

Comparaison entre un automate et un arbre

Introduction

On va dans cet exposé comparer les automates et les arbres, dire les points communs et les points divergents.

Table des matières

I- Qu'est-ce qu'un automate ?

II- Qu'est-ce qu'un arbre ?

III- Quel est le point commun entre un automate et un arbre ?

IV- Qu'est-ce qui distingue les arbres des automates ?

V- Quel est le point commun entre une expression régulière et une règle de réécriture?

VI- Qu'est-ce qui distingue une expression régulière d'une règle de réécriture?

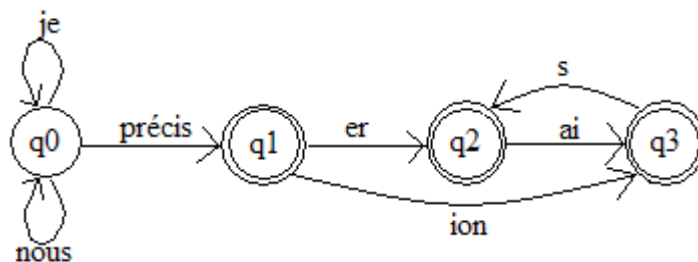
VII- Qu'est-ce qu'une expression régulière?

VIII- Qu'est-ce qu'une règle de réécriture?

Conclusion

Bibliographie

I- Qu'est-ce qu'un automate ?



Un automate à états finis est un quintuplet $M = (V, Q, q_0, F, \Lambda)$ où

V est le vocabulaire par exemple $V = \{je, nous, précis, er, ai, s, ion\}$

Q est l'ensemble des états par exemple $Q = \{q_0, q_1, q_2, q_3\}$

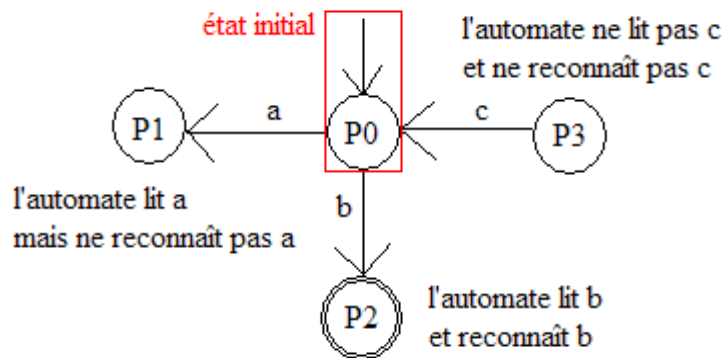
q_0 est l'état initial

F est l'ensemble des états accepteurs par exemple $F = \{q_1, q_2, q_3\}$

Λ est l'ensemble des transitions par exemple $\Lambda = \{(q_0, je, q_0), (q_0, nous, q_0), (q_0, précis, q_1), (q_1, er, q_2), (q_2, ai, q_3), (q_3, s, q_2), (q_1, ion, q_3)\}$

Un automate est un algorithme de reconnaissance de mots qui lit toutes les lettres d'un mot une par une et qui accepte ce mot si sa lecture se termine dans un état accepteur, également dit état final.

Exemple illustrant la définition de automate.



Chacune des transitions lit les symboles qu'il y a entre un état de départ et un état d'arrivée.
Par exemple $\lambda = \{(q_1, er, q_2)\}$

Les transitions peuvent être entre autres de nature:

- Réflexive, lorsqu'une boucle relie un état à lui-même. Par exemple $\lambda = \{(q_0, je, q_0), (q_0, nous, q_0)\}$
- Transitive, lorsqu'il est possible de sauter des états liés entre eux. Par exemple $\lambda = \{(q_1, er, q_2), (q_2, ai, q_3), (q_1, ion, q_3)\}$
- Symétrique, lorsque on peut faire un va et vient entre deux états consécutifs. $\lambda = \{(q_3, s, q_1), (q_1, ion, q_3)\}$
- Cyclique, lorsqu'après avoir parcouru un certain nombre d'états on revient au point de départ.
 $\lambda = \{(q_0, je, q_0); (q_3, s, q_1), (q_1, ion, q_3)\}$

Et les automates peuvent être de nature:

- Déterministe, lorsque une seule lecture est possible pour chaque symbole. Par exemple pour lire "er" on ne peut que passer par q_1 et q_2 .
- Non déterministe, lorsque plusieurs lectures sont possibles pour chaque symbole. Par exemple si pour lire "er" on avait le choix entre passer par q_1 et q_2 , ou passer par q_1 et q_4 .

Un mot appartient au langage d'un automate si on peut former ses parties avec le vocabulaire qui est donné et si il est reconnu dans un état accepteur.

