

## 1 Tense logic

This is Exercise 3 on Exercise Sheet 3.

## 2 Intensional logic

This is Exercise 4 on Exercise Sheet 3.

## 3 \* Branching time

This is Exercise 5 on Exercise Sheet 3.

## 4 Nested Cooper Storage

The following sentence contains a scope ambiguity involving three quantifiers.

Every researcher of a company sees a sample.

- (a) Use Nested Cooper Storage to derive as many readings of the sentence as possible.

The syntactic structure of the subject is (NP (Det Every) ( $\bar{N}$  (N researcher) (PP (P of) (NP a company)))). You can use a constant **see\*** of type  $\langle e, \langle e, t \rangle \rangle$  to represent the transitive verb and a term  $\mathbf{of}' = \lambda x \lambda F \lambda y. \mathbf{of}^*(y)(x) \wedge F(y)$  of type  $\langle e, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle$  to represent the preposition. Abbreviate recurring terms appropriately.

- (b) Give an intuitively plausible reading of the sentence that you couldn't derive this way. Try to explain why it isn't possible to derive it.

## 5 Extending Nested Cooper Storage

One limitation of Nested Cooper Storage is that it can only deal with scope-bearing elements that are noun phrases, and not with adverbs, negation, etc. For instance, it will only derive one reading for the sentence “Every student didn't pass-the-exam.”

- (a) Add new rules to Nested Cooper Storage that allow you to store and retrieve negations. You can analyse the word “didn't” syntactically as a VP modifier; its semantic representation should be of type  $\langle t, t \rangle$ .

- (b) Use your extended Nested Cooper Storage to derive the two readings of the sentence “Every student didn’t pass-the-exam.” You may abbreviate recurring subterms appropriately.
- (c) Your analysis probably suffers from the insensitivity of Nested Cooper Storage towards scope islands, and predicts six readings for the sentence “John says that every student didn’t pass-the-exam,” where “says” is a sentence-embedding verb whose semantic representation has type  $\langle t, \langle e, t \rangle \rangle$ . One of these (clearly wrong) extra readings is synonymous with “John doesn’t say that every student passed-the-exam.”

Fix this by modifying the rules of Nested Cooper Storage, in such a way that the quantifier store must be emptied (of all quantifiers and negations) at each sentence node.

---

To be turned in by 19/05/2005, 11:15 am