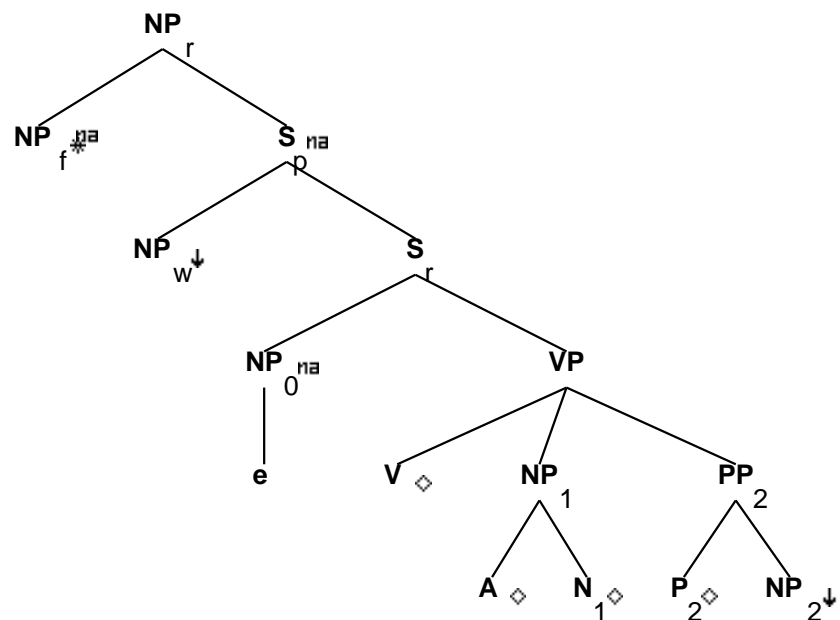


Family "Tnx0VAN1Pnx2"

March 5, 2008

1 Tree "betaN0nx0VAN1Pnx2"

1.1 graphe



1.2 comments

Idiom with V, A, N, and Prep anchors.
Relative clause on the subject.

EX: [The man] who made short work of your objections...

1.3 features

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

```

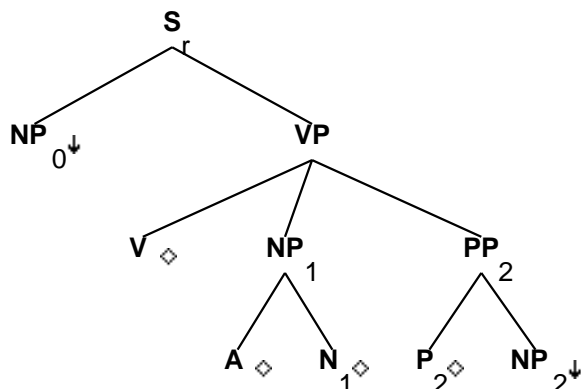
S_r.t:<inv> = -
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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:<compar> = -
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
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
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>

```

2 Tree "alphanx0VAN1Pnx2"

2.1 graphe



2.2 comments

Idiom with V, A, N, and Prep anchors.
Declarative.

EX: Ed made short work of the feast.

2.3 features

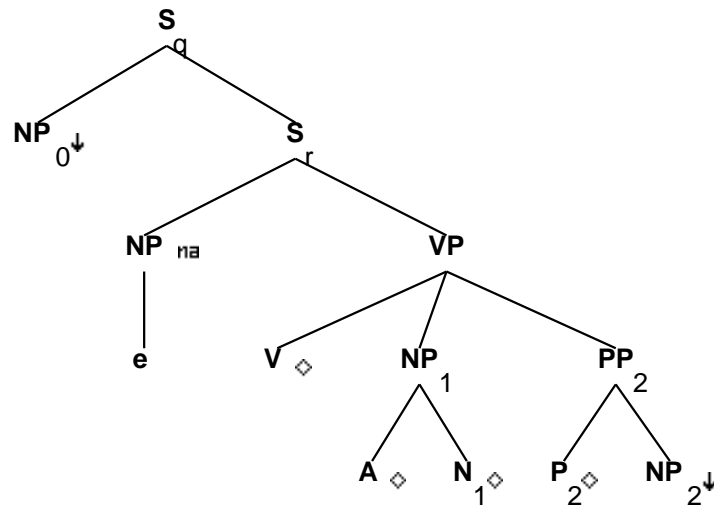
S_r.b:<extracted> = -

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_1:<case> = acc
NP_0:<wh> = -
S_r.b:<wh> = NP_0:<wh>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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:<inv> = -
 N_1:<case> = nom/acc
 NP_1.b:<agr> = N_1.t:<agr>
 S_r.b:<control> = NP_0.t:<control>
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2:<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<case> = PP_2.b:<assign-case>

3 Tree "alphaW0nx0VAN1Pnx2"

3.1 graphe



3.2 comments

Idiom with V, A, N, and Prep anchors.
 Wh-question on the subject.

EX: Who made short work of the feast?

3.3 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

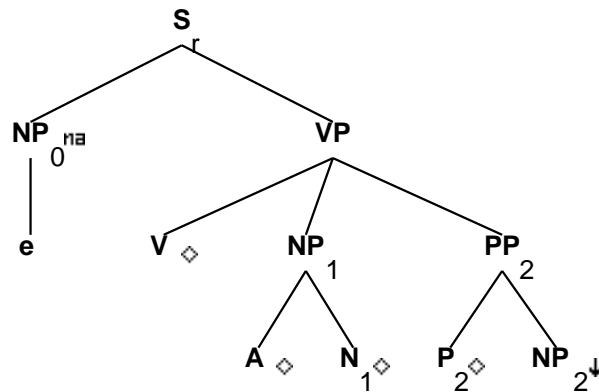
```

S_q.b:<wh> = NP_0:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP.t:<trace> = NP_0.t:<trace>
NP.t:<agr> = NP_0.t:<agr>
NP.t:<case> = NP_0.t:<case>
NP.t:<wh> = NP_0.t:<wh>
NP_0:<wh> = +
NP.t:<agr> = S_r.b:<agr>
NP.t:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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.b:<agr> = N_1.t:<agr>
N_1:<case> = nom/acc
S_r.t:<conj> = nil
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>

```

4 Tree "alphaInx0VAN1Pnx2"

4.1 graphe



4.2 comments

Idiom with V, A, N, and Prep anchors.
Imperative.

