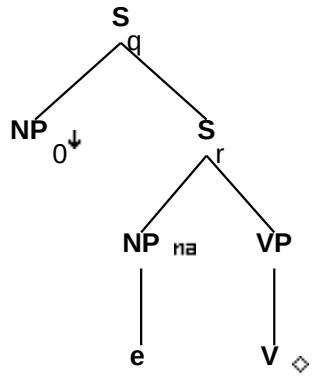
Family "Tnx0V"

March 5, 2008

1 Tree "alphaW0nx0V"

1.1 graphe



1.2 comments

```
Wh on the subject.  Need to decide what VP agrees with.
```

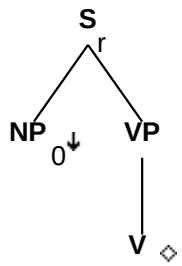
1.3 features

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

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

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

S_q.b:<wh> = NP_0.t:<wh>

S_r.t:<conj> = nil

S_r.t:<comp> = nil


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

```
S_r.b:<inv> = -
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm

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

S_r.b:<mode> = VP.t:<mode>
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:<compar> = -

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:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>

V.t:<passive> = -
V.t:<punct struct> = nil

NP.t:<agr> = S_r.b:<agr>
NP.t:<case> = S_r.b:<assign-case>

NP.t:<wh> = NP_0.t:<wh>

NP_0.t:<wh> = +

NP_0.t:<trace> = NP.t:<trace>
NP_0.t:<case> = NP.t:<case>
NP_0.t:<agr> = NP.t:<agr>
```

2 Tree "alphanx0V"

2.1 graphe



2.2 comments

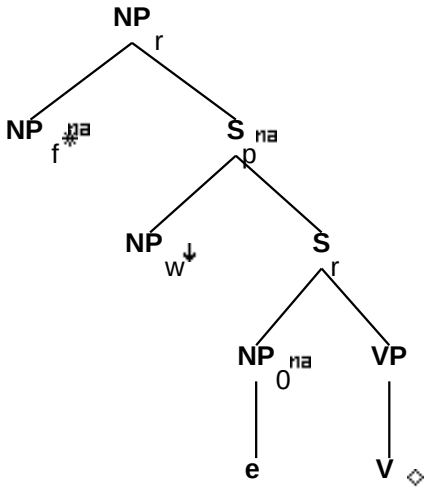
```
Intransitive declarative tree

'Anoop slept.'
'Fei ate.'
```

2.3 features

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

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>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>


S_r.b:<wh> = NP_0.t:<wh>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.b:<control> = NP_0.t:<control>

NP_0.t:<wh> = -

VP.b:<compar> = -


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:<passive> = V.t:<passive>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>

V.t:<passive> = -
V.t:<punct struct> = nil
```

3 Tree "betaN0nx0V"

3.1 graphe



3.2 comments

```
Relative Clause on the subject.  Need to decide what VP agrees with.

The foll. feature equation
NP_w.t:<select-mode> = ind
is to make sure that NP Comps only co-occur with
finite clauses. Thus (a) is ok while (b-d) are *.
(a) The boy [NP whose mother] Bill likes
(b) *The boy [NP whose mother] PRO to like
(c) *The boy [NP whose mother]  liked by Bill
(d) *The boy [NP whose mother]  reading the book
(a-d are not specific to this family, read/like have been used
to make a general point)

Adding Null Comp node to represent the [Spec, CP], Comp analysis of
relative clauses.
```

3.3 features

```
NP_r.b:<rel-clause> = +
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_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

NP_f.b:<case> = nom/acc
NP_w.t:<wh> = +
```

NP_w.t:<agr> = NP_0.t:<agr>
NP_w.t:<case> = NP_0.t:<case>
NP_w.t:<trace> = NP_0.t:<trace>
S_r.t:<inv> = -
S_r.t:<comp> = nil
S_r.t:<conj> = nil
S_r.t:<mode> = ind/inf

S_r.b:<comp> = nil

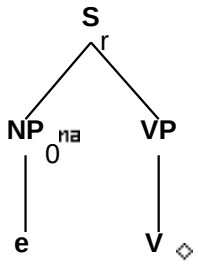
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>

VP.b:<compar> = -

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:<passive> = V.t:<passive>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
V.t:<passive> = -
V.t:<punct struct> = nil

4 Tree "alphaInx0V"

4.1 graphe



4.2 comments

Intransitive Imperative

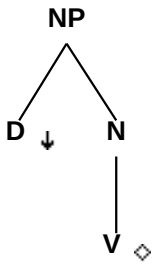
'Sleep!'

4.3 features

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

S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<comp> = nil

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


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

VP.t:<tense> = pres
VP.t:<neg> = -
VP.t:<mode> = base
VP.b:<mode> = V.t:<mode>


VP.b:<compar> = -

VP.b:<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:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>

V.t:<passive> = -
V.t:<punct struct> = nil

NP_0.t:<wh> = -
NP_0.t:<agr pers> = 2
NP_0.t:<agr 3rdsing> = -
NP_0.t:<agr num> = plur/sing
NP_0.t:<case> = nom

NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
```

