

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

<b>Eksamen i:</b>	<b>INF2820: Datalingvistik</b>
<b>Eksamensdag:</b>	<b>8. juni 2012</b>
<b>Tid for eksamen:</b>	<b>0900 – 1300</b>
<b>Oppgavesettet er på 6 side(r)</b>	
<b>Vedlegg:</b>	<b>Ingen</b>
<b>Tillatte hjelpemidler:</b>	<b>Ingen</b>

*Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.*

Der du synes det mangler forutsetninger, gjør dine egne forutsetninger og redegjør for dem!

## 1 Regulære uttrykk og kontekstfrie grammatikker (10 + 10 %)

- (a) La  $A$  være mengden

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

La  $r$  være det regulære uttrykket

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

(Vi bruker her notasjonen fra forelesningene og fra JFLAP. Andre kilder bruker annen notasjon som ‘ $\vee$ ’ eller ‘ $|$ ’ for ‘+’.) Da beskriver  $r$  en mengde strenger  $L \subseteq A^*$  og  $\langle A, L \rangle$  er et språk. Lag en endelig tilstandsautomat (“finite state automaton”) som beskriver dette språket.

- (b) Lag en kontekstfri grammatikk som beskriver dette språket.

## 2 Parsing (10 + 10 %)

- (a) Vi skal bruke grammatikken  $G$  gitt ved:

S  $\rightarrow$  NP VP  
VP  $\rightarrow$  V NP  
NP  $\rightarrow$  DET N  
V  $\rightarrow$  'fisker'  
VP  $\rightarrow$  'krabber'  
N  $\rightarrow$  'gutt' | 'fisk'  
NP  $\rightarrow$  'fisker' | 'krabber' | 'Ola'  
DET  $\rightarrow$  'en'

Vis hvordan en “recursive descent” parser vil anerkjenne strengen

(1) *Ola fisker krabber*

- (b) Vis hvordan en chartparser foretar et fullstendig chartparse for samme streng. Forklar hvilke valg du foretar med hensyn til strategi.

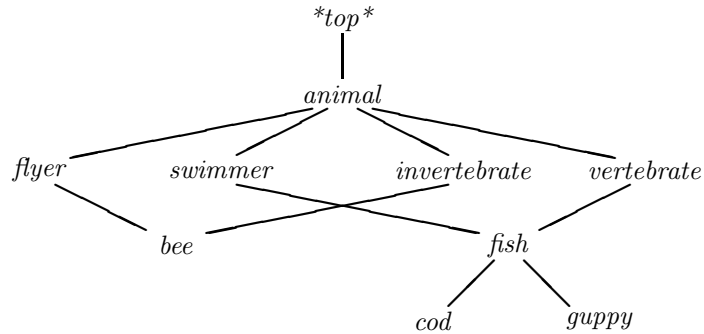
### 3 Mer parsing (10 %)

En del parsere for kontekstfrie grammatikker har problemer med en del konstruksjoner. De virker bare med en begrenset type grammatikker. Hvilke begrensninger - om noen - legger følgende parsere på de grammatikkene de skal virke med? For hvert spørsmål forklar kort hvorfor parserne må legge de begrensningene de gjør.

- (i) Recursive descent parser
- (ii) Shift-Reduce parser
- (iii) CKY parser
- (iv) Chartparser der aktive kanter innføres ovenifra og ned (“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  $t_2$ :  $t_1 \not\sqsubseteq t_2 \wedge t_2 \not\sqsubseteq t_1$ ).
- 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  $t_2$ ).
  - 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.
  - Is it possible for two types to *not* stand in the subsumption relation but nevertheless be *unifiable* with each other (i.e.  $t_1 \not\sqsubseteq t_2 \wedge t_2 \not\sqsubseteq t_1 \wedge t_1 \sqcap t_2 \neq \perp$ )? 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 contrast 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):

$$\textit{phrase} \left[ \begin{array}{l} \text{HEAD} \ \boxed{1} \\ \text{SPR} \ \boxed{2} \\ \text{COMPS} \ \langle \rangle \end{array} \right] \longrightarrow \textit{word} \left[ \begin{array}{l} \text{HEAD} \ \boxed{1} \\ \text{SPR} \ \boxed{2} \\ \text{COMPS} \ \langle \boxed{i}, \dots, \boxed{n} \rangle \end{array} \right], \boxed{i}, \dots, \boxed{n}$$

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 Structures (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*.

$$\begin{aligned} &the(i) \wedge dog(i) \wedge chase(s,i,j) \wedge past(s) \wedge that(j) \wedge cat(j) \\ &\quad \wedge near(s,k) \wedge the(k) \wedge aardvark(k) \end{aligned}$$

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* 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