THLR - Théorie des langages rationnels

Maxime Bridoux

Septembre 2020

Compilation

But

Transformer du texte en langage machine en deux étapes:

1. identification des mots-clés et variables;
2. reconnaissance de la grammaire

Analyse lexicale

Conversion du texte en une séquence de symboles

Analyse syntaxique

Analyse des symboles pour mettre en évidence les règles de grammaire utilisées et vérifier la validité du programme

Compilation

Programme initial

a = 2 + 3

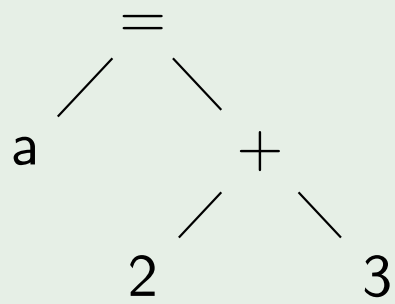
Analyse lexicale

[(id, a); (aff, =); (lit, 2); (op, +), (lit, 3)]

Analyse syntaxique

Règles disponibles:

1. id aff lit → programme
2. lit op lit → lit



Bio-informatique

Exemples d'objets naturels

ADN Succession de nucléotides dénotés par A,C,G ou T

Protéines Succession d'acides aminés

Modélisation informatique

Modélisation possible par un texte écrit sur un alphabet de 4 lettres (respectivement 20 lettres pour les protéines)

Applications

- Recherche d'une séquence particulière
- Mesurer la distance entre deux séquences

Langages humains

A l'oral

Une phrase orale est une suite linéaire de sons interprétés

A l'écrit

- Les langues écrites disposent d'un ou plusieurs alphabets qui permettent d'écrire des mots
- L'ensemble des mots d'une langue sont ceux qui sont disponibles dans son dictionnaire

Vocabulaire

Définitions

- On appelle lettres les symboles utilisés
- Un alphabet Σ un ensemble fini de lettres
- Un mot sur un alphabet Σ est une suite finie de lettres de Σ

Exemples

On pose l'alphabet $\Sigma = \{a, b\}$ composé des deux lettres a et b. Le mot $u = abba$ est un mot de 4 lettres sur l'alphabet Σ

Longueur d'un mot

Définition

La longueur d'un mot u, noté $|u|$ correspond au nombre total de lettres de u

Mot vide

L'unique mot de longueur 0 est appelé mot vide et est noté ϵ

Occurrences

Le nombre d'occurrences d'un symbole a dans un mot u est noté $|u|_a$. On a immédiatement:

$$|u| = \sum_{a \in \Sigma} |u|_a$$

Langages

Ensemble de tous les mots

Étant donné un alphabet Σ, on note Σ^* l'ensemble de tous les mots (de taille finie) formés sur Σ

Pour un alphabet de deux lettres

Pour $\Sigma = \{a, b\}$, on a $\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$

Définition

On appelle un langage L sur un alphabet Σ un sous ensemble de Σ^*, i.e. $L \subseteq \Sigma^*$. Cela correspond à un ensemble particulier de mots sur Σ

Langages

Exemples de langages

On pose $\Sigma = \{a, b\}$. Sont des langages sur Σ:

- Σ^*, l'ensemble de tous les mots sur Σ;
- $L_1 = \{\epsilon, a, b\} = \{u \in \Sigma^* \mid |u| < 2\}$;
- $L_2 = \{ab, aab, abb, aaab, aabb, \dots\}$, l'ensemble des mots commençant par a et finssant par b;
- $L_3 = \{\epsilon, ab, aabb, aaabbb, \dots\}$ le langage des mots ayant un certain nombre de a suivis par autant de b;
- $L_4 = \{aaa, bba\}$ un langage avec seulement deux mots

Calculabilité

Idée générale

L'ensemble des solutions d'un problème est un ensemble de mots: résoudre un problème revient à reconnaître un langage

Langage récursivement énumérable

Un langage L est dit récursivement énumérable s'il existe un algorithme A qui énumère tous les mots de L
i.e pour chaque mot u de L, l'énumérateur A produit u en un nombre fini d'étapes de calcul

Calculabilité

Définition alternative

Un langage L est dit récursivement énumérable (RE) ou semi-décidable s'il existe un algorithme A tel que:

$$\begin{cases} \text{si } u \in L, & A(u) \text{ s'arrête et répond oui} \\ \text{si } u \notin L, & A(u) \text{ peut boucler à l'infini} \end{cases}$$

Langage récursif

Un langage L est dit récursif (R) ou décidable s'il existe un algorithme A tel que:

$$\begin{cases} \text{si } u \in L, & A(u) \text{ s'arrête et répond oui} \\ \text{si } u \notin L, & A(u) \text{ s'arrête et répond non} \end{cases}$$

