LASF – Grammaires

Christophe Mouilleron

école nationale supérieure d'informatique pour l'industrie et l'entreprise

Problèmes liés aux langages rationnels

Représentation des langages
$\rightsquigarrow$ expressions rationnelles

Déterminer efficacement si :

- L est vide, fini ou infini — finitude
- $u \in A^*$ appartient à L — reconnaissance
- deux représentations correspondent au même langage — égalité
- L est rationnel — reconnaissabilité

$\rightsquigarrow$ automates finis

Génération des mots u de L
$\rightsquigarrow$ grammaires, grammaires régulières

Plan

1. Grammaires
   - Définition
   - Génération d'un langage
2. Grammaires régulières
   - Grammaires régulières à droite
   - Automate fini $\rightarrow$ grammaire régulière
3. Bilan sur les grammaires

Plan

Motivation

Limites des expressions régulières pour représenter un langage :

Motivation

Limites des expressions régulières pour représenter un langage :

- peu adapté aux langages complexes
- absence totale de structuration
- exprime les langages rationnels uniquement $(^n)^n$ non géré

Motivation

Limites des expressions régulières pour représenter un langage :

- peu adapté aux langages complexes
- absence totale de structuration
- exprime les langages rationnels uniquement $(^n)^n$ non géré

Alternative = représentation sous forme d'une grammaire :

- en linguistique,
- description de formats de fichiers ex : HTML5
- syntaxe d'un langage programmation
- documentation des commandes Unix pages man

Exemple informel : les phrases en français

Une phrase est une suite de mots organisée selon des règles.

Exemple de règles :

- `phrase` $\rightarrow$ `sujet verbe complément .`
- `phrase` $\rightarrow$ `pronomInterrogatif verbe sujet ?`

Exemple informel : les phrases en français

Une phrase est une suite de mots organisée selon des règles.

Exemple de règles :

- `phrase` $\rightarrow$ `sujet verbe complément .`
- `phrase` $\rightarrow$ `pronomInterrogatif verbe sujet ?`
- `sujet` $\rightarrow$ `prénom`

Exemple informel : les phrases en français

Une phrase est une suite de mots organisée selon des règles.

Exemple de règles :

- `phrase` $\rightarrow$ `sujet verbe complément .`
- `phrase` $\rightarrow$ `pronomInterrogatif verbe sujet ?`
- `sujet` $\rightarrow$ `prénom` | `article nom` | `article adjectif nom`

Exemple informel : les phrases en français

Une phrase est une suite de mots organisée selon des règles.

Exemple de règles :

- `phrase` → `sujet verbe complément .`
- `phrase` → `pronomInterrogatif verbe sujet ?`
- `sujet` → `prénom` | `article nom` | `article adjectif nom`
- `article` → le | la | … `nom` → chat | souris | …
- `verbe` → mange | dort | … `adjectif` → gros | …
- `pronomInterrogatif` → qui | que | quand | où | …

Grammaires

Définition

Une grammaire est un quadruplet (A, N, S, R) où :

- A est un ensemble de symboles terminaux
- N est un ensemble de symboles non terminaux
- $S \in N$ est l'axiome symbole de départ
- R est un ensemble de règles de la forme

$$u \to v$$

avec $u, v \in (A \cup N)^*$, u contient au moins un symbole dans N.

Grammaires non-contextuelles

Définition

Une grammaire (non-contextuelle) est un quadruplet (A, N, S, R) où :

- A est un ensemble de symboles terminaux
- N est un ensemble de symboles non terminaux
- $S \in N$ est l'axiome — symbole de départ
- R est un ensemble de règles de la forme

$$u \to v$$

avec $v \in (A \cup N)^*$, $u \in N$.

$\rightsquigarrow$ **règle** = change un symbole non terminal en une suite de symboles.

