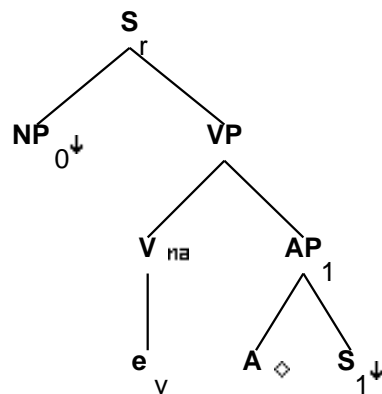


Family "Tnx0A1s1"

March 5, 2008

1 Tree "alphanx0A1s1"

1.1 graphe



1.2 comments

Tree for predicational adjectives which take sentential complements:
Bill is sure that he will be elected.

The farmers seem positive that the artichokes will
survive the flood.

1.3 features

S_r.b:<extracted> = -

S_r.b:<inv> = -

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

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>

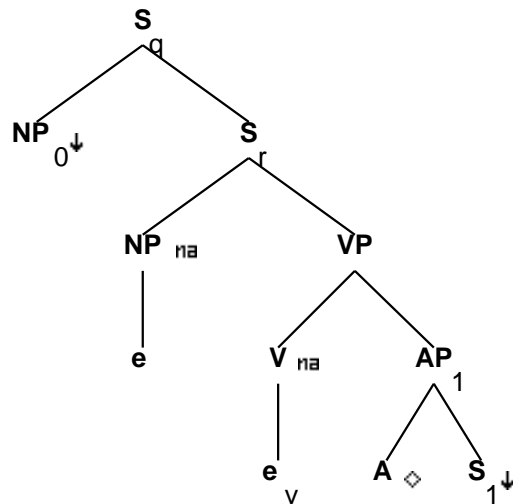
NP_0:<agr> = S_r.b:<agr>

NP_0:<case> = S_r.b:<assign-case>
 NP_0:<wh> = -
 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:<assign-case> = acc
 VP.b:<compar> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

 S_r.b:<control> = NP_0.t:<control>
 A.t:<compar> = AP_1.b:<compar>

2 Tree "alphaW0nx0A1s1"

2.1 graphe



2.2 comments

Subject extraction tree for predicational adjective which take sentential complements:

Which senators are sure that Bill will be approve?

Who seems postive that Max left?

2.3 features

S_q.b:<extracted> = +

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

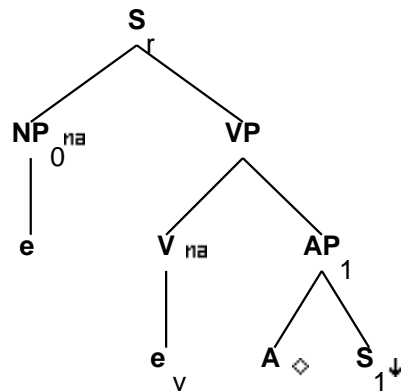
S_q.b:<wh> = NP_0.t:<wh>
 S_r.t:<comp> = nil
 S_r.t:<assign-comp> = inf_nil/ind_nil
 S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
 S_r.b:<assign-comp> = VP.t:<assign-comp>

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:<trace> = NP_0.t:<trace>
 NP:<agr> = NP_0.t:<agr>
 NP:<case> = NP_0.t:<case>
 NP.t:<wh> = NP_0.t:<wh>
 NP_0:<wh> = +
 NP.t:<agr> = S_r.b:<agr>
 NP.t:<case> = S_r.b:<assign-case>
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 VP.b:<mode> = nom
 VP.b:<assign-case> = acc
 VP.b:<compar> = -
 VP.t:<passive> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

S_r.t:<conj> = nil
 A.t:<compar> = AP_1.b:<compar>

3 Tree "alphaInx0A1s1"

3.1 graphe



3.2 comments

Imperative tree for predicational adjectives which take sentential complements:
Be certain that you take the right jacket.

