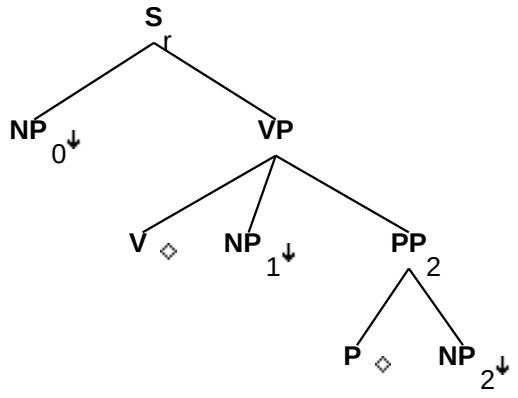
Family "TRnx0Vnx1Pnx2"

March 5, 2008

1 Tree "alphaRnx0Vnx1Pnx2"

1.1 graphe



1.2 comments

```
Resultative with prepositional predicate.

'John pounded the dough into a pancake'
'Mary ran her shoes into pieces'
```

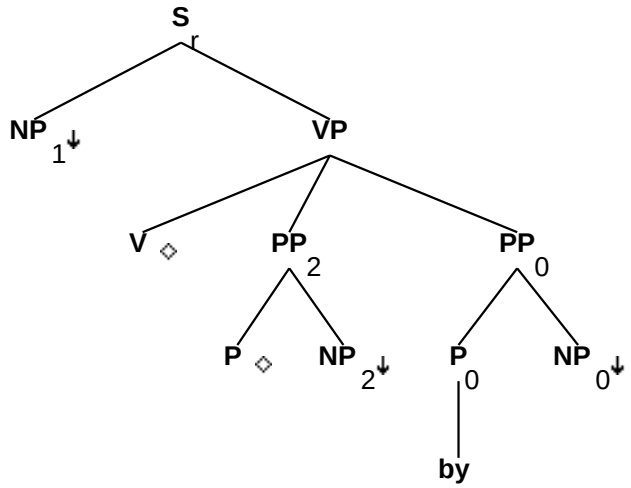
1.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<wh> = NP_0.t:<wh>
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:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
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>

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

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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

2 Tree "alphaRnx1VPnx2bynx0"

2.1 graphe



2.2 comments

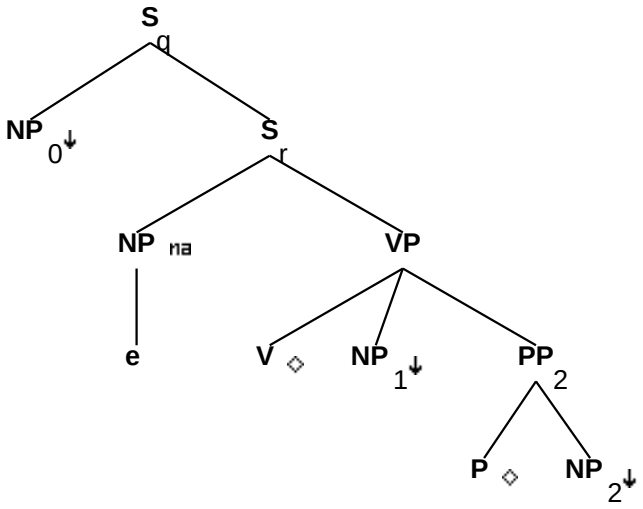
```
Passive on a prepositional resultative:

'The dough was pounded into a pancake by Max'
'The shoes were run into pieces by Mary'
```

2.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<wh> = NP_1.t:<wh>
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:<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>
NP_1.t:<wh> = -
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>

VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

3 Tree "alphaRW0nx0Vnx1Pnx2"

3.1 graphe



3.2 comments

```
Wh on the subject of a prepositional resultative:

 'Who pounded the dough into a pancake'
 'Who ran the shoes into pieces'

check the agr equation on  NP0
```

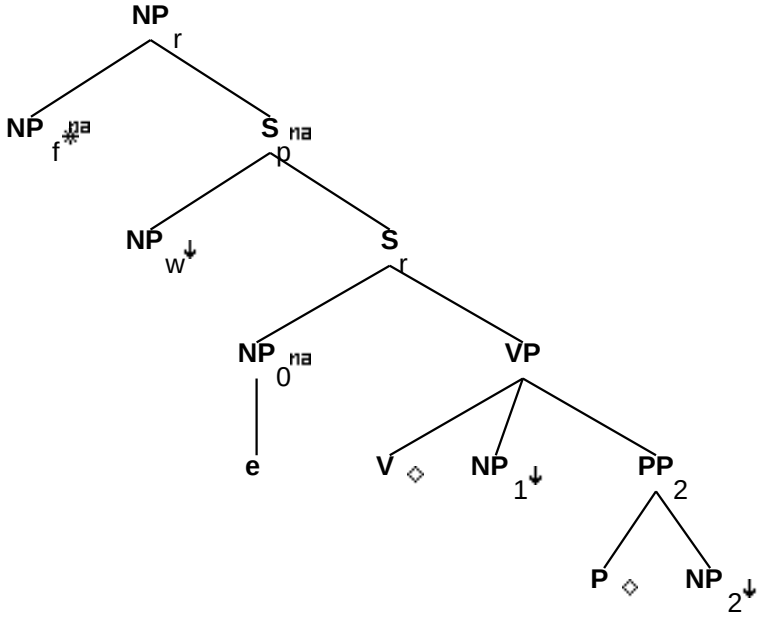
3.3 features

```
S_q.b:<extracted> = +
S_q.b:<comp> = nil
S_q.b:<wh> = NP_0.t:<wh>
S_q.b:<mode> = S_r.t:<mode>

S_q.b:<inv> = S_r.t:<inv>
NP_0.t:<wh> = +
S_r.t:<comp> = nil

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

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>

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.t:<agr> = S_r.b:<agr>
NP.t:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc

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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

4 Tree "betaRN0nx0Vnx1Pnx2"

4.1 graphe



4.2 comments

```
(wh) Relative clause on the subject of a prepositional resultative with a relative pronoun

  '(I saw) the man who pounded the dough into a pancake'
  '(I saw) the people who ran their shoes into pieces'
```

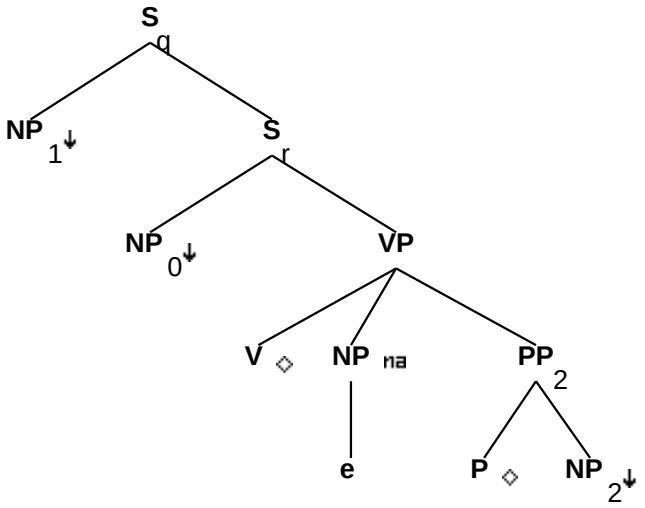
4.3 features

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

NP_f.b:<case> = nom/acc
NP_f.b:<refl> = -
NP_w.t:<wh> = +
NP_w.t:<trace> = NP_0.t:<trace>
NP_w.t:<case> = NP_0.t:<case>
NP_w.t:<agr> = NP_0.t:<agr>
S_r.t:<conj> = nil
S_r.t:<comp> = nil

S_r.t:<mode> = ind
S_r.t:<inv> = -

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

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

VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
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>
V.t:<passive> = -
NP_1:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
```

PP.b:<wh> = NP.t:<wh>

5 Tree "alphaRW1nx0Vnx1Pnx2"

5.1 graphe



5.2 comments

Wh question on the object of a prepositional resultative:

'What did Mary pound into a pancake'
'What did the runners run into pieces'

5.3 features

S_q.b:<extracted> = +
S_q.b:<comp> = nil
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = NP_1.t:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>

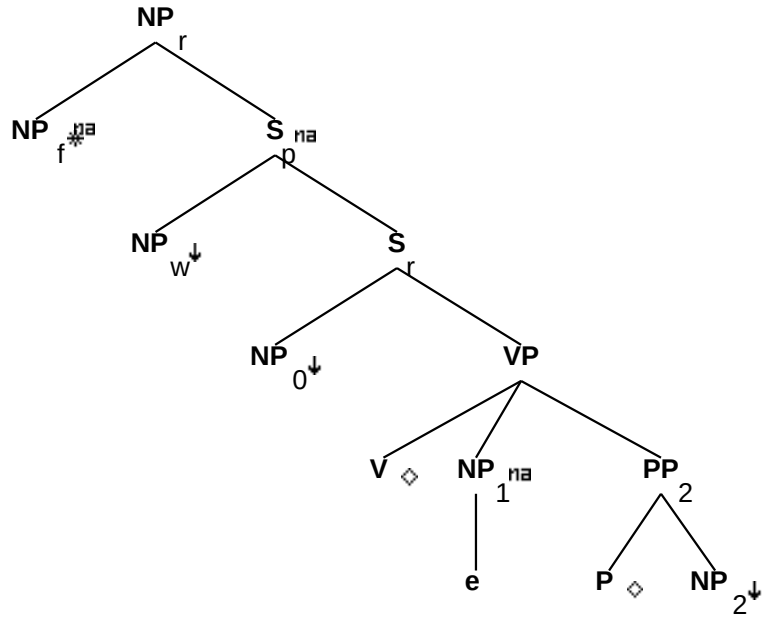
S_r.t:<comp> = nil
S_r.t:<conj> = nil
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<control> = NP_0.t:<control>