Par exemple l'automate M peut produire les mots: {précis, jeprécis, nousprécis, jejeprécis, nousnousprécis, préciser, préciserai, préciserais, préciseraisais, jenouspréciseraisais, précision, etc.}

On peut compter le nombre de lectures possible pour un mot. Par exemple pour l'automate non déterministe, on n'a qu'une possibilité de lecture possible pour lire {précision} mais deux pour lire {préciser}.

Et on peut compter la longueur des mots.

0 mots de longueur 1, 2, 3, 4, 5, 7

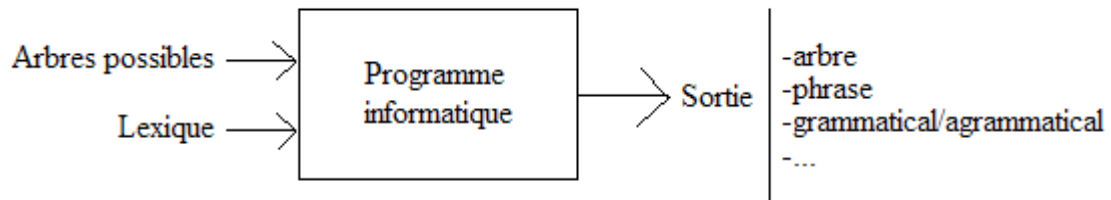
1 mot de longueur 6

1 mot de longueur 8

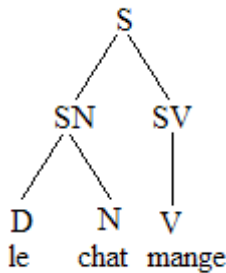
1 mot de longueur 9 etc...

II- Qu'est-ce qu'un arbre ?

Un arbre peut servir tout comme le lexique de donnée d'entrée d'un programme informatique comme Prolog (Par exemple: [SWI-Prolog](#) a [SQL](#) et est également disponible dans les [packages de cygwin](#)). Il représente l'analyse syntaxique que nous locuteurs avons décidé d'entrer après une longue réflexion. Un arbre doit pouvoir couvrir un grand nombre d'énoncés possibles, les catégories grammaticales d'un arbre doivent donc être les plus générales possible.



Remarque: Pour un linguiste, il vaut mieux que l'arbre soit une donnée d'entrée sinon il ne pourra pas avoir de prise sur la grammaire qu'il utilise.



Dans un arbre il y a toujours une racine, une catégorie initiale, c'est presque toujours la phrase noté S pour sentence (angl = phrase).

Des catégories grammaticales en relation de hiérarchie, la catégorie du haut inclue celle du bas et celle du bas dépend de celle du haut, et on applique les règles de gauche à droite.

Des constituants immédiats qui dans les grammaires hors contexte peuvent être soit le point de départ de l'analyse syntaxique soit le point d'arrivée. Le lexique de ces constituants immédiats est appelé vocabulaire terminal, et correspond aux feuilles de l'arbre. On n'est pas obligé de le représenter dans les arbres syntagmatiques, car ces arbres doivent pouvoir générer un plus grand nombre de phrases que celles de nos exemples.

Des inconvénients: les mots en deux morceaux comme ne...pas ne peuvent pas être réunis en une seule catégorie car deux arcs d'un arbre ne se recoupent jamais.

Des méthodes: pour décider comment on va regrouper les constituants immédiats en catégories grammaticales, on utilise la méthode de commutation qui consiste à remplacer un groupe d'éléments par un autre, et pour décider quelles sont les catégories grammaticales que l'on va définir, on utilise le critère de grammaticalité qui consiste à comparer deux phrases dont la structure grammaticale est identique.

Ex 1: Le chat mange. Le chien mange. La souris court.

Ex 2: Le chat va dans la cuisine. *Le chat va.

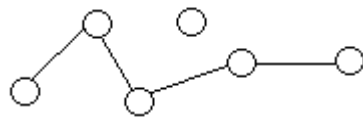
Une unité des conventions et de l'analyse:

| | |
|------------------------|---|
| S = Phrase. | A = Adjectif. |
| SN = Syntagme nominal. | V = Verbe. |
| SV = Syntagme verbal. | B = Adverbe (Cette convention peut être changée). |

| | |
|---|------------------|
| SA = Syntagme adjectival SB = Syntagme adverbial. SP = Syntagme prépositionnel. | P = Préposition. |
|---|------------------|

$S \rightarrow SN SV$
 $SN \rightarrow D N$
 $SV \rightarrow V GN$
 $SP \rightarrow P GN$
 $SA \rightarrow B A$
 $SB \rightarrow B B$

III- Quel est le point commun entre un automate et un arbre ?



Un graphe quelconque

On peut définir les automates et les arbres par la théorie des graphes où les noeuds correspondent aux états des automates ou aux catégories grammaticales des arbres syntagmatiques, et où les arcs correspondent aux transitions des automates ou aux relations de dépendance des arbres.

