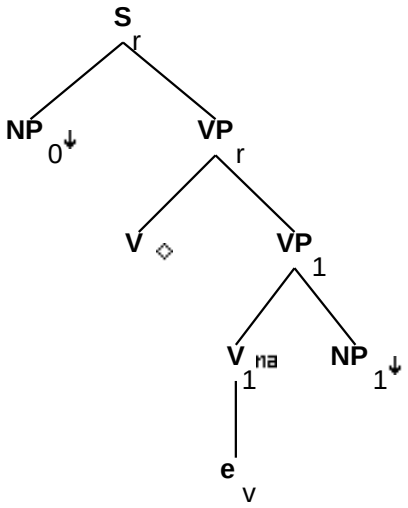
Family "Tnx0BEnx1"

March 5, 2008

1 Tree "alphanx0BEnx1"

1.1 graphe



1.2 comments

This is the tree for 'equative be': i.e. the 'be' that you get in sentences like:

My uncle is that man over there.
That man over there is my uncle.

The two NP's are 'equated' hence their interchangibility in contrast to predicative 'be' sentences (e.g. Tnx0Nx1) in which NP's are not interchangible.

1.3 features

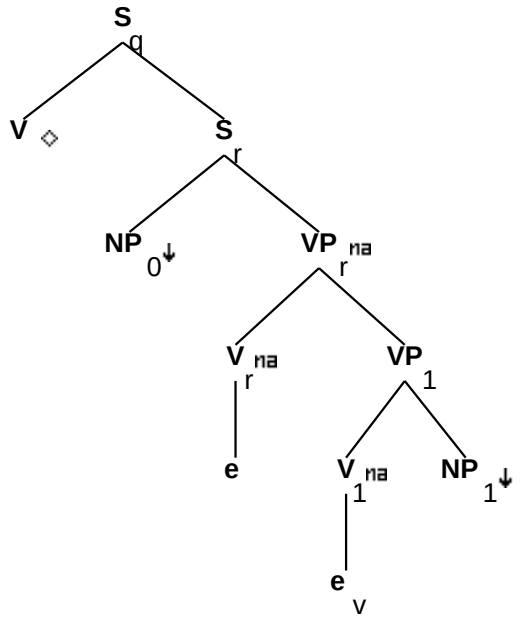
S_r.b:<extracted> = -
S_r.b:<inv> = -

S_r.b:<assign-comp> = VP_r.t:<assign-comp>

```
VP_1.t:<mode> = VP_r.b:<mode>
VP_1.t:<mode> = VP_1.b:<mode>
S_r.b:<mode> = VP_r.t:<mode>
S_r.b:<mainv> = VP_r.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP_r.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = nom/acc
S_r.b:<agr> = VP_r.t:<agr>
S_r.b:<assign-case> = VP_r.t:<assign-case>
V.t:<passive> = -
V.t:<mainv> = -
VP_r.b:<passive> = V.t:<passive>
VP_r.b:<agr> = V.t:<agr>
VP_r.b:<assign-case> = V.t:<assign-case>
VP_r.b:<assign-comp> = V.t:<assign-comp>
VP_r.b:<mode> = V.t:<mode>
VP_r.b:<tense> = V.t:<tense>
VP_r.b:<mainv> = V.t:<mainv>
S_r.b:<control> = NP_0.t:<control>
NP_0:<gerund> = -
NP_1:<gerund> = -
VP_r.b:<compar> = -
VP_1.b:<compar> = -
```

2 Tree "alphaInvnx0BEnx1"

2.1 graphe



2.2 comments

```
Inverted equative 'be' constructions.
This tree is necesary because 'be' anchors the tree
rather than adjoining as an auxiliary.


 Who is that man over there?
 Which one is your uncle?
```

2.3 features

```
S_q.b:<inv> = +
S_q.b:<extracted> = -
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<mode> = V.t:<mode>
V.t:<agr> = NP_0:<agr>
V.t:<assign-case> = NP_0:<case>
S_r.b:<assign-comp> = VP_r.t:<assign-comp>
S_r.b:<comp> = nil
S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<agr> = V.t:<agr>
S_r.b:<assign-case> = V.t:<assign-case>
S_r.b:<tense> = V.t:<tense>
```

```
V.t:<mode> = V_r.b:<mode>
V.t:<trace> = V_r.b:<trace>
V_r.t:<mode> = VP_r.b:<mode>
VP_r.b:<mode> = VP_1.t:<mode>
VP_1.b:<mode> = VP_1.t:<mode>
VP_r.b:<compar> = -
VP_1.b:<compar> = -
NP_1:<case> = nom/acc
S_r.t:<conj> = nil
NP_0:<gerund> = -
NP_1:<gerund> = -
```