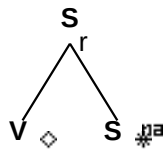
Family "auxs"

March 5, 2008

1 Tree "betaVs"

1.1 graphe



1.2 comments

```
Auxiliary tree for inversion
 'Do (you think  S)'
 'Has John thought S'

Note that when this is adjoined the non-finite
verb in the S that it adjoins to will inherit
the value for <agr>. This doesn't have any ill-
effect, but looks strange.
```

1.3 features

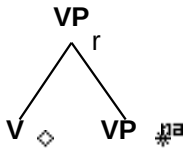
```
S_r.b:<inv> = +
S_r.b:<mode> = ind
S.b:<inv> = -
S.b:<comp> = nil

S.t:<agr> = V.b:<agr>
S.t:<conj> = and/or/but/nil
S_r.b:<assign-case> = S.t:<assign-case>
S_r.b:<conditional> = S.t:<conditional>
S_r.b:<perfect> = S.t:<perfect>
S_r.b:<progressive> = S.t:<progressive>
V.t:<assign-case> = S_r.b:<assign-case>
V.t:<mode> = S_r.b:<mode>
V.t:<tense> = S_r.b:<tense>
V.t:<agr> = S_r.b:<agr>
```

V.t:<neg> = S_r.b:<neg>
V.t:<assign-case> = S.t:<assign-case>
S_r.b:<nocomp-mode> = S.t:<nocomp-mode>

2 Tree "betaVvx"

2.1 graphe



2.2 comments

Auxiliary tree
'has (loved)'
'has been (loving)'

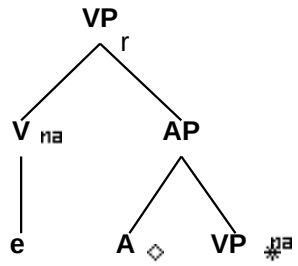
2.3 features

VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>

V.t:<assign-case> = VP_r.b:<assign-case>
V.t:<mode> = VP_r.b:<mode>
V.t:<tense> = VP_r.b:<tense>
V.t:<mainv> = VP_r.b:<mainv>
V.t:<agr> = VP_r.b:<agr>
V.t:<neg> = VP_r.b:<neg>
V.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -

3 Tree "betaVvx-adj"

3.1 graphe



3.2 comments

```
Auxiliary tree
 'has (loved)'
 'has been (loving)'
```

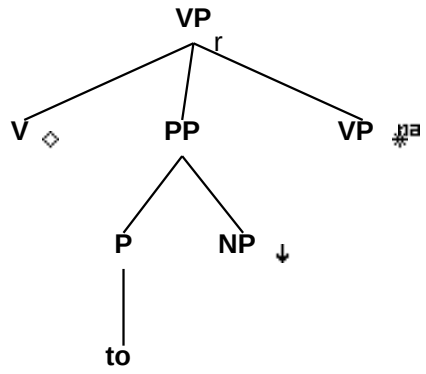
3.3 features

```
VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>


A.t:<assign-case> = VP_r.b:<assign-case>
A.t:<mode> = VP_r.b:<mode>
A.t:<tense> = VP_r.b:<tense>
A.t:<mainv> = VP_r.b:<mainv>
A.t:<agr> = VP_r.b:<agr>
A.t:<neg> = VP_r.b:<neg>
A.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -
VP.t:<mode> = inf
VP.t:<assign-comp>=ecm
A.b:<mode> = nom
AP.b:<equiv> = A.t:<equiv>
AP.b:<compar> = A.t:<compar>
```

4 Tree "betaVpxvx"

4.1 graphe



4.2 comments

NIL

4.3 features

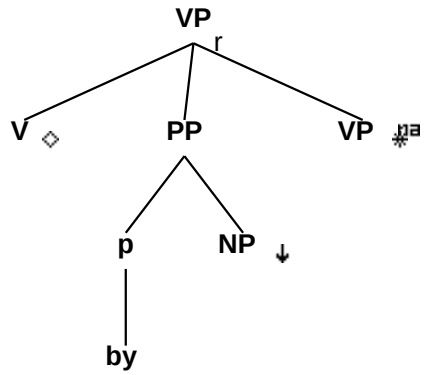
```
VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>

VP.t:<assign-comp> = inf_nil/ind_nil

V.t:<assign-case> = VP_r.b:<assign-case>
V.t:<mode> = VP_r.b:<mode>
V.t:<tense> = VP_r.b:<tense>
V.t:<mainv> = VP_r.b:<mainv>
V.t:<agr> = VP_r.b:<agr>
V.t:<neg> = VP_r.b:<neg>
V.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -
PP.b:<wh> = NP:<wh>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
P.t:<assign-case> = acc
```

