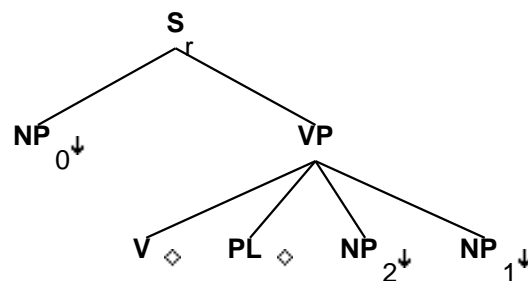


Family "Tnx0Vplnx2nx1"

March 5, 2008

1 Tree "alphanx0Vplnx2nx1"

1.1 graphe



1.2 comments

no comments

1.3 features

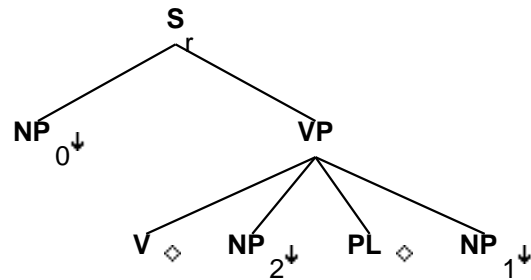
```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<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:<wh>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
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_r.b:<control> = NP_0.t:<control>
 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 "alphanx0Vnx2plnx1"

2.1 graphe



2.2 comments

no comments

2.3 features

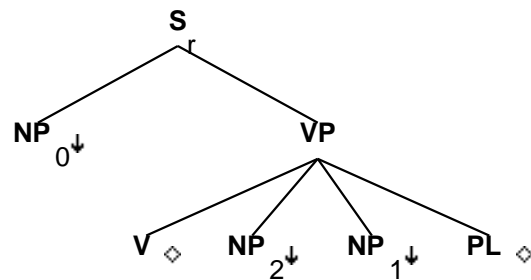
S_r.b:<extracted> = -
 S_r.b:<inv> = -
 S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
 S_r.b:<comp> = nil
 S_r.b:<tense> = VP.t:<tense>
 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_1:<case> = acc
 NP_2:<case> = acc
 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_r.b:<control> = NP_0.t:<control>
 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>

3 Tree "alphax0Vnx2nx1pl"

3.1 graphe



3.2 comments

no comments

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>
 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> = -

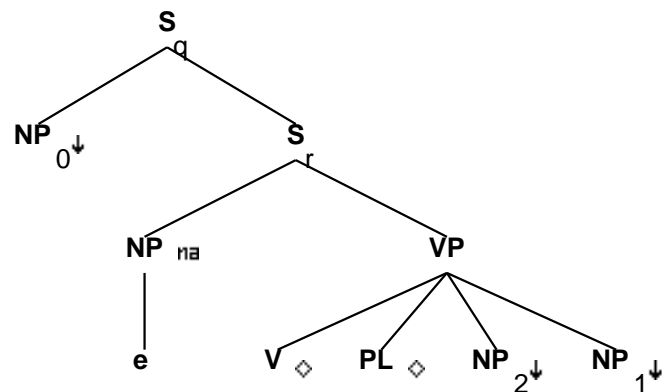
```

NP_1:<case> = acc
NP_2:<case> = acc
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_r.b:<control> = NP_0.t:<control>
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>

```

4 Tree "alphaW0nx0Vplnx2nx1"

4.1 graphe



4.2 comments

Need to decide what VP agrees with.

4.3 features

```

S_q.b:<extracted> = +
S_q.b:<inv> = S_r.t:<inv>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>

```

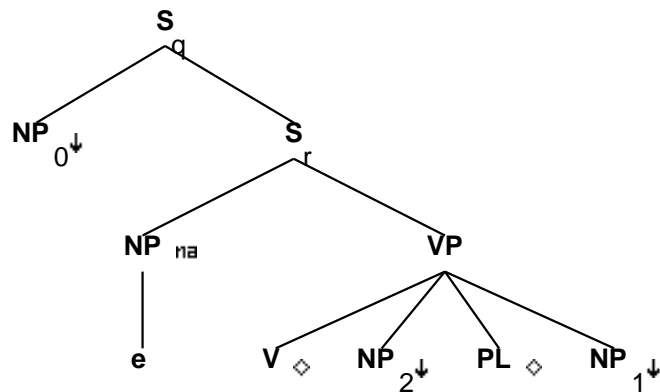
```

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:<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:<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>
NP_1:<case> = acc
NP_2:<case> = acc
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_r.t:<conj> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm

```

5 Tree "alphaW0nx0Vnx2plnx1"

5.1 graphe



5.2 comments

Need to decide what VP agrees with.

5.3 features

S_q.b:<extracted> = +

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

S_r.t:<comp> = nil

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

S_q.b:<wh> = NP₀:<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:<trace> = NP₀:<trace>

NP:<agr> = NP₀:<agr>

NP:<case> = NP₀:<case>

NP:<wh> = NP₀:<wh>

NP₀:<wh> = +

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>

NP₁:<case> = acc

NP₂:<case> = acc

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>

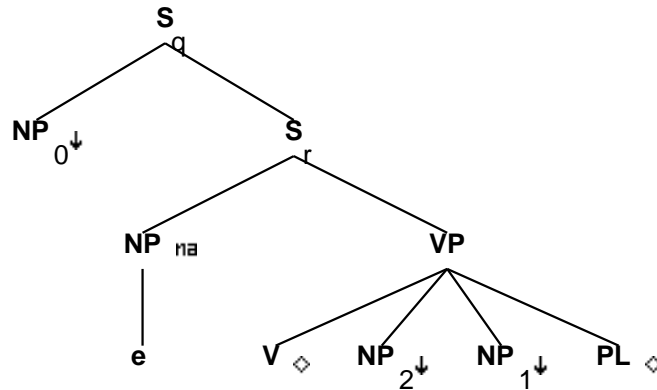
VP.b:<compar> = -

S_r.t:<conj> = nil

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

6 Tree "alphaW0nx0Vnx2nx1pl"

6.1 graphe



6.2 comments

Need to decide what VP agrees with.

6.3 features

S_q.b:<extracted> = +

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

S_r.t:<comp> = nil

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

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:<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:<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>

NP_1:<case> = acc

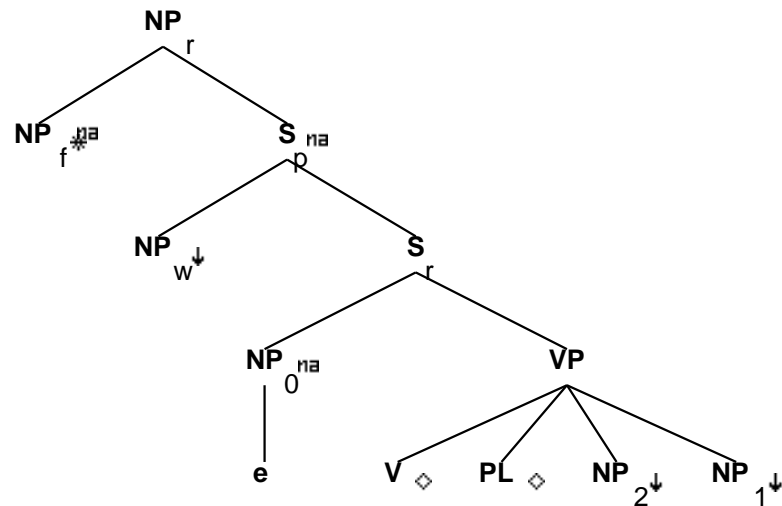
NP_2:<case> = acc

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_r.t:<conj> = nil
 S_r.b:<assign-comp> = inf_nil/ind_nil/ecm

7 Tree "betaN0nx0Vplnx2nx1"

7.1 graphe



7.2 comments

Need to decide what VP agrees with.

7.3 features

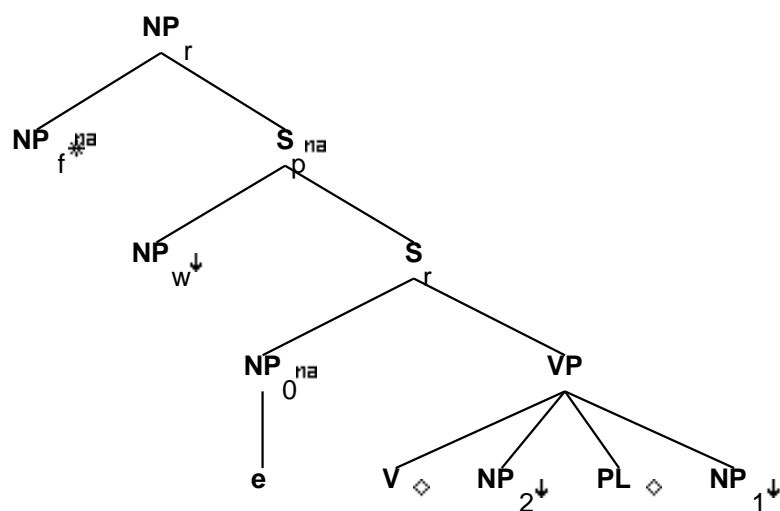
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
 S_r.b:<tense> = VP.t:<tense>
 S_r.t:<mode> = ind/inf
 S_r.t:<inv> = -
 NP_f.t:<wh> = NP_r.b:<wh>
 NP_f.t:<agr> = NP_r.b:<agr>