Exemple de grammaire

Exemple : $G = (A, N, S, R)$ avec

- $A = \{0, 1, \ldots, 9, +, \times\}$
- $N = \{S, A, P, C\}$
- R formé de

$$S \to A \mid P \mid C \qquad C \to 0 \mid 1 \mid \ldots \mid 9$$
$$A \to S + S \qquad P \to S \times S$$

Exemple de grammaire

Exemple : $G = (A, N, S, R)$ avec

- $A = \{0, 1, \ldots, 9, +, \times\}$
- $N = \{S, A, P, C\}$
- R formé de

$$S \to A \mid P \mid C \qquad\qquad C \to 0 \mid 1 \mid \ldots \mid 9$$
$$A \to S + S \qquad\qquad P \to S \times S$$

Par convention :

- l'axiome est noté S, Start
- les symboles non-terminaux (N) sont notés en majuscules,
- les symboles terminaux (A) sont notés en minuscules.

Ainsi, définir une grammaire = donner un ensemble de règles R.

Plan

Génération d'un mot

On se donne une grammaire $G = (A, N, S, R)$.

Dérivation = applications successives de règles dans R
Départ de l'axiome S
Arrêt quand le résultat est dans A^*

Génération d'un mot

On se donne une grammaire $G = (A, N, S, R)$.

Dérivation = applications successives de règles dans R
Départ de l'axiome S
Arrêt quand le résultat est dans A^*

Exemple : Avec G définie par

$$S \rightarrow S + S \mid S \times S \mid 0 \mid 1 \mid \ldots \mid 9$$

on peut dériver $2 + 3 \times 5$ via :

$$\begin{array}{lllllll} S & \rightarrow & S + S & \rightarrow & 2 + S & \rightarrow & 2 + S \times S \\ & \rightarrow & 2 + 3 \times S & \rightarrow & 2 + 3 \times 5 & & \end{array}$$

Langage engendré par une grammaire

Notation : $u \rightarrow_R^* v$ s'il existe une dérivation menant de u à v.

Définition

Le langage généré par la grammaire $G = (A, N, S, R)$ est le langage $L(G) = \{u \in A^*, S \rightarrow_R^* u\}$.

Exemple : avec $S \rightarrow aSb \mid \varepsilon$

Langage engendré par une grammaire

Notation : $u \to_R^* v$ s'il existe une dérivation menant de u à v.

Définition

Le langage généré par la grammaire $G = (A, N, S, R)$ est le langage $L(G) = \{u \in A^*, S \to_R^* u\}$.

Exemple : avec $S \to aSb \mid \varepsilon$ $\qquad L(G) = \{a^n b^n, n \in \mathbb{N}\}$

Questions soulevées :

- calculer $L(G)$ pour G donnée
- pour L donné, trouver G telle que $L(G) = L$
- tester si $u \in A^*$ est dans $L(G)$

Plan

Un peu de régularité

Grammaires plus expressives que les automates cf $a^n b^n$ + Compil.

$\rightsquigarrow$ ajout de contraintes pour se limiter à des grammaires simples

Motivations :

- trouver un modèle comparable aux automates finis,
- échauffement pour le cours de compilation.

Idée : imposer des règles de la forme

$$U \to \varepsilon \qquad U \to v \qquad U \to vV$$

avec $U, V \in N$ et $v \in A$

Simplifications possibles

1. Limitation à 1 terminal avant le non-terminal V.

   $\leadsto$ remplacer $\quad U \to a_1 a_2 a_3 V$
   par $\quad U \to a_1 U_1 \quad U_1 \to a_2 U_2 \quad U_2 \to a_3 V$.

2. Suppression des règles du type $U \to V$.

   $\leadsto$ ajouter $U \to \alpha$ pour chaque $\alpha \notin N$ tel que $\begin{cases} U \to V \to^* \alpha, \\ \alpha \neq U. \end{cases}$

Simplifications possibles

1. Limitation à 1 terminal avant le non-terminal V.

   $\leadsto$ remplacer $\quad U \to a_1 a_2 a_3 V$

   par $\quad U \to a_1 U_1 \qquad U_1 \to a_2 U_2 \qquad U_2 \to a_3 V$.

2. Suppression des règles du type $U \to V$.

   $\leadsto$ ajouter $U \to \alpha$ pour chaque $\alpha \notin N$ tel que $\begin{cases} U \to V \to^* \alpha, \\ \alpha \neq U. \end{cases}$

Exemple :

$$\begin{array}{l} S \to M \mid T \\ M \to a \\ T \to bM \mid S \end{array} \qquad \leadsto$$

Simplifications possibles

1. Limitation à 1 terminal avant le non-terminal V.

   $\rightsquigarrow$ remplacer $U \to a_1 a_2 a_3 V$

   par $U \to a_1 U_1 \quad U_1 \to a_2 U_2 \quad U_2 \to a_3 V$.

