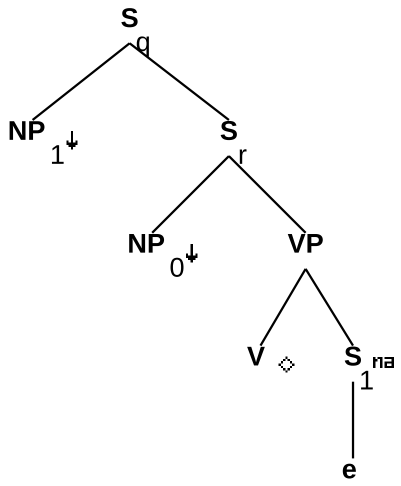
Family ”Tnx0Vs1”

March 5, 2008

1 Tree ”alphaW1nx0Vs1”

1.1 graphe



1.2 comments

```
Sentential complement verbs with question on complement:
What did Max expect?
What did the emu insist?
```

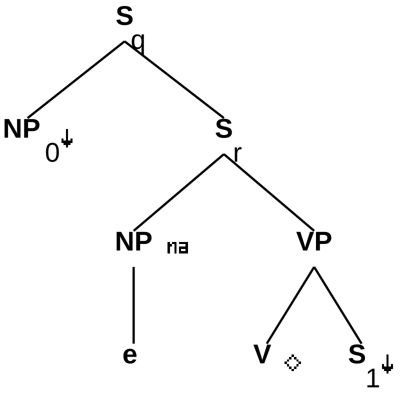
1.3 features

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

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

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

```
S_q.b:<wh> = NP_1:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_1:<trace> = S_1.b:<trace>
NP_1:<wh> = +
S_r.b:<tense> = VP.t:<tense>

S_1.t:<inv> = -
S_r.t:<conj> = nil
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

2 Tree "alphaW0nx0Vs1"

2.1 graphe



2.2 comments

Sentential complement verbs with subject extracted:

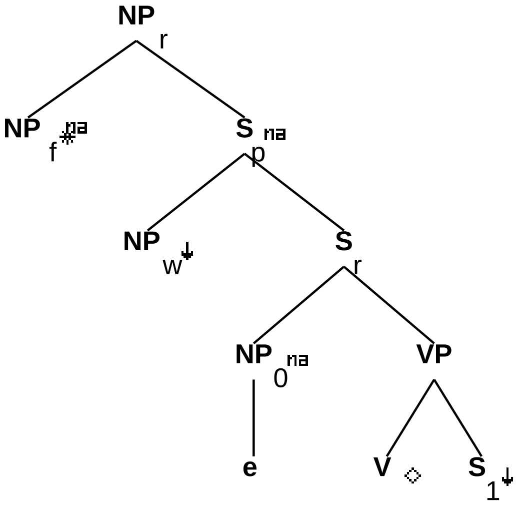
```
Who expects (for the emu) to leave
what insisted (that) Max leave
who wondered whether/if the emu would stay
which animal wondered why Bill left
```

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



S_q.b:<wh> = NP_0:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
NP:<trace> = NP_0:<trace>
NP:<agr> = NP_0:<agr>
NP:<case> = NP_0:<case>
NP:<wh> = NP_0:<wh>
NP_0:<wh> = +
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
S_r.t:<conj> = nil
```

3 Tree ”betaN0nx0Vs1”

3.1 graphe



3.2 comments

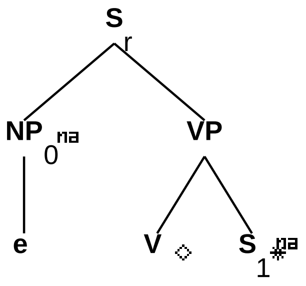
```
Relative clauses w/sentential complement verbs, subject gap:
The person who expects (for the emu) to leave
The animal that insisted (that) Max leave
The man who wondered whether/if the emu would stay
The emu which wondered why Bill left
```

3.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.b:<comp> = nil
S_r.t:<inv> = -
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
S_r.t:<conj> = nil

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
NP_r.b:<pron> = NP_f.t:<pron>
```

4 Tree "betaInx0Vs1"

4.1 graphe



4.2 comments