Un automate est un graphe orienté qui a une racine et qui est connexe et un arbre syntagmatique est un graphe orienté qui a une racine et qui est acyclique et projectif et ordonné et connexe.

Graphe = il y a des noeuds et des arcs
 Orienté = il y a un sens de lecture
 Une racine = un état initial
 Acyclique = il n'y a pas de cycle
 Projectif = deux arcs ne se recoupent jamais
 Ordonné = il y a une hiérarchie
 Connexe = chaque noeud est relié à un autre

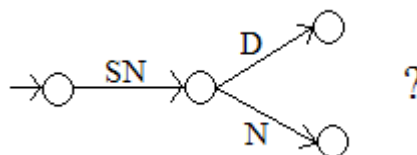
IV- Qu'est-ce qui distingue les arbres des automates ?

On ne peut pas concevoir un automate qui génère le langage d'un arbre syntaxique mais on peut concevoir un arbre syntaxique qui génère le langage d'un automate. Donc un arbre représente mieux le langage humain écrit qu'un automate car on a plus de liberté d'expression.

Avec un automate on ne peut pas:

-Inclure des catégories grammaticales à l'intérieur d'autres catégories grammaticales car on ne fait pas la distinction entre symbole terminal et symbole non terminal, i.e. entre le vocabulaire et la catégorie grammaticale.

Par exemple on ne peut pas représenter une simple règle de récursivité:
 $SN \rightarrow D + N$



-Illustrer le phénomène de récursivité qui consiste à donner à un syntagme la possibilité de pouvoir se dominer un nombre infini de fois à l'intérieur d'une phrase.

Par exemple on ne peut pas représenter des syntagmes prépositionnels emboîtés:

$SN \rightarrow D + N + SP$

$SP \rightarrow P + SN$

La fille du fils du frère de mon arrière-grand-père de la ville des lumières.

-Illustrer le phénomène d'enchâssement qui consiste à donner à une phrase la possibilité de pouvoir se dominer un nombre infini de fois à l'intérieur d'une autre phrase.

Par exemple on ne peut pas représenter les complétives:

$S \rightarrow GN GV$

$GV \rightarrow V S'$

$S' \rightarrow Conj S$

Je pense que tu penses que il pense que tout le monde pense qu'il ne neigera plus en antarctique.

Ni les relatives:

$S \rightarrow GN GV \mid GN S'$

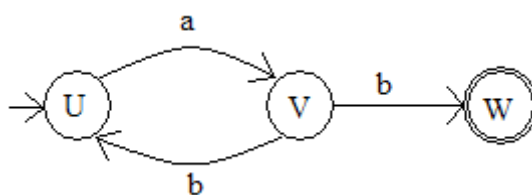
$GN \rightarrow GN S' \mid \Sigma$

$S' \rightarrow Pron S$

Le livre qui est gris qui est sur la table que tu m'avais apporté que tu voulais que je lise, je l'ai feuilleté.

Par contre on peut représenter un automate avec une règle de réécriture en donnant un nom à chaque état et en écrivant chaque symbole lu devant le nom de l'état.

Exemple: $V = \{a, b\}$ $Cat = \{U, V, W\}$ $q_0 = U$



Règles de réécriture

$U \rightarrow aV$
 $V \rightarrow bU \mid bW$
 $W \rightarrow \Sigma$

Expression régulière

$L = \{ab\}^*.b$

V- Quel est le point commun entre une expression régulière et une règle de réécriture?

Ce sont tous des langages formels car à partir d'un vocabulaire fini, on produit un nombre fini ou infini de mots.

Et de même à partir d'un nombre de mots fini, on produit un nombre fini ou infini de phrases.

Mais pour simplifier ce qu'on dit, on peut considérer qu'une phrase est un grand mot avec des blancs et donc qu'on part d'un vocabulaire fini pour produire un nombre fini ou infini de mots.

VI- Qu'est-ce qui distingue une expression régulière d'une règle de réécriture?

L'usage: une expression régulière décrit le langage d'un automate et une règle de réécriture décrit le langage d'un être humain d'un point de vue syntaxique. Donc on ne part pas de la même réalité et on n'a pas les mêmes notations.

