UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences

Exam in	**INF 2820: Computational Linguistics**
Day of exam:	**June 8th, 2012**
Exam hours:	**0900 – 1300**

This examination paper consists of 6 page(s).

Appendices:	**None**
Permitted materials:	**None**

Make sure that your copy of this examination paper
is complete before answering.

**If you think some assumptions are missing,
make your own and explain them!**

1 Regular Expressions and Context Free Grammars (10 + 10 %)

(a) Let A be the set

> {*Anne, Espen, Ola, Kari, et, barn, dyr, uhyre, smilte, lo, danset, likte, så, hørte, kjente, og, eller*}

Let r be the regular expression

> *((Ola + Kari) + (et (barn + dyr + uhyre)))*
> *((og + eller) ((Ola + Kari) + (et (barn + dyr + uhyre))))**
> *(lo + danset + smilte + ((likte + så + hørte)*
> *((Ola + Kari) + (et (barn + dyr + uhyre)))*
> *((og + eller) ((Ola + Kari) + (et (barn + dyr + uhyre))))*))*

(We use the notation from the lectures and JFLAP. Other sources use alternative notations like '$\vee$' or '|' for '+'.) Then r describes a set of strings $L \subseteq A^*$ and $\langle A, L\rangle$ is a language. Make a finite state automaton which descibes this language.

(b) Make a context free grammar which describes this language.

2 Parsing (10 + 10 %)

(a) We will consider the grammar G given by:

```
S -> NP VP
VP -> V NP
NP -> DET  N
V ->  'fisker'
VP -> 'krabber'
N -> 'gutt' | 'fisk'
NP -> 'fisker'| 'krabber' | 'Ola'
DET -> 'en'
```

Show how a recursive descent parser recognizes the string

(1) *Ola fisker krabber*

(b) Show how a chart parser performs a complete chart parse for the same string. Explain the choices you make with respect to strategy.

3 More Parsing (10 %)

Some parsers for context free grammars have problems with certain constructions. They only function with restricted types of grammar. Which restrictions—if any—do the following parsers place on the grammars they can implement? For each question explain shortly why the parser must invoke the restrictions.

(i) Recursive Descent Parser

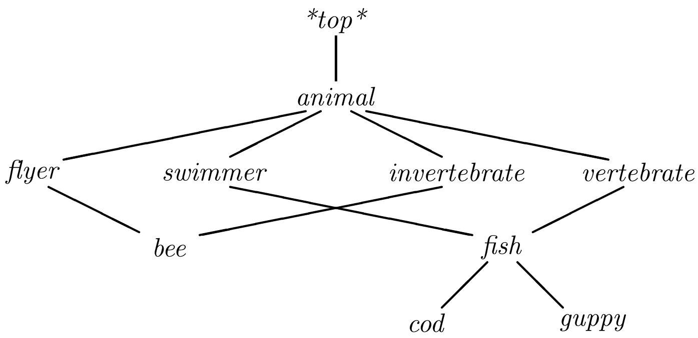
(ii) Shift-Reduce Parser

(iii) CKY Parser

(iv) Chart Parser where active edges are invoked top-down

4 Typed Feature Structures (3 + 4 + 3 %)

- Assume the following type hierarchy (which should look familiar):



- We introduced *subsumption* as a partial order among types, i.e. a binary relation, which for two types t_1 and t_2 will either order the types with respect to each other ($t_1 \sqsubseteq t_2$ or $t_2 \sqsubseteq t_1$) or not (i.e. there is no subsumption in either direction holding between t_1 and t_2: $t_1 \not\sqsubseteq t_2 \wedge t_2 \not\sqsubseteq t_1$).

(a) Name two pairs of types (i.e. two examples for each relation) from the above hierarchy for each of the following configurations: (i) $t_1 \sqsubseteq t_2$ (t_1 subsumes t_2) and (ii) $t_1 \not\sqsubseteq t_2$ (there is no subsumption relation between t_1 and t_2).

(b) In our logic of typed feature structures (as implemented in the LKB), the type hierarchy determines whether or not two types can be unified, and if so, what the result of type unification should be. We introduced the term *greatest lower bound* (glb) in this context; in one sentence, what is the result of unifying two types t_1 and t_2, and how is it related to the glb concept.

(c) Is it possible for two types to *not* stand in the subsumption relation but nevertheless be *unifiable* with each other (i.e. $t_1 \not\sqsubseteq t_2 \wedge t_2 \not\sqsubseteq t_1 \wedge t_1 \sqcap t_2 \neq \bot$)? If so, name two pairs of types from the above hierarchy for which this is the case.

5 Unification Grammar (3 + 3 + 3 + 3 + 3 %)

- In our unification-based LKB grammars (following Sag, Wasow, & Bender 2003), the three top-level features `HEAD`, `SPR`, and `COMPS` jointly determine the syntactic category of expressions, i.e. in contract to the simple, atomic category labels used in context-free grammar, we take advantage of (typed) feature structures to 'break up' syntactic categories into bundles of feature – value pairs.