```
Imperative tree for sentential complement verbs:
prepare to leave the ship!
insist that he leave immediately!
```

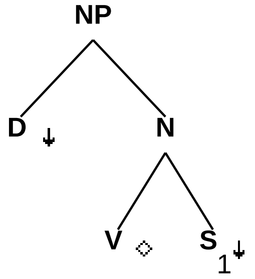
4.3 features

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

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


S_r.b:<comp> = nil
S_r.b:<mode> = imp
```

```
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
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:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<neg> = -
VP.t:<mode> = base
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.t:<tense> = pres
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

5 Tree "alphaDnx0Vs1"

5.1 graphe



5.2 comments

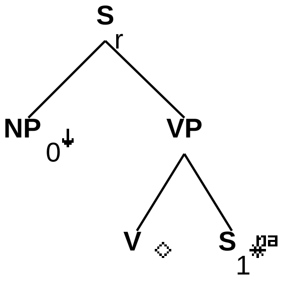
```
Gerunds with determiners on sentential complement verbs:
        the thinking that all is lost (is not surprising)
```

5.3 features

```
NP.b:<const> = D.t:<const>
NP.b:<definite> = D.t:<definite>
NP.b:<quan> = D.t:<quan>
NP.b:<card> = D.t:<card>
NP.b:<gen> = D.t:<gen>
NP.b:<decreas> = D.t:<decreas>
NP.b:<wh> = D.t:<wh>
V.b:<mode> = ger
NP.b:<case> = nom/acc
NP.b:<agr num> = sing
NP.b:<agr pers> = 3
NP.b:<agr 3rdsing> = +
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
```

6 Tree "betanx0Vs1"

6.1 graphe



6.2 comments

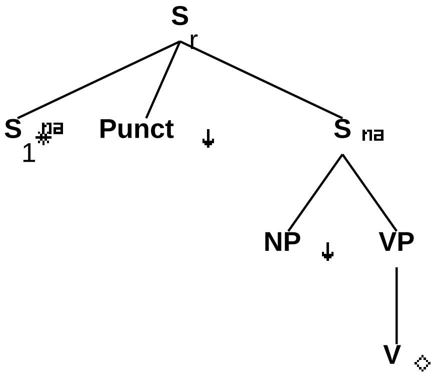
```
Sentential complement verbs:
Max expects (for the emu) to leave
the emu insisted (that) Max leave
Max wondered whether/if the emu would stay
the emu wondered why Bill left

Gerunds in this position are treated as NPs and get the transitive trees.
```

6.3 features

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



S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<punct contains> = VP.t:<punct contains>
VP.b:<punct contains> = S_1.t:<punct contains>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

7 Tree ”betaspunxV”

7.1 graphe



7.2 comments

Verb of saying as VP modifier (also made NP modifier version):

```
  John is ill, Mary says
```

I think that 'Mary says, John is ill' is weird and that string order would be a regular S-comp

This verb will be selected by all of the verbs of saying. It is quite weird, though. The sentential complement of the verb (the propositional argument) is really *distributed* in what it adjoins onto.

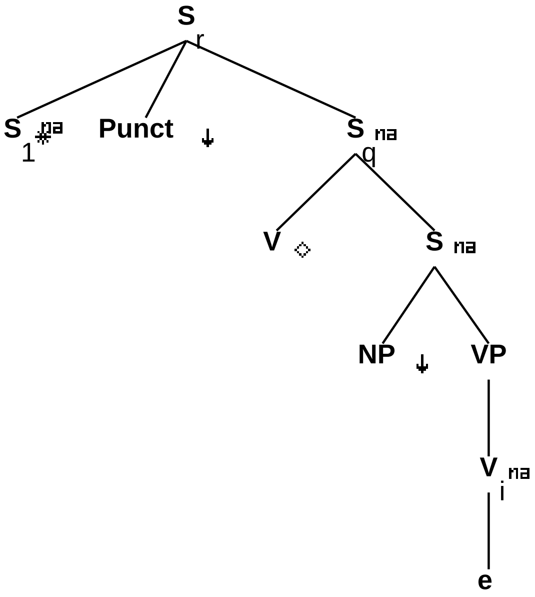
Not sure about interanal S features, probably should be just like rel clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of foot)

7.3 features