```
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>
```

```
8  Tree "betaN0nx0Vnx2plnx1"
```

8.1 graphe



8.2 comments

Need to decide what VP agrees with.

8.3 features

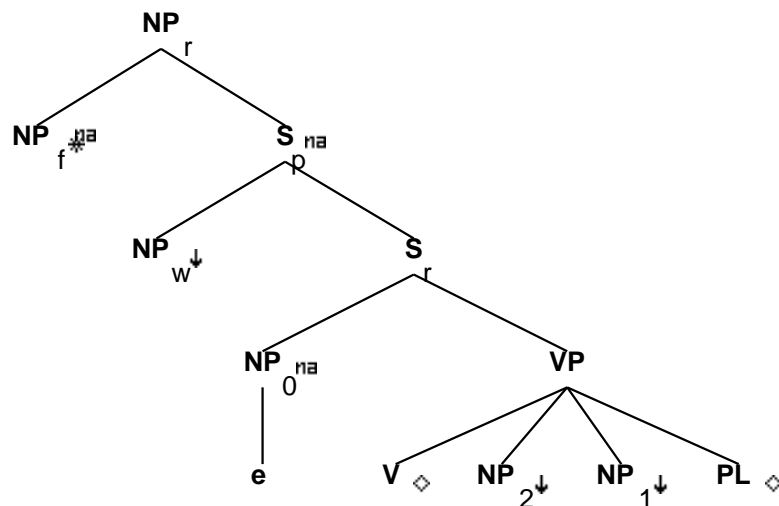
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
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>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

9 Tree "betaN0nx0Vnx2nx1pl"

9.1 graphe



9.2 comments

Need to decide what VP agrees with.

9.3 features

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

```

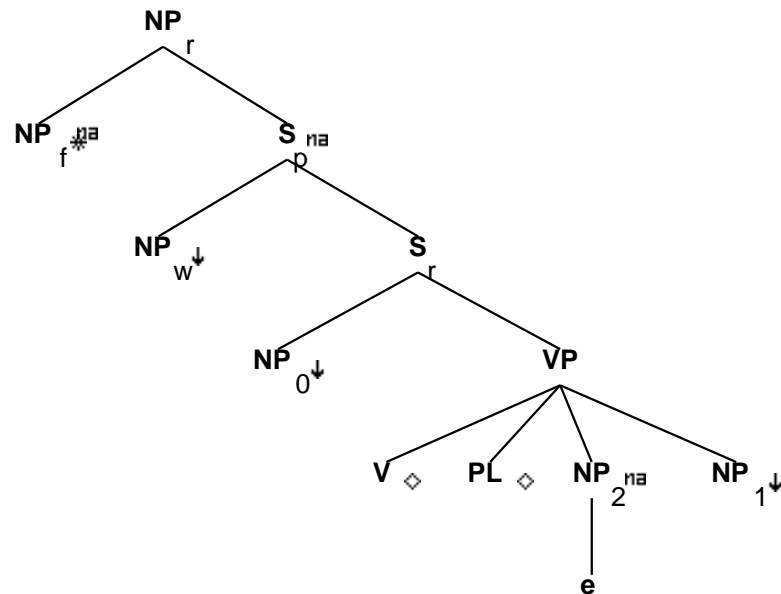
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

10 Tree "betaN1nx0Vplnx2nx1"

10.1 graphe



10.2 comments

No original comments.

10.3 features

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

```

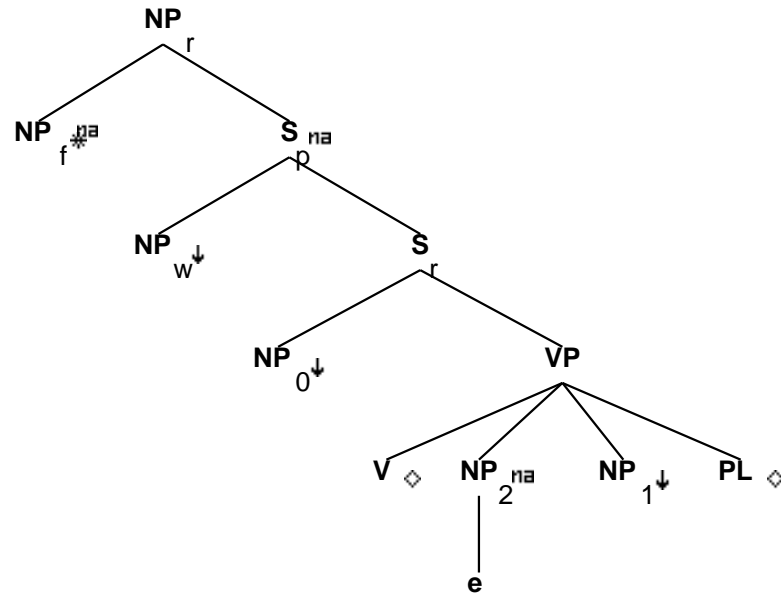
S_r.b:<mode> = VP.t:<mode>
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
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.t:<case> = acc
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.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>

```

11 Tree "betaN1nx0Vnx2nx1pl"

11.1 graphe



11.2 comments

No original comments.