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

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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
V.t:<punct struct> = nil
NP.t:<case> = acc
NP.t:<trace> = NP_1.t:<trace>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<wh> = NP_1.t:<wh>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

6 Tree "betaRN1nx0Vnx1Pnx2"

6.1 graphe



6.2 comments

```
(wh) Relative clause on the object of a prepositional resultative.

 'The dough which Max pounded into a pancake'
 'The shoes which Tobi ran into pieces'
```

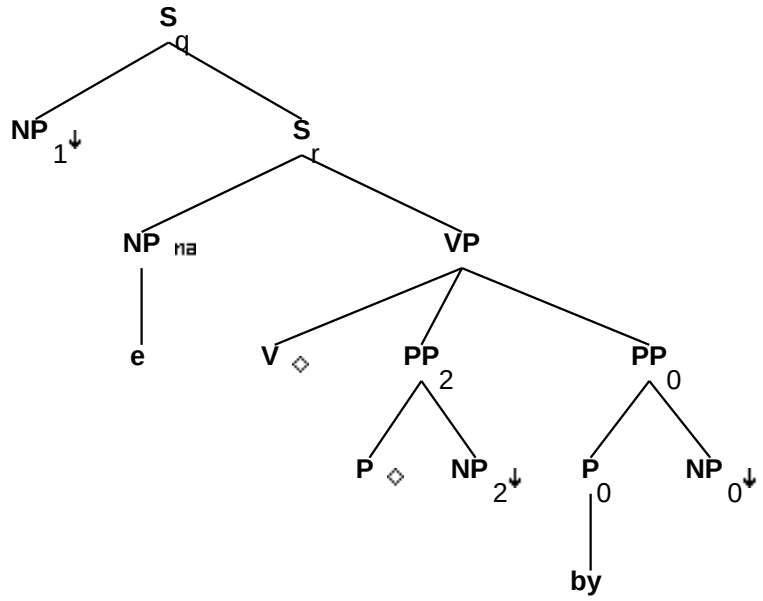
6.3 features

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

NP_f.b:<case> = nom/acc
NP_f.b:<refl> = -
NP_w.t:<wh> = +
NP_w.t:<trace> = NP_1.t:<trace>
NP_w.t:<case> = NP_1.t:<case>
NP_w.t:<agr> = NP_1.t:<agr>

S_r.t:<mode> = ind
S_r.t:<conj> = nil
S_r.t:<comp> = nil
```

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

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

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

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:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
V.t:<punct struct> = nil
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

7 Tree "alphaRW1nx1VPnx2bynx0"

7.1 graphe



7.2 comments

```
Wh question on NP1 in passivized prepositional resultative construction with by-phrase.

'What was pounded into pancake by Max'
'What was run into pieces by Mary'
```

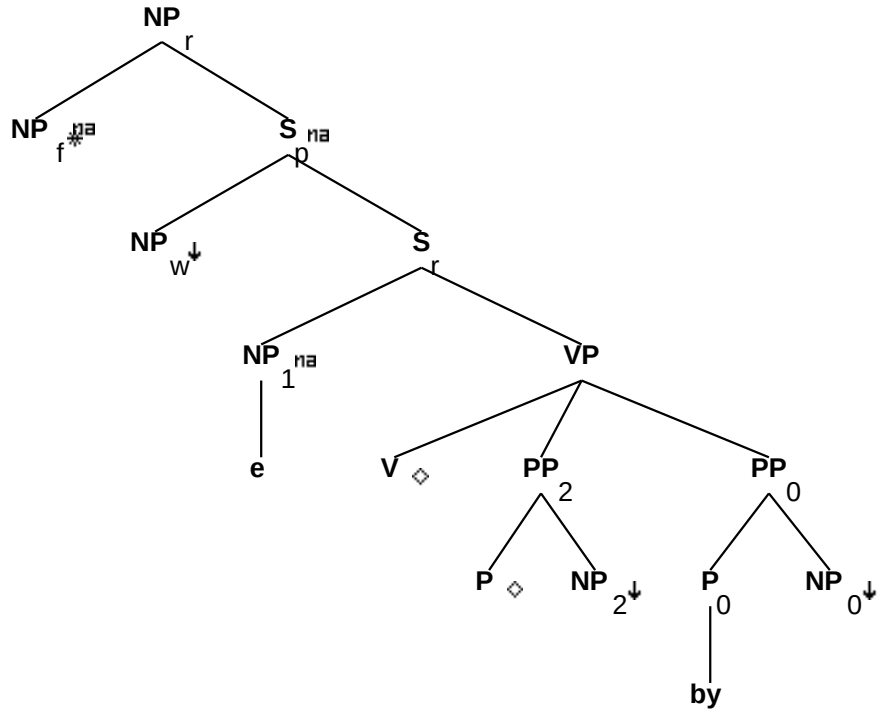
7.3 features

```
S_q.b:<extracted> = +
S_q.b:<comp> = nil
S_q.b:<wh> = NP_1.t:<wh>

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>
NP_1.t:<wh> = +
S_r.t:<comp> = nil

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

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

NP.t:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<wh> = NP_1.t:<wh>

VP.b:<passive> = +
VP.b:<compar> = -
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:<passive> = V.t:<passive>
V.t:<mode> = ppart
V.t:<passive> = +
V.t:<punct struct> = nil
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

8 Tree "betaRN1nx1VPnx2bynx0"

8.1 graphe



8.2 comments

```
(wh) Relative Clause on subject of prepositional resultative passive with by-phrase.

'the dough which was pounded into a pancake by the cook'
```

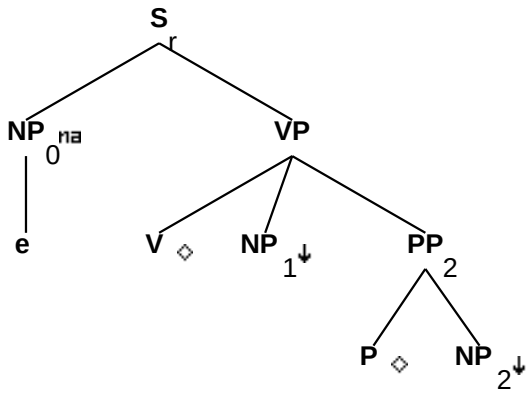
8.3 features

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

NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = nom/acc
NP_f.b:<refl> = -
NP_w.t:<wh> = +
NP_w.t:<case> = NP_1.t:<case>
NP_w.t:<agr> = NP_1.t:<agr>
NP_w.t:<trace> = NP_1.t:<trace>
```

```
S_r.t:<inv> = -
S_r.t:<mode> = ind
S_r.t:<conj> = nil
S_r.t:<comp> = nil

S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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:<compar> = -
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
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>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
V.t:<punct struct> = nil
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

9 Tree "alphaRInx0Vnx1Pnx2"

9.1 graphe



9.2 comments

```
Imperative on prepositional resultative:
 'Pound the dough into a pancake'
 'Run your shoes into pieces'
```

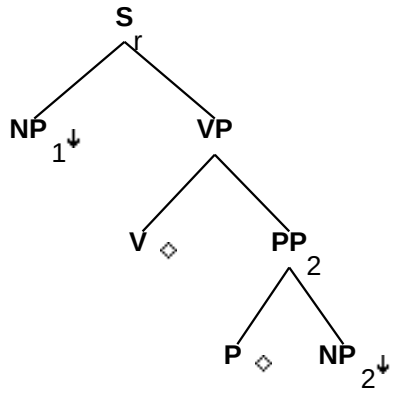
9.3 features

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

S_r.b:<mode> = imp
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:<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>
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>
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> = -
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>

10 Tree "alphaRnx1VPnx2"

10.1 graphe



10.2 comments

Passive on prepostional resultative w/out by-phrase:
'The dough was pounded into a pancake'
'The shoes were broken into pieces'

10.3 features

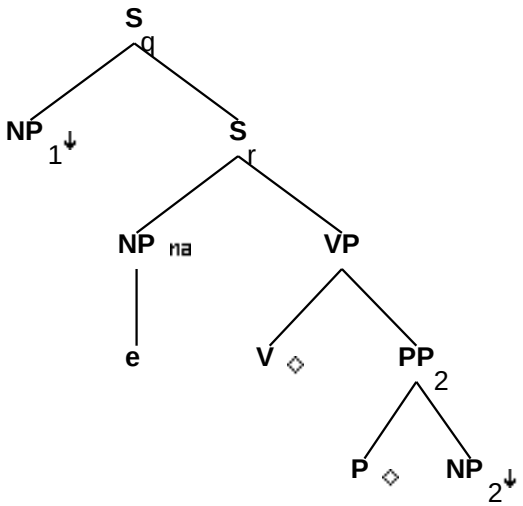
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.b:<wh> = NP_1.t:<wh>
S_r.b:<mode> = VP.t:<mode>

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:<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>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
NP_1.t:<wh> = -

VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

11 Tree "alphaRW1nx1VPnx2"

11.1 graphe



11.2 comments

Wh question on NP1 in passive prepositional resultative constructions, w/o by-phrase:

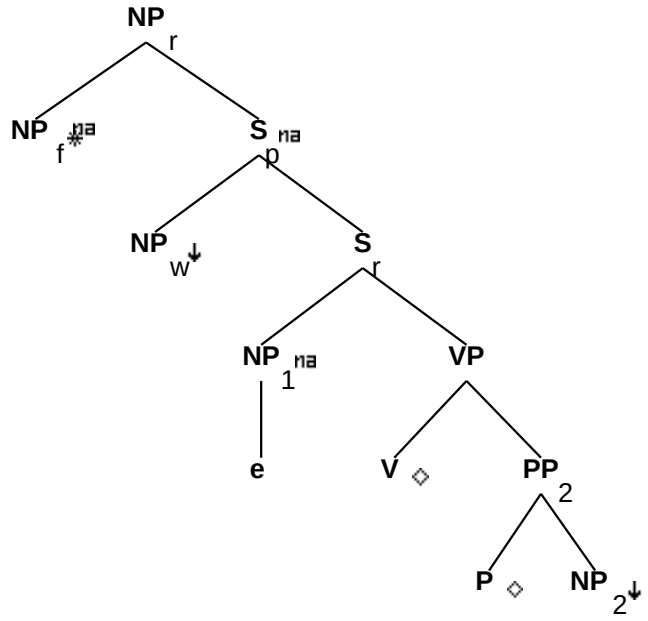
```
'What was pounded into a pancake'
'What was run into pieces'
```

