

# Travaux pratiques de mathématiques appliquées

## Langages formels et automates finis

### Solutions des exercices JFLAP

## 2. Grammaires régulières

### 2.2. Visualisation de la séquence de dérivation et de l'arbre de dérivation d'un mot appartenant au langage engendré par une grammaire

JFLAP : (ex3.1a.jff)

File Input Test Convert Help

Editor Brute Parser

Start Pause Step Derivation Table

Input: aqv<sub>x</sub>

String accepted! 7 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing the text field)

Table Text Size

LHS	RHS
S	→ aA
S	→ bA
S	→ aC
A	→ B
B	→ qvC
C	→ x
B	→ y
A	→ ε

Table Text Size

S	aA
A	B
B	qvC
C	x

Derived x from C. Derivations complete.

JFLAP : (ex3.1a.jff)

File Input Test Convert Help

Editor Brute Parser

Start Pause Step Noninverted Tree

Input: aqv<sub>x</sub>

String accepted! 7 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing the text field)

Table Text Size

LHS	RHS
S	→ aA
S	→ bA
S	→ aC
A	→ B
B	→ qvC
C	→ x
B	→ y
A	→ ε

Table Text Size

```

graph TD
    S((S)) --- a((a))
    S --- A((A))
    A --- B((B))
    B --- q((q))
    B --- v((v))
    B --- C((C))
    C --- x((x))
  
```

Derived x from C. Derivations complete.

### 2.3. Déterminer le langage engendré par une grammaire

Exercice 3.1 du livre « JFLAP Book »	
a)	<p>Acceptés : bb, bbb, bbbbbb, aabb, aaaabb, ...  Rejetés : aab, abb, aaabb, abbb, aaaaaabbbbbbbbbbb, ...  Langage des mots commençant un nombre pair (éventuellement nul) de 'a', suivis d'au moins deux 'b'.</p> $L(G) = \{a^{2n}b^p, n \geq 0, p \geq 2\}$ $ER_G = (aa)^*bb^*b$
b)	<p>Acceptés : baba, babab, babaaba, babaabab, babaabaaba, ...  Refusés : aba, abab, bbaba, bababb, babababa, ...  Langage des mots commençant par 'b', suivi d'une ou plusieurs fois 'aba' et terminés éventuellement par un seul 'b'.</p> $L(G) = \{b(aba)^nb^p, n \geq 1, p \leq 1\}$ $ER_G = b(aba)^*aba(b \lambda)$
c)	<p>Acceptés : bb, aabb, bbaa, abab, bbaaaa, aabaab, ...  Refusés : aa, aab, aabba, ababb, aabbaabb, ...  Langage des mots contenant exactement deux 'b' et un nombre pair (éventuellement nul) de 'a'.</p> $L(G) = \{a^nba^pba^q, (n+p+q) \bmod 2 = 0\}$ $ER_G = (aa)^*((aba b)(aa)^*(aba b) abba)(aa)^*$
d)	<p>Acceptés : b, bbb, bbbbb, aba, abababa, aabaabba, ...  Refusés : bb, bbbb, abba, ababa, aabbaabb, ...  Langage des mots contenant un nombre impair de 'b' et un nombre quelconque de 'a'.</p> $L(G) = \{(a^pba^qb)^na^rba^s, n, p, q, r, s \geq 0\}$ $ER_G = (a^*ba^*b)^*a^*ba^*$
e)	<p>Acceptés : a, aa, aaa, bb, bbb, bbbb, abba, abbabbba, aaaaaa, ...  Refusés : b, ab, aba, aabaab, abbaababb, ...  Langage des mots composés de 'a' et de 'b' dans lesquels chaque séquence de 'b' comporte au minimum deux 'b'.</p> $L(G) = \{(a^nba^qa^r)^p, n, q \geq 0, p = 0 \text{ ou } p \geq 2, r \geq 0\}$ $ER_G = (a^* bb^*b)^*$
f)	<p>Acceptés : ba, babbb, bbabb, bbabbbbb, bbbab, ...  Refusés : bab, bbab, bbba, bbbabb, abb, ...  Langage des mots comportant un seul caractère 'a' entouré d'un certain nombre de 'b', selon trois configurations avant/après possibles :</p> <ul style="list-style-type: none"> <li>• <math>\bmod 3 = 1 / \bmod 3 = 0</math></li> <li>• <math>\bmod 3 = 0 / \bmod 3 = 1</math></li> <li>• <math>\bmod 3 = 2 / \bmod 3 = 2</math></li> </ul> $L(G) = \{b^{3n+1}ab^{3p}\} \cup \{b^{3n+2}ab^{3p+2}\} \cup \{b^{3n}ab^{3p+1}\}, \text{ avec } n, p \geq 0$ $ER_G = (bbb)^*(ba(bbb)^* bba(bbb)^*bb ab(bbb)^*)$