```
S_r.b:<punct struct> = Punct.t:<punct struct>
Punct.t:<punct struct> = comma/dash
S_1.t:<comp> = nil
S_1.t:<comp> = S_r.b:<comp>
S_1.t:<extracted> = S_r.b:<extracted>
S_1.t:<assign-comp> = S_r.b:<assign-comp>
S_1.t:<tense> = S_r.b:<tense>
S_1.t:<wh> = S_r.b:<wh>
S_1.t:<inv> = S_r.b:<inv>
S_1.t:<invlink> = S_r.b:<invlink>
S_1.t:<mode> = S_r.b:<mode>
S_1.t:<assign-case> = S_r.b:<assign-case>
S_1.t:<agr> = S_r.b:<agr>


S.t:<inv> = -
S.t:<mode> = ind
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<assign-case> = VP.t:<assign-case>
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>

NP:<agr> = S.b:<agr>
NP:<case> = S.b:<assign-case>

NP:<wh> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
```

8 Tree "betaspuVnx"

8.1 graphe



8.2 comments

```
Verb of saying, inverted,  as VP modifier (also made NP modifier version):
  John is ill, says Mary

This verb will be selected by all of the verbs of saying. It is quite
weird, though. The sentential complement of the verb (the
propositional argument) is really *distributed* in what it adjoins
onto.

Not sure about internal S features, probably should be just like rel
clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of
foot)
```

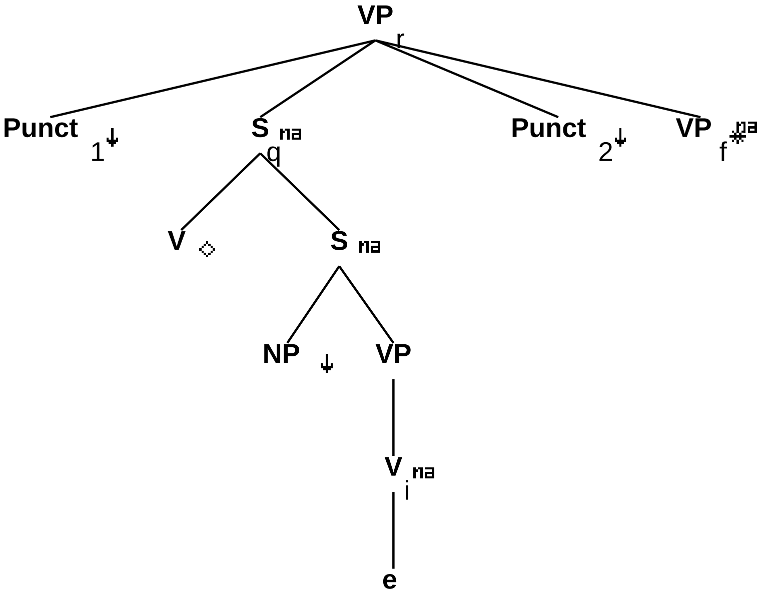
8.3 features

```
Punct.t:<punct struct> = comma/dash/colon
S_1.t:<comp> = nil
S_1.t:<comp> = S_r.b:<comp>
```

```
S_1.t:<extracted> = S_r.b:<extracted>
S_1.t:<assign-comp> = S_r.b:<assign-comp>
S_1.t:<tense> = S_r.b:<tense>
S_1.t:<wh> = S_r.b:<wh>
S_1.t:<inv> = S_r.b:<inv>
S_1.t:<invlink> = S_r.b:<invlink>
S_1.t:<mode> = S_r.b:<mode>
S_1.t:<assign-case> = S_r.b:<assign-case>
S_1.t:<agr> = S_r.b:<agr>


S_q.t:<mode> = ind
S_q.t:<comp> = nil
S_q.b:<comp> = nil
S_q.b:<agr> = V.t:<agr>
S_q.b:<tense> = V.t:<tense>
S_q.b:<mode> = V.t:<mode>
S_q.b:<assign-case> = V:<assign-case>
S_q.b:<assign-case> = S.t:<assign-case>
S_q.b:<agr> = S.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = NP:<case>
S.b:<agr> = NP:<agr>


VP.b:<mode> = V_i.t:<mode>
VP.b:<tense> = V_i.t:<tense>
VP.b:<mainv> = V_i.t:<mainv>
VP.b:<passive> = V_i.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
V:<assign-comp> = V.t:<assign-comp>
V:<mode> = V.t:<mode>
V:<tense> = V.t:<tense>
V:<passive> = V.t:<passive>
V.t:<trace> = V_i.t:<trace>
```

9 Tree "betapuVnxpuvx"

9.1 graphe



9.2 comments

```
Verb of saying, inverted, as VP modifier (also made NP modifier version):
  John, says Mary, is ill

This verb will be selected by all of the verbs of saying. It is quite
weird, though. The sentential complement of the verb (the
propositional argument) is really *distributed* in what it adjoins
onto.