11.3 features

```
S_q.b:<extracted> = +
S_q.b:<comp> = nil
S_q.b:<wh> = NP_1.t:<wh>

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>
NP_1.t:<wh> = +
S_r.t:<comp> = nil

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

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

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>
NP.t:<agr> = S_r.b:<agr>
NP.t:<case> = S_r.b:<assign-case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<wh> = NP_1.t:<wh>

VP.b:<passive> = +
VP.b:<compar> = -
VP.b:<passive> = V.t:<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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

12 Tree "betaRN1nx1VPnx2"

12.1 graphe



12.2 comments

```
(wh) relative clause on the subject in passivized prepositional resultative, w/o by-phrase

'(I saw) the dough which was pounded into a pancake'
'(I saw) the shoes which were run into pieces'
```

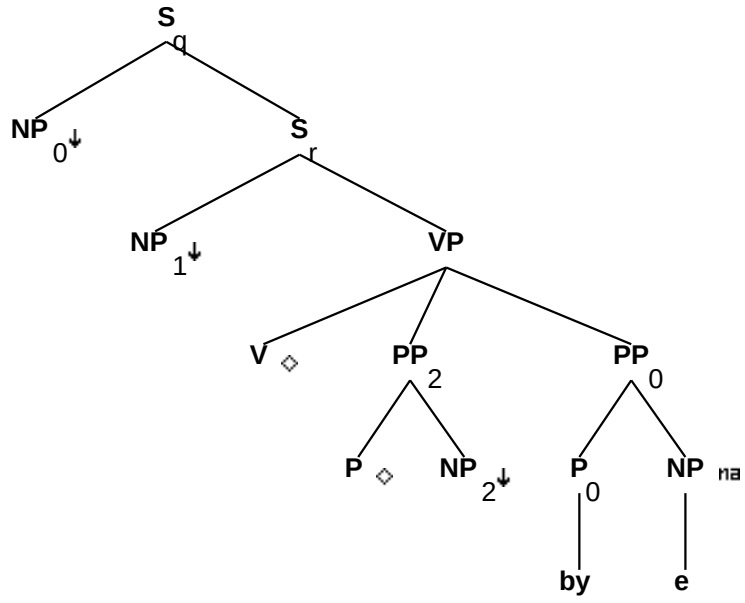
12.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind
S_r.t:<inv> = -
S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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_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>
NP_r.b:<compar> = NP_f.t:<compar>

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

13 Tree "alphaRW0nx1VPnx2bynx0"

13.1 graphe



13.2 comments

Wh question, extraction from by-phrase of nx0 in passivized prepositional resultative constructions:

'Who was the dough pounded into a pancake by'

13.3 features

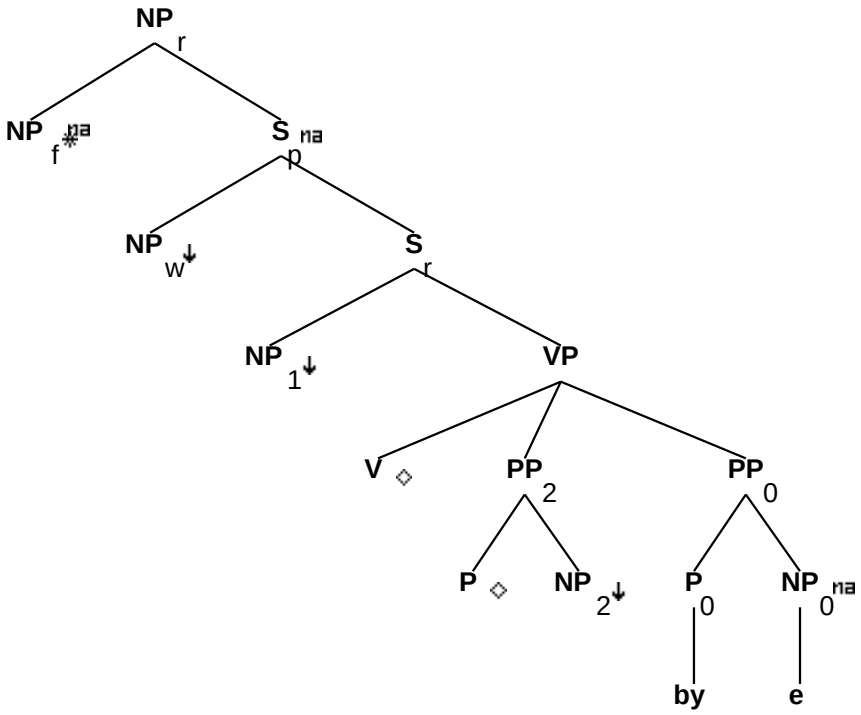
S_r.t:<comp> = nil
S_q.b:<extracted> = +

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>
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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>

14 Tree "betaRN0nx1VPnx2bynx0"

14.1 graphe



14.2 comments

(wh) relative clause on demoted argument of passivized prepositional resultative.

'(Mary saw) the cook which the dough was pounded into a pancake by'

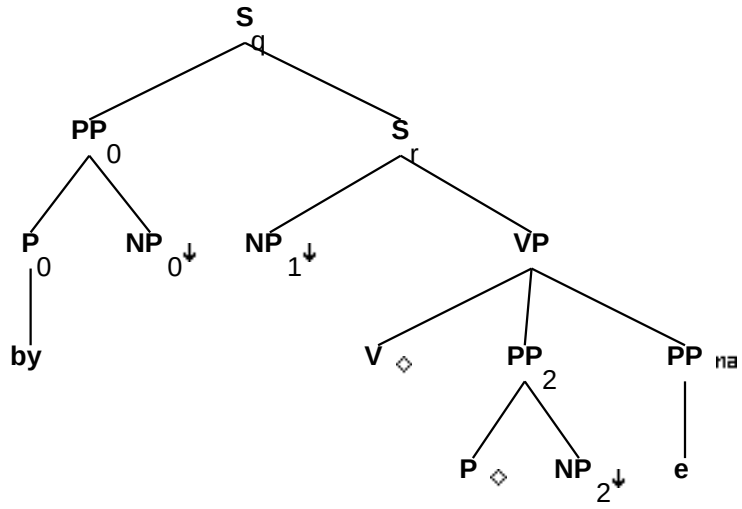
14.3 features

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind

```
S_r.t:<inv> = -
S_r.b:<comp> = nil
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>
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.t:<mode> = ind
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:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
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.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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

15 Tree "alphaRpW0nx1VPnx2bynx0"

15.1 graphe



15.2 comments

```
Wh question on NP0 in passivized prepositional resultative  constructions, by-phrase extra

'By whom was the dough pounded into a pancake'
```

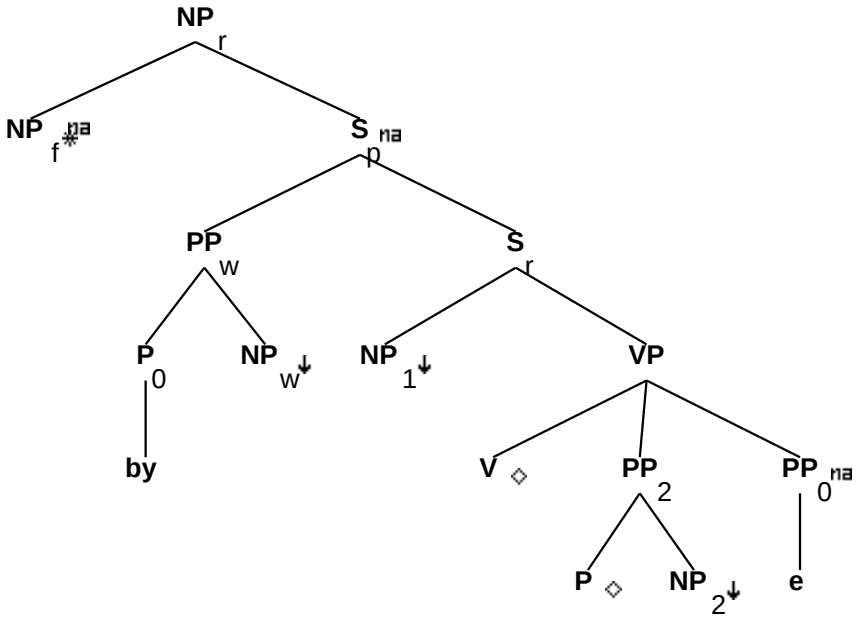
15.3 features

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

16 Tree "betaRNbynx0nx1VPnx2bynx0"

16.1 graphe



16.2 comments

PP Pied piping on relative clause in passivized prepositional resultative.