11.3 features

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

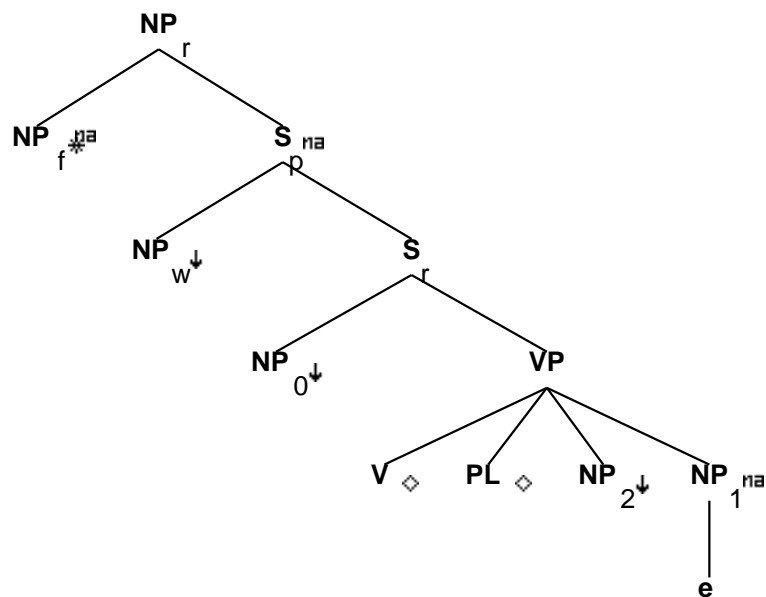
S_r.b:<mode> = VP.t:<mode>
 S_r.t:<inv> = -
 S_r.t:<mode> = ind/inf
 S_r.b:<inv> = -
 NP_0:<agr> = S_r.b:<agr>
 NP_0:<case> = S_r.b:<assign-case>
 NP_2:<case> = acc
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 NP_r.b:<wh> = NP_f.t:<wh>
 NP_r.b:<agr> = NP_f.t:<agr>
 NP_r.b:<case> = NP_f.t:<case>
 S_r.b:<tense> = VP.t:<tense>
 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.t:<case> = acc
 S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
 NP_w.t:<trace> = NP_1.b:<trace>
 NP_w.t:<case> = NP_1.b:<case>
 NP_w.t:<agr> = NP_1.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>

12 Tree "betaN2nx0Vplnx2nx1"

12.1 graphe



12.2 comments

No original comments.

12.3 features

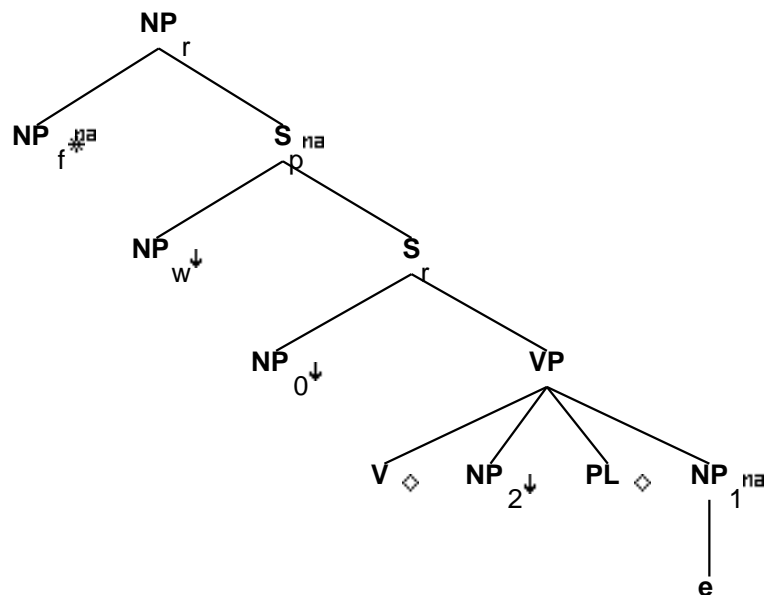
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
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>
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_2.t:<case> = acc
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.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>

13 Tree "betaN2nx0Vnx2plnx1"

13.1 graphe



13.2 comments

No original comments.

13.3 features

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

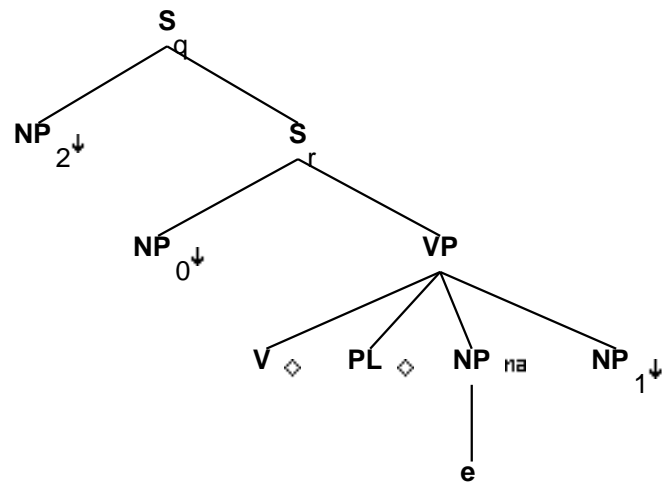
S_r.b:<mode> = VP.t:<mode>
 S_r.t:<mode> = ind/inf
 S_r.t:<inv> = -
 S_r.b:<inv> = -
 NP_0:<agr> = S_r.b:<agr>
 NP_0:<case> = S_r.b:<assign-case>
 NP_1:<case> = acc
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 NP_r.b:<wh> = NP_f.t:<wh>
 NP_r.b:<agr> = NP_f.t:<agr>
 NP_r.b:<case> = NP_f.t:<case>
 S_r.b:<tense> = VP.t:<tense>
 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>
 S_r.b:<tense> = VP.t:<tense>
 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_2.t:<case> = acc
 S_r.t:<conj> = nil

 S_r.b:<control> = NP_0.t:<control>
 NP_w.t:<trace> = NP_2.b:<trace>
 NP_w.t:<case> = NP_2.b:<case>
 NP_w.t:<agr> = NP_2.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>

14 Tree "alphaW1nx0Vplnx2nx1"

14.1 graphe



14.2 comments

No original comments.

14.3 features

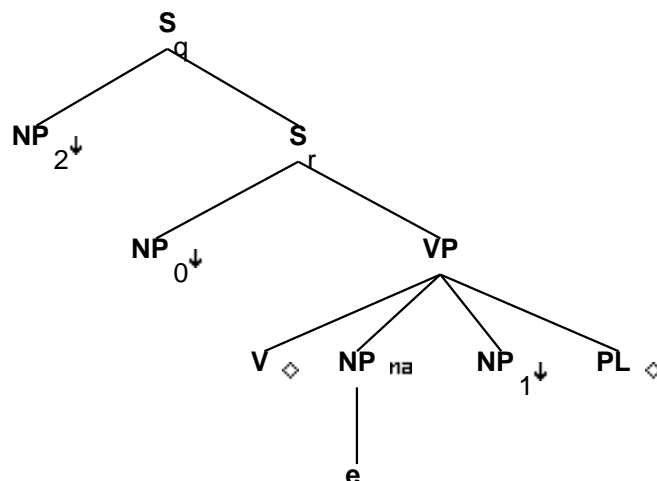
```
S_q.b:<extracted> = +

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_q.b:<comp> = nil
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>
NP_2:<case> = NP:<case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP:<trace> = NP_1:<trace>
NP:<agr> = NP_1:<agr>
NP:<case> = NP_1:<case>
NP:<wh> = NP_1:<wh>
S_r.b:<tense> = VP.t:<tense>
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.t:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
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 "alphaW1nx0Vnx2nx1pl"

15.1 graphe



15.2 comments

No original comments.

15.3 features

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_q.b:<comp> = nil

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>

NP_2:<case> = NP:<case>

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

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

NP:<trace> = NP_1:<trace>

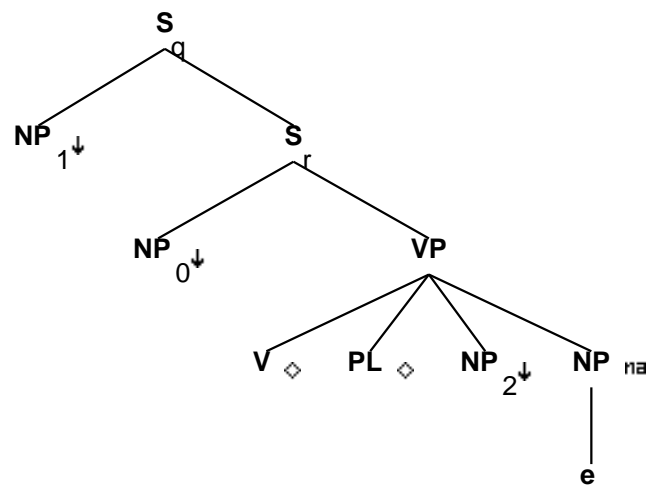
NP:<agr> = NP_1:<agr>

NP:<case> = NP_1:<case>

NP:<wh> = NP_1:<wh>
 S_r.b:<tense> = VP.t:<tense>
 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.t:<case> = acc
 S_r.t:<conj> = nil
 S_r.b:<control> = NP_0.t:<control>
 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>

16 Tree "alphaW2nx0Vplnx2nx1"

16.1 graphe



16.2 comments

No original comments.

16.3 features

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>

