

Merci de ne pas rédiger les parties 1 et 2 sur les mêmes feuilles; elles seront corrigées par deux examinateurs différents.

Le soin apporté à la qualité de rédaction et de présentation des réponses sera pris en compte.

Partie 1

Exercice 1.1 – Syntaxe (3 points)

Les automates à nombre fini d'états ne permettent pas de reconnaître les langages palindromes, c'est-à-dire les langages ne contenant que des séquences qui peuvent se lire indifféremment de gauche à droite ou de droite à gauche. Cet argument est défendu par Noan Chomsky en 1957 pour introduire les grammaires algébriques (les grammaires CFG) comme modèle syntaxique.

1. Expliquer pourquoi il serait impossible de construire un automate à nombre fini d'états permettant de générer un langage palindrome, et seulement celui-ci.

Correction

Faisons l'hypothèse qu'il existe un automate à nombre fini d'états qui reconnaît les langages palindromes. Cet automate doit permettre de construire un chemin $c_1 = (q_1, q_2, \dots, q_i)$ reconnaissant w où q_1 est l'état initial. Il doit également permettre de construire un chemin $c_2 = (q_{i+1}, q_{i+2}, \dots, q_{2i})$ reconnaissant w^{-1} où q_{2i} est un état final.

Si les chemins c_1 et c_2 ne contiennent pas de cycles, alors il doit y avoir une infinité de chemins possibles pour reconnaître tous les ww^{-1} , qui sont en nombre infini. Soit ils en contiennent, et il est alors impossible de s'assurer que c_1 et c_2 soient de mêmes longueurs. Les deux cas sont contradictoires avec nos hypothèses.

2. Étant admis l'impossibilité de se satisfaire des grammaires rationnelles pour exprimer la syntaxe d'une langue, est-il toutefois possible de se satisfaire des grammaires algébriques? Pour quelles raisons?

Correction

Les grammaires algébriques, encore appelées *Contex-Free Grammars* permettent de décrire la structure syntagmatique des phrases (les groupes de mots). Ceci est insuffisant pour décrire l'ensemble des propriétés syntaxiques d'une phrase pour les raisons suivantes :

- Deux phrases dont les sens sont proches peuvent être représentés par des structures syntagmatiques très différentes. C'est par exemple le cas d'une phrase à l'actif et son équivalent au passif en français.
- Si l'on étend une grammaire CFG pour décrire des phénomènes d'accord, il est alors impossible de traiter indépendamment les régularités syntagmatiques.
- Certaines langues comme le suisse-allemand (cette connaissance précise n'était pas demandée à l'examen) permettent de construire des structures qui se formalisent par les langages $\{\omega\omega/\omega \in \Sigma^*\}$. Il n'est pas possible de construire une grammaire CFG qui engendre un tel langage.

Exercice 1.2 – Logique du 1^{er} ordre (3,5 points)

1. Montrer que la formule suivante est une tautologie :

$$((a \Rightarrow b) \wedge ((c \vee \neg b) \Rightarrow a)) \Rightarrow (b \vee c)$$

Correction

			D	E	F	G	H	I	
<i>a</i>	<i>b</i>	<i>c</i>	$a \Rightarrow b$	$\neg b$	$c \vee \neg b$	$\mathbf{F} \Rightarrow a$	$\mathbf{D} \wedge \mathbf{G}$	$b \vee c$	$\mathbf{H} \Rightarrow \mathbf{I}$
0	0	0	1	1	1	0	0	0	1
0	0	1	1	1	1	0	0	1	1
0	1	0	1	0	0	1	1	1	1
0	1	1	1	0	1	0	0	1	1
1	0	0	0	1	1	1	0	0	1
1	0	1	0	1	1	1	0	1	1
1	1	0	1	0	0	1	1	1	1
1	1	1	1	0	1	1	1	1	1

2. Traduire sous forme de clauses de Horn les expressions suivantes :
 - (a) *Luc* est un homme ou un garçon

Correction

male(Luc)

- (b) *Jean* est le frère de *Marie*

Correction

sibling(Jean, Marie)

- (c) *Luc* est le fils de *Marie*

Correction

parent(Marie, Luc)

- (d) *Jeanne* est la fille de *Jean*

Correction

parent(Jean, Jeanne)

- (e) X est le cousin de Y si X est un homme ou un garçon et si Q est le père ou la mère de X , si R est le père ou la mère de Y et si Q et R sont frères ou sœurs.