'(Mary saw) the man by whom the dough was pounded into a pancake'

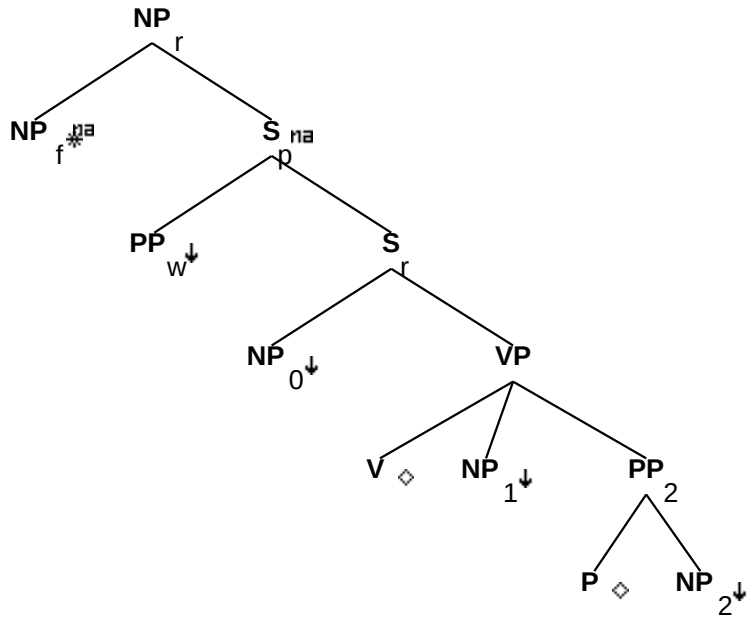
16.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind
S_r.b:<comp> = nil
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>
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:<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>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
P_0.b:<assign-case> = acc
S_r.t:<conj> = nil

NP_w.t:<wh> = +
S_r.t:<comp> = nil
PP_w.t:<trace> = PP_0.b:<trace>
PP_w.t:<case> = PP_0.b:<case>
PP_w.t:<agr> = PP_0.b:<agr>
PP_w.b:<assign-case> = P_0.t:<assign-case>
PP_w.b:<assign-case> = NP_w.t:<case>
PP_w.b:<wh> = NP_w.t:<wh>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>
```

PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>

17 Tree "betaRNpxnx0Vnx1Pnx2"

17.1 graphe



17.2 comments

Relative clause on an adjunct PP (pied-piped) in prepositional resultative.

'The day on which Max pounded the dough into a pancake'

17.3 features

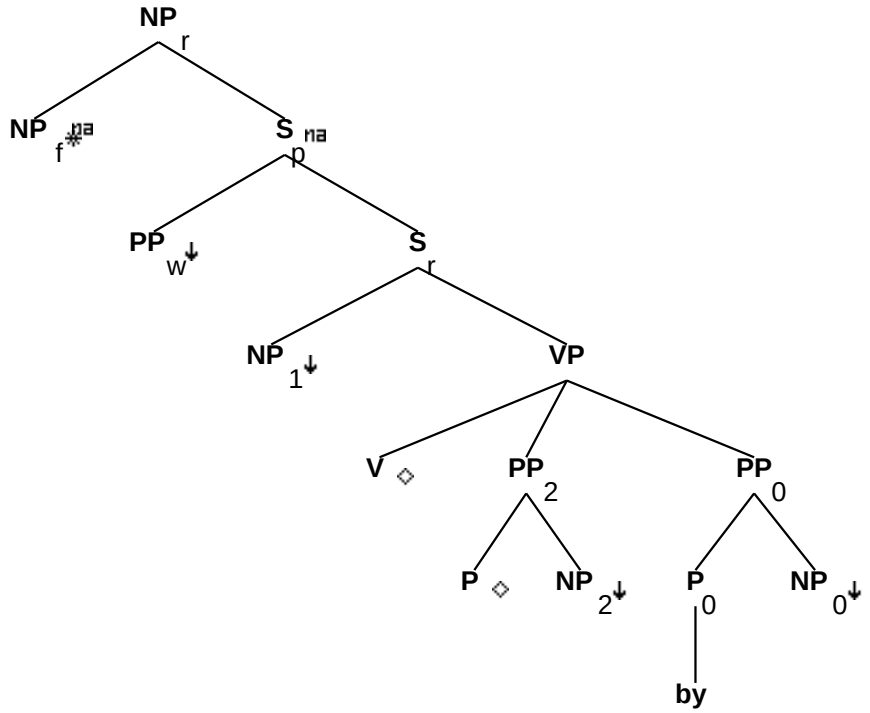
S_r.b:<extracted> = -

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:<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> = -
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> = -
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
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

18 Tree "betaRNpxnx1VOnx2bynx0"

18.1 graphe



18.2 comments

```
Relative clause on (pied-piped) adjunct PP
in passivized (with by-phrase) prepositional resultatives

'The day on which the dough was pounded into a pancake by Max'
'The town in which the shoes were run into pieces by Mary'
```

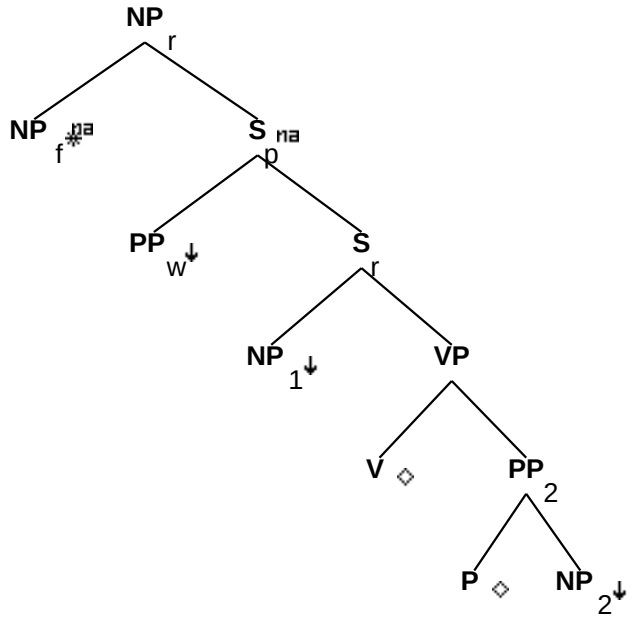
18.3 features

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<tense> = VP.t:<tense>
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> = -
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>
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
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

19 Tree "betaRNpxnx1VPnx2"

19.1 graphe



19.2 comments

```
Relative clause on (pied-piped) adjunct PP
in passivized (w/o by-phrase) prepositional resultatives.

'The day on which the dough was pounded into a pancake'
'The town in which the shoes were run into pieces'
```

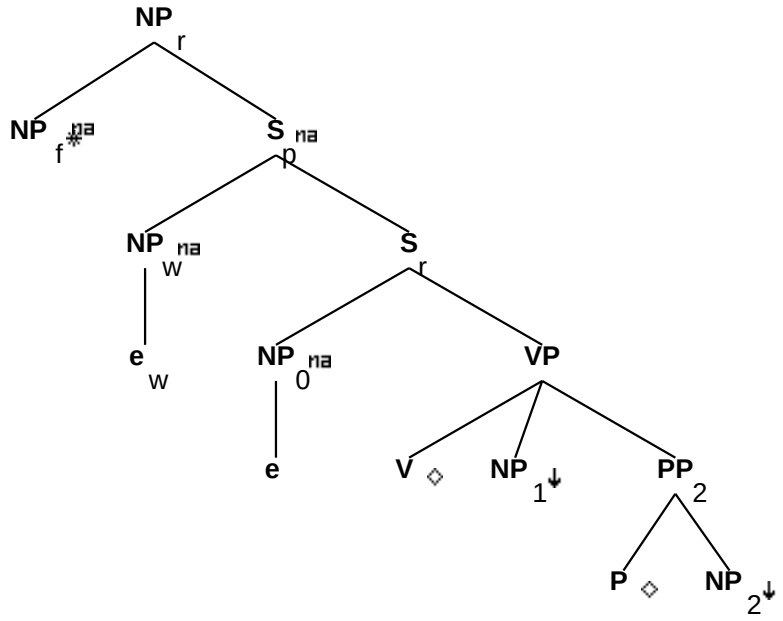
19.3 features

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



S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_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>
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
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

20 Tree "betaRNc0nx0Vnx1Pnx2"

20.1 graphe



20.2 comments

```
(COMP) relative clause on the subject of a prepositional resultative.

 'The guy that pounded the dough into a pancake'
 'the runners that ran their shoes into pieces'
```

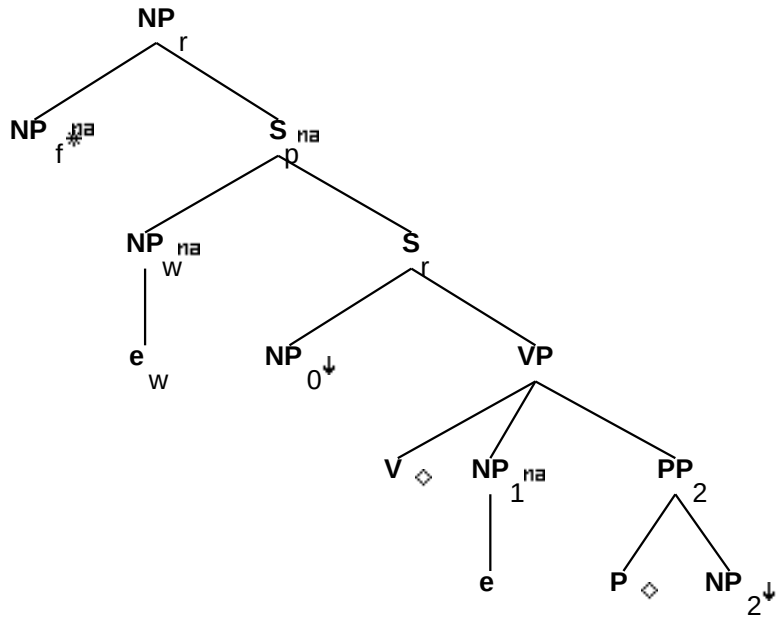
20.3 features

```
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> = -
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
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<refl> = -
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

21 Tree "betaRNc1nx0Vnx1Pnx2"

21.1 graphe



21.2 comments

(COMP) relative clause on the object of a prepositional resultative.