```

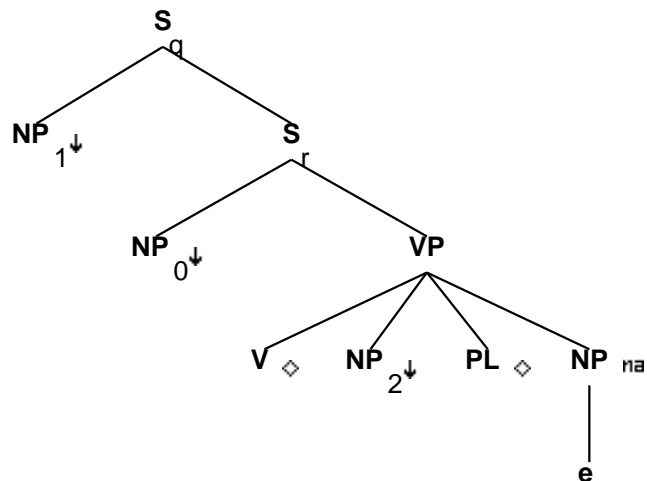
```

S_q.b:<wh> = NP_2:<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_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = NP:<case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP:<trace> = NP_2:<trace>
NP:<agr> = NP_2:<agr>
NP:<case> = NP_2:<case>
NP:<wh> = NP_2:<wh>
S_r.b:<tense> = VP.t:<tense>
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.t:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
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>

```

17 Tree "alphaW2nx0Vnx2plnx1"

17.1 graphe



17.2 comments

No original comments.

17.3 features

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_2:<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_0:<agr> = S_r.b:<agr>

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

NP_1:<case> = NP:<case>

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

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

NP:<trace> = NP_2:<trace>

NP:<agr> = NP_2:<agr>

NP:<case> = NP_2:<case>

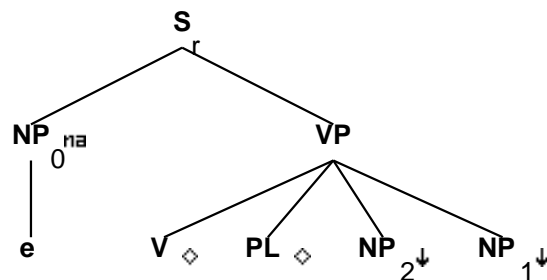
```

NP:<wh> = NP_2:<wh>
S_r.b:<tense> = VP.t:<tense>
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.t:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
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>

```

18 Tree "alphaInx0Vplnx2nx1"

18.1 graphe



18.2 comments

no comments

18.3 features

```

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

```

```

S_r.b:<mode> = imp
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>

```



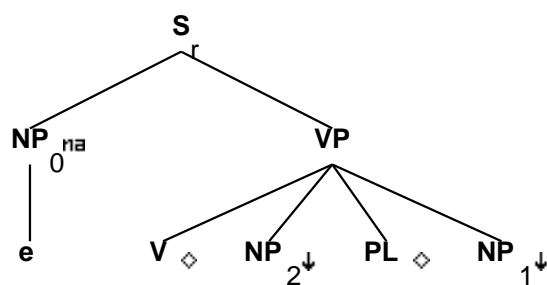
```

NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

```

19 Tree "alphaInx0Vnx2plnx1"

19.1 graphe



19.2 comments

no comments

19.3 features

S_r.b:<extracted> = -

```

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

```

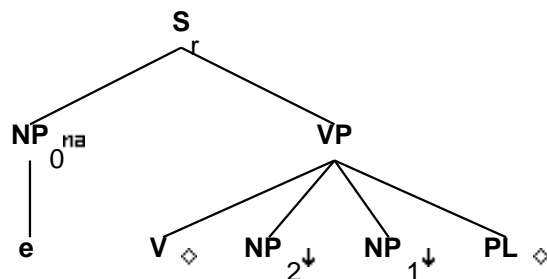
```

S_r.b:<mode> = imp
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:<wh>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

```

20 Tree "alphaInx0Vnx2nx1pl"

20.1 graphe



20.2 comments

no comments

20.3 features

```

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

```

```

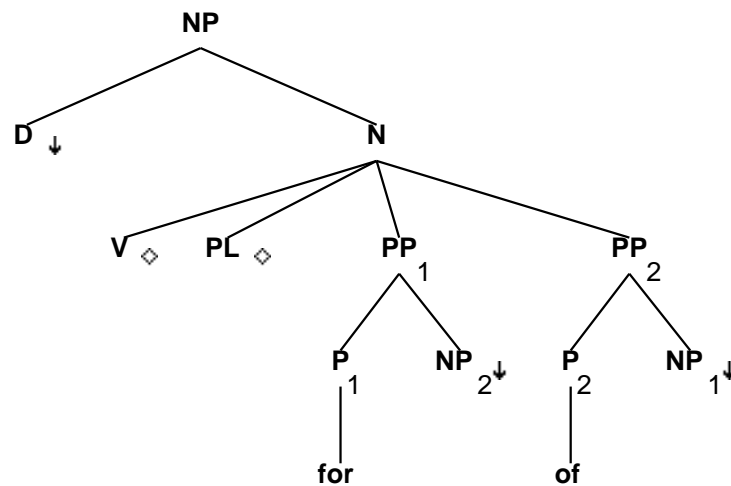
S_r.b:<mode> = imp
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
NP_1:<case> = acc
NP_2:<case> = acc
S_r.b:<agr> = 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_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>

21 Tree "alphaDnx0Vplnx2nx1"

21.1 graphe



21.2 comments

Ditransitive verb particle - Determiner Gerund

'the opening up for Michelle of the new bank account'

21.3 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> = +

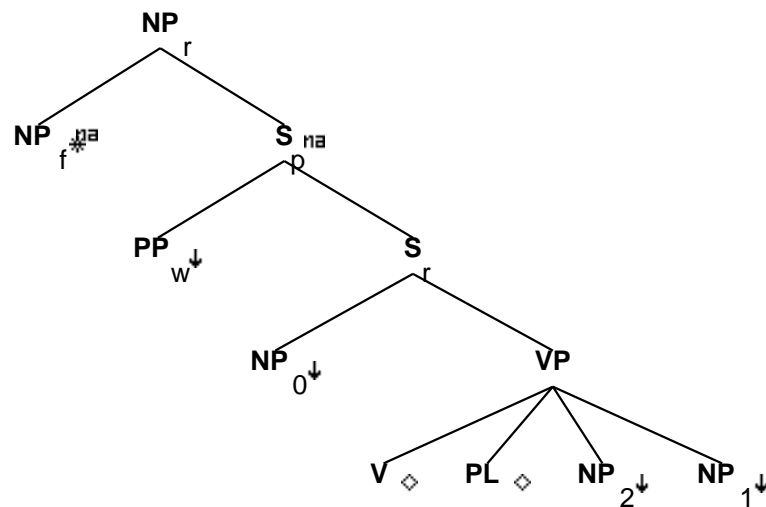
```