Pour écrire une expression régulière on prend en compte le quintuplet (V, Q, q_0, F, Λ) relatif aux automates et pour écrire une règle de réécriture, on prend en compte le quadruplet (S, V, Cat, R) relatif aux grammaires de réécriture.

| (V, Q, q_0, F, Λ) | (S, V, Cat, R) |
|---|---|
| V est le vocabulaire Q est l'ensemble des états q_0 est l'état initial F est l'ensemble des états accepteurs Λ est l'ensemble des transitions | S est la catégorie initiale V est le vocabulaire Cat sont les catégories grammaticales R sont les règles de réécriture |

L'acceptabilité:

Soit P une phrase, toutes les phrases acceptables sont reconnues par l'automate. Mais toutes les phrases acceptables et certaines phrases inacceptables sont reconnues par le vérificateur syntaxique pourvu que ces phrases soient grammaticales. Car les expressions régulières définissent toutes les phrases qui sont acceptables dans un langage. Et les règles de réécriture définissent toutes les phrases qui sont grammaticales. Or la grammaticalité n'est pas l'acceptabilité, car ce qui est grammatical est bien construit, et ce qui est acceptable a un sens.

Donc:

Pierre mange une pomme.

Pierre mange une maison.

Sont des phrases grammaticales car bien construites.

VII- Qu'est-ce qu'une expression régulière?

Une expression régulière sert à décrire un nombre fini ou infini de mots acceptables appartenant au langage d'un automate.

Au niveau des symboles on a pour les expressions régulières:

$\{ \}$ = les accolades qui séparent les mots lus par les automates.

, = la virgule, parfois aussi notée U ou + qui indique qu'on a le choix entre deux lectures possibles.

. = le point qui symbolise la concaténation, qui indique qu'on lit d'abord tel mot puis tel autre.
 * = l'étoile de Kleene qui signifie que la séquence concernée peut se répéter un nombre de fois compris entre 0 et l'infini.

Σ = le sigma qui est un mot du vocabulaire, qui peut être engendré par l'étoile de Kleene, ou qui peut servir de transition lorsqu'on réunit deux automates, ou qui peut être la conséquence de la reconnaissance de aucun mot lorsque l'état initial est un état accepteur.

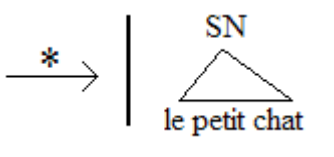
VIII- Qu'est-ce qu'une règle de réécriture?

Une règle de réécriture appartient à un ensemble de règles de réécriture noté R. Un ensemble de règles de réécriture sert à décrire un nombre fini ou infini de phrases grammaticales appartenant au langage humain d'un point de vue syntaxique.

Au niveau des symboles on a pour les règles de réécriture:

→ = la flèche qui signifie que x se réécrit y ou soit que y est inclus dans x.

| = le séparateur d'instructions qui signifie qu'un même symbole peut se réécrire de deux manières différentes

| | |
|--|---|
|  <p>The diagram shows a horizontal arrow pointing to the right with an asterisk (*) above it. To the right of the arrow is a vertical line. To the right of the vertical line is a triangle with the letters 'SN' above it and the text 'le petit chat' below it.</p> | <p>= la flèche avec une étoile dessus, ou le triangle dans les parties du bas d'un arbre, qui signifie que plusieurs étapes de réécriture se sont succédées</p> |
|--|---|

De plus on essaye de ne pas générer trop de phrases agrammaticales quand on écrit ses règles de réécriture:

*Pierre marche Marie (verbe intransitif)

Pierre connaît Marie (verbe transitif)

Donc $GV \rightarrow VGN$ n'est pas toujours vrai et il faut faire un GV_i et un GV_t pour pouvoir avoir ces deux cas.

Conclusion

Un automate et un arbre peuvent être décrits comme étant des graphes orientés et connexes avec une racine et des noeuds. Cependant un arbre est acyclique, projectif et ordonné et représente des relations d'inclusion. On peut représenter le langage d'un automate par des règles de réécriture mais on ne peut pas faire l'inverse. Les règles de réécriture sont tout comme les expressions régulières des langages formels grâce auxquels à partir d'un nombre fini de mots on peut générer un nombre infini de phrases. Cependant les règles de réécriture servent à vérifier si une phrase est grammaticale ou pas sans prendre en compte l'aspect sémantique de la phrase alors qu'une expression régulière ne fait que vérifier si une phrase peut être acceptée ou non par l'automate.

Bibliographie

Cours de grammaires formelles de M. Kahane de 2006 à 2008.

La grammaire d'aujourd'hui, M. Arrivé, F. Gadet, M. Galmiche, Flammarion, 1986

Auteur

Del Socorro Françoise, 2008

Copyrights:



Comparaison entre un automate et un arbre by *Del Socorro Françoise* est mis à disposition selon les termes de la [licence Creative Commons Paternité-Pas d'Utilisation Commerciale 2.0 France](https://creativecommons.org/licenses/by-nc/2.0/fr/).

Remarque: Où trouver des copyrights gratuits pour diffuser vos créations sur le web ?

<http://fr.creativecommons.org/>

"Ces contrats d'accès ouvert peuvent être utilisés pour tout type de création : texte, film, photo, musique, site web... "