'The dough that Max pounded into a pancake'
'The shoes that the runners ran into pieces'

21.3 features

S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<comp> = nil
S_r.t:<inv> = -
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<case> = NP_f.t:<case>
NP_r.b:<agr> = NP_f.t:<agr>
NP_1.t:<case> = acc
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
V.t:<punct struct> = nil
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:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_f.b:<refl> = -
S_r.t:<conj> = nil

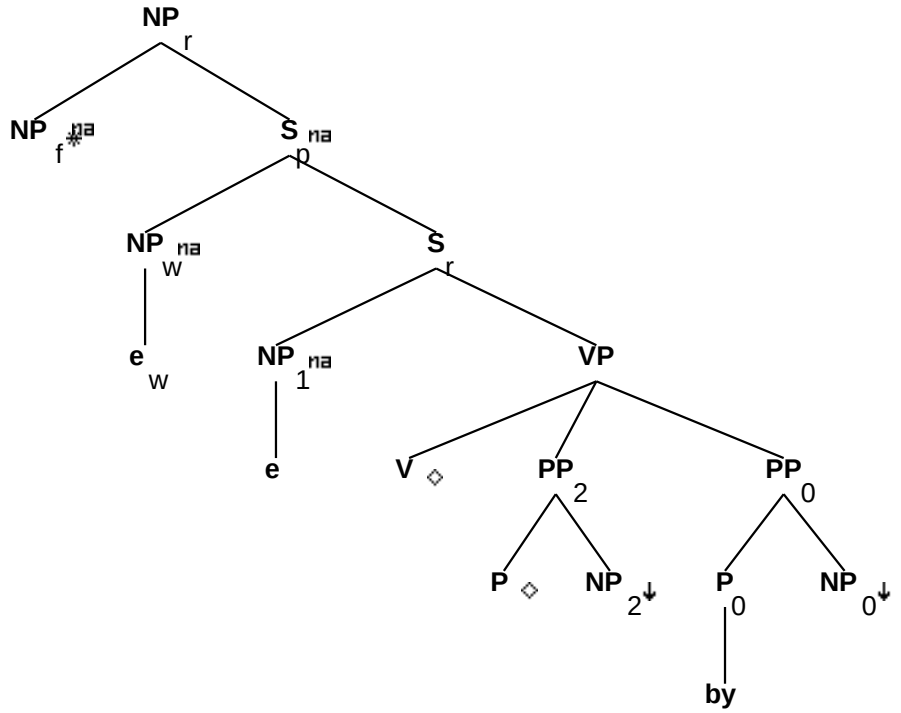
S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_f.b: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

PP.b:<assign-case> = P.t:<assign-case>

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

22 Tree "betaRNc1nx1VPnx2bynx0"

22.1 graphe



22.2 comments

(COMP) relative clause on subject in passivized prepositional relative clause with by-phrase:

'I saw the dough that was pounded into a pancake by Max'

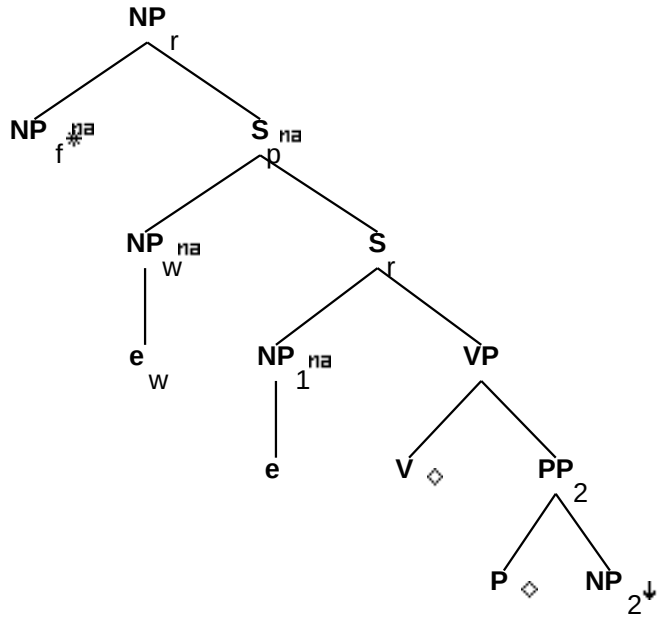
22.3 features

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<inv> = -
S_r.b:<comp> = nil
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>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
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>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
V.t:<punct struct> = nil
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<compar> = -
NP_f.b:<refl> = -
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.t:<conj> = nil

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/ger/ind/ppart
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_f.b:<case> = nom/acc
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

23 Tree "betaRNc1nx1VPnx2"

23.1 graphe



23.2 comments

```
(COMP)relative clause on subject in a
passivized prepositional resultative, w/o by-phrase:

'I saw the dough that was pounded into a pancake'
```

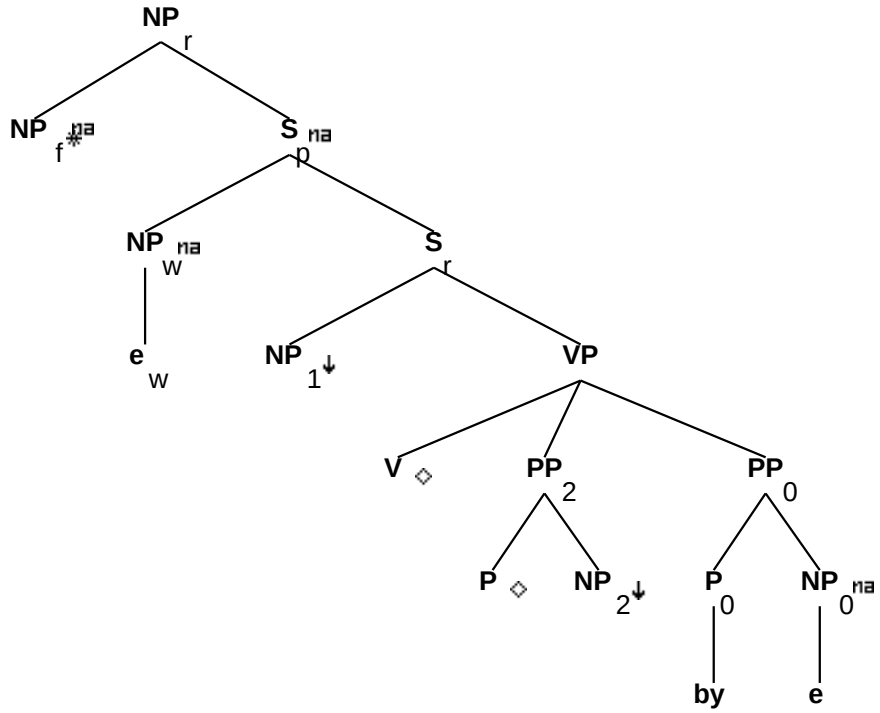
23.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<inv> = -
S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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_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/ppart/ger/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_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

24 Tree "betaRNc0nx1VPnx2bynx0"

24.1 graphe



24.2 comments

(COMP) relative clause on demoted argument from by-phrase of a passivized prepositional resultative:

'(I saw) the man that the dough was pounded into a pancake by'

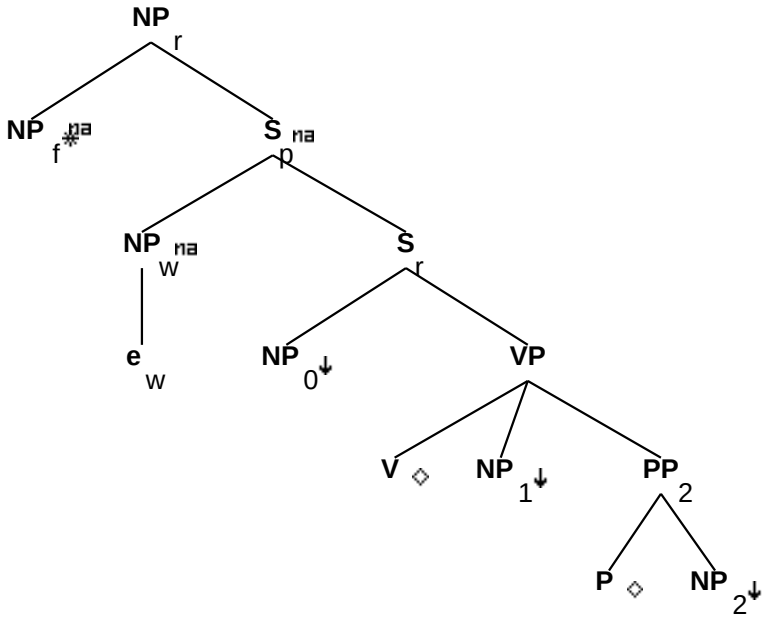
24.3 features

NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<inv> = -
S_r.b:<comp> = nil
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>
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.t:<mode> = ind
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:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
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.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/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_f.b:<refl> = -
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

25 Tree "betaRNcnx0Vnx1Pnx2"

25.1 graphe



25.2 comments

(COMP) relative clause on an adjunct of a prepositional resultative:

'The day that I pounded the dough into a pancake'

25.3 features

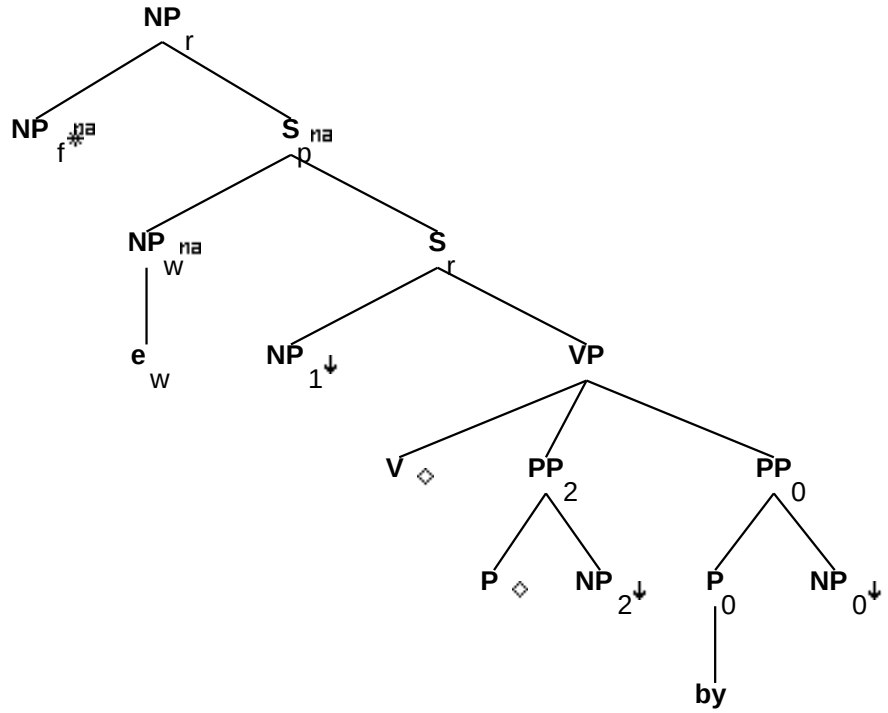
S_r.b:<extracted> = -

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:<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> = -
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> = -
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>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

