The task of the previous assignment was to make PSG rules utilize feature constraints. Now we will see a way of solving that task and apply it to further problems. The feature structures we have been dealing with so far are sets of attribute value pairs, where values can themselves be such sets. The standard way of imposing feature checks for PSG rules is to use unification. Here we will see a simpler method.

The simplification is taking feature structures as sets of complex symbols rather than sets of attribute value pairs. For example, instead of specifying the category feature as $\begin{bmatrix}\text{CAT} & \text{v}\end{bmatrix}$ we will use [V] or, if you like to make this a binary feature, $[\text{V}^+]$. For another example, number can be represented as $[\text{P}^-]$ for $\begin{bmatrix}\text{NUM} & \text{sg}\end{bmatrix}$ and $[\text{P}^+]$ for $\begin{bmatrix}\text{NUM} & \text{plu}\end{bmatrix}$. If you like you can use multi-character symbols for the sake of readability – use $[\text{PLU}^+]$ instead of $[\text{P}^+]$, for instance. For a given feature X distinguish between the interpretable versus uninterpretable instances of it as X versus $\hat{\text{X}}$.

In writing our PSG rules we will use possibly partial feature specifications. The following is an example verb phrase rule:

$$[\text{V}^+,\text{FIN}^-] \quad \rightarrow \quad [\text{V}^+] \quad [\text{N}^+,\text{D}^+,\hat{\text{V}}^+]$$

Here the feature specification of the object is partial in the following sense. When you want to apply this rule to, say, *love* and *him*, the latter will have the feature specification $[\text{N}^+,\text{D}^+,3,\text{PLU}^-,\hat{\text{V}}^+]$, including entries for person and number. The rule will simply ignore these and will not complain about having some extra features that are not specified in the rule, of course, as long as there is no clash.

As a working assumption, we will assume that any feature that does not appear in the result category to the left of the arrow are deleted. In this regard, the above rule dictates that the uninterpretable category feature of the object is checked by the verb.

One additional notch of power can be added to this system by utilizing variables. Take the following schematic rule:

$$[\text{X},\text{Y}^\alpha] \quad \rightarrow \quad [\text{Z}^\beta,\text{T}] \quad [\text{Z}^\beta,\text{Y}^\alpha]$$

This rule dictates that the items to be merged must agree on their Z feature, and the result category should inherit the value of Y feature of the category on the right.

If you like, you are free to add your own conventions or mechanisms as long as you apply them consistently.

Question 1

Use the present system to capture the fragment of English consisting of the following lexical items:

(1) a. Verbs: love, loves;
b. Pronouns: he, him, she, her;
c. Proper names: John, Mary;
d. Nouns: cat, dog, cats, dogs;
e. Determiners: a, the, |the two| (as a single lexical item)

Your grammar should be able to derive grammatical sentences that can be constructed with these items, and it should be unable to derive ungrammatical ones.

Question 2

Specify a PSG with feature specifications to capture the Turkish fragment including the following items:

(2) a. çocuk;

b. -lAr (plural number marker);
c. -lAr (plural agreement marker);
d. Ahmet;
e. -(y)H (accusative case);
f. ∅ (nominative case);
g. her, bir, |her bir|, |her iki|, bütün;
h. uyudu, gördü.

Your grammar should be able to derive grammatical sentences that can be constructed with these items, and it should be unable to derive ungrammatical ones.

Note that:

- Assume things like *Ahmetler* is ungrammatical, plural cannot attach to proper nouns.
- Treat suffixes as normal lexical items.
- Do not worry about phonology, *(y)H* adapts to its environment automatically.
- Be careful about two *-lAr*s, one attaches to nouns or noun phrases, the other attaches to verbs, their functions differ – and be careful not to derive things like **Çocuk Ahmeti gördüler*, while accepting both *Çocuklar Ahmeti gördü* and *Çocuklar Ahmeti gördüler*.

Question 3

How can we handle structures involving "movement" within the present system of PSG rules with feature specifications? Try to implement "movement as remerge" idea in the present system. Demonstrate your solution over deriving:

(3) Which chair has Adrian always liked?