3.3 features

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

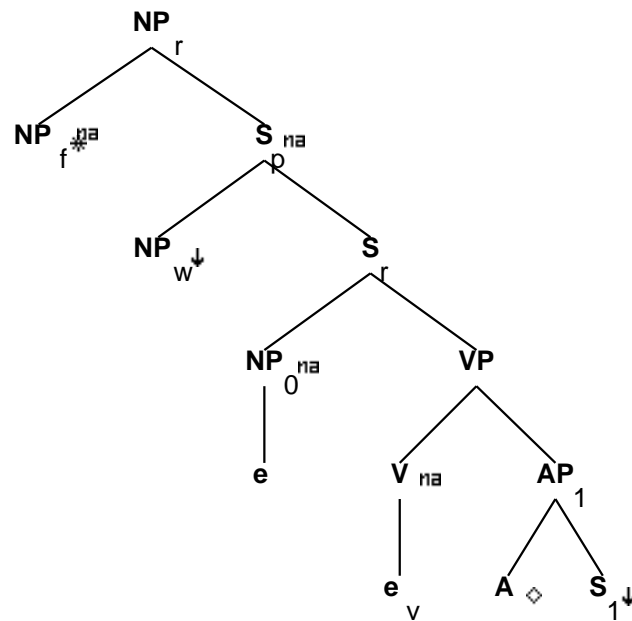
```
S_r.b:<mode> = imp
S_r.b:<mainv> = VP.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<passive> = VP.t:<passive>
VP.t:<passive> = -
VP.t:<mode> = base
VP.t:<neg> = -
VP.t:<tense> = pres
VP.b:<mode> = nom
```

VP.b:<assign-case> = acc
 VP.b:<compar> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

 A.t:<compar> = AP_1.b:<compar>

4 Tree "betaN0nx0A1s1"

4.1 graphe



4.2 comments

Subject relative-clause tree for predicational bare NPs which take
 sentential complements:
 the person who is positive (that) Bill has left

4.3 features

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

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

```

S_r.t:<inv> = -
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = nom
VP.b:<assign-case> = acc
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_1.t:<control> = NP_0.t:<control>

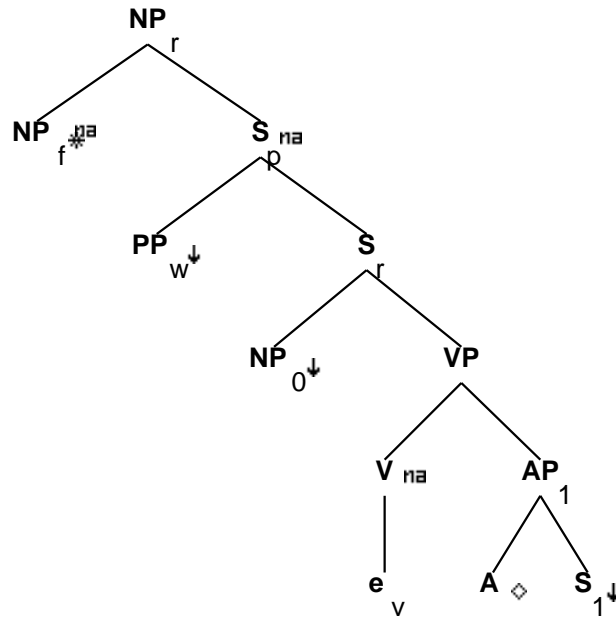
S_r.t:<conj> = nil
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>

NP_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
NP_w.t:<wh> = +
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>

```

5 Tree "betaNpxnx0A1s1"

5.1 graphe



5.2 comments

Tree for predicational adjectives which take sentential complements:

Bill is sure that he will be elected.

The farmers seem positive that the artichokes will survive the flood.