26 Tree "betaRNcnx1VPnx2bynx0"

26.1 graphe



26.2 comments

```
(COMP) relative clause on an adjunct
of passivized prepositional
resultative w/ by-phrase:

'The day that the dough was pounded into a pancake by Max'
```

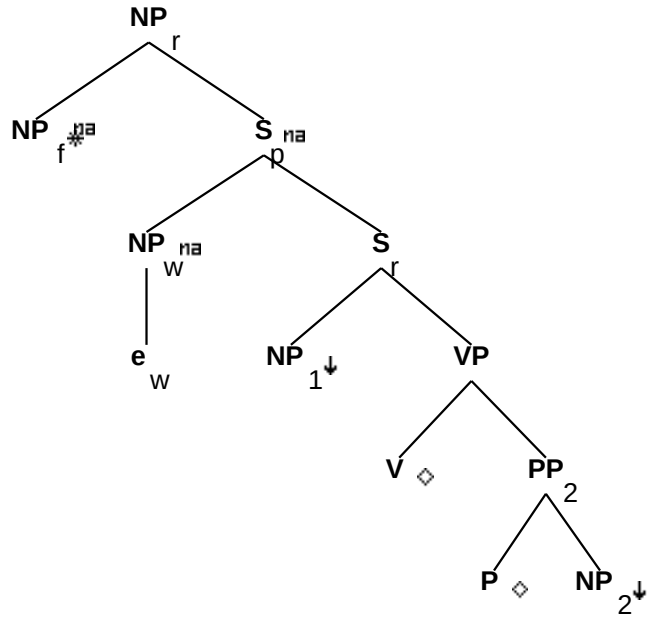
26.3 features

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<tense> = VP.t:<tense>
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:<mode> = ppart
V.t:<passive> = +
V.t:<punct struct> = nil
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>
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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

27 Tree "betaRNcnx1Vpnx2"

27.1 graphe



27.2 comments

```
(COMP) relative clause on an adjunct of a passivized prepositional
resultative w/out  by-phrase:

'The day that the dough was pounded into a pancake'
```

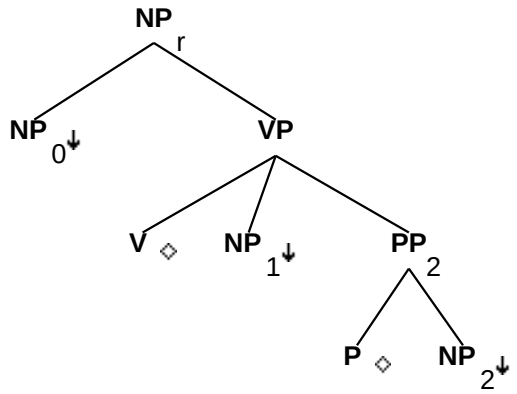
27.3 features

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



S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_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_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>
NP_r.b:<compar> = NP_f.t:<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>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

28 Tree "alphaRGnx0Vnx1Pnx2"

28.1 graphe



28.2 comments

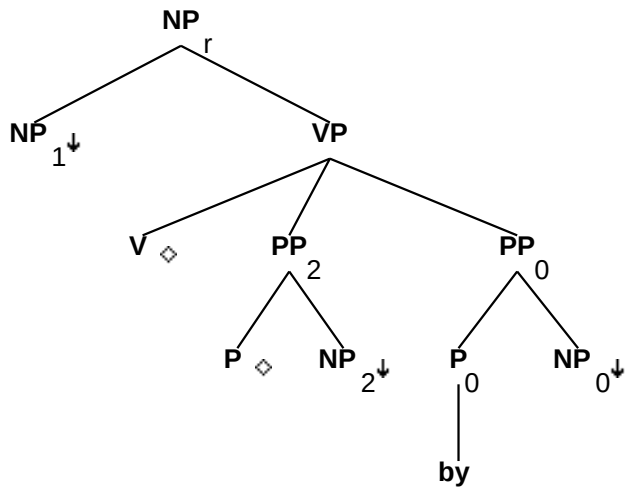
```
Prepositional resultative NP gerund tree:
'Mary approved of 'Peter pounding the dough into a pancake''
'Mary approved of 'Peter's pounding the dough into a pancake''
'Mary approved of 'pounding the dough into a pancake''
```

28.3 features

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


VP.t:<mode> = ger

VP.b:<compar> = -
NP_r.b:<gerund> = +
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

29 Tree "alphaRGnx1VPnx2bynx0"

29.1 graphe



29.2 comments

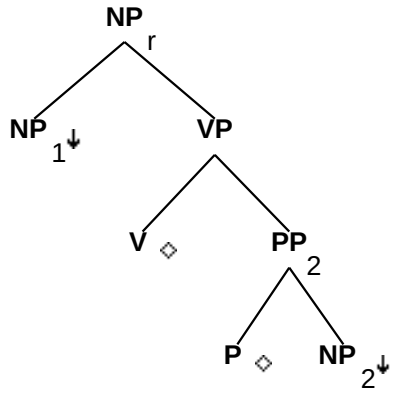
Transitive gerund passive with the 'by' phrase from a perp. rsultative: 'Mary approved of 'The dough being pounded into a pancake by Max''

29.3 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_1:<case> = acc/none/gen
NP_1:<wh> = NP_r.b:<wh>
NP_r.b:<compar> = NP_1:<compar>
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>

NP_r.b:<gerund> = +

PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
VP.t:<mode> = ger
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

30 Tree "alphaRGnx1VPnx2"

30.1 graphe



30.2 comments

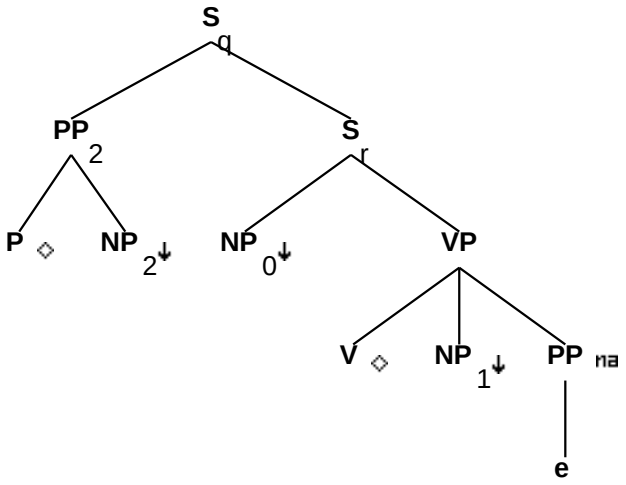
Transitive gerund passive without the 'by' phrase from a prep. resultative: 'John was devastated at 'THe dough being pounded into a pancake''

30.3 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:<wh> = NP_1:<wh>
NP_r.b:<compar> = NP_1:<compar>
NP_1:<case> = acc/none/gen

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

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

31 Tree "alphaRpW2nx0Vnx1Pnx2"

31.1 graphe



31.2 comments

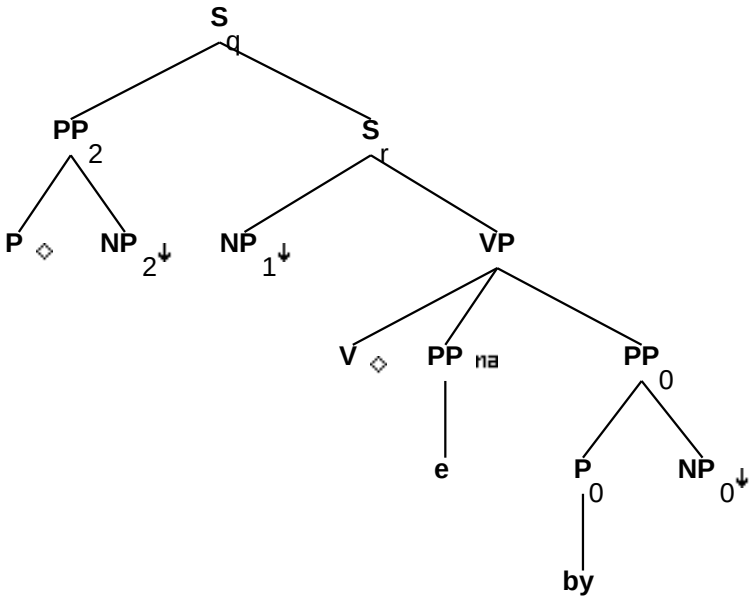
NIL

31.3 features

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

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

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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<wh> = PP.t:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.t:<comp> = nil
S_r.t:<conj> = nil
PP.t:<trace> = PP.t:<trace>
```

32 Tree "alphaRpW2nx1VPnx2bynx0"

32.1 graphe



32.2 comments

NIL

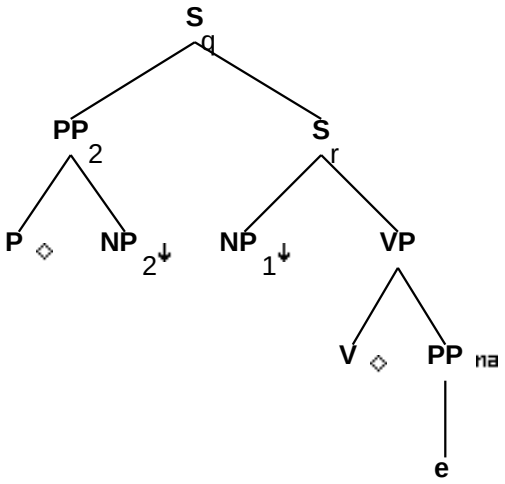
32.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<control> = NP_1.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>

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>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>

VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<wh> = PP.t:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.t:<comp> = nil
S_r.t:<conj> = nil
PP.t:<trace> = PP.t:<trace>
```

