

**Structures informatiques et logiques pour la modélisation
linguistique (C-2-27-1)
Examen 2008**

A

Soit la syntaxe abstraite suivante, à laquelle est associée une sémantique à la Montague :

ALICE : np
 SOMEONE : np
 LEFT : $np \rightarrow s$
 THINK : $cc \rightarrow np \rightarrow s$
 THAT : $(np \rightarrow s) \rightarrow (np \rightarrow cc)$

$\llbracket \text{ALICE} \rrbracket = \lambda P. P \text{ alice}$
 $\llbracket \text{SOMEONE} \rrbracket = \lambda P. \exists x. P x$
 $\llbracket \text{LEFT} \rrbracket = \lambda S. S (\lambda x. \text{left } x)$
 $\llbracket \text{THINK} \rrbracket = \lambda C. \lambda S. S (\lambda x. C (\text{think } x))$
 $\llbracket \text{THAT} \rrbracket = \lambda P. \lambda s. \lambda f. P (\lambda p. s (\lambda x. f (p x)))$

où :

alice : ι
left : $\iota \rightarrow o$
think : $\iota \rightarrow (o \rightarrow o)$

1. Calculer la représentation sémantique de la phrase *Alice thinks that someone left*, dont la syntaxe abstraite est donnée par le terme suivant :

THINK (THAT LEFT SOMEONE) ALICE

2. Le terme sémantique assigné à THINK induit une interprétation *de re*. Donner un terme sémantique alternatif qui induit l'interprétation *de dicto*.

3. Soit les trois phrases suivantes :

Alice thinks that every man is mortal (1)
Alice thinks that Socrates is a man (2)
Alice thinks that Socrates is mortal (3)

Donner un axiome sur la modalité **think** qui permette d'inférer (3) à partir de (1) et (2).

B

Un texte est une suite de phrases terminées par un point. Une phrase est une suite de mots séparés par des espaces, le premier étant capitalisé. Un mot est une suite non vide de lettres minuscules, un mot capitalisé est une lettre majuscule suivie d'une suite de lettres minuscules, le mot associé est obtenu en remplaçant la majuscule initiale par la minuscule correspondante.

1.

Donner un automate fini reconnaissant un texte.

En déduire une machine qui construise le vocabulaire d'un texte (ensemble des mots qui le composent) sous forme d'un arbre lexical. Cette machine est-elle finie?

2.

La modifier pour qu'elle produise en sortie la liste de ses phrases, chaque phrase étant précédée d'une astérisque ssi un même mot y apparaît deux fois.

3.

On rajoute des parenthèses ouvrantes '(' '[' '{' et fermantes ')' ']' '}'. Une phrase bien construite est une phrase où les parenthèses sont correctement emboîtées. Construire une machine reconnaissant les phrases bien construites. Cette machine est-elle finie?

Modifier cette machine pour qu'elle produise en sortie la liste de ses phrases bien construites, les phrases mal construites étant tronquées par le caractère '!' suivi par les parenthèses minimales nécessaires à la bonne construction du segment initial maximal ainsi complété.

4.

On suppose un lexique des mots (fléchis) de la langue, donné sous forme d'un arbre lexical décoré par une catégorie lexicale (nom, adjectif, pronom, verbe, adverbe, article). Transformer la machine donnée en 1 pour qu'elle produise un texte étiqueté par les catégories des mots qui la composent. Exemple : <Voici : adverbe> <une : article> <phrase : nom>.