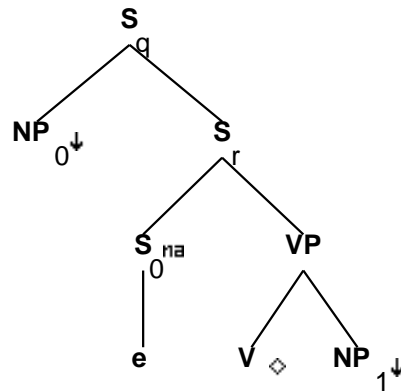# Family "Ts0Vnx1"

March 5, 2008

1 Tree "alphaW0s0Vnx1"

1.1 graphe

1.2 comments

```
Sentential subject with wh question on the subject:
  What annoys the emu? (That Bill ate all of the artichokes)
(These also get a W0nx0Vnx1 parse)
```

1.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_r.b:<assign-comp> = inf_nil/ind_nil
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<assign-case> = VP.t:<assign-case>




S_q.b:<wh> = NP_0:<wh>
S_q.b:<comp> = nil
```

```
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<inv> = -
NP_0:<trace> = S_0.t:<trace>
NP_0:<wh> = +
S_r.b:<agr> = VP.t:<agr>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<compar> = -
NP_1.t:<case> = acc
S_r.t:<conj> = nil
```
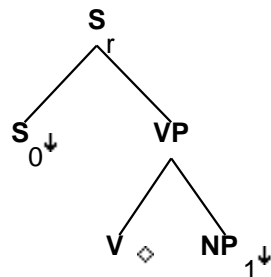
2 Tree "alphas0Vnx1"

2.1 graphe

2.2 comments

```
Sentential subject:
  John eating artichokes annoyed Bill
to eat artichokes thrills Max
for the emu to eat artichokes worried the gorilla
whether to eat artichokes worried Max
```

2.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<assign-case> = VP.t:<assign-case>
```

```
S_r.b:<mainv> = VP.t:<mainv>
S_r.b:<passive> = VP.t:<passive>

S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
VP.t:<agr pers> = 3
S_0.t:<mode> = inf/ind
S_0.t:<comp> = that/for/whether/nil
S_0.t:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_1.t:<case> = acc
```
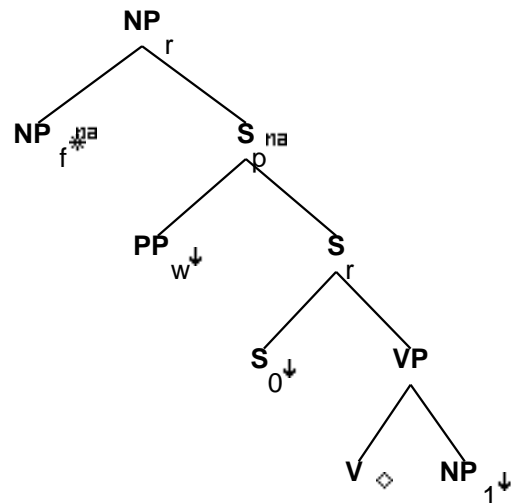
3   Tree "betaNpxs0Vnx1"

3.1   graphe

3.2   comments

```
Sentential subject:
```

```
   John eating artichokes annoyed Bill
to eat artichokes thrills Max
for the emu to eat artichokes worried the gorilla
whether to eat artichokes worried Max
```

3.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
VP.b:<agr num> = sing
VP.b:<agr pers> = 3
VP.b:<agr 3rdsing> = +
S_0.t:<mode> = inf/ind
S_0.t:<comp> = that/for/whether/nil
S_0.t:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_1.t:<case> = acc
S_r.t:<inv> = -
PP_w.t:<wh> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```
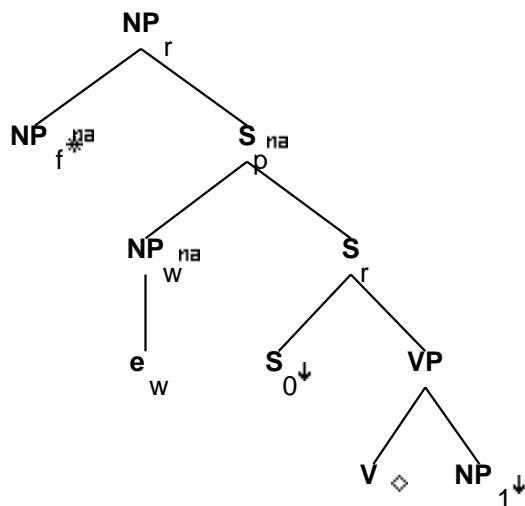
4 Tree "betaNcs0Vnx1"

4.1 graphe

```
                        NP
                          r
              /                    \
         NP                      S
           f*na                    na
                              p
                      /              \
                  NP                    S
                    na                    r
                  w               /          \
              |              S              VP
             e                0↓          /      \
              w                        V          NP
                                        ◇            1↓
```

4.2 comments

```
Sentential subject:
  John eating artichokes annoyed Bill
to eat artichokes thrills Max
for the emu to eat artichokes worried the gorilla
whether to eat artichokes worried Max
```

4.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
VP.b:<agr num> = sing
VP.b:<agr pers> = 3
VP.b:<agr 3rdsing> = +
S_0.t:<mode> = inf/ind
S_0.t:<comp> = that/for/whether/nil
S_0.t:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
```

```
VP.b:<agr> = V.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_1.t:<case> = acc
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```