**Name:** _____

**Section:**_____

**Linguistics 001**
Spring 2009
Homework 8
**Due: Wed., April 22nd for 15 points**

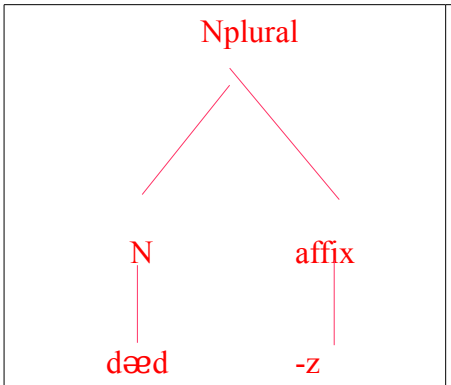
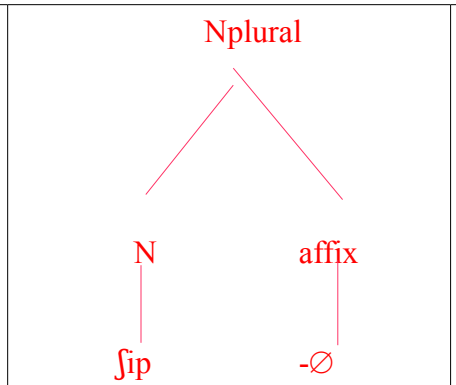
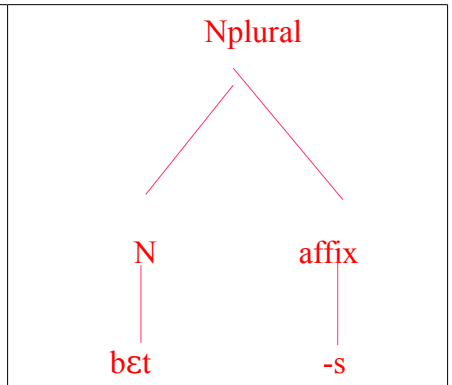
**PART I. CFG's and Chomsky Normal Form**
The close relationship between modern linguistics and computer science is no accident. These grammars are instances of the literal, "input/output" conceptualization of production and comprehension that computer scientists and linguists share.

We've worked with two types of formal grammars so far this course. They involve surface forms and more abstract forms. You've learned to implement rules that "rewrite" abstract forms as surface forms.

Here's how rewrite rules work. Context-Free rules like (i) and (ii) below apply everywhere, rewriting the abstract symbol on the left-hand side as the abstract symbol or surface form on the right. Context-Sensitive rules like (iii) only apply in specific environments, which are either described or listed.

(i) $N_{plural}$ → N $affix_{plural}$
(ii) N → {dÊd , bɛt , ʃip , ... }
(iii) $affix_{plural}$ → -z / [voiced] __
-s / [unvoiced] __
-∅ / { ʃip, dir, fiʃ }

1. **Translate the words below into their surface phonological forms (in IPA). Then, use the rules above to construct the morphological trees underlying the surface forms.**

| | | |
|---|---|---|
|  |  |  |
| / dædz / "dads" | / ʃip / "sheep" | / bɛts / "bets" |

Below is a (simplified) categorization of Context-Free and Context-Sensitive Grammars.

| | Input (left-hand side) | Output (right-hand side) |
|---|---|---|
| ***Context-Free Grammars (CFG)*** | Apply to all | zero or more nonterminal symbols and<br>one or zero terminal symbols |
| ***Context-Sensitive Grammars (CSG)*** | Apply in specifically defined environment | zero or more nonterminal symbols and<br>one or zero terminal symbols |

**2.**

**Identify each of the following as either a CFG or CSG. Give an example of a rule to support your answer.**

| **Syntax** | **Phonology** |
|---|---|
| CFG or CSG | CFG or CSG |
| example rule:<br>NP -> DET N | example rule:<br>/t/ -> [flap] / Vstress_V |

**PART II**. Normal forms and limitations

Below is a variant on the English "ABC song" that has traditionally been used to teach phonics. It is a call-and-response song wherein the teacher asks about a written letter and the children respond with the appropriate phonemes.