Not sure about internal S features, probably should be just like rel
clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of
foot)
```

9.3 features

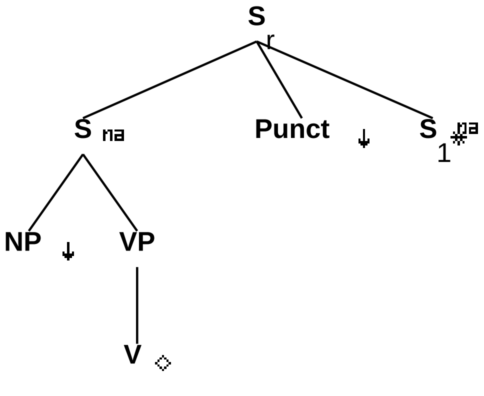
```
VP_r.b:<tense> = VP_f.t:<tense>
VP_r.b:<mode> = VP_f.t:<mode>
VP_r.b:<agr> = VP_f.t:<agr>
VP_r.b:<assign-case> = VP_f.t:<assign-case>

VP_r.b:<assign-comp> = VP_f.t:<assign-comp>
Punct_1.t:<punct struct> = Punct_2.t:<punct struct>
Punct_1.t:<punct struct> = VP_r.b:<punct struct>
Punct_1.t:<punct struct> = comma/dash
```

```
S_q.t:<mode> = ind
S_q.t:<comp> = nil
S_q.b:<comp> = nil
S_q.b:<agr> = V.t:<agr>
S_q.b:<tense> = V.t:<tense>
S_q.b:<mode> = V.t:<mode>
S_q.b:<assign-case> = V:<assign-case>
S_q.b:<assign-case> = S.t:<assign-case>
S_q.b:<agr> = S.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = NP:<case>
S.b:<agr> = NP:<agr>
```

```
VP.b:<mode> = V_i.t:<mode>
VP.b:<tense> = V_i.t:<tense>
VP.b:<mainv> = V_i.t:<mainv>
VP.b:<passive> = V_i.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
V:<assign-comp> = V.t:<assign-comp>
V:<mode> = V.t:<mode>
V:<tense> = V.t:<tense>
V:<passive> = V.t:<passive>
V.t:<trace> = V_i.t:<trace>
```

10 Tree ”betanxVpus”

10.1 graphe



10.2 comments

```
Verb of saying as pre-S modifier (also made NP modifier version):
  John is ill, Mary says
```

Also, for:

```
and/Conj_CONJs , they_NXN assert_punxVpuvx , any/Det_Ddx further/A_An drop/N_NXN in/Prep_n
```

This verb will be selected by all of the verbs of saying. It is quite weird, though. The sentential complement of the verb (the propositional argument) is really *distributed* in what it adjoins onto.

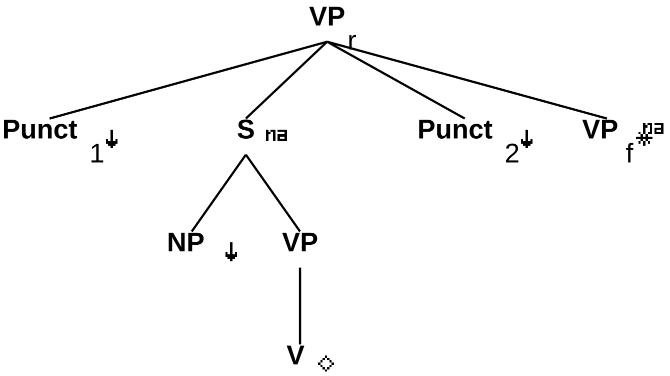
Not sure about internal S features, probably should be just like rel clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of foot)

10.3 features

```
S_r.b:<punct struct> = Punct.t:<punct struct>
Punct.t:<punct struct> = comma/dash/colon
S_1.t:<comp> = nil
S_1.t:<comp> = S_r.b:<comp>
S_1.t:<extracted> = S_r.b:<extracted>
S_1.t:<assign-comp> = S_r.b:<assign-comp>
S_1.t:<tense> = S_r.b:<tense>
S_1.t:<wh> = S_r.b:<wh>
S_1.t:<inv> = S_r.b:<inv>
S_1.t:<invlink> = S_r.b:<invlink>
S_1.t:<mode> = S_r.b:<mode>
S_1.t:<assign-case> = S_r.b:<assign-case>
S_1.t:<agr> = S_r.b:<agr>


