CH.9 La hiérarchie de Chomsky

- 9.1 Les grammaires générales
- 9.2 Les grammaires contextuelles
- 9.3 La classification des langages

9.1 Les grammaires générales

Encore appelées grammaires de type 0.

Exemple :

$S \rightarrow TZ$ $T \rightarrow 0T1C$ $T \rightarrow \varepsilon$

$C1 \rightarrow 1C$ $CZ \rightarrow Z2$ $1Z \rightarrow 1$

Cette grammaire engendre les mots de la forme $0^i1^i2^i$.

Ce langage n'est pas algébrique, donc les langages engendrés par les grammaires générales contiennent strictement les langages algébriques.

Théorème (Chomsky 1959):
Le langage L est engendré par une grammaire générale si et seulement s'il est accepté par une machine de Turing (ou un automate à deux piles).

Autre formulation :
Les langages engendrés par les grammaires de type 0 sont les langages récursivement énumérables.

La démonstration est **constructive**.
Néanmoins (voir chapitre précédent), les machines peuvent boucler en si le mot testé n'est pas dans le langage. De plus, le passage entre les deux énoncés du théorème n'est pas constructif.

Il s'ensuit que le problème de trouver si un mot est engendré par une grammaire générale est indécidable.

On a en fait un résultat beaucoup plus général :

Théorème (Rice) :
On désigne par une grammaire générale G.
Tous les problèmes de la forme P(G) sont indécidables sauf les deux problèmes triviaux (les constantes vrai et faux).

On peut formuler le même énoncé avec des machines de Turing au lieu des grammaires.

La conséquence est que les grammaires de type 0 ne peuvent être utilisées dans la pratique.

Enfin, les langages récursivement énumérables sont fermés par union, intersection, concaténation, substitution, homomorphisme inverse, mais pas par complémentation.

9.2 Les grammaires contextuelles

On restreint les règles en obligeant le membre droit à être au moins aussi long que le membre de gauche.
Ceci oblige à exclure le mot vide.
Les grammaires contextuelles sont aussi appelées grammaires de type 1.

Exemple :

$S \to TZ$ $T \to 0U1$ $T \to 01$
$U \to 0U1C$ $U \to 01C$
$C1 \to 1C$ $CZ \to Z2$ $1Z \to 12$

Cette grammaire engendre encore les mots de la forme $0^i1^i2^i$.

Etant donnée une grammaire contextuelle et un mot de longueur n, il est possible de vérifier si ce mot est engendré par la grammaire :
On fabrique toutes les dérivations à partir de l'axiome. Ces dérivations produisent des mots de longueur de plus en plus grande. On vérifie si le mot donné est obtenu.

On vient de démontrer le théorème (constructif) suivant :

Théorème :
Tout langage engendré par une grammaire contextuelle est récursif.

La réciproque est fausse : il existe des langages récursifs qui ne sont pas engendrables par une grammaire non contextuelle.

La démonstration de ce fait utilise un "argument diagonal".

On peut énumérer les grammaires non contextuelles sur l'alphabet {0,1}. Pour chacune d'entre elles, on peut fabriquer une machine de Turing (= un programme) qui s'arrête sur toutes les entrées en acceptant les mots engendrés par cette grammaire (voir théorème précédent).

On a donc fabriqué une suite de machines M_i. Toute grammaire non contextuelle a une machine M_i qui accepte le langage qu'elle engendre.

Soit maintenant l'entier k. On l'écrit en base 2 ; on le garde s'il est rejeté par la machine M_k et on rejette s'il est accepté par la machine M_k. On définit ainsi un langage récursif L, avec l'algorithme permettant de tester si un mot appartient ou non à L.

Si L était engendré par une grammaire, il existerait un programme n de la liste ci-dessus pour tester les mots. La contradiction vient de l'examen de l'écriture binaire de n par ce programme M_n.

Il existe un dispositif mécanique de reconnaissance des langages dépendant du contexte, appelé **automate borné linéairement**.

C'est une machine de Turing non déterministe qui n'utilise de sa mémoire infinie qu'une portion dépendant linéairement de la taille du mot testé.

Théorème:
Un langage est dépendant du contexte si et seulement s'il est accepté par un automate borné linéairement.

La démonstration est constructive dans les deux sens.

A noter qu'on ne sait pas si l'on peut se passer de l'hypothèse de non déterminisme.

Les autres problèmes sur les langages contextuels sont indécidables, y compris celui de savoir s'il est vide.

Enfin, les langages contextuels sont fermés par les mêmes opérations que les langages récursivement énumérables. La fermeture par complémentation n'est pas connue (liée au déterminisme).

9.3 La classification des langages

On peut dresser le tableau de la hiérarchie des langages, encore appelée **hiérarchie de Chomsky**.

On dispose pour chaque classe de langages d'un dispositif de génération et d'un dispositif de reconnaissance. Le passage de l'un à l'autre est toujours constructif.

Les inclusions sont toujours strictes.

Nom	Dispositif de génération	Dispositif de reconnaissance
Langages récursivement énumérables	Grammaires générales = Grammaires de type 0	Machines de Turing = Automates à 2 piles (pouvant boucler)
Langages récursifs	?	Machines de Turing = Automates à 2 piles (s'arrêtant toujours)
Langages contextuels	Grammaires contextuelles = Grammaires de type 1	Automates bornés linéairement
Langages non contextuels	Grammaires algébriques = Grammaires de type 2	Automates à pile non déterministes
Langages non contextuels déterministes	?	Automates à pile déterministes
Langages rationnels	Grammaires linéaires droites = Grammaires de type 3 Expressions régulières	Automates finis