2. Suppression des règles du type $U \to V$.

   $\rightsquigarrow$ ajouter $U \to \alpha$ pour chaque $\alpha \notin N$ tel que $\begin{cases} U \to V \to^* \alpha, \\ \alpha \neq U. \end{cases}$

Exemple :

$$\begin{array}{l} S \to M \mid T \\ M \to a \\ T \to bM \mid S \end{array} \quad \rightsquigarrow \quad \begin{array}{l} S \to a \mid bM \\ M \to a \\ T \to bM \mid a \end{array}$$

Grammaires régulières à droite

Définition

Une grammaire $G = (A, N, S, R)$ est dite régulière à droite lorsque toutes les règles de R sont de l'une des formes suivantes :

- $U \to \varepsilon$
- $U \to a$
- $U \to aV$

avec $U \in N$, $V \in N$ et $a \in A$.

Exemple : grammaire engendrant le langage a^+b^+

$$S \to aS \mid aB$$
$$B \to bB \mid b$$

Test de générabilité par analyse descendante

Principe de l'analyse descendante :

- recherche d'une dérivation $S \to_R^* u$ en partant de S,
- réduction la plus à gauche possible, utilisant la première lettre non générée de u.

Exemple : $S \to aS \mid aB \mid \varepsilon \qquad B \to bB \mid \varepsilon$

Est-ce que $abb \in L(G)$?

$S \to aS$
$S \to aB$

Test de générabilité par analyse descendante

Principe de l'analyse descendante :

- recherche d'une dérivation $S \to_R^* u$ en partant de S,
- réduction la plus à gauche possible, utilisant la première lettre non générée de u.

Exemple : $S \to aS \mid aB \mid \varepsilon \qquad B \to bB \mid \varepsilon$

Est-ce que $abb \in L(G)$?

$S \to aS \to \ldots$ impossible de commencer par *ab*

$S \to aB$

Test de générabilité par analyse descendante

Principe de l'analyse descendante :

- recherche d'une dérivation $S \rightarrow_R^* u$ en partant de S,
- réduction la plus à gauche possible, utilisant la première lettre non générée de u.

Exemple : $S \rightarrow aS \mid aB \mid \varepsilon \qquad B \rightarrow bB \mid \varepsilon$

Est-ce que $abb \in L(G)$? **OUI**

$S \rightarrow aS \rightarrow \ldots$ impossible de commencer par *ab*

$S \rightarrow aB \rightarrow abB \rightarrow abbB \rightarrow abb$ ok

Test de générabilité par analyse descendante

Principe de l'analyse descendante :

- recherche d'une dérivation $S \to_R^* u$ en partant de S,
- réduction la plus à gauche possible, utilisant la première lettre non générée de u.

Exemple : $S \to aS \mid aB \mid \varepsilon \qquad B \to bB \mid \varepsilon$

Est-ce que $abb \in L(G)$? **OUI**

$S \to aS \to \dots$ impossible de commencer par *ab*

$S \to aB \to abB \to abbB \to abb$ ok

Remarques :

- principe valable pour tout type de grammaire …
- … et particulièrement adapté si G régulière à droite

Grammaire régulière à droite $\to$ automate

Idée = passer par la grammaire régulière réduite

$\leadsto$ changer $U \to a$ en $U \to aE \quad E \to \varepsilon$
où E est un nouveau symbole non-terminal.