33 Tree "alphaRpW2nx1VPnx2"

33.1 graphe



33.2 comments

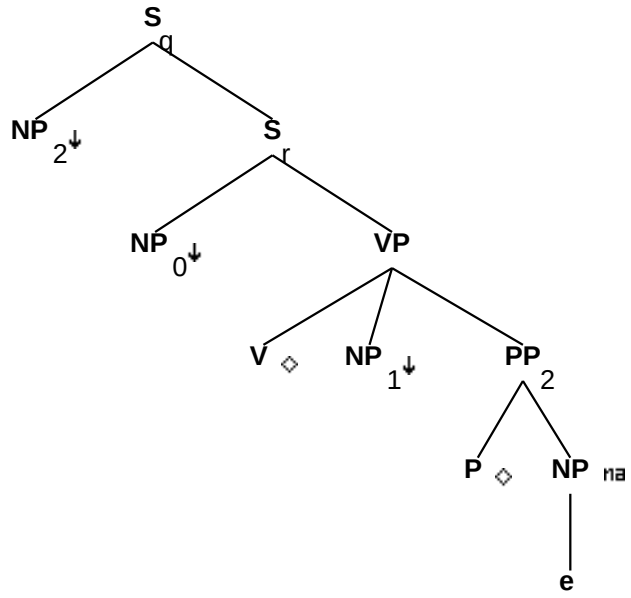
NIL

33.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.b:<mode> = VP.t:<mode>

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:<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>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>

VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<wh> = PP.t:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.t:<comp> = nil
S_r.t:<conj> = nil
PP.t:<trace> = PP.t:<trace>
```

34 Tree "alphaRW2nx0Vnx1Pnx2"

34.1 graphe



34.2 comments

NIL

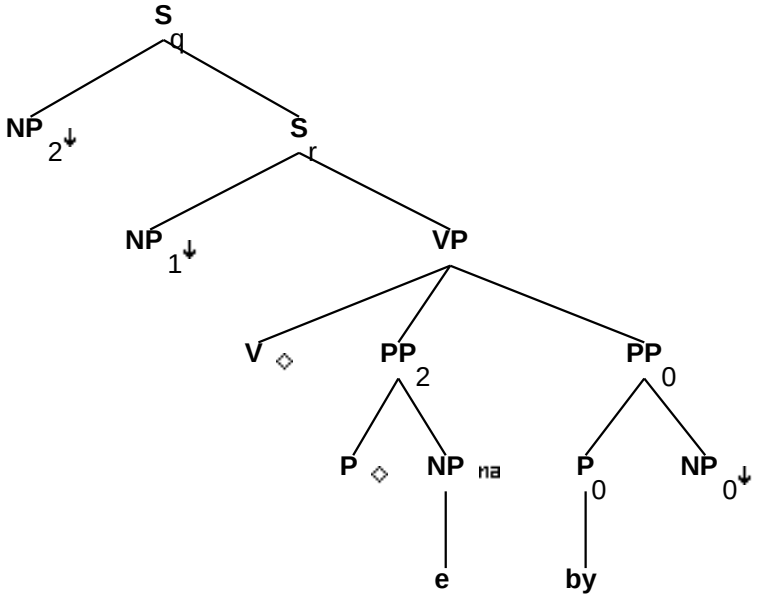
34.3 features

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

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

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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = NP.t:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_r.t:<comp> = nil
S_r.t:<conj> = nil
NP:<trace> = NP:<trace>
NP:<agr> = NP:<agr>
NP:<case> = NP:<case>
NP:<wh> = NP:<wh>
```

35 Tree "alphaRW2nx1VPnx2bynx0"

35.1 graphe



35.2 comments

NIL

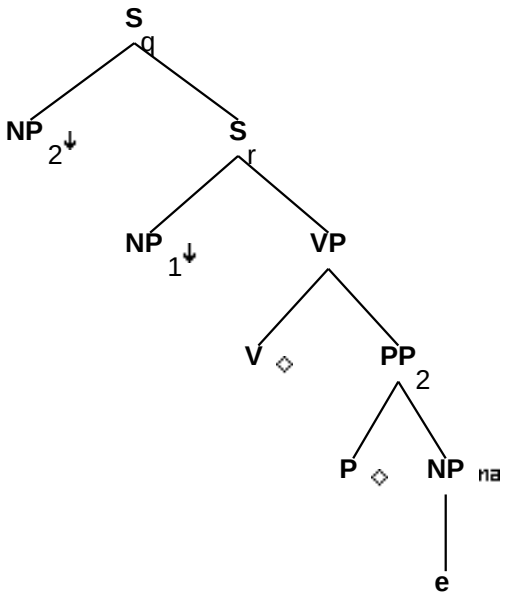
35.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<control> = NP_1.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>

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>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>

VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = NP.t:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_r.t:<comp> = nil
S_r.t:<conj> = nil
NP:<trace> = NP:<trace>
```

NP:<agr> = NP:<agr>
NP:<case> = NP:<case>
NP:<wh> = NP:<wh>

36 Tree "alphaRW2nx1VPnx2"

36.1 graphe



36.2 comments

NIL

36.3 features

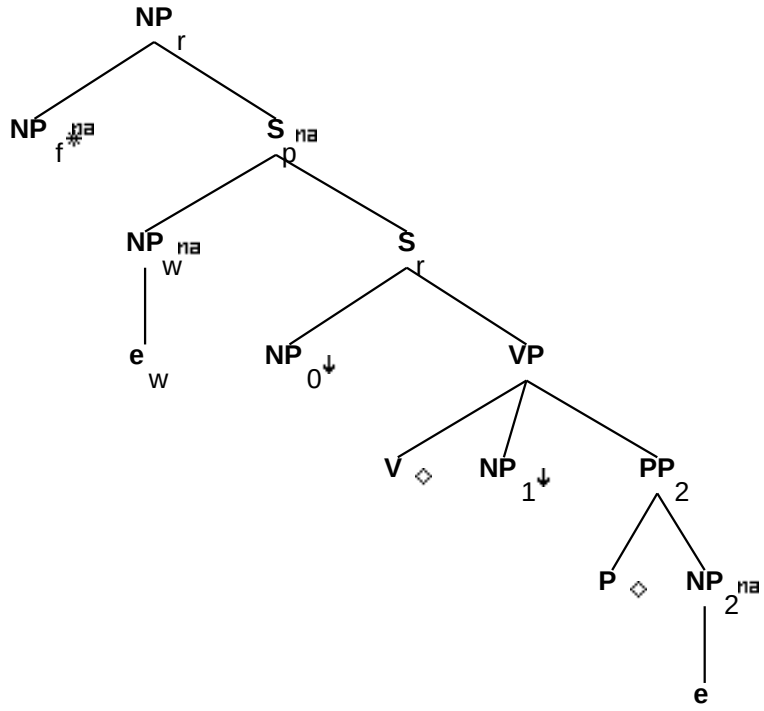
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.b:<mode> = VP.t:<mode>

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:<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>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>

```
VP.b:<compar> = -
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>
V.t:<punct struct> = nil
V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = NP.t:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_r.t:<comp> = nil
S_r.t:<conj> = nil
NP:<trace> = NP:<trace>
NP:<agr> = NP:<agr>
NP:<case> = NP:<case>
NP:<wh> = NP:<wh>
```

37 Tree "betaRNc2nx0Vnx1Pnx2"

37.1 graphe



37.2 comments

(COMP) relative clause on the (prepositional) object of a prepositional resultative.

'The shape that Max pounded the dough into'

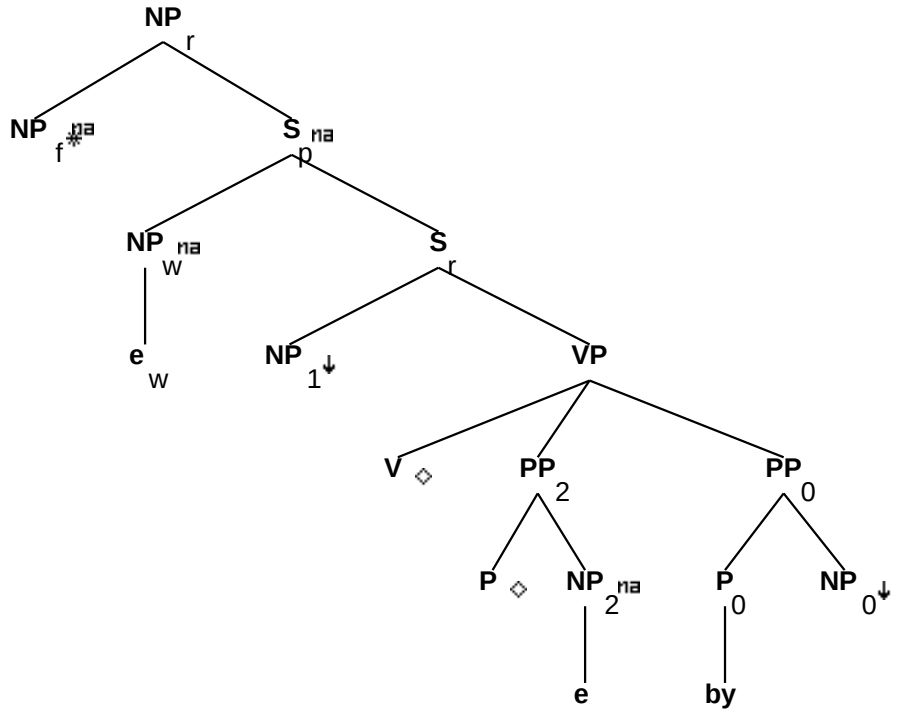
37.3 features

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