S.t:<inv> = -
S.t:<mode> = ind/inf
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<assign-case> = VP.t:<assign-case>
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>

NP:<agr> = S.b:<agr>
NP:<case> = S.b:<assign-case>

NP:<wh> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
```

11 Tree ”betapunxVpuvx”

11.1 graphe



11.2 comments

```
Verb of saying, inverted, as VP modifier (also made NP modifier version):
  John, says Mary, is ill

This verb will be selected by all of the verbs of saying. It is quite
weird, though. The sentential complement of the verb (the
propositional argument) is really *distributed* in what it adjoins
onto.

Not sure about internal S features, probably should be just like rel
clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of
foot)
```

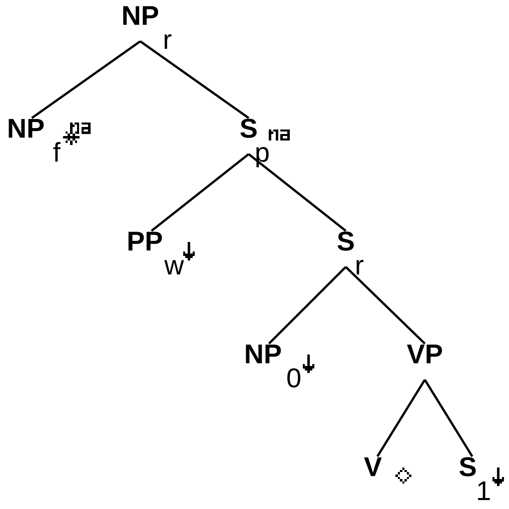
11.3 features

```
VP_r.b:<tense> = VP_f.t:<tense>
VP_r.b:<mode> = VP_f.t:<mode>
VP_r.b:<agr> = VP_f.t:<agr>
VP_r.b:<assign-case> = VP_f.t:<assign-case>

VP_r.b:<assign-comp> = VP_f.t:<assign-comp>
Punct_1.t:<punct struct> = Punct_2.t:<punct struct>
Punct_1.t:<punct struct> = VP_r.b:<punct struct>
Punct_1.t:<punct struct> = comma/dash
S.b:<agr> = NP.t:<agr>
```

```
S.b:<assign-case> = NP.t:<case>
NP.t:<wh> = -

S.t:<mode> = ind
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = VP:<assign-case>


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

12 Tree "betaNpxnx0Vs1"

12.1 graphe



12.2 comments

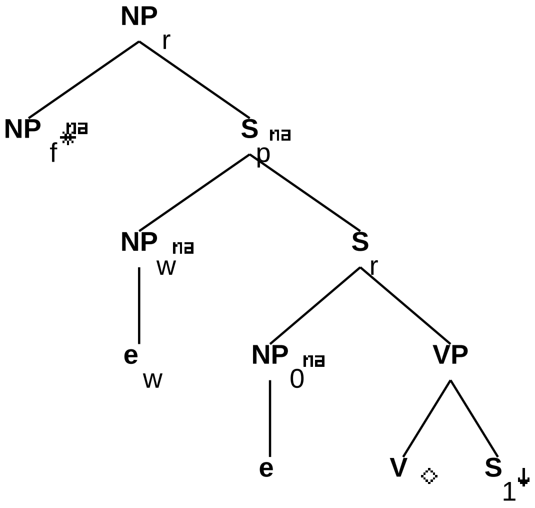
```
Sentential complement verbs:
Max expects (for the emu) to leave
the emu insisted (that) Max leave
Max wondered whether/if the emu would stay
the emu wondered why Bill left

Gerunds in this position are treated as NPs and get the transitive trees.
```