Automate associé à $G = (A, N, S, R)$ régulière réduite :

1. définir un état par symbole non-terminal $\quad Q = N$
2. état initial = axiome S $\quad I = \{S\}$
3. état U final lorsque $U \to \varepsilon$ dans R $\quad F = \{U, U \to \varepsilon\}$
4. transition $U \xrightarrow{a} V$ lorsque $U \to aV$ dans R $\quad E = \{(U, a, V), U \to aV\}$

Grammaire régulière à droite $\rightarrow$ automate – Exemple

si G définie par $\quad S \rightarrow aB \mid b \quad B \rightarrow bS \mid \varepsilon$

grammaire régulière réduite =

Grammaire régulière à droite $\rightarrow$ automate – Exemple

si G définie par $\quad S \rightarrow aB \mid b \quad B \rightarrow bS \mid \varepsilon$

grammaire régulière réduite = $\quad S \rightarrow aB \mid bE \quad B \rightarrow bS \mid \varepsilon \quad E \rightarrow \varepsilon$

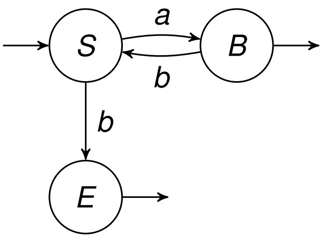
automate associé à G =

Grammaire régulière à droite $\rightarrow$ automate – Exemple

si G définie par $\quad S \rightarrow aB \mid b \quad B \rightarrow bS \mid \varepsilon$

grammaire régulière réduite = $\quad S \rightarrow aB \mid bE \quad B \rightarrow bS \mid \varepsilon \quad E \rightarrow \varepsilon$

automate associé à G =



Plan

1. Grammaires
   - Définition
   - Génération d'un langage

2. Grammaires régulières
   - Grammaires régulières à droite
   - Automate fini $\rightarrow$ grammaire régulière

3. Bilan sur les grammaires

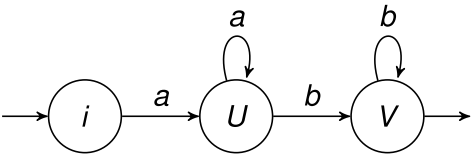
Automate fini $\rightarrow$ grammaire régulière

Étant donné un automate standard $\mathcal{A} = (A, Q, \{i\}, F, E)$, on peut définir la grammaire régulière à droite $G_{\mathcal{A}} = (A, Q, i, R)$ par :

- si $p \xrightarrow{a} q$, créer la règle $p \rightarrow aq$,
- si $p \in F$, créer la règle $p \rightarrow \varepsilon$.

Exemple :

symbole de départ = i

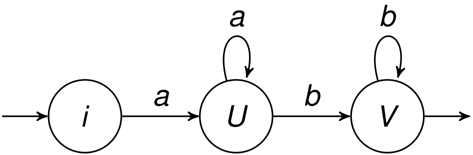


Automate fini $\rightarrow$ grammaire régulière

Étant donné un automate standard $\mathcal{A} = (A, Q, \{i\}, F, E)$, on peut définir la grammaire régulière à droite $G_{\mathcal{A}} = (A, Q, i, R)$ par :

- si $p \xrightarrow{a} q$, créer la règle $p \rightarrow aq$,
- si $p \in F$, créer la règle $p \rightarrow \varepsilon$.

Exemple :



symbole de départ = i

$$i \rightarrow aU$$
$$U \rightarrow aU \mid bV$$
$$V \rightarrow bV \mid \varepsilon$$

Plan

Répondre à des questions de générabilité

Pour montrer que $u \in L(G)$, on peut :

- trouver une dérivation valide ...
- ... en faisant si besoin une analyse descendante

Pour montrer que $u \notin L(G)$, on peut :

- montrer que l'analyse descendante échoue dur si G pas rég. droite
- raisonner sur le nombre de lettres
- raisonner sur les occurrences de certaines lettres

Manipulation de grammaires

Simplification de grammaires :

- $U \to a_1 \dots a_k V \rightsquigarrow U \to a_1 U_1 \quad \dots \quad U_{k-2} \to a_{k-1} U_{k-1} \quad U_{k-1} \to a_k V$
- $U \to V a_1 \dots a_k \rightsquigarrow U \to U_1 a_1 \quad \dots \quad U_{k-2} \to U_{k-1} a_{k-1} \quad U_{k-1} \to V a_k$
- $U \to V \rightsquigarrow U \to a \mid b W$ (si $V \to^* a \mid b W$)

- $U \to a \rightsquigarrow U \to a E \quad E \to \varepsilon$ réduction

Conversion entre grammaires et automates :

- grammaire régulière à droite $\to$ automate penser à réduire
- automate $\to$ grammaire régulière à droite standardiser l'automate