EX: Make short work of it!

4.3 features

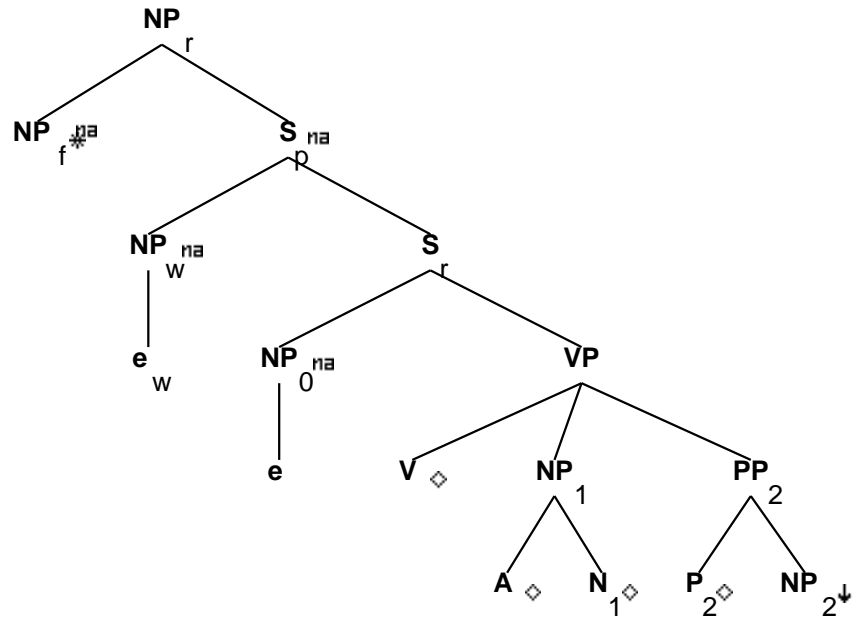
S_r.b:<extracted> = -
S_r.b:<comp> = nil

S_r.b:<inv> = -
S_r.b:<mode> = imp
S_r.b:<tense> = VP.t:<tense>
VP.t:<tense> = pres
S_r.b:<wh> = NP_0:<wh>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<control> = NP_0.t:<control>
VP.t:<neg> = -
VP.t:<mode> = base

VP.b:<mode> = V.t:<mode>
 VP.b:<passive> = V.t:<passive>
 V.t:<passive> = -
 V.t:<contr> = -
 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> = -
 NP_1.b:<agr> = N_1.t:<agr>
 N_1.<case> = nom/acc
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2.<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2.<case> = PP_2.b:<assign-case>

5 Tree "betaNc0nx0VAN1Pnx2"

5.1 graphe



5.2 comments

Idiom with V, A, N, and Prep anchors.
 Relative clause on the subject, with overt Comp.

EX: [The man] that made short work of the wood...

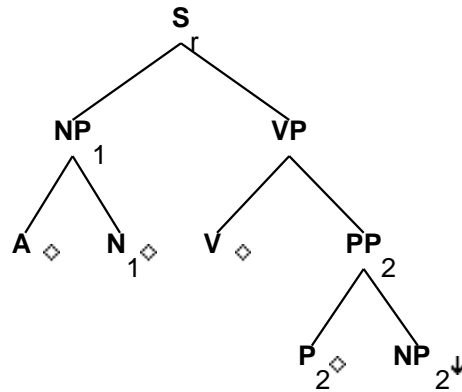
5.3 features

```
S_r.b:<comp> = nil
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.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>
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<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>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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:<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_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>
```


6 Tree "alphaAN1VPnx2"

6.1 graphe



6.2 comments

Idiom with V, A, N, and Prep anchors.
Passive without by-phrase.

EX: Short work was made of the report.

6.3 features

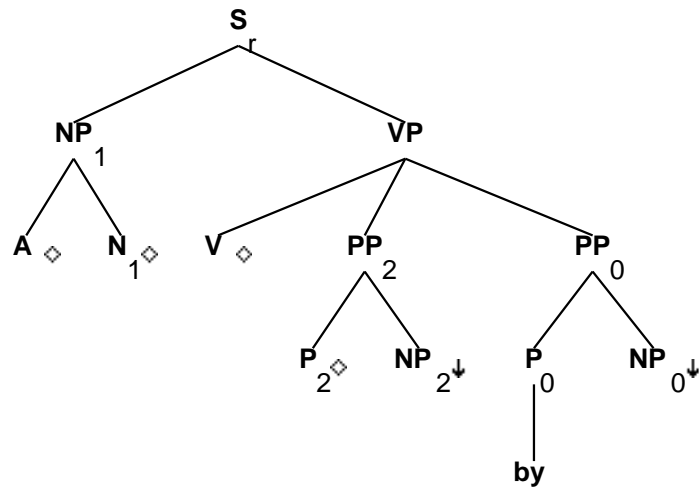
S_r.b:<extracted> = -
S_r.b:<mode> = VP.t:<mode>

S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_1:<wh>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
V.t:<mode> = ppart

V.t:<passive> = +
 S_r.b:<inv> = -
 S_r.b:<control> = NP_1.t:<control>
 NP_1.b:<agr> = N_1.t:<agr>
 N_1.t:<case> = nom/acc
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2:<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<case> = PP_2.b:<assign-case>

7 Tree "alphaAN1VPnx2bynx0"

7.1 graphe



7.2 comments

Idiom with V, A, N, and Prep anchors.
 Passive with by-phrase.

EX: Short work was made of the final by the rambunctious Brazilian team.