Conséquence

On a l'inclusion stricte suivante: $R \subsetneq RE$

Hiérarchie de Chomsky (1956)

Type 0	Langages récursivement énumérables
	Machines de Turing

Langages récursifs

Type 1	Langages contextuels
Type 2	Langages algébriques
	Automates à pile
Type 3	Langages rationnels
	Automates finis

Concaténation

Définition

La concaténation de deux mots $u = (u_1 \dots u_k)$ et $v = (v_1 \dots v_l)$ est le mot $u \bullet v = uv = (u_1 \dots u_k v_1 \dots v_l)$

Propriétés

La concaténation:

- est une opération interne de Σ^*;
- est associative: $(uv)w = u(vw)$ pour $u, v, w \in \Sigma^*$;
- possède un élément neutre ϵ: $u\epsilon = u = \epsilon u$

Structure

$(\Sigma^*, \bullet)$ est donc un monoïde

Liberté

Décomposition unique

Chaque mot possède une décomposition unique:

$$(u_1 \dots u_n) = (v_1 \dots v_p) \implies \begin{cases} n = p \\ \forall i, u_i = v_i \end{cases}$$

Autrement dit, $(u_1 \dots u_n) = u_1 \bullet u_2 \bullet \cdots \bullet u_n = u_1 \dots u_n$

Structure

$(\Sigma^*, \bullet)$ est le monoïde libre créé à partir de Σ

Notations

Puissance

On pose par induction:

- $u^0 = \epsilon$;
- $u^{n+1} = u^n \bullet u$ pour $n \geq 0$

Exemples

- $\{\epsilon, ab, aabb, aaabbb, \dots\} = \{a^n b^n, n \geq 0\}$
- $\{\epsilon, ab, abab, ababab, \dots\} = \{(ab)^n, n \geq 0\}$

Factorisation

Pour $u = xy$, on note $y = x^{-1}u$ et $x = uy^{-1}$

Facteurs et sous-mots

Définitions

- Un mot u est un facteur de v s'il existe deux mots u_1 et u_2 tels que $v = u_1 u u_2$;
- u est un préfixe de v s'il existe w tel que $v = uw$;
- u est un suffixe de v s'il existe w tel que $v = wu$;
- u est un sous-mot de w s'il existe une factorisation de w en $v_1 u_1 v_2 u_2 \dots v_n u_n v_{n+1}$ avec pour tout i, u_i, v_i des mots tels que $u = u_1 u_2 \dots u_n$;
- un facteur u (resp. préfixe, suffixe, sous-mot) de v est propre s'il diffère de v
- on note $|pref|_k(u)$ (resp. $|suff|_k(u)$) le préfixe (resp. suffixe) de taille k de u (égal à u si $k > |u|$)

Relations sur les mots

Définitions

- Un mot u est primitif si l'équation $u = v^i$ n'admet pas de solutions pour $i > 1$;
- Deux mots x et y sont conjugés s'il existe u et v des mots tels que $x = uv$ et $y = vu$;

Exemples

- $abab$ n'est pas primitif car peut se réécrire $(ab)^2$;
- $aabb = a^2b^2$ est primitif
- $aabb$ est conjugué à $abba$, $bbaa$, $baab$

Mot miroir

Définitions

- Le miroir ou transposé de $u = u_1 \dots u_n$ est le mot $u^R = u_n \dots u_1$
- u est un palindrome si $u = u^R$

Exemples

- le miroir de *cabab* est *babac*
- *ababa* est un palindrome

Quotient

Définition

Le quotient droit de u par v, noté uv^{-1} est:

$$uv^{-1} = \begin{cases} w & \text{si } u = wv \\ \text{indéfini} & \text{si v n'est pas suffixe de u} \end{cases}$$

On définit de manière similaire de le quotient gauche de u par v, noté $v^{-1}u$

Exemples

- $abcde(cde)^{-1} = ab$
- $abcde(ab)^{-1}$ est indéfini
- $(ab)^{-1}abcde = cde$

Relations d'ordre sur les mots

Préfixe

La relation $u \leq_p v$ si u est un préfixe de v est une relation d'ordre partielle sur Σ^*

Ordres totaux sur Σ^*

On suppose disposer d'un ordre total $\leq$ sur Σ et l'on définit:

- l'ordre lexicographique, noté $\leq_l$ tel que $u \leq_l v$ ssi u est un préfixe de v ou $(u = tu'$ et $v = tv'$ avec $u'_1 < v'_1)$
- l'ordre radiciel, noté $\leq_r$ tel que $u \leq_r v$ ssi $|u| < |v|$ ou $(|u| = |v|$ et $u \leq_l v)$