Correction

$\text{cousin}(x, y) \Leftarrow \text{male}(x) \wedge \text{parent}(x, q) \wedge \text{parent}(x, r) \wedge \text{sibling}(q, r)$

Exercice 1.3 – Prolog (3,5 points)

1. Soit le code Prolog suivant :

```
mysterious([], L2, L2).  
mysterious([X|L1], L2, [X|R]) :- mysterious(L1, L2, R).
```

- (a) Montrer qu'il est possible de prouver la clause suivante.

mysterious([a, b, c], [d, e, f], X).

Correction

Modifions la clause au nom de variable près :

$\text{mysterious}([a, b, c], [d, e, f], Z)$.

Appliquons à

$\text{mysterious}([X|L1], L2, [X|R]) :- \text{mysterious}(L1, L2, R)$.

$X = a$

$L1 = [b, c]$

$L2 = [d, e, f]$

$Z = [a|R]$

Cette clause est vérifiée si celle-ci l'est :

$\text{mysterious}([b, c], [d, e, f], R)$

Appliquons à

$\text{mysterious}([X'|L1'], L2', [X'|R']) :- \text{mysterious}(L1', L2', R')$.

$X' = b$

$L1' = [c]$

$L2' = [d, e, f]$

$R = [b|R']$

$\text{mysterious}([c], [d, e, f], R')$

Appliquons à

$\text{mysterious}([X''|L1''], L2'', [X''|R'']) :- \text{mysterious}(L1'', L2'', R'')$.

$X'' = c$

$L1'' = []$

$L2'' = [d, e, f]$

$R' = [c|R'']$

$\text{mysterious}([], [d, e, f], R'')$

Appliquons à

$\text{mysterious}([], L2''', L2''')$

$\text{mysterious}([], L2''', L2''')$

$L2''' = [d, e, f]$

(b) Quelle est la valeur de X?

Nous l'avons noté Z

$L2''' = [d, e, f]$

$R'' = L2'''$

$R' = [c|R'']$

$R = [b|R']$

$Z = [a|R]$

$Z = [a, b, c, d, e, f]$

(c) Est-il possible de prouver la clause suivante? Pourquoi?

$mysterious([a, b, c], -, [d, e, f])$.

Correction

Non, ce n'est pas possible car il est impossible d'instancier $mysterious([], L2, L2)$.
ou $mysterious([X|L1], L2, [X|R])$ par $mysterious([a, b, c], -, [d, e, f])$

2. Exprimer sous forme d'une phrase en langage naturel le programme Prolog suivant :

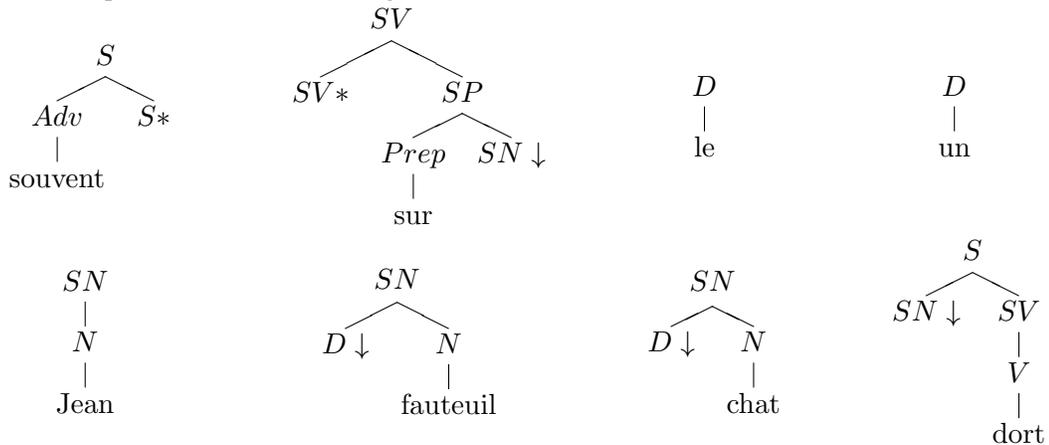
```
decompose_deux(X) :- zero(X) , ! , fail .
decompose_deux(X) :- pair(X) .
```