12.3 features

```
S_r.b:<extracted> = -
S_r.b:<wh> = NP_0.t:<wh>




S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

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

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

13 Tree "betaNc0nx0Vs1"

13.1 graphe



13.2 comments

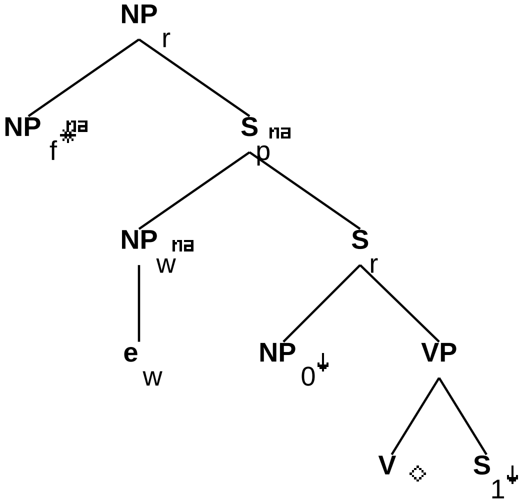
```
Relative clauses w/sentential complement verbs, subject gap:
The person who expects (for the emu) to leave
The animal that insisted (that) Max leave
The man who wondered whether/if the emu would stay
The emu which wondered why Bill left
```

13.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.t:<inv> = -
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
S_r.t:<conj> = nil

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
NP_r.b:<pron> = NP_f.t:<pron>
```

14 Tree "betaNcnx0Vs1"

14.1 graphe



14.2 comments

Sentential complement verbs:
Max expects (for the emu) to leave
the emu insisted (that) Max leave
Max wondered whether/if the emu would stay
the emu wondered why Bill left

Gerunds in this position are treated as NPs and get the transitive trees.

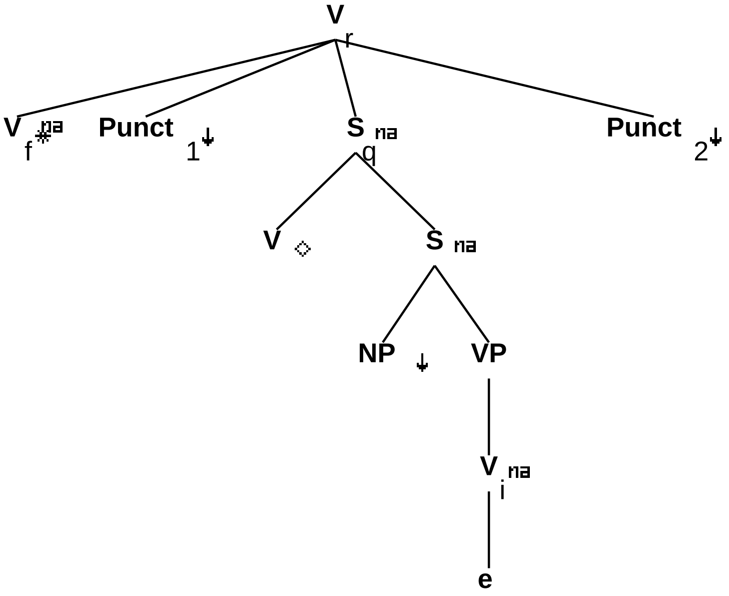
14.3 features

```
S_r.b:<extracted> = -
S_r.b:<wh> = NP_0.t:<wh>



S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

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

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

15 Tree "betavpuVnxpu"

15.1 graphe



15.2 comments

```
Parenthetical verb of saying, inverted, between subject and verb:
  John, says Mary, is ill

Features on internal S are the same as in a relative clause.
```

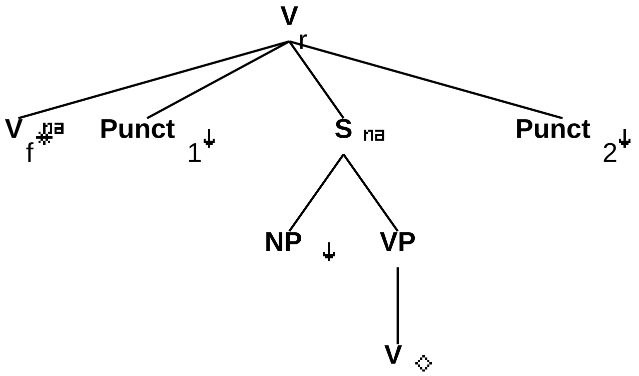
15.3 features

```
V_r.b:<tense> = V_f.t:<tense>
V_r.b:<mode> = V_f.t:<mode>
V_r.b:<agr> = V_f.t:<agr>
V_r.b:<assign-case> = V_f.t:<assign-case>