Relations d'ordre sur les mots

Exemples

- $a \leq_{p,l,r} ab$
- $aaaaaaaaaaa$ et b ne sont pas comparables avec $\leq_p$
- $aaaaaaaaaaa \leq_l b$
- $aaaaaaaaaaa \geq_r b$

Propriétés de l'ordre radiciel

- ordre bien fondé: il n'existe pas de chaîne infinie décroissante
- $\forall w, w' \in \Sigma^*, u \leq_r v \implies wuw' \leq_r wvw'$

Distances

Axiomes de la distance

- $d(u, v) \geq 0$
- $d(u, v) = 0 \Leftrightarrow u = v$
- $d(u, w) \leq d(u, v) + d(v, w)$

Distance préfixe

On note $\text{plpc}(u, v)$ le plus long préfixe commun à u et v.

$$d_p(u, v) = |u| + |v| - 2|plpc(u, v)|$$

définit une distance nommée distance préfixe

Distances

Distance d'édition

La distance d'édition ou distance de Levenshtein entre u et v est le nombre d'opérations élémentaires (insertion et suppression) pour passer de u à v

Exemple

La distance de Levenshtein de *chien* à *chameau* est de 6:
$chien \xrightarrow{-i} chen \xrightarrow{+a} chaen \xrightarrow{+m} chamen \xrightarrow{-n} chame \xrightarrow{+a} chamea \xrightarrow{+u} chameau$

Applications

Reconnaissance de texte et de formes; reconnaissance vocale

Opérations ensemblistes

Définitions ensemblistes

$$L_1 \cup L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ ou } u \in L_2\}$$
$$L_1 \cap L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ et } u \in L_2\}$$
$$\overline{L} = \{u \in \Sigma^* \mid u \notin L\}$$

Exemples

Soit $L_1 = \{aa, ab\}$ et $L_2 = \{aa, aaaa\}$

- $L_1 \cup L_2 = \{aa, aaaa, ab\}$
- $L_1 \cap L_2 = \{aa\}$
- $\overline{L_1} = \Sigma^* \setminus \{aa, ab\}$

Autres opérations

Concaténation

La concaténation de deux langages L_1 et L_2 est le langage:

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

Propriétés

C'est une opération associative, ce qui permet de définir l'itération d'un langage:

- $L^0 = \{\epsilon\}$ par convention
- $L^{n+1} = L^n L$ pour $n \geq 0$

On a bien $L^1 = L$ avec ces définitions

Étoile de Kleene

Définition

La fermeture (ou étoile) de Kleene d'un langage L est le langage:

$$L^* = \bigcup_{i \geq 0} L^i$$

Il s'agit de tous les mots qu'il est possible de construire en concaténant un nombre fini de mots de L

Propriété

L^* est le plus petit langage contenant ϵ et L qui soit stable par concaténation

Autre langage utile

Définition

On définit aussi:

$$L^+ = \bigcup_{i \geq 1} L^i$$

On a alors $L^+ = LL^* = L^*L$

Remarque

On a toujours $\epsilon \in L^*$, alors que $\epsilon \in L^+ \Leftrightarrow \epsilon \in L$

Exemples

Autour de Σ

Pour $L = \Sigma = \{a, b, c\}$:

- $L^0 = \{\epsilon\}$
- $L^1 = \{a, b, c\}$
- $L^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$
- $L^* = \Sigma^*$
- $L^+ = \Sigma^* \setminus \{\epsilon\}$

Langages des préfixes

Définitions

Soit L un langage sur Σ, on appelle langages des préfixes (resp. suffixes, facteurs):

- $\text{Pref}(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, uv \in L\}$
- $\text{Suff}(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, vu \in L\}$
- $\text{Fact}(L) = \{v \in \Sigma^* \mid \exists u, w \in \Sigma^*, uvw \in L\}$

Applications

L est un langage préfixe si pour tout $u \neq v$ dans L, $u \notin \mathit{Pref}(v)$
Applications : codages préfixes

Morphismes

Définition

Un morphisme de monoïde d'un monoïde $(M, \bullet)$ dans un monoïde $(N, \times)$ est une application $\phi : M \to N$ vérifiant:

- $\phi(\epsilon_M) = \epsilon_N$
- $\phi(u \bullet v) = \phi(u) \times \phi(v), \forall u, v \in M$

Codage

Soit Σ un alphabet et $B = \{0, 1\}$, on peut coder les mots de Σ en binaire par une application $C : \Sigma \to B^*$. C se prolonge alors d'une unique façon en un morphisme $\tilde{C} : \Sigma^* \to B^*$

Code

Un code est un morphisme injectif i.e $\phi(u) = \phi(v) \implies u = v$