G.

Caractérisation des langages linéaires droits/gauches

Théorème

Un langage est rationel ssi il est linéaire droit (ou gauche)

Preuve

Par les deux lemmes précédents.