Correction

Tous les pairs sont *décomposable_deux*, sauf zéro. Il s'agit de la décomposition d'un nombre pair par une puissance de deux.

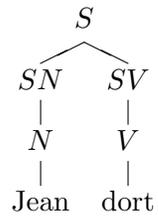
Exercice 1.4 – TAG (3 points)

Soit la grammaire d'arbres adjoints suivante :



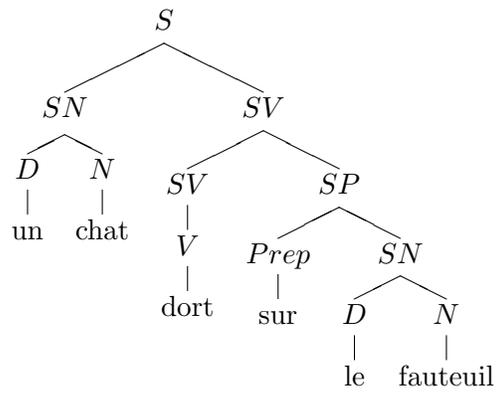
1. Donner l'arbre dérivé pour les phrases suivantes :
— *Jean dort.*

Correction



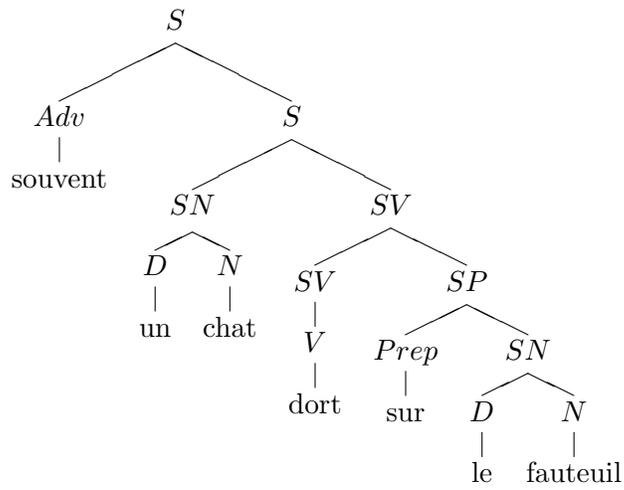
— *Un chat dort sur le fauteuil.*

Correction



— *Souvent, Jean dort sur le fauteuil.*

Correction



2. Donner des exemples de phrases n'ayant pas de sens (courant) en français engendrées par cette grammaire.

Correction

- un fauteuil dort sur le chat
- un chat dort sur un fauteuil sur un chat sur Jean sur un fauteuil

3. Le langage des phrases engendrées par cette grammaire est-il fini ? Pourquoi ?

Correction

Non, il est infini car quelque soit la phrase engendrée par la grammaire, il est toujours possible d'adjoindre un arbre auxiliaire supplémentaire pour produire une phrase plus longue.

Partie 2

Exercice 2.1 – CYK (3 points)

Soit la grammaire suivante, reconnaissant le langage $L_7 = \{uvw\}$ avec u un mot comportant un nombre pair et non nul de a , v comportant au moins un a et un b , w comportant n a suivis de n b (pour $n > 0$). Cette grammaire est donnée sous forme normale de Chomsky :

$$G_7 = \left\{ \begin{array}{l} S \rightarrow UR \\ U \rightarrow UU \\ U \rightarrow AA \\ R \rightarrow VW \\ V \rightarrow VV \\ V \rightarrow VA \\ V \rightarrow VB \\ V \rightarrow AB \\ V \rightarrow BA \\ W \rightarrow AX \\ W \rightarrow AB \\ X \rightarrow WB \\ A \rightarrow a \\ B \rightarrow b \end{array} \right.$$

- Donner trois exemples de mots engendrés par cette grammaire.

