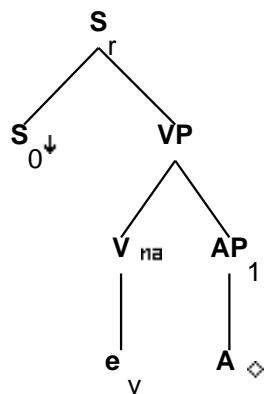# Family "Ts0Ax1"

March 5, 2008

1 Tree "alphas0Ax1"

1.1 graphe

1.2 comments

```
Sentential subject with adjectival predicate:
That Max continues to drive a Jaguar is incredible.
For the Jaguar to be towed seems unlikely.
Whether Max will get a Jeep is uncertain.
To drive a Jaguar is decadent.
```
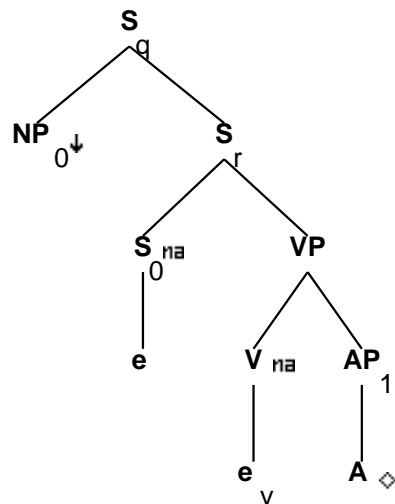
1.3 features

```
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:<mainv> = VP.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_0:<mode> = inf/ind
```

```
S_0:<comp> = that/whether/for/nil
S_0:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<passive> = VP.t:<passive>
VP.t:<passive> = -
VP.t:<agr pers> = 3
VP.b:<mode> = nom
VP.b:<assign-case> = acc
VP.b:<compar> = -
A.t:<compar> = AP_1.b:<compar>
```

2   Tree "alphaW0s0Ax1"

2.1   graphe

2.2   comments

```
Sentential subject with adjectival predicate, subject extracted:
What is incredible?
Which event seems unlikely?
Who is decadent?

(These will also get a W0nx0Ax1 parse.)
```
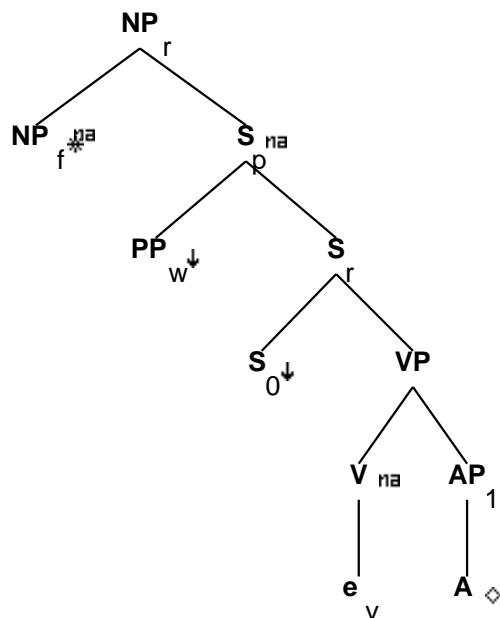
2.3   features

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

S_q.b:<inv> = S_r.t:<inv>
```

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




VP.b:<compar> = -
VP.t:<passive> = -
VP.t:<agr pers> = 3
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:<wh> = +
S_0:<assign-case> = NP_0:<case>
S_0:<trace> = NP_0:<trace>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<assign-case> = acc
VP.b:<mode> = nom
S_r.t:<conj> = nil
A.t:<compar> = AP_1.b:<compar>
```

3   Tree "betaNpxs0Ax1"

3.1   graphe

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

3.2   comments

```
Sentential subject with adjectival predicate:
That Max continues to drive a Jaguar is incredible.
For the Jaguar to be towed seems unlikely.
Whether Max will get a Jeep is uncertain.
To drive a Jaguar is decadent.
```
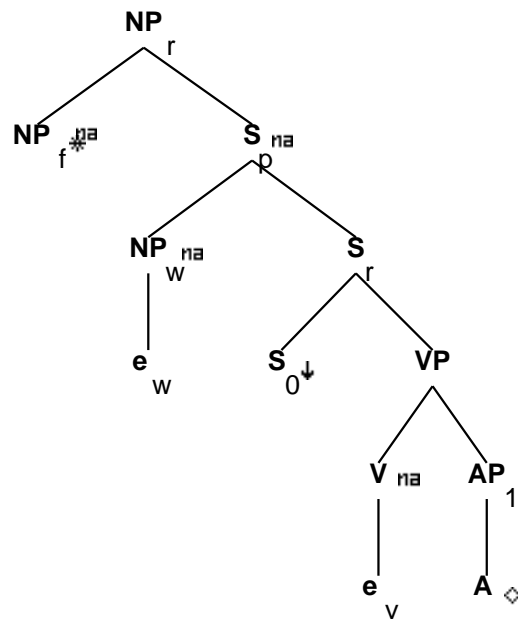
3.3   features

```
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:<mainv> = VP.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_0:<mode> = inf/ind
S_0:<comp> = that/whether/for/nil
S_0:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
S_r.b:<agr> = VP.t:<agr>
```

```
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<passive> = VP.t:<passive>
VP.t:<passive> = -
VP.b:<mode> = nom
VP.b:<compar> = -
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
A.t:<compar> = AP_1.b:<compar>
NP_r.b:<pron> = NP_f.t:<pron>
```

4 Tree "betaNcs0Ax1"

4.1 graphe

4.2 comments

```
Sentential subject with adjectival predicate:
That Max continues to drive a Jaguar is incredible.
For the Jaguar to be towed seems unlikely.
Whether Max will get a Jeep is uncertain.
To drive a Jaguar is decadent.
```

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:<mainv> = VP.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_0:<mode> = inf/ind
S_0:<comp> = that/whether/for/nil
S_0:<assign-comp> = inf_nil
S_0.t:<inv> = -
S_0.t:<extracted> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<passive> = VP.t:<passive>
VP.t:<passive> = -
VP.b:<mode> = nom
VP.b:<compar> = -
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
A.t:<compar> = AP_1.b:<compar>
NP_r.b:<pron> = NP_f.t:<pron>
```