V_r.b:<assign-comp> = V_f.t:<assign-comp>
```

```
Punct_1.t:<punct struct> = Punct_2.t:<punct struct>
Punct_1.t:<punct struct> = V_r.b:<punct struct>
Punct_1.t:<punct struct> = comma/dash

S_q.t:<mode> = ind
S_q.t:<comp> = nil
S_q.b:<comp> = nil
S_q.b:<agr> = V.t:<agr>
S_q.b:<tense> = V.t:<tense>
S_q.b:<mode> = V.t:<mode>
S_q.b:<assign-case> = V:<assign-case>
S_q.b:<assign-case> = S.t:<assign-case>
S_q.b:<agr> = S.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = NP:<case>
S.b:<agr> = NP:<agr>
VP.b:<mode> = V_i.t:<mode>
VP.b:<tense> = V_i.t:<tense>
VP.b:<mainv> = V_i.t:<mainv>
VP.b:<passive> = V_i.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
V:<assign-comp> = V.t:<assign-comp>
V:<mode> = V.t:<mode>
V:<tense> = V.t:<tense>
V:<passive> = V.t:<passive>
V.t:<trace> = V_i.t:<trace>
```

16 Tree "betavpunxVpu"

16.1 graphe



16.2 comments

Parenthetical verb of saying, between subject and verb:

```
  John, Mary says, is ill
```

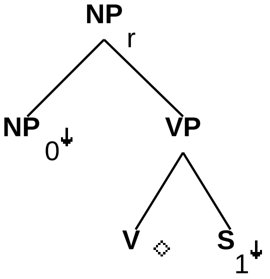
Features on internal S are the same as in a relative clause.

16.3 features

```
V_r.b:<tense> = V_f.t:<tense>
V_r.b:<mode> = V_f.t:<mode>
V_r.b:<agr> = V_f.t:<agr>
V_r.b:<assign-case> = V_f.t:<assign-case>

V_r.b:<assign-comp> = V_f.t:<assign-comp>
Punct_1.t:<punct struct> = Punct_2.t:<punct struct>
Punct_1.t:<punct struct> = V_r.b:<punct struct>
Punct_1.t:<punct struct> = comma/dash
S.b:<agr> = NP.t:<agr>
S.b:<assign-case> = NP.t:<case>
NP.t:<wh> = -

S.t:<mode> = ind
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = VP:<assign-case>


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

17 Tree "alphaGnx0Vs1"

17.1 graphe



17.2 comments

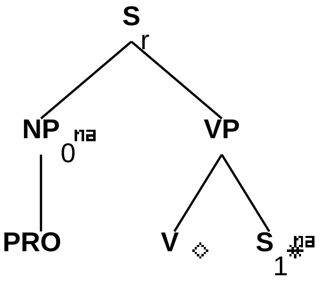
```
Sentential Complement - NP Gerund

[Max('s) believing that it will rain] is surprising.
```

17.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +

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

S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
NP_0:<case> = acc/gen
```

18 Tree ”betanx0Vs1-PRO”

18.1 graphe



18.2 comments

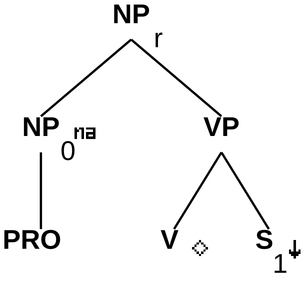
```
Sentential Complement w/ PRO subject

Max wants [PRO to believe that the emu will leave].
While [PRO insisting that the emu leave] Max got kicked.
```

18.3 features

```
S_r.b:<extracted> = -
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_0.t:<case> = none
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_1.t:<assign-comp> = inf_nil/ind_nil
S_1.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<punct contains> = VP.t:<punct contains>
VP.b:<punct contains> = S_1.t:<punct contains>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.t:<mode> = inf/ger
```

19 Tree ”alphaGnx0Vs1-PRO”

19.1 graphe



19.2 comments

```
Sentential Complement - NP gerund w/ PRO subject

[PRO expecting the emu to leave] helps maintain John’s sanity.
```

19.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
NP_0:<wh> = NP_r.b:<wh>
NP_0.t:<case> = none
NP_0.t:<wh> = -
VP.t:<mode> = ger
S_1.t:<assign-comp> = inf_nil/ind_nil

S_1.t:<inv> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
```