(a) For convenience, we have allowed ourselves to continue talking about syntactic categories as 'noun phrases' (NP), 'verb phrases' (VP), or 'sentences' (S)—but have always insisted that such names are merely abbreviations for specific typed feature structure (TFS) configurations. In terms of the above three top-level features, show the TFSs that (in our LKB grammars) correspond to the following abbreviatory category labels: (i) NP and (ii) VP.

(b) In no more than two sentences per feature, say which type of syntactic information is encoded by each feature, and sketch the typical range of values for each of the above three features.

(c) Sag, Wasow, & Bender (2003) operate with a head – complement rule that has the formal property we called *variable arity* (i.e. a right-hand side that can vary in length):

$$
{}_{\textit{phrase}}\left[\begin{array}{ll} \texttt{HEAD} & \boxed{1} \\ \texttt{SPR} & \boxed{2} \\ \texttt{COMPS} & \langle\rangle \end{array}\right] \longrightarrow {}_{\textit{word}}\left[\begin{array}{ll} \texttt{HEAD} & \boxed{1} \\ \texttt{SPR} & \boxed{2} \\ \texttt{COMPS} & \left\langle \boxed{\mathrm{i}}, \ldots, \boxed{\mathrm{n}} \right\rangle \end{array}\right], \boxed{\mathrm{i}}, \ldots, \boxed{\mathrm{n}}
$$

In no more than two sentences, comment on the flow of information (regarding our three top-level features) between the elements of the right-hand side of the rule and its left-hand side. For example, how do we know which right-hand side element is the syntactic head in this rule?

(d) Somewhat technically, what do we mean by calling the above rule *variable arity*? To answer this question, think about how it would be used in combining an intransitive, transitive, or ditransitive verb with its complements. Furthermore, in a sentence or two, speculate about the algorithmic challenges that lead the LKB chart parser to not allow syntactic rules of variable arity.

(e) In our implementation, we recast the head – complement rule of Sag, Wasow, & Bender (2003) as a binary rule that could apply recursively, i.e. combine a head with exactly one of its complements at a time, for each application of the rule. Draw the feature structure representation (for example in the format we used above for the rule of Sag, Wasow, & Bender, 2003) of our binary head – complement variant. Make sure to show explicitly (i) the top-level type of the head daughter and (ii) the decomposition of its `COMPS` list.

6 Semantics in Feature Structues (4 + 5 + 4 %)

(a) Following is the (simplified) logical-form semantics that our final grammars assign to one of the readings of the sentence *the dog chased that cat near the aardvark.*

$$\textit{the(i)} \wedge \textit{dog(i)} \wedge \textit{chase(s,i,j)} \wedge \textit{past(s)} \wedge \textit{that(j)} \wedge \textit{cat(j)} \\ \wedge\ \textit{near(s,k)} \wedge \textit{the(k)} \wedge \textit{aardvark(k)}$$

Recall that there is a PP attachment ambiguity in this sentence: where in the syntactic structure does the PP attach to yield the above semantics?

(b) Note the *past(s)* predication in the above, which records the grammatical tense of the chasing event. In our course grammars, tense was actually not reflected in the semantics. At which point in the analysis of the above sentence should the *past(s)* predication be introduced, i.e. which of our grammar rules would have to be augmented to contribute the tense information to the semantics (i.e. insert it into the `RSTR` list)?

(c) In embedding semantics in unification-based grammars, what do we mean by *compositionality*? In terms of semantic composition, what happens when we form a new phrase, say by using the *head – complement rule* rule to analyze the verb phrase *chased that cat.*

7 Formal Syntax (7 + 5 %)

- Consider the following sentences:
  - (i) *The professor rested on the terrace.*
  - (ii) *Those students depend on that book.*
  - (iii) *The manager reported on the desk.*
  - (iv) *The students rest.*
  - (v) **The manager depends.*
  - (vi) *That professor reported.*

(a) We make the claim that the PPs in (i) and (ii) have quite different syntactic functions; in a couple of sentences, can you think of possible ways to test that claim? What happens, for example, when the PP is omitted, or when another preposition is substituted for *on*? We further claim that (iii) is ambiguous—in the sense of having at least two distinct interpretations—because here the PP could have either one of the two possible syntactic functions. In your own words, what is the difference between the various usages of the PPs in these examples?

(b) Sketch constituent trees for each of the possible readings you have identified for sentence (iii) above, using abbreviatory node labels like 'Det', 'N', 'NP', 'VP', et al., and annotating each branch as to whether the constituent dominated by it acts as a *head*, *specifier*, *complement*, or *modifier*.

SLUTT