7.3 features

S_r.b:<mode> = VP.t:<mode>
 S_r.b:<comp> = nil
 S_r.b:<extracted> = -
 S_r.b:<tense> = VP.t:<tense>
 S_r.b:<wh> = NP_1:<wh>
 NP_1:<agr> = S_r.b:<agr>
 NP_1:<case> = S_r.b:<assign-case>
 NP_1.b:<case> = N_1.t:<case>
 NP_1:<wh> = -

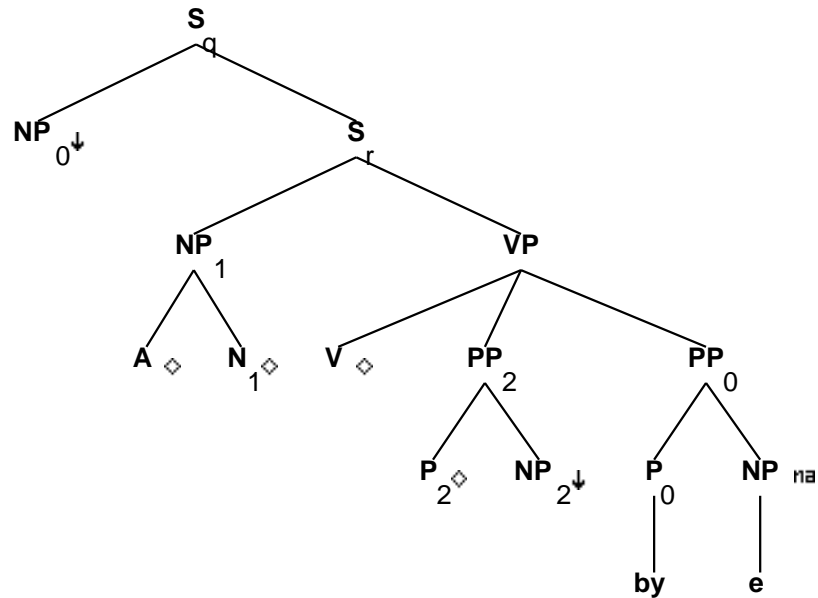
```

S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
S_r.b:<control> = NP_1.t:<control>
PP_0.b:<wh> = NP_0:<wh>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>

```

8 Tree "alphaW0AN1VPnx2bynx0"

8.1 graphe



8.2 comments

Idiom with V, A, N, and Prep anchors.

Wh-question extracted from by-phrase in passive construction.

EX: Who was short work made of the digging by?

Topicalization:

EX: Madeline short work was made of the digging by.

8.3 features

S_r.t:<comp> = nil

S_q.b:<extracted> = +

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

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

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

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

S_q.b:<comp> = nil

S_r.b:<inv> = -

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

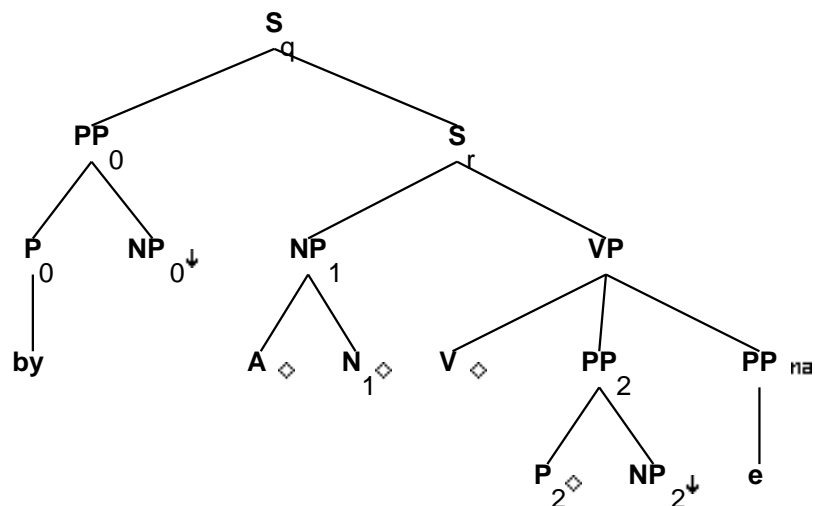
```

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:<assign-comp> = VP.t:<assign-comp>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.b:<control> = NP_1.t:<control>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
V.t:<punct struct> = nil
NP.t:<agr> = NP_0.t:<agr>
NP.t:<case> = NP_0.t:<case>
NP.t:<trace> = NP_0.t:<trace>
NP.t:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP:<case> = PP_0.b:<assign-case>
S_r.t:<conj> = nil
PP_0.b:<wh> = NP:<wh>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>

```

9 Tree "alphaw0AN1VPnx2bynx0"

9.1 graphe



9.2 comments

Idiom with V, A, N, and Prep anchors.

Wh-question on object of extracted by-phrase in passive construction.

EX: By whom was short work made of the digging?

Topicalization:

EX: By Madeline short work was made of the digging.

9.3 features

P_0.b:<assign-case> = acc

PP_0.b:<assign-case> = P_0.t:<assign-case>

S_q.b:<extracted> = +

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

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

NP_0.<case> = PP_0.b:<assign-case>

PP_0.b:<wh> = NP_0.<wh>

S_q.b:<wh> = PP_0.t:<wh>

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

S_q.b:<comp> = nil

S_r.b:<inv> = -

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

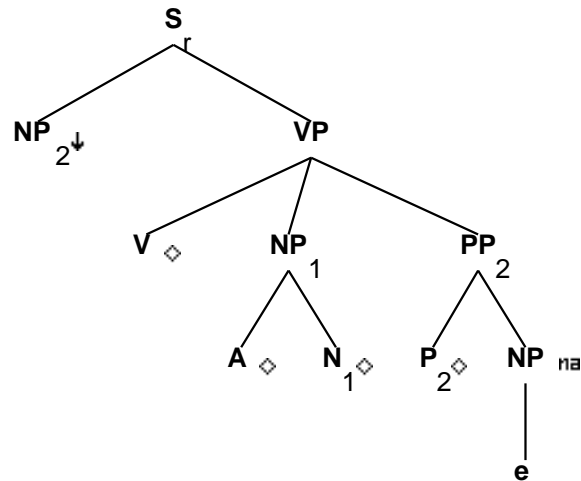
```

S_r.t:<comp> = nil
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:<assign-comp> = VP.t:<assign-comp>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.b:<control> = NP_1.t:<control>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
V.t:<punct struct> = nil
VP.b:<passive> = V.t:<passive>
PP_0.t:<trace> = PP.t:<trace>
S_r.t:<conj> = nil
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>

```

10 Tree "alphanx2VAN1P"

10.1 graphe



10.2 comments

Idiom with V, A, N, and Prep anchors.
Outer passive without by-phrase.

EX: The chopping was made short work of.