P_1.b:<assign-case> = acc
PP_1.b:<assign-case> = P_1.t:<assign-case>
PP_1.b:<assign-case> = NP_1.t:<case>
P_2.b:<assign-case> = acc
PP_2.b:<assign-case> = P_2.t:<assign-case>
PP_2.b:<assign-case> = NP_2.t:<case>

```

22 Tree "betaNpxnx0Vplnx2nx1"

22.1 graphe



22.2 comments

no comments

22.3 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>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>

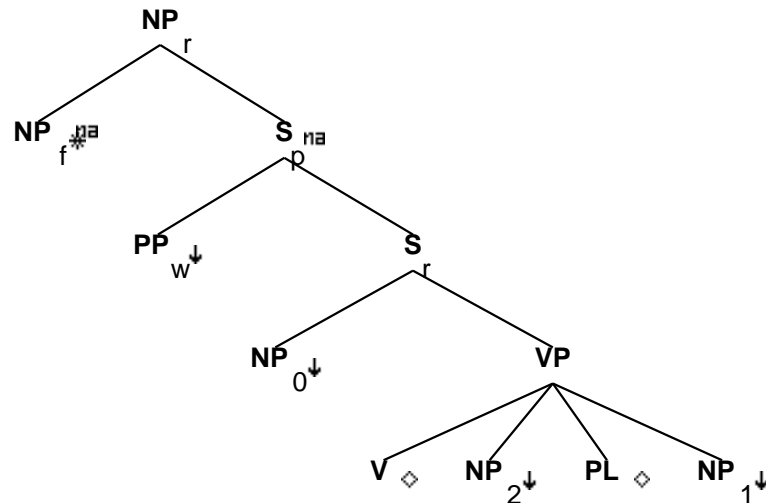
```

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_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
 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>

23 Tree "betaNpxnx0Vnx2plnx1"

23.1 graphe



23.2 comments

no comments

23.3 features

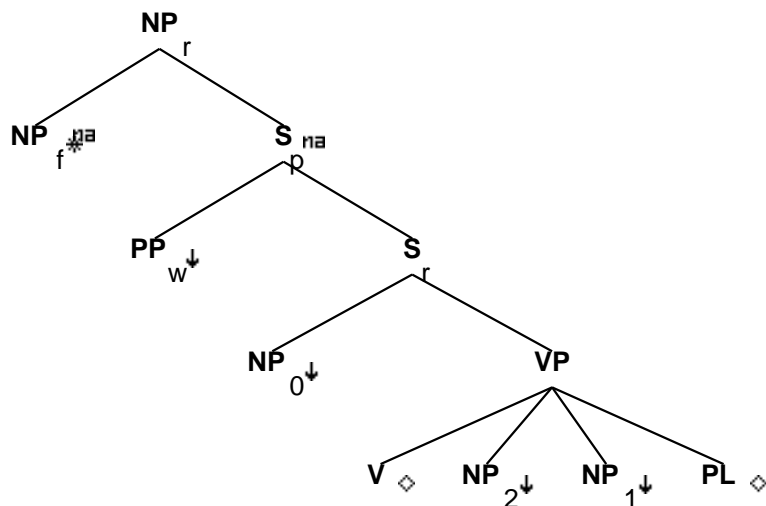
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
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_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
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>

24 Tree "betaNpxnx0Vnx2nx1pl"

24.1 graphe



24.2 comments

no comments

24.3 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>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
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>

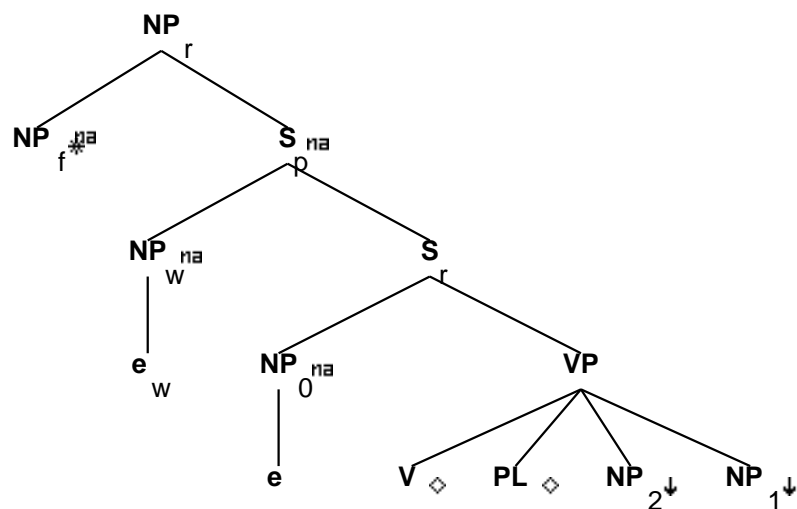
```


VP.b:<mainv> = V.t:<mainv>
 VP.b:<compar> = -
 S_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
 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>

25 Tree "betaNc0nx0Vplnx2nx1"

25.1 graphe



25.2 comments

Need to decide what VP agrees with.

25.3 features

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

```

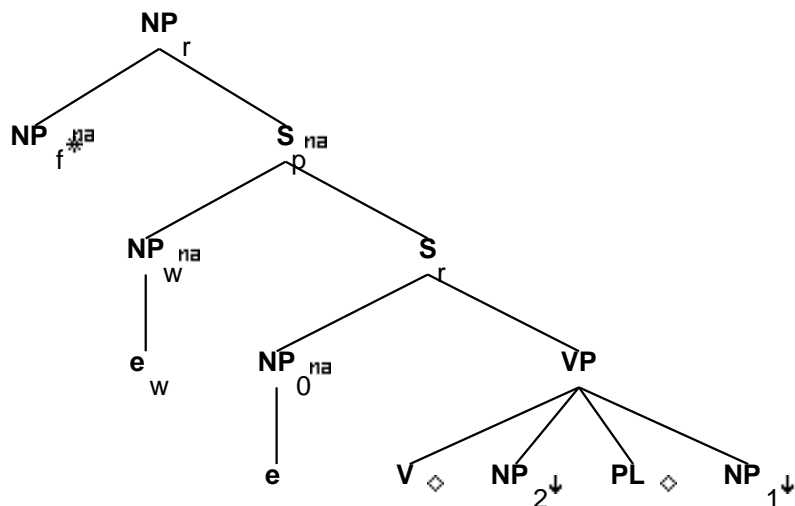
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

```

26 Tree "betaNc0nx0Vnx2plnx1"

26.1 graphe



26.2 comments

Need to decide what VP agrees with.

26.3 features

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

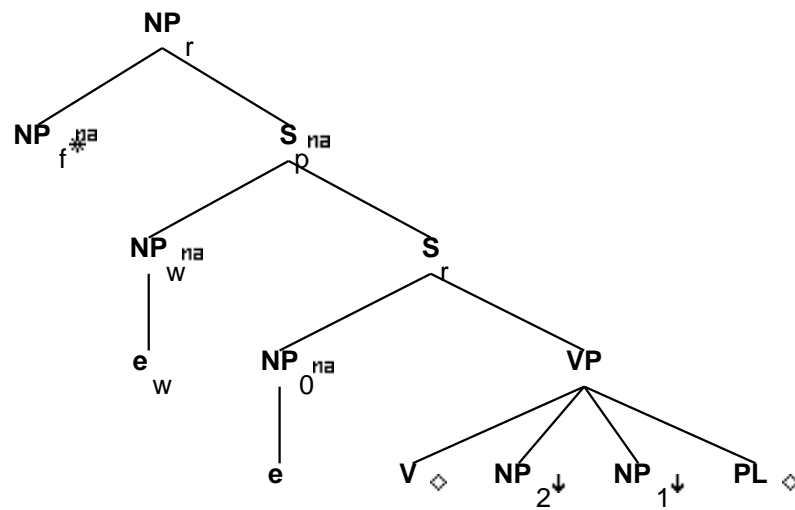
S_r.b:<mode> = VP.t:<mode>
 S_r.b:<tense> = VP.t:<tense>
 S_r.t:<inv> = -
 NP_f.t:<wh> = NP_r.b:<wh>
 NP_f.t:<agr> = NP_r.b:<agr>
 NP_f.t:<case> = NP_r.b:<case>
 S_r.b:<comp> = nil
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 S_r.b:<agr> = NP_0.t:<agr>
 S_r.b:<assign-case> = NP_0.t:<case>
 NP_1:<case> = acc
 NP_2:<case> = acc
 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_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>

27 Tree "betaNc0nx0Vnx2nx1pl"

27.1 graphe



27.2 comments

Need to decide what VP agrees with.

27.3 features

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

