Section for: LIN120, Syntactic Theory – A formal approach
TA: T. Florian Jaeger (tiflo@stanford.edu)
Office hours: We 11:15-12:00 in 460-113 (Margaret Jacks Hall, first floor)
Office phone: (650) 723 0924
This handout is available at: www.stanford.edu/~tiflo/

Section for LIN120, 10/21/02

Next homework due 10/25/02: Chapter 5 – Problem 2,3
Chapter 6 – Problem 2,3,4

Summary of some important Concepts of last weeks lectures:

- Last week we introduced Semantics into the formalism.
  - A feature structure of the type *expression* does now immediately contain **two** feature structures
    - SYN with a value of the type *syn-cat*
    - SEM with a value of the type *sem-cat*
  - The new feature SEM contains three features
    - MODE (*quest, prop, ref*, etc.)
    - INDEX which has as its value an index variable
    - RESTR which has as its value a list of *pred(icate)*s
- We defined two general principles (like the HFP, SHAC, or Valence Principle) which make reference to semantics:
  - SIP – Semantic Inheritance Principle (p. 148 in the reader)
    - *In any headed phrase, the mother's MODE and INDEX values are identical to those of the head daughter.*
  - SCP - Semantic Compositionality Principle (p. 147 in the reader):
    - *In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters,*
    - Note that the SCP differs in two important respects from the syntactic principles (and the SIP) we got to know so far.
      - The SCP includes all daughters not only the head daughter.
      - Furthermore, the SCP does not build hierarchies, it only adjoins lists to lists.
- The introduction of semantics in interaction with already known principles and grammar rules allowed us to revise our (4) grammar rules, so that we now know the final versions of the
  - Head-Complement rule
    - Heads select for complements, not vice versa.
    - The semantics of the resulting phrase is done by the SIP and SCP.
    - HEAD comes from the head of the phrase
    - VAL | COMPS of the resulting phrase is satisfied (empty)
  - Head-Specifier rule
    - Heads select for specifiers, not vice versa.
    - The semantics of the resulting phrase is done by the SIP and SCP.
    - HEAD comes from the head of the phrase
    - VAL | COMPS of the head has to be satisfied (empty)
    - VAL | SPR of the resulting phrase is satisfied (empty)
  - Head-Modifer rule (p. 150)
    - Modifiers select for their heads, not vice versa.
    - The semantics of the resulting phrase is done by the SIP and SCP.
    - VAL and HEAD come from the head of the phrase
- and a revised version of the
  - Coordination-rule (p. 153) – to get the semantics in the only non-headed phrase right.

        - The RESTR collection is done by SCP.
        - The coordination rule itself has to do the job of the SIP since their is no head.
        - VAL of all daughters except for the CONJ has to be the same and is identified with the VAL of the resulting phrase.
        - Nothing is said about the HEAD feature of the resulting phrase.
- We also stressed that
    - (well-formed) trees are models – i.e. not underspecified
    - lexical entries are not models but descriptions – i.e. they *can* be underspecified
    - underspecified descriptions can license up to an infinite number of models!

Example exercises:

-