## 2.4. Concevoir une grammaire qui engendre un langage donné

### Exercice 3.2 du livre « JFLAP Book »

Les fichiers « sol-ex3.2-a », « sol-ex3.2-b », « sol-ex3.2-b-bis », « sol-ex3.2-c », « sol-ex3.2-d », « sol-ex3.2-e », « sol-ex3.2-f » disponibles sur HELMo Learn donnent un exemple de solution possible pour chacun des six énoncés.

## 2.5. Transformer une grammaire algébrique en une grammaire régulière

### Exercice 3.4 du livre « JFLAP Book »

Les fichiers « sol-ex3.4-b », « sol-ex3.4-c », « sol-ex3.4-d / sol-ex3.4-d-bis », « sol-ex3.4-e » disponibles sur HELMo Learn donnent un exemple de solution possible pour chacun des quatre énoncés nécessitant une modification.

Dans le cas particulier de l'énoncé d, la première solution correspond au remplacement de la première production de la grammaire par  $S \rightarrow AB$ , la solution « bis » correspond à l'ajout d'une production  $S \rightarrow \varepsilon$ .

- a) Oui
- b) Non. Langage =  $(aa)^*(bb^*)b$
- c) Non. Langage =  $a^*b^*$
- d) Non. Langage =  $aa(aa)^*(bb)^*$       Langage bis =  $(aa(aa)^*(bb)^*)^*$
- e) Non. Langage =  $a \mid bb^*abb^*$

# Automates finis

## 2.2. Concevoir un automate qui reconnaît un langage donné

### Exercice 1.1 du livre « JFLAP Book »

Les fichiers « sol-ex1.1-a », « sol-ex1.1-b », « sol-ex1.1-c », « sol-ex1.1-d », « sol-ex1.1-e » disponibles sur HELMo Learn donnent un exemple de solution possible pour chacun des cinq énoncés.

## 2.3. Valider et modifier un automate

### Exercice 1.2 du livre « JFLAP Book »

Les fichiers « sol-ex1.2-a », « sol-ex1.2-b-DFA », « sol-ex1.2-b-NFA », « sol-ex1.2-c-DFA », « sol-ex1.2-c-NFA », « ex1.2-d » disponibles sur HELMo Learn donnent un exemple de solution possible pour chacun des quatre énoncés.

Pour les énoncés b et c, deux solutions sont proposées : une solution déterministe (DFA) et une solution non déterministe (NFA).

**Exercice 1.3 du livre « JFLAP Book »**

Les fichiers « sol-ex1.3-a », « sol-ex1.3-b », « sol-ex1.3-c-NFA », « sol-ex1.3-c-DFA » disponibles sur HELMo Learn donnent un exemple de solution possible pour chacun des trois énoncés.

Pour l'énoncé c, deux solutions sont proposées : une solution non déterministe (NFA) et une solution déterministe (DFA).

## 2.4. Équivalence entre grammaires régulières et automates finis

**Exercice 3.5 du livre « JFLAP Book »**

L'outil « Convert Right-Linear Grammar to FA » de JFLAP vous permet d'obtenir la solution finale automatiquement. Il suffit de cliquer sur le bouton « Show All ».

**Exercice 3.6 du livre « JFLAP Book »**

L'outil « Convert to Grammar » de JFLAP vous permet d'obtenir la solution finale automatiquement. Il suffit de cliquer sur le bouton « Show All ».

## 2.5. Équivalence de langages

**Exercice 1.4 du livre « JFLAP Book »**

Les fichiers « sol-ex1.4-a », « sol-ex1.4-b », « sol-ex1.4-c », « sol-ex1.4-d », « sol-ex1.4-e » disponibles sur HELMo Learn donnent un exemple d'automate possible pour chacun des cinq langages.

On constate que les automates a, b et d sont équivalents. Il en va de même des langages correspondants. En revanche, les automates (et langages) c et e ne sont équivalents à aucun autre.

## 2.6. Automates non déterministes

**Exercice 2.1 du livre « JFLAP Book »**

L'outil « Convert to DFA » de JFLAP vous permet d'obtenir la solution finale automatiquement. Il suffit de cliquer sur le bouton « Complete ».

**Exercice 2.3 du livre « JFLAP Book »**

Le fichier « sol-ex2.3 » disponible sur HELMo Learn contient la solution de cet exercice.