```

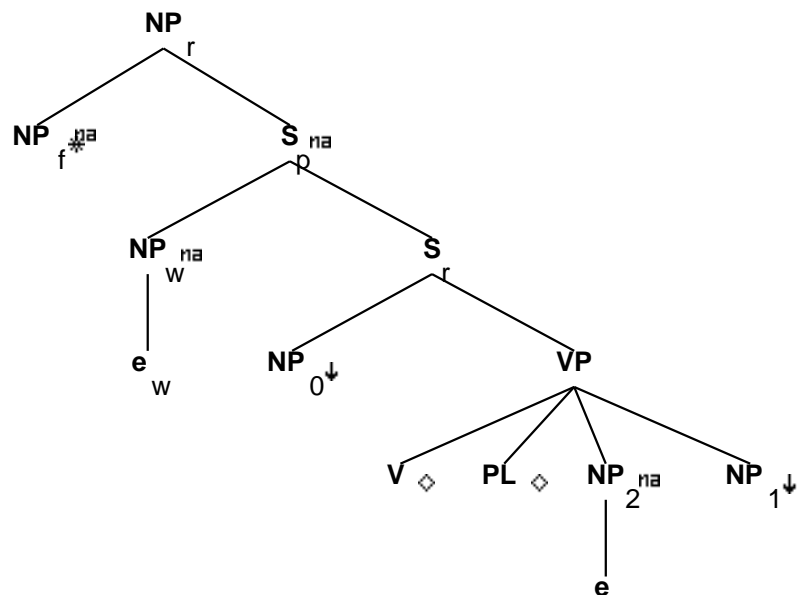
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

```

28 Tree "betaNc2nx0Vplnx2nx1"

28.1 graphe



28.2 comments

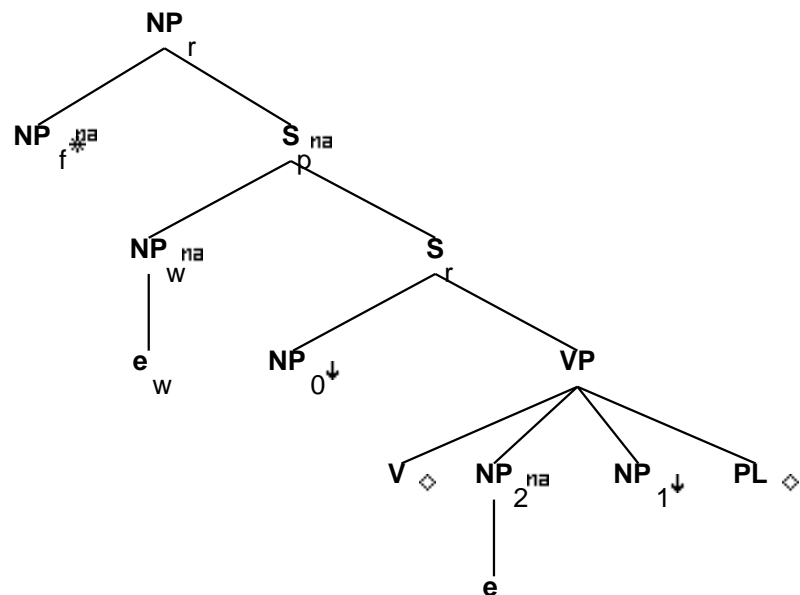
No original comments.

28.3 features

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

S_r.b:<mode> = VP.t:<mode>
 S_r.t:<inv> = -
 S_r.b:<inv> = -
 NP_0:<agr> = S_r.b:<agr>
 NP_0:<case> = S_r.b:<assign-case>
 NP_2:<case> = acc
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 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>
 VP.b:<passive> = V.t:<passive>
 V.t:<passive> = -
 VP.b:<agr> = V.t:<agr>

```
S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```



29.3 features

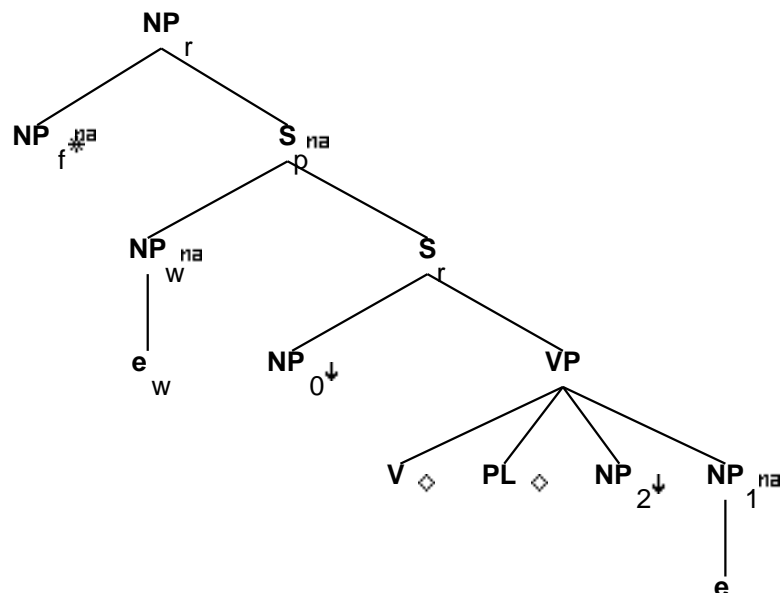
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
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>
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.t:<case> = acc
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

30 Tree "betaNc1nx0Vplnx2nx1"

30.1 graphe



30.2 comments

No original comments.