Correction

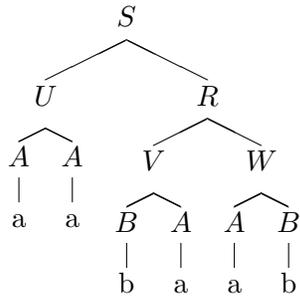
- $S \Rightarrow UR \Rightarrow UUR \Rightarrow AAUR \Rightarrow AAAAR \Rightarrow AAAAVW \Rightarrow$
 $AAAAABW \Rightarrow AAAAABAB \Rightarrow^* aaaaabab$
- $S \Rightarrow UR \Rightarrow AAR \Rightarrow AAVW \Rightarrow AABAW \Rightarrow AABAAB \Rightarrow^*$
 $aabaab$
- $S \Rightarrow UR \Rightarrow AAR \Rightarrow AAVW \Rightarrow AAVBW \Rightarrow AABABW \Rightarrow$
 $AABABAX \Rightarrow AABABAWB \Rightarrow AABABAABB \Rightarrow aababaabb$

— Analyser le mot *aabaab* en utilisant l'algorithme de Cocke-Kasami-Younger (CKY ou CYK).

Correction

6	S					
5		V/R				
4		V	V/R			
3		V	V			
2	U	V/W	V	U	V/W	
1	A	A	B	A	A	B
1	a	a	b	a	a	b

Ce qui donne l'arbre de dérivation suivant :



Exercice 2.2 – Ambiguïté, Modèles de Markov Cachés (4 points)

Voici un exemple d'ambiguïté syntaxique en Anglais : considérons les mots *time*, *fly*, *like*. Chacun de ces trois mots dispose (au moins) de deux catégories syntaxiques associées à des sens différents : *time* est le nom commun *temps*, le verbe *chronométrer*, ainsi que l'adjectif *temporel* ; *fly* est le nom commun *mouche* et le verbe *to fly* (voler, fuir, passer), *like* est le verbe *aimer* et la préposition *comme*. Ainsi, la phrase *Time flies like an arrow* Est (normalement) traduite par *le temps vole comme une flèche*, mais dispose également de trois autres interprétations (moins intuitives).

Pour cet exercice, voici une grammaire qui engendre cette phrase dans toutes ses interprétations :

$$G_8 = \left\{ \begin{array}{l} S \rightarrow SN SV \\ S \rightarrow SV \\ SN \rightarrow N \\ SN \rightarrow DET N \\ SN \rightarrow A N \\ SN \rightarrow N SP \\ SV \rightarrow V SP \\ SV \rightarrow V SN \\ SV \rightarrow V SN SP \\ SP \rightarrow P SN \\ N \rightarrow time \\ N \rightarrow flies \\ N \rightarrow arrow \\ DET \rightarrow an \\ V \rightarrow flies \\ V \rightarrow like \\ V \rightarrow time \\ P \rightarrow like \\ A \rightarrow time \end{array} \right.$$

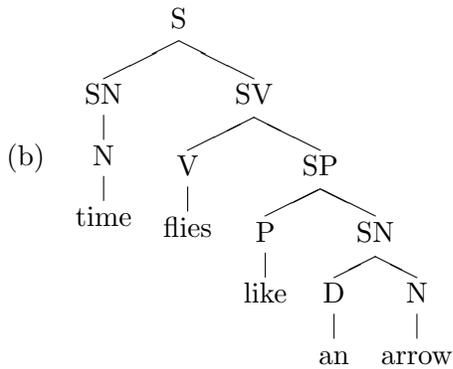
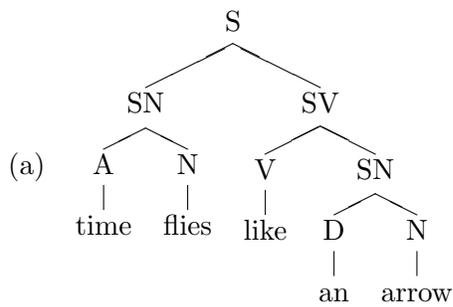
1. Donner les trois autres interprétations possibles de la phrase (notez que *an arrow* est univoque et sera toujours interprété comme *une flèche*).