5 Tree "alphaDnx0V"

5.1 graphe



5.2 comments

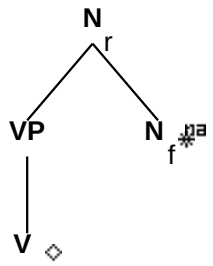
```
Intransitive Determiner gerund tree:

John disapproves of [the drinking]
John disapproves of [Mary's drinking]
```

5.3 features

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

NP.b:<compar> = N.t:<compar>

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

N.t:<compar> = -
```

6 Tree "betaVintransn"

6.1 graphe



6.2 comments

This tree handles things like 'the following items', i.e. -ing adjectives. All intransitive verbs allow this use, while only a limited number of other verb classes do. We are retaining a set of these others as adjectives (in adjectives.txt), on the assumption that this is lexicalized and not fully productive for non-intransitive verbs.

6.3 features

N_r.b:<case> = N_f.t:<case>
N_r.b:<agr> = N_f.t:<agr>
N_r.b:<wh> = N_f.t:<wh>
N_r.b:<pron> = N_f.t:<pron>
N_r.b:<conj> = N_f.t:<conj>

N_r.b:<const> = N_f.t:<const>
N_r.b:<gen> = N_f.t:<gen>
N_r.b:<definite> = N_f.t:<definite>
N_r.b:<quan> = N_f.t:<quan>
N_r.b:<card> = N_f.t:<card>
N_r.b:<decreas> = N_f.t:<decreas>
N_r.b:<compar> = N_f.t:<compar>

N_f.t:<case> = nom/acc
N_f.t:<compar> = -

VP.t:<mode> = VP.b:<mode>

VP.b:<compar> = -

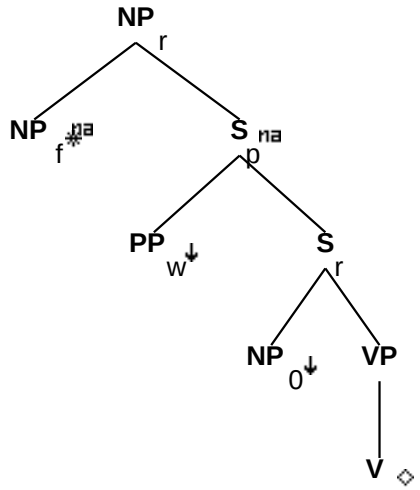
VP.b:<mode> = V.t:<mode>

V.t:<mode> = ger

V.t:<punct struct> = nil

7 Tree "betaNpxnx0V"

7.1 graphe



7.2 comments

Adding Null Comp node to represent
the [Spec, CP], Comp analysis of
relative clauses.

Also adding select-mode specification of ind/inf on the PP_w

7.3 features

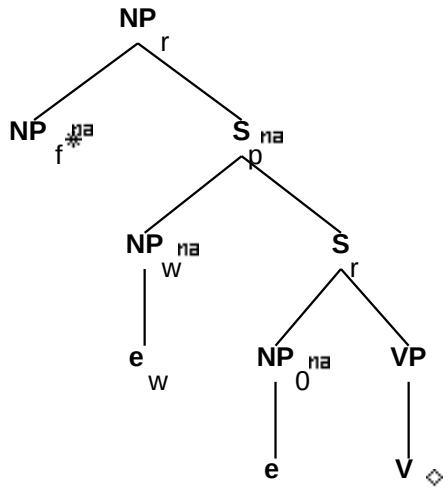
NP_r.b:<rel-clause> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>
NP_f.b:<case> = acc/nom
PP_w.t:<wh> = +

S_r.t:<comp> = nil
S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<agr> = VP.t:<agr>

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

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>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0.t:<wh> = -
VP.b:<compar> = -

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:<passive> = V.t:<passive>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
V.t:<passive> = -
V.t:<punct struct> = nil
```

8 Tree "betaNc0nx0V"

8.1 graphe



8.2 comments

```
Wh on the subject.  Need to decide what VP agrees with.
```

Adding null NP_w node ([Spec, CP] node) to host the null operator.

The eqn. S_r.t:<mode> = inf/ind is replaced by S_r.t:<mode> = inf/ger/ind to allow reduced relatives with progressives.