5.3 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>

NP_0:<agr> = S_r.b:<agr>

NP_0:<case> = S_r.b:<assign-case>

NP_0:<wh> = -

S_r.b:<agr> = VP.t:<agr>

S_r.b:<assign-case> = VP.t:<assign-case>

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

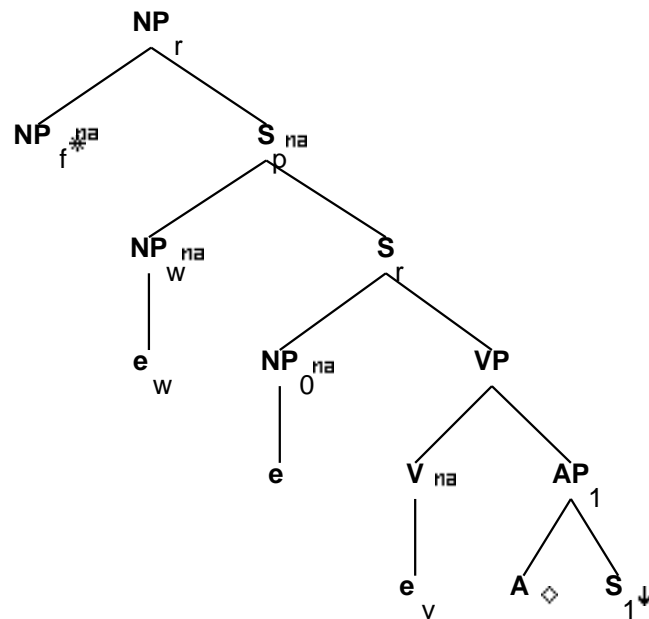
VP.t:<passive> = -

VP.b:<mode> = nom
 VP.b:<assign-case> = acc
 VP.b:<compar> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

 S_r.b:<control> = NP_0.t:<control>
 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>

6 Tree "betaNc0nx0A1s1"

6.1 graphe



6.2 comments

Subject relative-clause tree for predicational bare NPs which take sentential complements:

the person who is positive (that) Bill has left

6.3 features

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

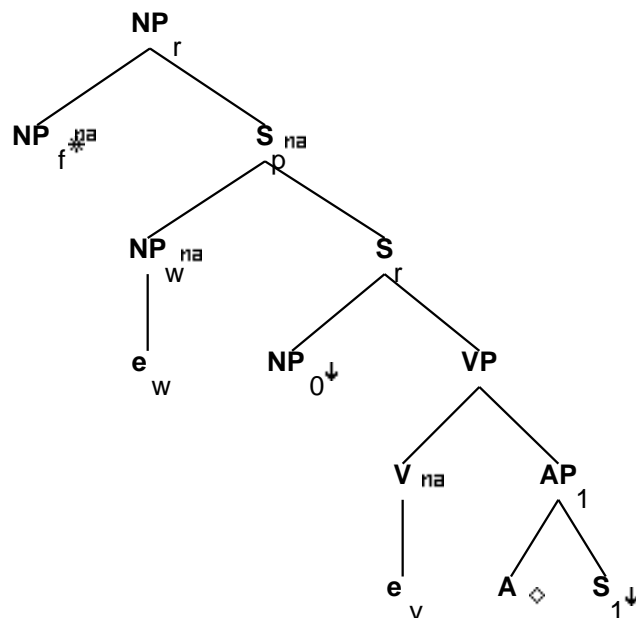
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = nom
VP.b:<assign-case> = acc
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_1.t:<control> = NP_0.t:<control>

S_r.t:<conj> = nil
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_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind
S_r.t:<nocomp-mode> = inf/ger
VP.t:<assign-comp> = that/ind_nil/inf_nil/ecm
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
A.t:<compar> = AP_1.b:<compar>
NP_r.b:<pron> = NP_f.t:<pron>

7 Tree "betaNcnx0A1s1"

7.1 graphe



7.2 comments

Tree for predicational adjectives which take sentential complements:
 Bill is sure that he will be elected.

The farmers seem positive that the artichokes will
 survive the flood.

7.3 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>

NP_0:<agr> = S_r.b:<agr>

NP_0:<case> = S_r.b:<assign-case>

NP_0:<wh> = -

S_r.b:<agr> = VP.t:<agr>

S_r.b:<assign-case> = VP.t:<assign-case>

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

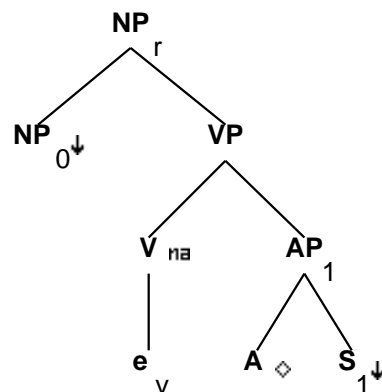
VP.t:<passive> = -

VP.b:<mode> = nom
 VP.b:<assign-case> = acc
 VP.b:<compar> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

 S_r.b:<control> = NP_0.t:<control>
 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>

8 Tree "alphaGnx0A1s1"

8.1 graphe



8.2 comments

NP Gerund tree for predicational adjectives which take sentential complements:

...Max('s) being sure that Bill stole the money...