10.3 features

```
S_r.b:<extracted> = -
S_r.b:<mode> = VP.t:<mode>

S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
```



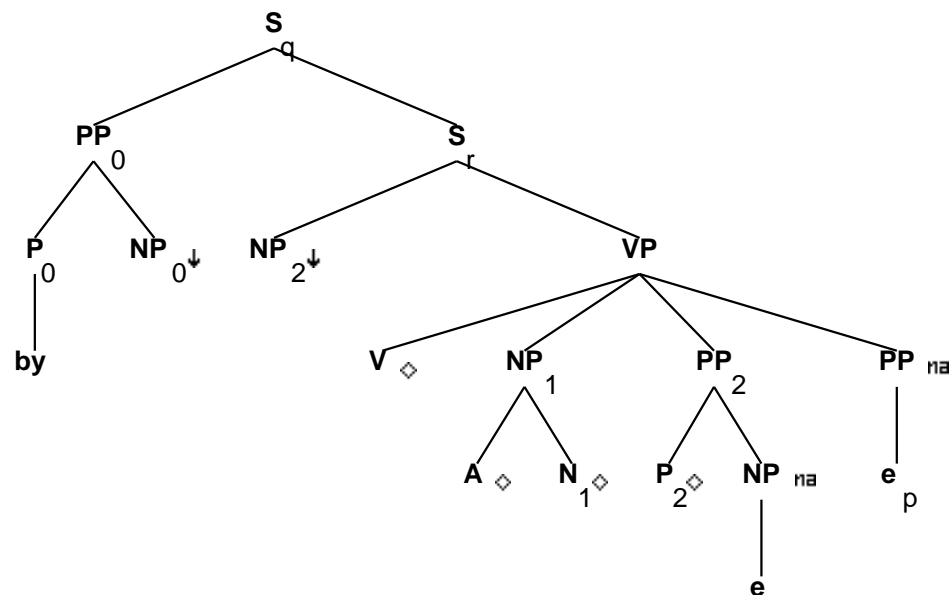
```

V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
S_r.b:<control> = NP_2.t:<control>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>

```

11 Tree "alphapW0nx2VAN1Pbynx0"

11.1 graphe



11.2 comments

Idiom with V, A, N, and Prep anchors.
Outer passive with a wh-moved by-phrase.

EX: By whom was the chopping made short work of?

11.3 features

```
PP_0.b:<assign-case> = P_0.t:<assign-case>
```

```

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

```

```

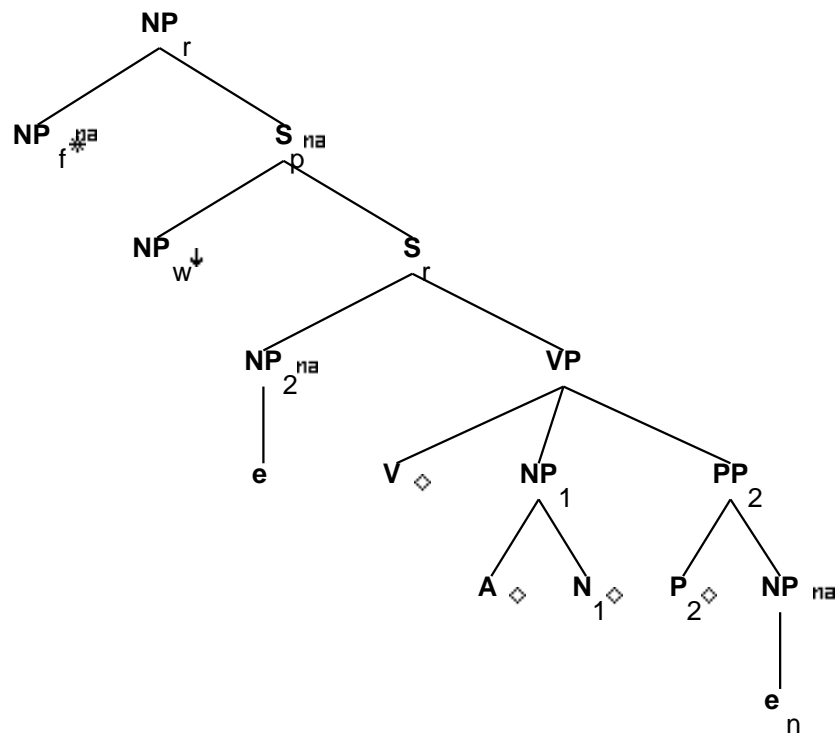
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
S_q.b:<wh> = PP_0.t:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
PP_0.t:<trace> = PP.t:<trace>
S_r.b:<extracted> = -
S_r.b:<mode> = VP.t:<mode>

S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
S_r.b:<control> = NP_2.t:<control>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>

```

12 Tree "betaN2nx2VAN1P"

12.1 graphe



12.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive without by-phrase, relative on the subject.

EX: The chopping which was made short work of...

12.3 features

```

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.b:<agr> = VP.t:<agr>
S_r.t:<mode> = ind/inf
S_r.b:<comp> = nil
S_r.b:<agr> = NP_2.t:<agr>
S_r.b:<assign-case> = NP_2.t:<case>
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:<assign-comp> = VP.t:<assign-comp>
  