30.3 features

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

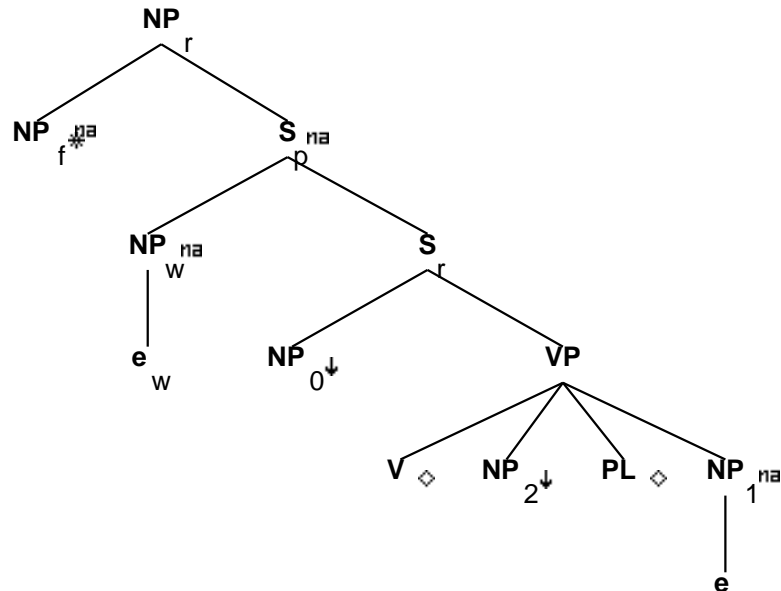
S_r.b:<mode> = VP.t:<mode>
 S_r.t:<inv> = -
 S_r.b:<inv> = -
 NP_0:<agr> = S_r.b:<agr>
 NP_0:<case> = S_r.b:<assign-case>
 NP_1:<case> = acc
 S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-case> = VP.t:<assign-case>
 NP_r.b:<wh> = NP_f.t:<wh>
 NP_r.b:<agr> = NP_f.t:<agr>
 NP_r.b:<case> = NP_f.t:<case>
 S_r.b:<tense> = VP.t:<tense>
 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_2.t:<case> = acc
 S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
 NP_w.t:<trace> = NP_2.b:<trace>
 NP_w.t:<case> = NP_2.b:<case>
 NP_w.t:<agr> = NP_2.b:<agr>
 NP_r.b:<rel-clause> = +
 S_r.t:<mode> = inf/ind
 S_r.t:<nocomp-mode> = ind
 VP.t:<assign-comp> = that/for/ind_nil
 S_r.b:<nocomp-mode> = S_r.b:<mode>
 NP_f.b:<case> = nom/acc
 NP_r.b:<pron> = NP_f.t:<pron>

31 Tree "betaNc1nx0Vnx2plnx1"

31.1 graphe



31.2 comments

No original comments.

31.3 features

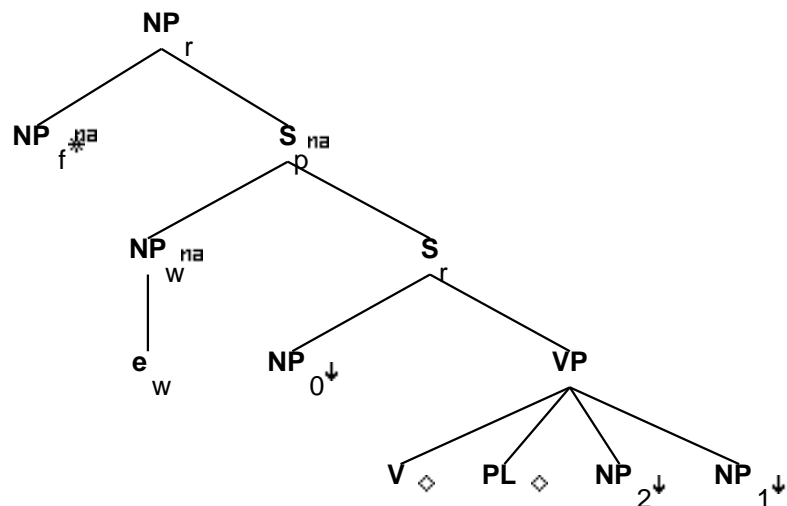
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
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>
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>
S_r.b:<tense> = VP.t:<tense>
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_2.t:<case> = acc
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

32 Tree "betaNcnx0Vplnx2nx1"

32.1 graphe



32.2 comments

no comments

32.3 features

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

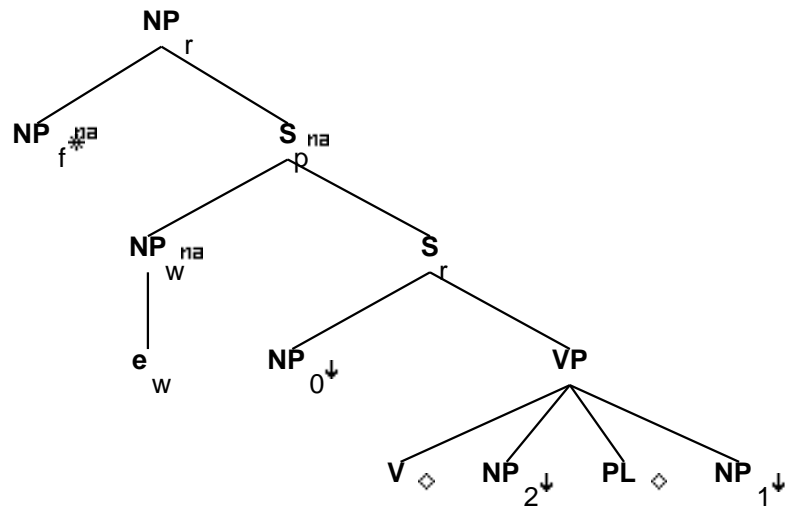
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<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_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
 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>

33 Tree "betaNcnx0Vnx2plnx1"

33.1 graphe



33.2 comments

no comments

33.3 features

S_r.b:<extracted> = -

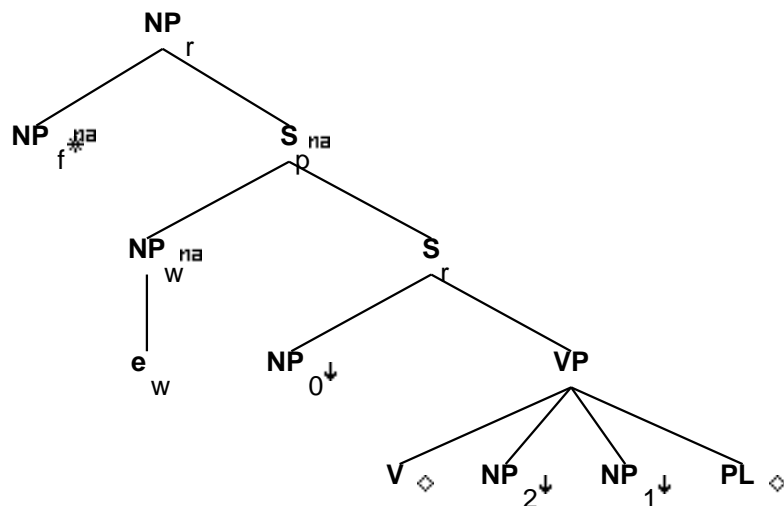
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
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_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
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>

34 Tree "betaNcnx0Vnx2nx1pl"

34.1 graphe



34.2 comments

no comments

34.3 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>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = acc
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>

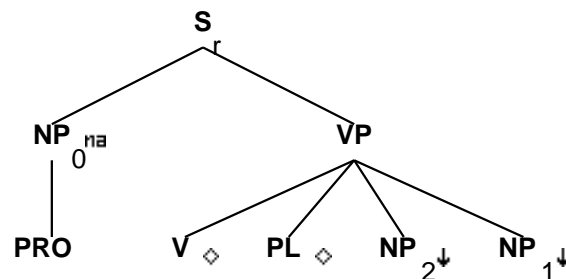
```

VP.b:<mainv> = V.t:<mainv>
 VP.b:<compar> = -
 S_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
 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>

35 Tree "alphax0Vplnx2nx1-PRO"

35.1 graphe



35.2 comments

Ditransitive Verb Particle - particle before NPs, w/ PRO subject

John wants [PRO to open up Mary a bank account].

While [PRO opening up Mary a bank account] John met all the tellers.

35.3 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

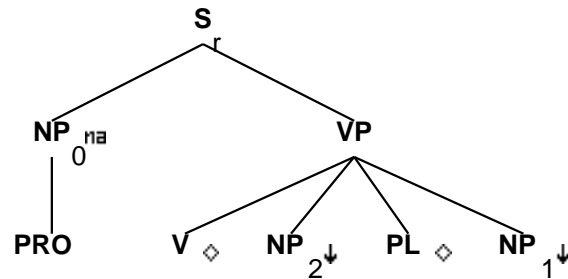