8.3 features

NP_0:<wh> = NP_r.b:<wh>
 VP.t:<mode> = ger

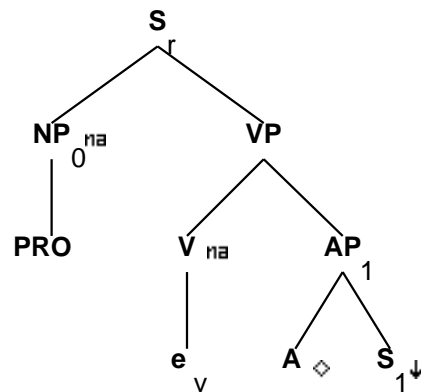
NP_r.t:<case> = nom/acc
 NP_r.t:<agr num> = sing
 NP_r.t:<agr pers> = 3
 NP_r.t:<agr 3rdsing> = +
 VP.b:<mode> = nom
 VP.b:<assign-case> = acc

VP.b:<compar> = -
 S_1.t:<assign-comp> = inf_nil/ind_nil
 S_1.t:<inv> = -
 S_1.t:<control> = NP_0.t:<control>

NP_r.b:<gerund> = +
 A.t:<compar> = AP_1.b:<compar>
 NP_0:<case> = acc/gen

9 Tree "alphanx0A1s1-PRO"

9.1 graphe



9.2 comments

Tree for predicational adjectives which take sentential complements - PRO subject

Bill wants [PRO to be sure that he will be elected].

9.3 features

S_r.b:<extracted> = -
 S_r.b:<inv> = -
 S_r.t:<assign-comp> = inf_nil/ind_nil
 S_r.b:<assign-comp> = VP.t:<assign-comp>
 S_r.b:<mode> = VP.t:<mode>

```

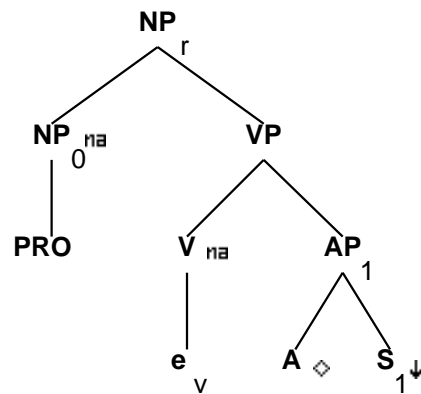
S_r.b:<mainv> = VP.t:<mainv>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = -
NP_0.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
S_r.b:<passive> = VP.t:<passive>
VP.t:<passive> = -
VP.b:<mode> = nom
VP.b:<assign-case> = acc
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_1.t:<control> = NP_0.t:<control>

S_r.b:<control> = NP_0.t:<control>
A.t:<compar> = AP_1.b:<compar>
VP.t:<mode> = inf/ger

```

10 Tree "alphaGnx0A1s1-PRO"

10.1 graphe



10.2 comments

NP Gerund tree for predicational adjectives which take sentential complements - PRO subject

[PRO being sure that Bill stole the money] is important to John.

10.3 features

```

NP_0:<wh> = NP_r.b:<wh>
NP_0.t:<case> = none

```

NP_0.t:<wh> = -
VP.t:<mode> = ger
NP_r.t:<case> = nom/acc
NP_r.t:<agr num> = sing
NP_r.t:<agr pers> = 3
NP_r.t:<agr 3rdsing> = +
VP.b:<mode> = nom
VP.b:<assign-case> = acc

VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_1.t:<control> = NP_0.t:<control>

NP_r.b:<gerund> = +
A.t:<compar> = AP_1.b:<compar>