The feature equation: S_r.b:<rel-pron> = S_r.b:<mode> is being added to make sure that adjunction is not forced with reduced relative subject extractions: the book written by Bill, the boy reading the book (these examples are not meant to be specific to this tree family) but is forced for indicative subject extractions with a covert Comp * the boy [t_i read the book] The rel-pron feature of the null Comp is ppart/ger/adjoined-clause

8.3 features

NP_r.b:<rel-clause> = +
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_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

NP_f.b:<case> = nom/acc
NP_w.t:<agr> = NP_0.t:<agr>
NP_w.t:<case> = NP_0.t:<case>
NP_w.t:<trace> = NP_0.t:<trace>
S_r.t:<nocomp-mode> = inf/ger
S_r.t:<inv> = -
S_r.t:<conj> = nil
S_r.t:<mode> = inf/ger/ind

S_r.b:<comp> = nil

S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.b:<mode> = S_r.b:<nocomp-mode>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>

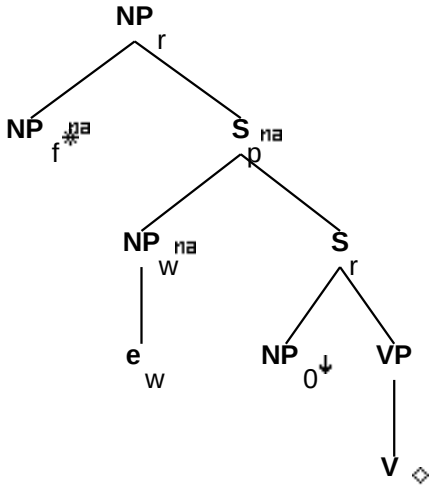
VP.t:<assign-comp> = that/inf_nil/ind_nil/ecm
VP.b:<compar> = -

VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>

VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
V.t:<passive> = -
V.t:<punct struct> = nil

9 Tree "betaNcnx0V"

9.1 graphe



9.2 comments

9.3 features

NP_r.b:<rel-clause> = +
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_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

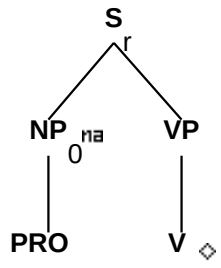
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -

S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
S_r.b:<extracted> = -
S_r.b:<comp> = nil
S_r.b:<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>
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
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:<nocomp-mode> = S_r.b:<mode>
S_r.b:<control> = NP_0.t:<control>
NP_0.t:<wh> = -
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
VP.t:<assign-comp> = that/for/ind_nil

VP.b:<compar> = -
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>
V.t:<passive> = -
V.t:<punct struct> = nil
```

10 Tree "alphanx0V-PRO"

10.1 graphe



10.2 comments

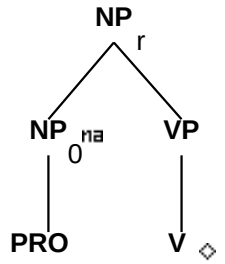
```
Intransitive declarative tree with PRO subject

I want [PRO to sleep].
While [PRO sleeping] I choked.
```

10.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<extracted> = -
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>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<wh> = NP_0.t:<wh>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<control> = NP_0.t:<control>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0.t:<wh> = -
NP_0.t:<wh> = -
NP_0.t:<case> = none
VP.b:<compar> = -
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:<passive> = V.t:<passive>
VP.b:<assign-comp> = V.t:<assign-comp>
V.t:<passive> = -
V.t:<punct struct> = nil
VP.t:<mode> = inf/ger
```

11 Tree "alphaGnx0V-PRO"

11.1 graphe



11.2 comments

```
Intransitive NP gerund tree w/ PRO subject
```

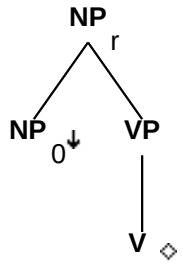
[PRO wandering] disturbs everyone.

11.3 features

NP_r.b:<gerund> = +
NP_r.b:<agr pers> = 3
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr 3rdsing> = +
NP_r.b:<wh> = NP_0.t:<wh>
NP_r.b:<compar> = NP_0.t:<compar>
NP_0.t:<wh> = -
NP_0.t:<case> = none
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<punct struct> = nil

12 Tree "alphaGnx0V"

12.1 graphe



12.2 comments

Intransitive NP gerund tree:

[Mr. Nolen's wandering] disturbs everyone

12.3 features

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

NP_r.b:<wh> = NP_0.t:<wh>

```
NP_r.b:<compar> = NP_0.t:<compar>

VP.t:<mode> = ger
VP.b:<compar> = -

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<punct struct> = nil
NP_0:<case> = acc/gen
```