```

```

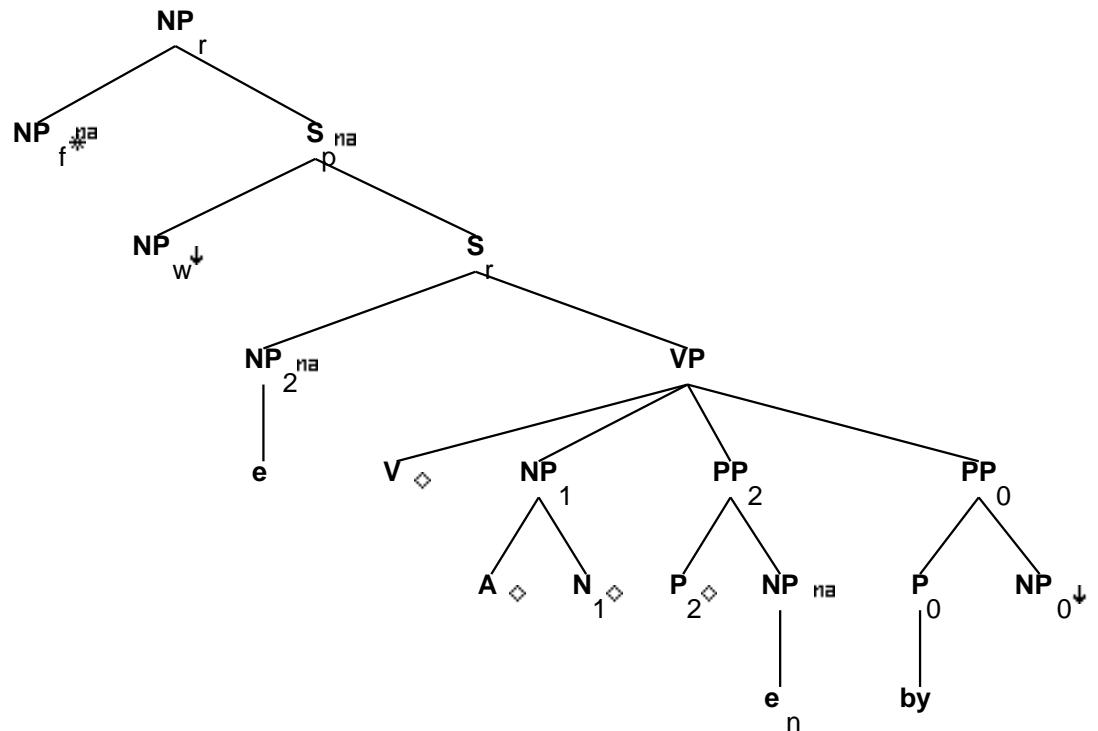
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
V.t:<punct struct> = nil
NP_f.b:<refl> = -
S_r.t:<conj> = nil

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_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
S_r.b:<control> = NP_1.t:<control>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>
NP_2:<agr> = NP.t:<agr>
NP_2:<trace> = NP.t:<trace>
NP.t:<case> = NP_2:<case>
NP.t:<wh> = NP_2:<wh>

```

13 Tree "betaN2nx2VAN1Pbyn0"

13.1 graphe



13.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive with by-phrase, relative clause on subject.

EX: Sentencing, which was made short work of by the magistrate...

13.3 features

```

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.b:<agr> = VP.t:<agr>
S_r.t:<mode> = ind/inf
S_r.b:<comp> = nil
S_r.b:<agr> = NP_2.t:<agr>
S_r.b:<assign-case> = NP_2.t:<case>
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:<assign-comp> = VP.t:<assign-comp>
  
```

```

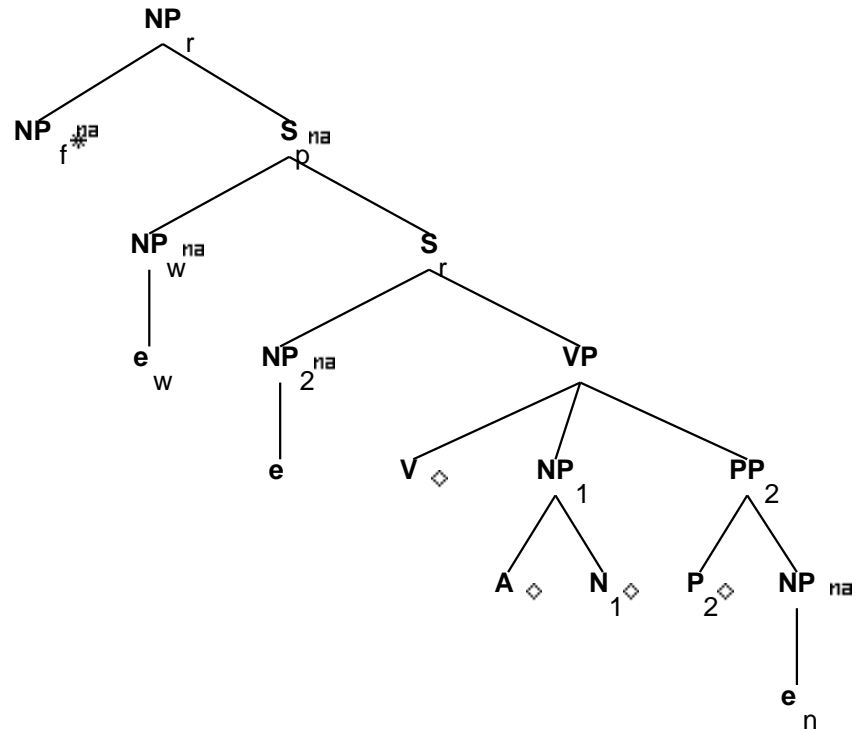
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
V.t:<punct struct> = nil
NP_f.b:<refl> = -
S_r.t:<conj> = nil

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
S_r.b:<control> = NP_2.t:<control>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>
NP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>

```

14 Tree "betaNc2nx2VAN1P"

14.1 graphe



14.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive without by-phrase, relative clause on the subject, with overt Comp.

EX: The chopping that was made short work of...

14.3 features

```

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.b:<agr> = VP.t:<agr>
S_r.b:<comp> = nil
S_r.b:<agr> = NP_2.t:<agr>
S_r.b:<assign-case> = NP_2.t:<case>
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:<assign-comp> = VP.t:<assign-comp>

