UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 1992/93

April 1993

CSI 504 - PEMPROSESAN BAHASA MANUSIA
[NATURAL LANGUAGE PROCESSING]

Masa: [3 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **5** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
- Jawab **SEMUA** soalan (terdapat **3** soalan).

Answer **ALL** questions (there are **3** questions altogether).

1. a. Given the following sentence:

Malaysians love durians

(i) Give suitable representation structures for the following levels of interpretation:

- morpho-syntactic categories (NP, VP, ..., etc.);
- syntactic functions (SUBJ, ..., etc.);
- logico-semantic relations (ARG0,..., AGENT,..., etc.).

(ii) Indicate the morphological information for each word in the sentence, giving at least the following information whenever appropriate (this would be as in the result of a morphological analysis):

occurence, lemma, syntactic category, tense, syntactic number, semantic number, semantic category, valency information.

(iii) Write a suitable set of phrase-structure rules to analyse the morpho-syntactic structure you have given above, and then define the corresponding context-free grammar G = [.....].

(iv) Modify your context-free grammar to include the following sentence in the language described:

Indonesians love Malaysians

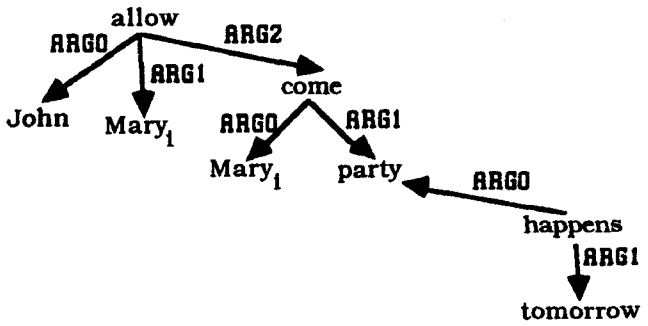
With this addition, enumerate all the sentences in the language L(G) described by the grammar.

[50/100]

b. Given that the following are logical predicates assigned to words:

allowed : allow(X, Y, **to** Z)
come : come(X, **to** Y)
tomorrow : happens(X, tomorrow)

Study the logical structure given below for the sentence:*John allowed Mary to come to the party tomorrow.* which makes use of the said predicate assignments. Note the index i to denote identical elements, and the direction of the arrows pointing to the arguments of a given predicate.



(i) Now suppose the following are additional logical predicate assignments to words:

gave : give(X, **to** Y, Z)
permission : allow(X, Y, **to** Z)
and : and(X, Y)

Using the approach in the example above, give the appropriate logical structures for each of the following sentences:

John gave a book to Mary
Mary allowed John and Susan to come to the party
John gave permission to Mary to come to the party tomorrow

(ii) Based on all the above, what would be the predicative assignments to the words involved as well as the logical structure for the following sentence:

Last year, the librarian gave permission to students to enter the library at anytime, but soon after, he caught a student stealing a book.

[note: you can use compound words whenever appropriate, for example,
last year : happens(X, last_year)]

[30/100]

c. Independent of all the above, what would be an appropriate predicate calculus logical expression(s) for the following sentence (here, you need to define your predicates and use quantifiers), and then suggest a tree structure representation(s) for the expression(s) you give:

All Malaysians love an Indonesian

[20/100]

2. a. The grammar with the following context free rules defines simple arithmetic expressions, where E is an axiom or start symbol, T and F are non-terminals, while a, +, *, (,) are terminals.

(1) **E -> E + T**
(2) **E -> T**
(3) **T -> T * F**
(4) **T -> F**
(5) **F -> (E)**
(6) **F -> a**

Construct the LR(1) parsing table for this grammar using the SLR method
[hint: follow(E) = { +,), - | } and follow(F) = { *, +,), - | }, ...but do check these out..]

[50/100]

b. (i) What is the difference between a recogniser and a parser?

(ii) What kind of grammar is said to be ambiguous?

(iii) In relation to LR parsing,

- how does one determine whether a given grammar is unambiguous based on the LR parsing table constructed for the grammar?
- is the condition you give necessary and/or sufficient?
- is the grammar for arithmetic expressions given above ambiguous?

(iv) Write a very short essay (not exceeding one page) on parsing natural languages as opposed to parsing computer languages. In your write-up, address at least the following issues:

- what is the main objective of parsing?
- what are the fundamental differences between parsing computer languages and parsing natural languages?
- what are the tools currently available for parsing?

[50/100]

3. a. Let R0 - R12 be transformational rules given in standard parantheses form, with $A and $B being variables instantiable to strings of trees and the terminals written in bold:

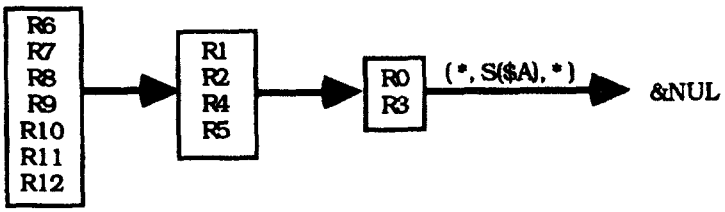
R0: S(NP($A), VP($B)) --> NP($A) VP($B)
R1: VP($A, NP($B)) --> v($A) NP($B)
R2: VP($A, AP($B)) --> v($A) AP($B)
R3: VP(works, ADVP($A, $B)) --> VP(works, NP(AP($A), $B))
R4: NP($A, $B) --> det($A) NP($B)
R5: NP(AP($A), $B) --> AP($A) NP($B)
R6: NP($A) --> n($A)
R7: AP($A) --> a($A)
R8: n(student) --> **student**
R9: n(hours) --> **hours**
R10: v(works) --> **works**
R11: a(odd) --> **odd**
R12: det(a) --> **A**

(i) Using these rules for analysis, draw the corresponding search tree indicating the rules applicable and applied (and not more) on the arcs, and the partial structures obtained in the nodes, for a bottom-up strategy on the sentence:

A student works odd hours.

