UNIVERSITY OF MALTA
BOARD OF STUDIES FOR INFORMATION TECHNOLOGY

Department of Computer Science & A.I.
B.Sc. (Hons.) I.T. III Year 1997/98

January 1998 Assessment Session

2nd February 1998

9.00 a.m.– 10.30 a.m.

CSA305 - Natural Language Processing (1 credit)

Answer **three** *questions. Be sure to provide* **adequate exemplification** *to illustrate your understanding of concepts discussed even if not explicitly asked to do so.*

1. (a) Give a critical account of Weizenbaum's ELIZA program. Your account should mention why it was written, how it works, what information is associated with keywords, and how its performance could be improved.

   (b) (i) What are the basic principles underlying the design of Winograd's SHRDLU?

   (ii) SHRDLU generated the responses in italics below. Go through the dialogue and identify the various kinds of problem (e.g. syntactic, semantic, other) resolved by the system in order to generate each response.

> Pick up a red block
> *OK.*
> Find a block which is taller than the one you are holding and put it in the box
> *OK.*
> What does the box contain?
> *The blue pyramid and the blue block.*
> What is the pyramid supported by?
> *The box.*
> How many blocks are not in the box?
> *Four of them.*
> Is at least one of them narrower than the one which I told you to pick up?
> *Yes, the red cube.*
> Is it supported?
> *Yes, by the table.*
> When did you pick it up?
> *While I was stacking up the red cube, a large red block and a large green cube.*
> Why?
> *To get rid of it.*
> Why did you do that?
> *To clean off the red cube.*

2. (a) In what ways do natural languages (i) resemble (ii) differ from artificial languages.

   (b) What is a context free grammar? What features of natural languages do you consider to be (i) easy to capture (ii) difficult to capture using context free grammar.

   (c) The following two grammars clearly generate the same strings. Discuss the ways in which they are not equivalent.

grammar 1:		**grammar 2:**
`np --> n`	`versus`	`np -> n`
`np --> np n`		`np -> n np`

   (d) What is the difference between a grammar, a recogniser, and a parser? Sketch the design of a general top-down recogniser (including descriptions of parse states) and show how it can be customised to yield depth first or breadth first behaviour. Describe circumstances which cause non-terminating behaviour with (i) a depth-first algorithm and (ii) a breadth-first algorithm

   (e) Describe the operation of a shift-reduce parser. Show how it is possible, in Prolog, to store the grammar rules in order to simplify the operation of this particular parsing algorithm. Give a *simple* grammar/lexicon for the sentence *Mike cooked spaghetti* and show how the shift/reduce algorithm would parse it.

3. (a) Of what use, in general, is a grammar formalism? What aspects of an NLP problem does the choice of a particular formalism usually determine?

   (b) What features are offered by the "Definite Clause Grammar" (DCG) formalism? What are its strongest and weakest points?

   (c) Explain how a DCG grammar rule is translated into a Prolog clause. Give translations for the following DCG rules into Prolog clauses.

      1. `s --> np, vp.`
      2. `n --> [X], { noun(X) }.`

   What is the significance of the curly brackets in the second rule?

   (d) How is text represented, and how is it connected to the grammar?

   (e) Write a DCG grammar able to derive noun phrases of the following class:
   a man
   three men
   three large men
   three large green men
   three large green mean ugly two-headed men
   the large green mean ugly two-headed man

   N.B. Indicate the lexical information available. Pay attention to the types of the adjectives, and try to take account of their "pecking order". The grammar should therefore exclude strings such as *a green large man*

   (f) Show how the grammar can be modified to create a parse tree that shows the syntactic structure of the constituents.

4 (a) Write short notes on the following

(i) Logical Form

(ii) Compositionality Principle

(iii) Lambda Calculus

(b) Explain the principles according to which a grammar can be designed to derive the logical form of sentences involving a proper noun and an intransitive verb, e.g. *Uther sleeps.*

(c) How might these principles be realised within a DCG?

(d) What are the problems associated with the semantics of quantified noun phrases and how can they be addressed?