

Interface syntaxe - linguistique computationnelle

Examen final

Aucun document autorisé ; Durée : 1 heure 30.

1. Soit le langage L_1 sur le vocabulaire $V = \{l', \text{homme}, \text{qui}, a, \text{vu}, \text{ours}\}$ formé de l'ensemble des phrases finies de la forme *l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu ... qui a vu l'ours*.

- (a) Donnez une grammaire algébrique (*context-free*) engendrant L_1 .
- (b) Donnez une grammaire régulière* engendrant L_1 .
- (c) Soit L_2 le langage engendré par la grammaire :

S	→	NP Rel
NP	→	l'homme
		l'ours
Rel	→	qui a vu NP Rel
		qui a vu l'ours

Décrire informellement les différences entre les langages L_1 et L_2 .

(d) **[Bonus (difficile)]** Donner la grammaire du langage $L_2 \setminus L_1$ (mots de L_2 qui ne sont pas dans L_1).

2. Soit la grammaire

S	→	NP VP	Det	→	un le
NP	→	Det N'	N	→	chat chien
VP	→	V_i	V_i	→	dort ronfle
N'	→	N' Rel			
		N			
Rel	→	qui V_i			

- (a) Donner l'arbre syntaxique de la phrase (1)
 - (1) le chat qui dort ronfle
- (b) Dessiner la partie pertinente de l'arbre d'exploration des solutions* que donnerait une analyse descendante de la phrase précédente. Commenter.
- (c) Un non-terminal comme N' dans cette grammaire n'est productif que s'il se dérive (exactement une fois) en N . En d'autres termes, toutes les chaînes produites par N' sont composées d'un N suivi d'un nombre quelconque de Rel . Redéfinir (éventuellement en introduisant un nouveau non-terminal) les règles de productions de N' pour que l'analyse descendante soit possible.

3. On s'intéresse maintenant à la construction d'une représentation sémantique (en logique) au moyen du lambda-calcul typé vu en cours.

- (a) Donner les représentations "lexicales" et le détail des combinaisons pertinentes pour la phrase *un chat ronfle* (avec la grammaire précédente).
- (b) La représentation de *le chat ronfle*, d'après Russel, est (2)*. Quelle λ -expression faut-il associer à *le* pour produire ce résultat ?
 - (2) $\exists x (\text{chat}(x) \wedge \forall y (\text{chat}(y) \rightarrow y = x) \wedge \text{ronfle}(x))$
- (c) Considérons à nouveau la phrase (1) et son arbre syntaxique. Indiquer, pour chaque nœud du sous-arbre de racine N' , le type qu'il doit avoir pour que N' soit de type $\langle e, t \rangle$.
- (d) On suppose que la représentation de cette phrase est (3). Quelle λ -expression doit-on associer à *qui* pour produire un tel résultat ?
 - (3) $\exists x (\text{chat}(x) \wedge \text{dort}(x) \wedge \forall y ((\text{chat}(y) \wedge \text{dort}(y)) \rightarrow y = x) \wedge \text{ronfle}(x))$
- (e) **[Bonus]** La représentation précédente suppose que l'on a affaire à une relative *restrictive*. Comment pourrait-on prendre en compte le cas d'une relative *appositive*, comme dans : *le chat, qui dort, ronfle* (en agissant éventuellement au niveau de la grammaire) ?

*** Rappels**

grammaire régulière grammaire dont toutes les productions engendrent exactement un symbole terminal, et au plus un symbole non terminal (sans compter les éventuelles ϵ -productions).

Arbre d'exploration des solutions : pour la grammaire $S \rightarrow aSbS|aS|c$, et le mot *aacbc*, l'arbre ressemble à :

notation de Russel $\exists x (\text{chat}(x) \wedge \forall y (\text{chat}(y) \rightarrow y = x))$ indique l'unicité du référent.

On ne demande pas de démonstrations, mais de brèves justifications seront fournies le cas échéant.