| | |
|---|---|
| *What does the A say?* | /e/ /a/ /Ê/ |
| *What does the B say?* | /b/ /b/ /b/ |
| *What does the C say?* | /s/ and /k/ |
| *What does the D say?* | /d/ /d/ /d/ |
| ... | |
| *What does the S say?* | /s/ /s/ /s/ |
| *What does the T say?* | /t/ /t/ /t/ |
| ... | |
| *What does the X say?* | /ks/ /ks/ /ks/ |
| *What does the Y say?* | /jə/ /i/ /ai/ |

**3.**

**In the space below, describe this excerpt of the song as a CFG which rewrites each letter as an appropriate sound. A few examples are included, to get you started. "A" is meant to represent the written form of a letter, and it's not important that it's capitalized. (ie, "A" = "a").**

| | | | |
|---|---|---|---|
| S ➞ {"A"or"B"or...or"Z"} | "B" ➞ /b/ | "E" ➞ /ɛ/ | "Y" ➞ /je/ |
| (ie, any single letter) | "C" ➞ /s/ | "K" ➞ /k/ | "Y" ➞ /i/ |
| "A" ➞ /e/ | "C" ➞ /k/ | "S" ➞ /s/ | "Y" ➞ /ai/ |
| "A" ➞ /a/ | "D" ➞ /d/ | "T" ➞ /t/ | |
| "A" ➞ /æ/ | "E" ➞ /i/ | "X" ➞ /ks/ | |

**4.**

**Which rule(s) above violate the output conditions of CFGs as listed in the above table?**

"X" → /ks/
"Y" → /je/
"Y" → /ai/

**5.**

**If children use the "Phonics song" rewriting rules to map letters to sounds, how will they read the following words out loud? List all possibilities and put an asterisk (*) next to the incorrect forms.**

| | | | |
|---|---|---|---|
| "CATS" | kæts | *sæts | |
| | *kets | *sets | |
| | *kats | *sats | |
| "CADS" | *kæds | *sæds | |
| | *keds | *seds | |
| | *kads | *sads | |
| "SAY" | *sæje | *seje | *saje |
| | *sæi | sei | *sai |
| | *sæai | *seai | *saai |

**6.**

**Refer the phrase structure rules that we used to do syntax earlier in the class. Forget about the "AND" rule for a second. It's just a hack anyway, and you might be able to see how it won't work with this algorithm. Prove that the string (a) is a grammatical sentence of English by filling out the table:**

"I will see the man with my new binoculars"

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | | | | | | | | |
| | TP | | | | | | | |
| S | | VP | | | | | | |
| | TP | | NP | | | | | |
| S | | VP | | | | | | |
| | TP | | | | PP | | | |
| | | VP | | | | NP | | |
| NP | | | NP | | | | N | |
| N | T | V | det | N | P | det | A | N |
| I | will | see | the | man | with | my | new | binoculars |

| | |
|---|---|
| S → NP TP | NP → N |
| TP → T VP | NP → det N |
| T → "will" | NP → NP PP |
| VP → V NP | N → A N |
| VP → VP PP | A → "*adjectives*" |
| V → "*verbs*" | PP → P NP |
| N → "*nouns*" | P → "*prepositions*" |

1 character = 1 syllable = 1 morpheme

**7.**

**Consider the following monomorphemic Mandarin words:**

zhāng láng
'cockroach'

pí pá
'flute'

**Is the equation above violated? If so, how?**

one morpheme doesn't equate to one syllable or one character.

Many languages, including Mandarin, have a system of *diminuitivization*, in which attaching a morpheme to a word expresses smallness, comfort, and familiarity. In German, for instance, the word for “girl” is *Mäd-chen*, literally 'girl-little'. In Chinese, the morpheme *-r* 'boy' indicates this same meaning, and is used in words like “ball”, which counts as a single character and is pronounced as a single syllable.

< qiúr >
*qiur*
“ball"

**8.**

**What parts of this equation above are violated? What parts are satisfied here?**

“1 morpheme =  1 character” is violated, but “1 syllable = 1 character” is satisfied.