```

```

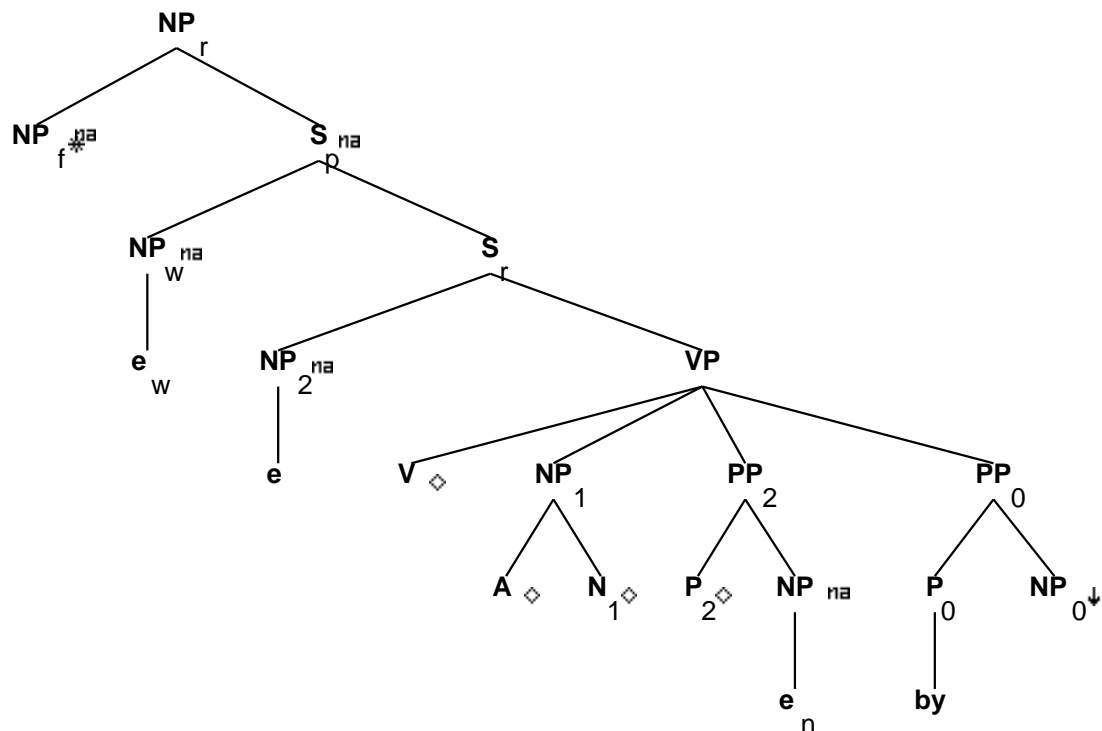
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
V.t:<punct struct> = nil
NP_f.b:<refl> = -
S_r.t:<conj> = nil

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> = +
NP_f.b:<case> = nom/acc
S_r.t:<mode> = inf/ger/ppart/ind
S_r.t:<nocomp-mode> = ind/ger/ppart
VP.t:<assign-comp> = that/inf_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>

```


15 Tree "betaNc2nx2VAN1Pbynx0"

15.1 graphe



15.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive with by-phrase, relative clause on the subject, with overt Comp.

EX: The game that was made short work of by the team...

15.3 features

```

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.b:<agr> = VP.t:<agr>
S_r.b:<comp> = nil
S_r.b:<agr> = NP_2.t:<agr>
S_r.b:<assign-case> = NP_2.t:<case>
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:<assign-comp> = VP.t:<assign-comp>

```

```

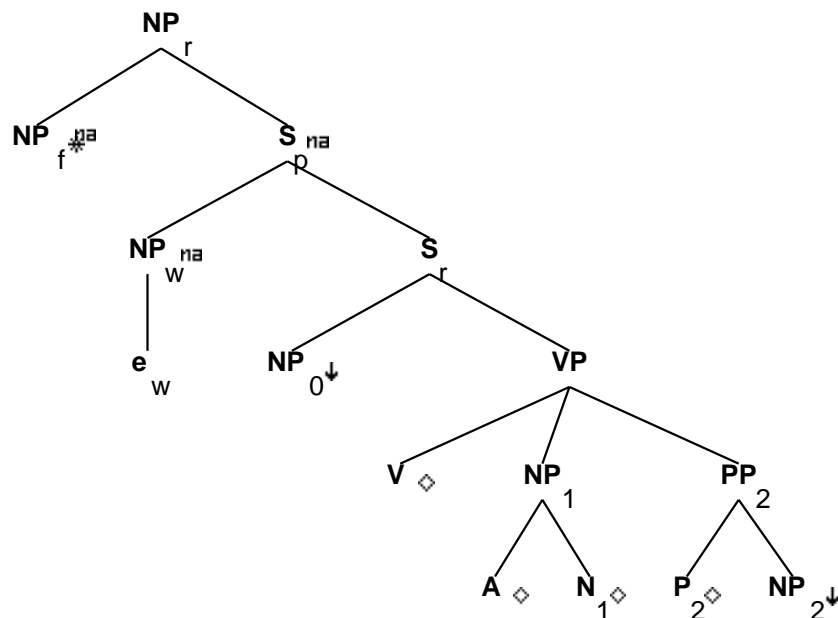
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
V.t:<punct struct> = nil
NP_f.b:<refl> = -
S_r.t:<conj> = nil

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> = +
NP_f.b:<case> = nom/acc
S_r.t:<mode> = inf/ger/ppart/ind
S_r.t:<nocomp-mode> = ind/ger/ppart
VP.t:<assign-comp> = that/inf_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
S_r.b:<control> = NP_2.t:<control>
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>

```

16 Tree "betaNcnx0VAN1Pnx2"

16.1 graphe



16.2 comments

Transitive idiom with V, A, N, and Prep anchors.
Adjunct relative clause, with overt Comp.

EX: [The time] that I made short work of it...