(ii) Show clearly in your search tree above all possible representation trees for the given sentence.

(iii) Show also in the search tree the sequence of application of rules if the following is a ROBRA-like control graph provided for the said set of rules. Indicate on the arcs the sequence 1,2,...,m,.....,n and then move to ma, (m+1)a, (m+2)a,..... if the analysis backtracks to the point just before the application of the m-th step. For instance, if step 4 is done by rule R9 and the system backtracks to undo R9 and then applies R10, then this latter application is step 4a since it replaces step 4. Recall also that a ROBRA-like control finds the first solution and then stops. Indicate which representation tree this is in your search tree.



Here, each elementary grammar is applied iteratively according to the sequence of rules given until no further rules are applicable. There are no conditions on the arcs except for the last one before &NUL, to mean that the analysis is only successful if the object matches the structure specified on the arc. A failure to satisfy an arc condition forces a backtracking which undoes the previous rule application and tries the next rule (or next grammar if no further rule is applicable).

(iv) Indicate how to make minimal changes to the control graph in order to obtain the other solution(s), also giving the sequence of application of rules.

[50/100]

b. Given the following entries in an English->Malay dictionary with annotations following each entry, the details of which are explained further below:

alluvial *adj.* aluvium, lanar: *alluvial dam*, empangan alluvium; *alluvial tin*, timah lanar; *alluvial plain*, dataran aluvium.
{ **ee** *cat.* me, me: *ex*, mx; *ex*, mx; *ex.*, mx }
alluvium *n.* lanar, aluvium.
{ **ee** *cat.* me, me. }

ally *n. one united to another for common purposes,* **a.** *(of country)* sekutu: *France was an ally of Britain during World War II,* Perancis adalah sekutu Britain semasa Perang Dunia Kedua; **b.** *(of person)* sekutu, bersekutu: *the two men were allies in the Union dispute,* kedua-dua lelaki itu bersekutu dalam pertikaian Kesatuan;
vi. bersekutu: *they allied to fight the enemy,* mereka bersekutu untuk menentang musuh;
vi. **1.** *unite (by treaty),* bersekutu: *a Third World country which allied itself with a superpower,* sebuah negara Dunia Ketiga yang bersekutu dengan dengan sebuah kuasa besar; **2.** *(usu. passive) connect (by likeness, resemblance, descent)* ada persamaannya: *a disease which is closely allied to dysentry,* penyakit yang ada persamaannya yang rapat dengan disentri; *English is allied to German,* bahasa Inggeris ada persamaannya dengan Jerman.

{ **ee** *cat. gl,* **lt.** *sgl* me: *ex,* mx; **lt.** *sgl* me, me: *ex,* mx;
cat. me: ex, mx;
cat. **nm.** *gl,* me: ex, mx; **nm.** *gl* me: ex, mx; ex, mx. }

allied *adj.* **1.** *joined,* **a.** *(by treaty)* bersekutu, berikat: *the joined forces,* kuasa-kuasa bersekutu; **b.** *usu. Allied,* Pihak Berikat; **2.** *connected,* berkait(an), berhubung(an): *closely allied to his lack of self-confidence was his fear of being ridiculed,* bahawa dia takut dipersendakan berkaitan dengan kurangnya kepercayaan terhadap dirinya sendiri; **3.** *connected by likeness, resemblance, descent,* berkaitan: *biology and allied sciences,* biologi dan bidang-bidang ilmu sains berkaitan.

{ **ee** *cat.* **nm.** *gl,* **lt.** *sgl* me, me: *ex,* mx; **lt.** *sgl* me; **nm.** *gl,* me, me: ex, mx; **nm.** *gl,* me: ex, mx. }

Annotations:

ee	English entry
cat	category
me	Malay equivalent
ex	English example
mx	Malay example (translation of *ex*)
gl	gloss
lt.	letter
sgl	subgloss
nm.	number

(i) Based on these examples, devise a hierarchical structure for the dictionary using the following notation at the nodes:
- [...] for optional elements,
- + for iterative/repeatable nodes with at least one element, and
- * for iterative/repeatable nodes with zero or more elements.

(ii) Based on the hierarchical structure given, construct both the following, each of which should be able to parse all the given dictionary entries:

- a finite-state automaton;
- a set of context-free rules,

and in doing so, give the derivation tree for the entry **alluvial** (here, there is no necessity for terminal rules, nor to include the terminals in the derivation tree - just terminate at the annotations, i.e. ee, cat, etc.)

[Some Help:
For the context-free rules, note that

x y*[z]q

with branching at all points can be written as

X -> x R	(R is the rest of the entry)
R -> Y | Z | Q	(alternatives for R)
Y -> y Y | y Z | y Z Y	(for * I continue without branch ! with brach)
Z -> z Q | Q	(for optionality)
Q -> q	(terminal)]

[50/100]

- oooOooo -