5 Tree "betaVbynxvx"

5.1 graphe



5.2 comments

```
Auxiliary tree
 'has (loved)'
 'has been (loving)'
```

5.3 features

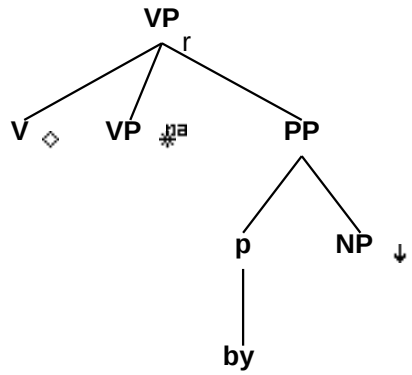
```
VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>

VP.t:<assign-comp> = ecm

V.t:<assign-case> = VP_r.b:<assign-case>
V.t:<mode> = VP_r.b:<mode>
V.t:<tense> = VP_r.b:<tense>
V.t:<mainv> = VP_r.b:<mainv>
V.t:<agr> = VP_r.b:<agr>
V.t:<neg> = VP_r.b:<neg>
V.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -
PP.b:<wh> = NP:<wh>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
P.t:<assign-case> = acc
```

6 Tree "betaVvxbynx"

6.1 graphe



6.2 comments

```
Auxiliary tree
 'has (loved)'
 'has been (loving)'
```

6.3 features

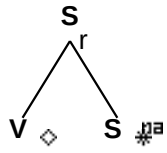
```
VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>

VP.t:<assign-comp> = ecm

V.t:<assign-case> = VP_r.b:<assign-case>
V.t:<mode> = VP_r.b:<mode>
V.t:<tense> = VP_r.b:<tense>
V.t:<mainv> = VP_r.b:<mainv>
V.t:<agr> = VP_r.b:<agr>
V.t:<neg> = VP_r.b:<neg>
V.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -
PP.b:<wh> = NP:<wh>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
P.t:<assign-case> = acc
```

7 Tree "betaIVs"

7.1 graphe



7.2 comments

NIL

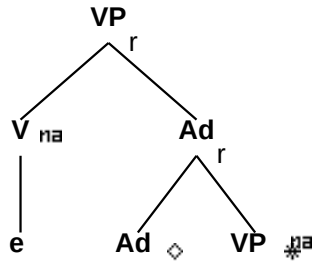
7.3 features

```
S_r.b:<inv> = +
S_r.b:<mode> = imp
S.b:<inv> = -
S.b:<comp> = nil

S.t:<agr> = V.b:<agr>
S_r.b:<conditional> = S.t:<conditional>
S_r.b:<perfect> = S.t:<perfect>
S_r.b:<progressive> = S.t:<progressive>
V.t:<tense> = S_r.b:<tense>
V.t:<agr> = S_r.b:<agr>
V.t:<neg> = S_r.b:<neg>
S_r.b:<nocomp-mode> = S.t:<nocomp-mode>
V.b:<mode> = base
```

8 Tree "betaVvx-arb"

8.1 graphe



8.2 comments

```
Auxiliary tree
 'has (loved)'
```

'has been (loving)'

8.3 features

```
VP_r.t:<mainv> = -
VP_r.b:<conditional> = VP.t:<conditional>
VP_r.b:<perfect> = VP.t:<perfect>
VP_r.b:<progressive> = VP.t:<progressive>


Ad.t:<assign-case> = VP_r.b:<assign-case>
Ad.t:<mode> = VP_r.b:<mode>
Ad.t:<tense> = VP_r.b:<tense>
Ad.t:<mainv> = VP_r.b:<mainv>
Ad.t:<agr> = VP_r.b:<agr>
Ad.t:<neg> = VP_r.b:<neg>
Ad.t:<assign-comp> = VP_r.b:<assign-comp>
VP_r.b:<compar> = -
VP.t:<compar> = -
VP.t:<mode> = inf
VP.t:<assign-comp>=ecm
Ad.b:<mode> = nom
Ad_r.b:<equiv> = Ad.t:<equiv>
Ad_r.b:<compar> = Ad.t:<compar>
```