16.3 features

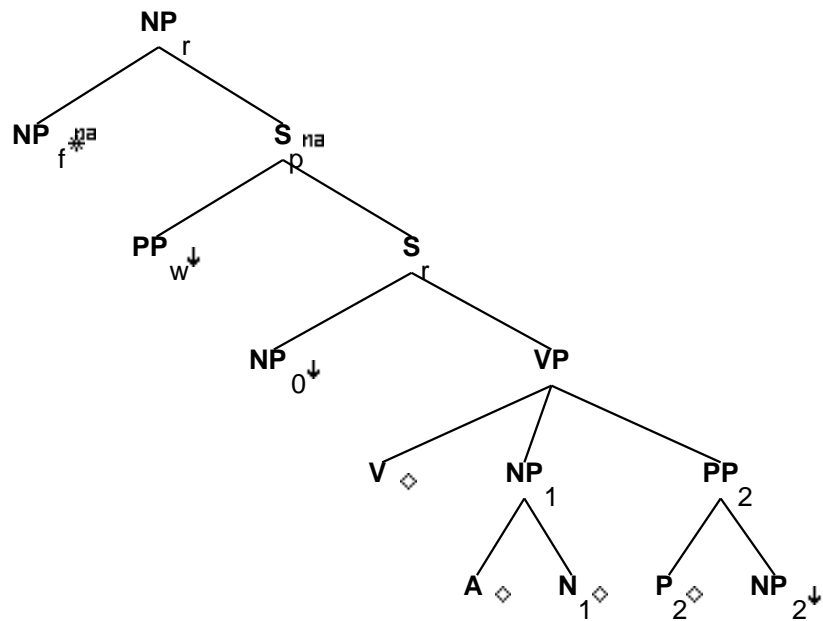
```

S_r.b:<comp> = nil
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:<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>
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<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>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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:<compar> = -
 S_r.b:<control> = NP_0:<control>
 S_r.b:<extracted> = -
 NP_0:<wh> = -
 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_1.b:<agr> = N_1.t:<agr>
 N_1.t:<case> = nom/acc
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2:<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<case> = PP_2.b:<assign-case>

17 Tree "betaNpnx0VAN1Pnx2"

17.1 graphe



17.2 comments

Transitive idiom with V, A, N, and Prep anchors.

Adjunct relative clause with PP.

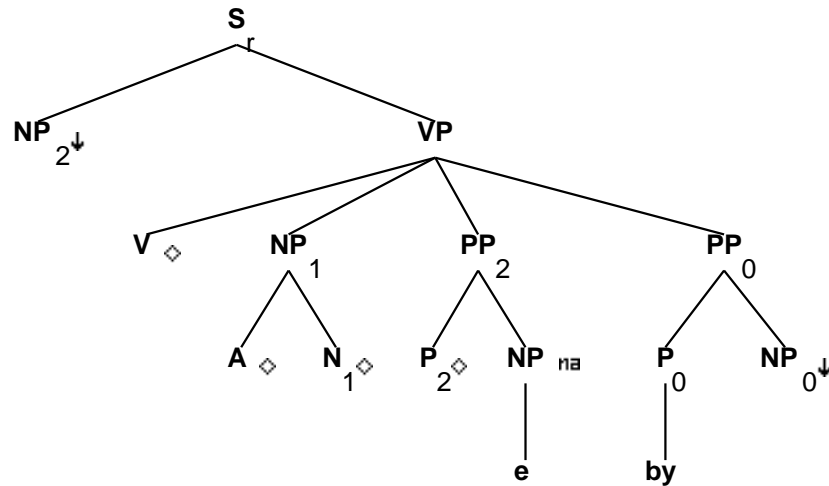
EX: [The place] where he made short work of it...

17.3 features

```
S_r.b:<comp> = nil
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:<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>
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<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>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<contr> = -
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:<compar> = -
S_r.t:<comp> = nil
S_r.b:<control> = NP_0:<control>
S_r.b:<extracted> = -
PP_w:<wh> = +
NP_0:<wh> = -
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
NP_f.b:<case> = nom/acc
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>
```

18 Tree "alphanx2VAN1Pbyn0"

18.1 graphe



18.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive with by-phrase.

EX: The ham was made short work of by Gerald.

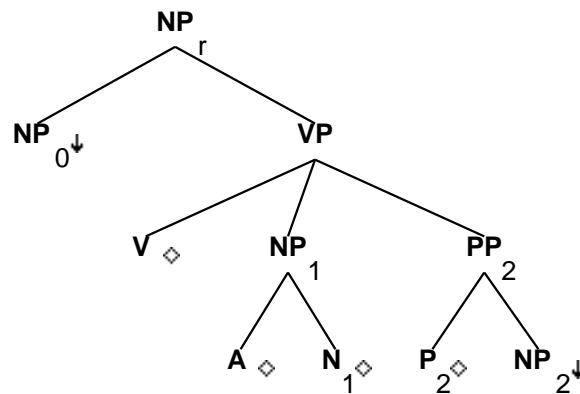
18.3 features