```

S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = S_r.b:<wh>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0:<wh> = -
NP_0.t:<case> = none
NP_1:<case> = acc
NP_2:<case> = acc
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_r.b:<control> = NP_0.t:<control>
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

```

36 Tree "alphanx0Vnx2plnx1-PRO"

36.1 graphe



36.2 comments

Ditransitive Verb Particle - particle between NPs, w/ PRO subject

John wanted [PRO to open Mary up a bank account].

While [PRO opening Mary up a bank account] John met all the tellers.

36.3 features

```

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

```

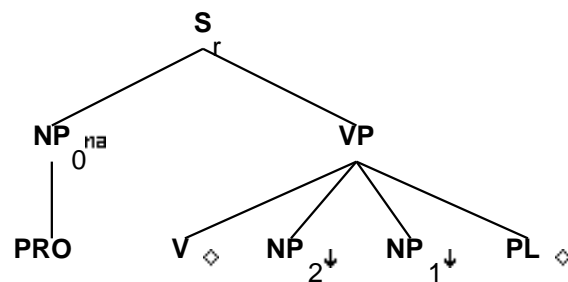
```

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>
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
NP_1:<case> = acc
NP_2:<case> = acc
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_r.b:<control> = NP_0.t:<control>
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

```

37 Tree "alphax0Vnx2nx1pl-PRO"

37.1 graphe



37.2 comments

Ditransitive Verb Particle - particle after NPs, w/ PRO subject

John wanted [PRO to open Mary a bank account up].

While [PRO opening Mary a bank account up] John met all the tellers.

37.3 features

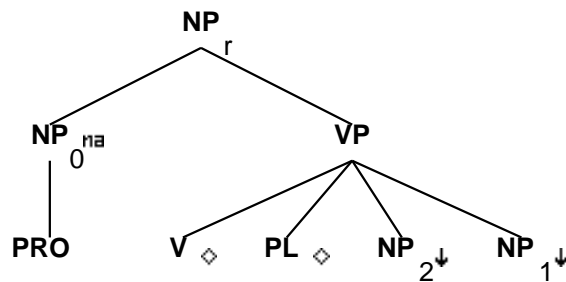
```

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>
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
NP_1:<case> = acc
NP_2:<case> = acc
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_r.b:<control> = NP_0.t:<control>
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

```

38 Tree "alphaGnx0Vplnx2nx1-PRO"

38.1 graphe



38.2 comments

Ditransitive Verb Particle - NP Gerund, particle before NPs, w/ PRO subject

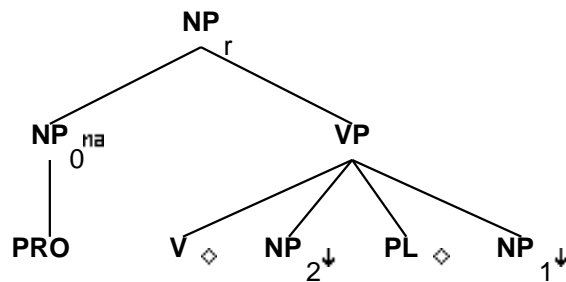
Michelle's father was opposed to [PRO opening up Michelle a new bank account].

38.3 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
NP_1:<case> = acc
NP_2:<case> = acc
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -

39 Tree "alphaGnx0Vnx2plnx1-PRO"

39.1 graphe



39.2 comments

Ditransitive Verb Particle - NP gerund with particle between NPs, w/ PRO subject

Michelle's father was opposed to [PRO opening Michelle up a new bank account].

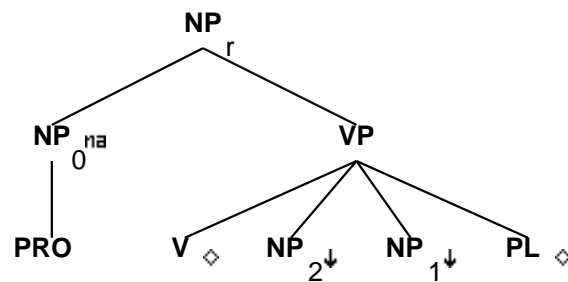
39.3 features

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
NP_1:<case> = acc
NP_2:<case> = acc

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

40 Tree "alphaGnx0Vnx2nx1pl-PRO"

40.1 graphe



40.2 comments

Ditransitive Verb Particle - NP Gerund with particle after NPs, w/ PRO subject

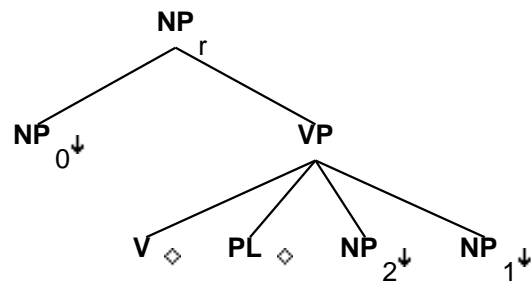
Michelle's father was opposed to [PRO opening Michelle a new bank account up].

40.3 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
 NP_1:<case> = acc
 NP_2:<case> = acc
 VP.b:<mode> = V.t:<mode>
 VP.b:<passive> = V.t:<passive>
 VP.b:<compar> = -
 V.t:<passive> = -

41 Tree "alphaGnx0Vplnx2nx1"

41.1 graphe



41.2 comments

Ditransitive Verb Particle - NP Gerund, particle before NPs

John('s) opening up Michelle a new bank account...

41.3 features

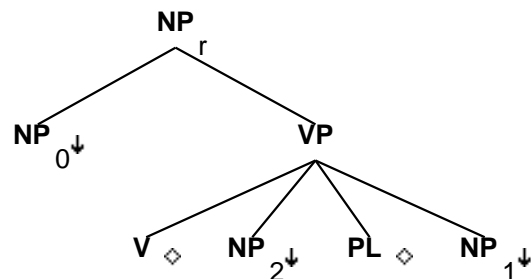
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +

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

NP_1:<case> = acc
NP_2:<case> = acc
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
NP_0:<case> = acc/gen

42 Tree "alphaGnx0Vnx2plnx1"

42.1 graphe



42.2 comments

Ditransitive Verb Particle - NP gerund with particle between NPs

John('s) opening Michelle up a new bank account...

42.3 features

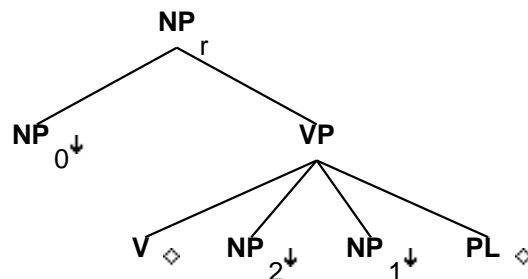
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +

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

NP_1:<case> = acc
NP_2:<case> = acc
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
NP_0:<case> = acc/gen

43 Tree "alphaGnx0Vnx2nx1pl"

43.1 graphe



43.2 comments

Ditransitive Verb Particle - NP Gerund with particle after NPs

... John('s) opening Michelle a new bank account up.

43.3 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

NP_1:<case> = acc

NP_2:<case> = acc

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

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

VP.b:<compar> = -

V.t:<passive> = -

NP_0:<case> = acc/gen