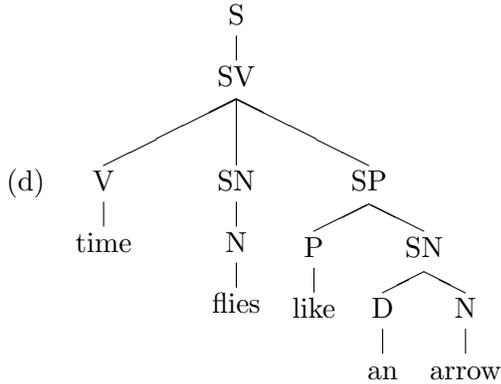
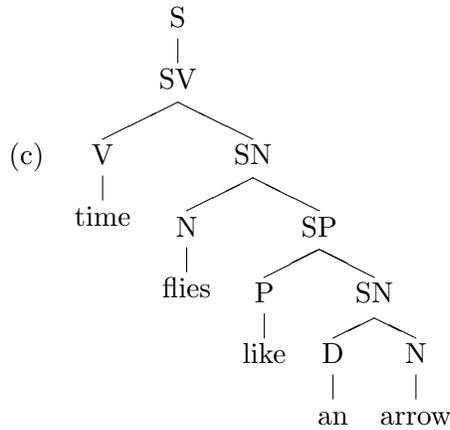
Correction

- (a) Les mouches fugaces apprécient une flèche (*Time(A) flies(N) like an arrow*)

- (b) Chronométrez les mouches qui sont comme une flèche (*Time(V)*
flies(N) like an arrow)
- (c) Chronométrez les mouches, à la manière d'une flèche (*Time(V)*
flies(N), like an arrow)
2. Donner les quatre arbres d'analyse syntaxique (différents) de la phrase,
 en utilisant la grammaire précédente.

Correction





3. Voici un Modèle de Markov Caché correspondant aux catégories mises en œuvre précédemment :
- En utilisant la méthode de Viterbi, donnez le chemin le plus probable de selon ce HMM pour cette phrase.

Correction

Les quatre arbres correspondent à ces trois séquences d'étiquettes (les deux derniers arbres ont la même frontière) :

- (a) A N V DET N
- (b) N V P DET N
- (c) V N P DET N

Calculons les probabilités de chaque chemin. Nous convertissons en fractions de 10 pour simplifier les calculs.

- (a) A N V DET N : $(2/10 \times 10/10) \times (10/10 \times 1/10) \times (5/10 \times 4/10) \times (4/10 \times 10/10) \times (10/10 \times 5/10) = \frac{8 \cdot 10^6}{10^{10}} = 8 \cdot 10^{-4}$

(b) N V P DET N : $(5/10 \times 4/10) \times (5/10 \times 5/10) \times (5/10 \times 10/10) \times (10/10 \times 10/10) \times (10/10 \times 5/10) = \frac{125 \cdot 10^6}{10^{10}} = 125 \cdot 10^{-4}$

(c) V N P DET N : $(3/10 \times 1/10) \times (1/10 \times 1/10) \times (2/10 \times 10/10) \times (10/10 \times 10/10) \times (10/10 \times 5/10) = \frac{3 \cdot 10^5}{10^{10}} = 3 \cdot 10^{-5}$

Le chemin de Viterbi de plus probable est celui qui maximalise la probabilité, c'est-à-dire le second.

— Donnez les probabilités des quatre étiquetages possibles.

Correction

Nous avons répondu à cette question pour répondre à la précédente.