```
S_r.b:<extracted> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
```

V.t:<mode> = ppart
 V.t:<passive> = +
 S_r.b:<inv> = -
 S_r.b:<control> = NP_2.t:<control>
 NP_1.b:<agr> = N_1.t:<agr>
 N_1.t:<case> = nom/acc
 P_2.b:<assign-case> = acc
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<trace> = NP.t:<trace>
 PP_0.b:<assign-case> = P_0.t:<assign-case>
 PP_0.b:<assign-case> = NP_0.t:<case>
 P_0.b:<assign-case> = acc
 PP_0.b:<wh> = NP_0:<wh>

19 Tree "alphaGnx0VAN1Pnx2"

19.1 graphe



19.2 comments

Idiom with V, A, N, and Prep anchors.
 NP gerund.

[Graham('s) making short work of it] is the last thing we expected.

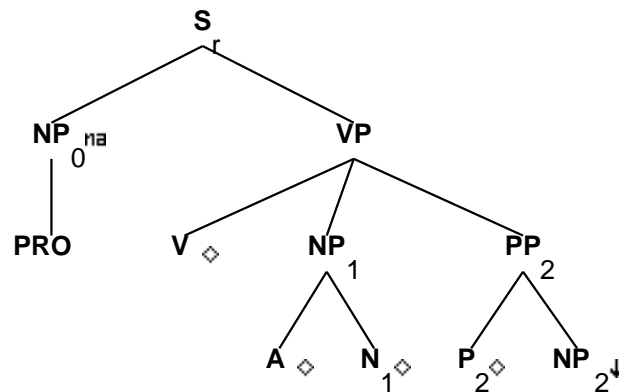
19.3 features

NP_0:<wh> = NP_r.b:<wh>
 NP_r.t:<case> = nom/acc
 NP_r.t:<agr num> = sing
 NP_r.t:<agr pers> = 3
 NP_r.t:<agr 3rdsing> = +
 NP_1:<case> = acc

VP.b:<mode> = none
 VP.b:<compar> = -
 NP_r.b:<gerund> = +
 V:<mode> = ger
 NP_1.b:<agr> = N_1.t:<agr>
 N_1:<case> = nom/acc
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2:<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<case> = PP_2.b:<assign-case>
 NP_0:<case> = acc/gen

20 Tree "alphax0VAN1Pnx2-PRO"

20.1 graphe



20.2 comments

Idiom with V, A, N, and Prep anchors.
w/ PRO subject

Ed wanted [PRO to make short work of the glazed ham].

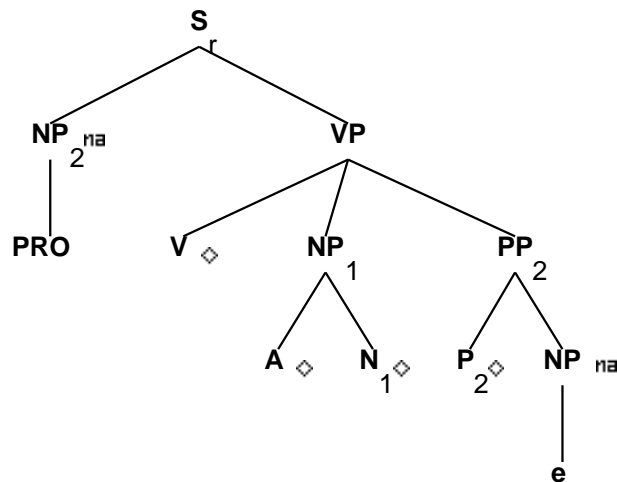
20.3 features

S_r.b:<extracted> = -
 S_r.b:<mode> = VP.t:<mode>
 S_r.b:<comp> = nil
 S_r.b:<inv> = -
 S_r.b:<tense> = VP.t:<tense>
 S_r.b:<assign-case> = NP_0.t:<case>
 S_r.b:<control> = NP_0.t:<control>
 NP_0:<agr> = S_r.b:<agr>
 NP_0:<wh> = -
 NP_0.t:<case> = none
 S_r.b:<wh> = NP_0:<wh>

S_r.b:<agr> = VP.t:<agr>
 S_r.b:<assign-comp> = VP.t:<assign-comp>
 VP.t:<mode> = inf/ger
 VP.b:<passive> = V.t:<passive>
 V.t:<passive> = -
 V.t:<contr> = -
 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> = -
 NP_1:<case> = acc
 N_1:<case> = nom/acc
 NP_1.b:<agr> = N_1.t:<agr>
 P_2.b:<assign-case> = acc
 PP_2.b:<wh> = NP_2:<wh>
 P_2.t:<assign-case> = PP_2.b:<assign-case>
 NP_2:<case> = PP_2.b:<assign-case>

21 Tree "alphanx2VAN1P-PRO"

21.1 graphe



21.2 comments

Idiom with V, A, N, and Prep anchors.
 Outer passive without by-phrase.
 w/ PRO subject

John didn't want [PRO to be made short work of].

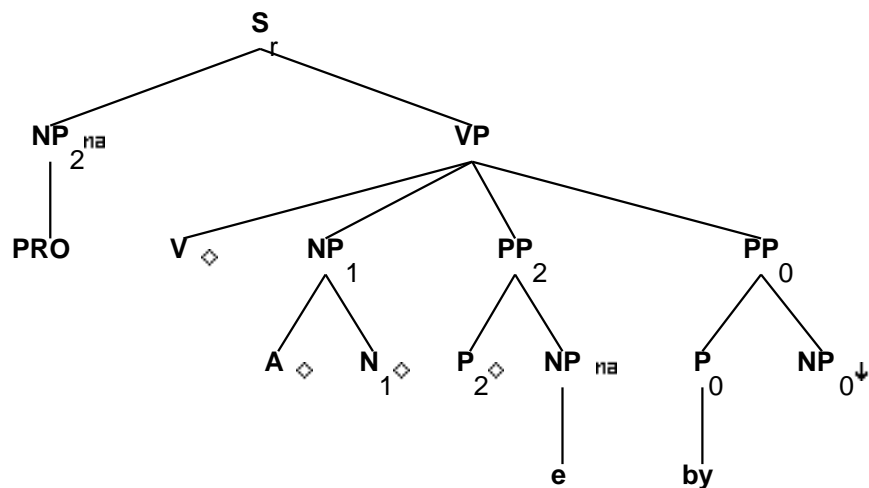
21.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
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_2.t:<case>
S_r.b:<wh> = NP_2:<wh>
S_r.b:<control> = NP_2.t:<control>
NP_2:<agr> = S_r.b:<agr>
NP_2:<wh> = -
NP_2.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.t:<mode> = inf/ger
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +

NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>
```

22 Tree "alphanx2VAN1Pbynx0-PRO"

22.1 graphe



22.2 comments

Idiom with V, A, N, and Prep anchors.

Outer passive with by-phrase.

w/ PRO subject

John didn't want [PRO to be made short work of by Gerald].

22.3 features

```

S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<control> = NP_2.t:<control>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_2.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.t:<mode> = inf/ger
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>

```

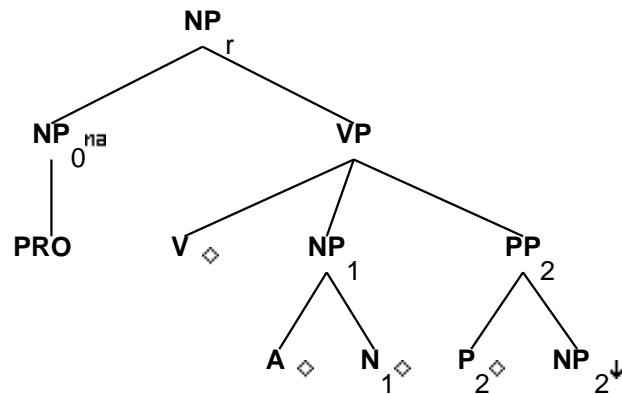
```

VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
NP_1.b:<agr> = N_1.t:<agr>
N_1.t:<case> = nom/acc
P_2.b:<assign-case> = acc
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<trace> = NP.t:<trace>
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>

```

23 Tree "alphaGnx0VAN1Pnx2-PRO"

23.1 graphe



23.2 comments

Idiom with V, A, N, and Prep anchors.
NP gerund w/ PRO subject.

[PRO making short work of it] is the last thing we expected of Graham.

23.3 features

```

NP_0:<wh> = NP_r.b:<wh>
NP_r.t:<case> = nom/acc
NP_r.t:<agr num> = sing
NP_r.t:<agr pers> = 3
NP_r.t:<agr 3rdsing> = +
NP_1:<case> = acc
NP_0.t:<case> = none
NP_0.t:<wh> = -

```

VP.b:<mode> = none
VP.b:<compar> = -
NP_r.b:<gerund> = +
V:<mode> = ger
NP_1.b:<agr> = N_1.t:<agr>
N_1:<case> = nom/acc
P_2.b:<assign-case> = acc
PP_2.b:<wh> = NP_2:<wh>
P_2.t:<assign-case> = PP_2.b:<assign-case>
NP_2:<case> = PP_2.b:<assign-case>