```
V.t:<passive> = -
V.t:<punct struct> = nil
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:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
NP_f.b:<refl> = -
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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_f.b:<refl> = -
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

38 Tree "betaRNc2nx1VPnx2bynx0"

38.1 graphe



38.2 comments

```
(COMP) relative clause on (prepositional) object in passivized
prepositional relative clause with
by-phrase:

 '(I saw) the shape that the dough was pounded into by Max'
```

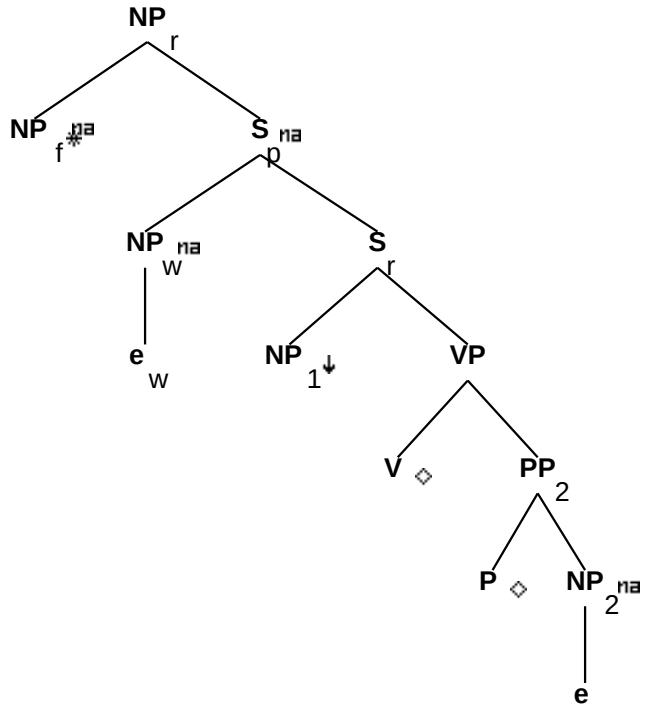
38.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<inv> = -
S_r.b:<comp> = nil
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>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
```

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

NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind/ppart
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_f.b:<case> = nom/acc
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

39 Tree "betaRNc2nx1VPnx2"

39.1 graphe



39.2 comments

```
(COMP)relative clause on (prepositional)
object in a passivized prepositional resultative,
w/o by-phrase:

'(I saw) the shape that the dough was pounded into'
```

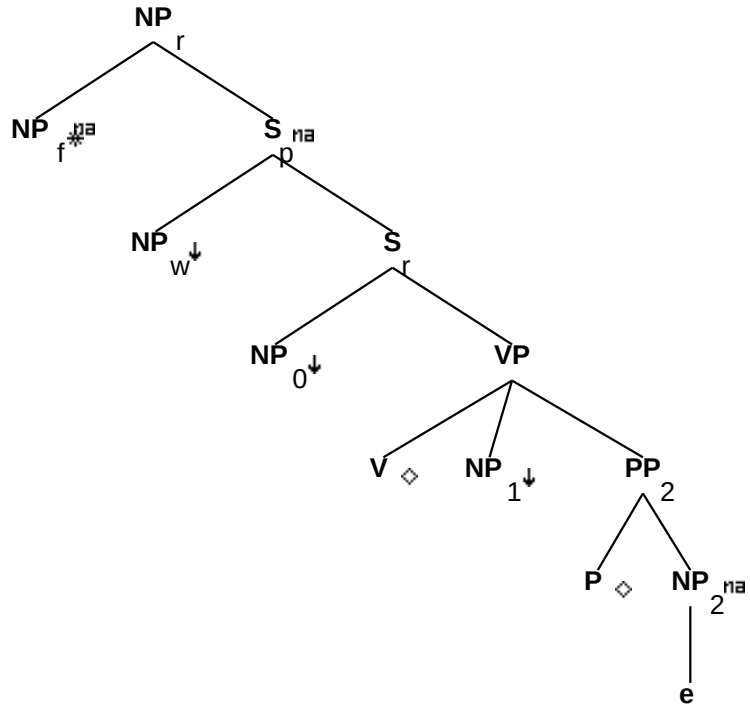
39.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<inv> = -
S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ppart/ger/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_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<compar>

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

40 Tree "betaRN2nx0Vnx1Pnx2"

40.1 graphe



40.2 comments

```
(wh) Relative clause on the (prepositional) object of
a prepositional resultative.

 'The shape which Max pounded the dough into'
```

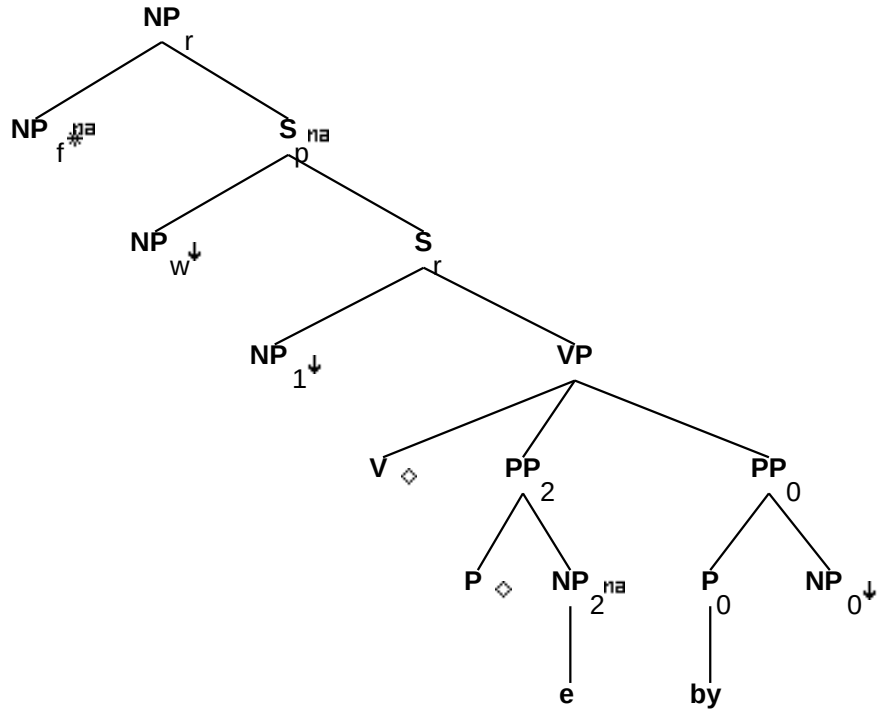
40.3 features

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

NP_f.b:<case> = nom/acc
NP_f.b:<refl> = -
NP_w.t:<wh> = +
NP_w.t:<trace> = NP.t:<trace>
NP_w.t:<case> = NP.t:<case>
NP_w.t:<agr> = NP.t:<agr>
```

```
S_r.t:<mode> = ind
S_r.t:<conj> = nil
S_r.t:<comp> = nil
S_r.t:<inv> = -

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

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

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:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
V.t:<passive> = -
V.t:<punct struct> = nil
NP_1.t:<case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

41 Tree "betaRN2nx1VPnx2bynx0"

41.1 graphe



41.2 comments

```
(wh) Relative Clause on (prepositional) object
of prepositional resultative passive with by-phrase.

'the shape which the dough was pounded into by the cook'
```

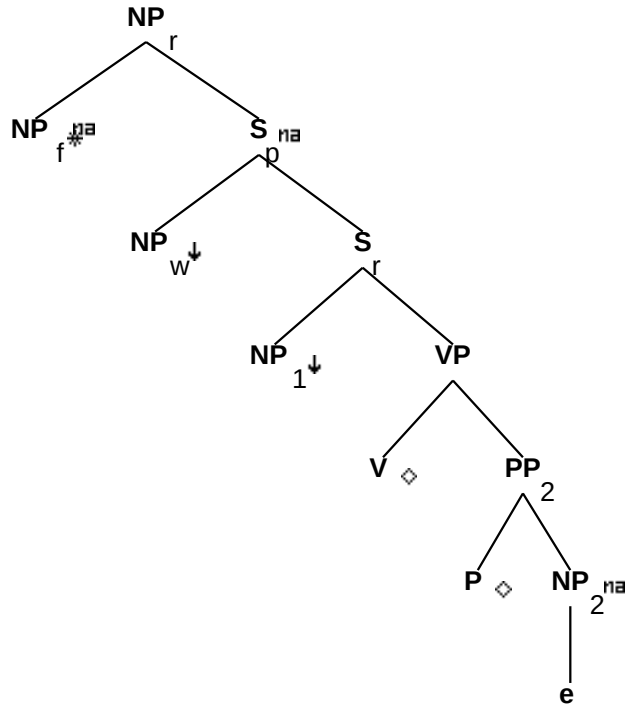
41.3 features

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

NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = nom/acc
NP_f.b:<refl> = -
NP_w.t:<wh> = +
NP_w.t:<case> = NP.t:<case>
NP_w.t:<agr> = NP.t:<agr>
NP_w.t:<trace> = NP.t:<trace>
```

```
S_r.t:<inv> = -
S_r.t:<mode> = ind
S_r.t:<conj> = nil
S_r.t:<comp> = nil

S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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:<compar> = -
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
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>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
V.t:<punct struct> = nil
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
```

42 Tree "betaRN2nx1VPnx2"

42.1 graphe



42.2 comments

```
(wh) relative clause on the (prepositional) object in
passivized prepositional resultative, w/o by-phrase:

'(I saw) the shape which the dough was pounded into'
```

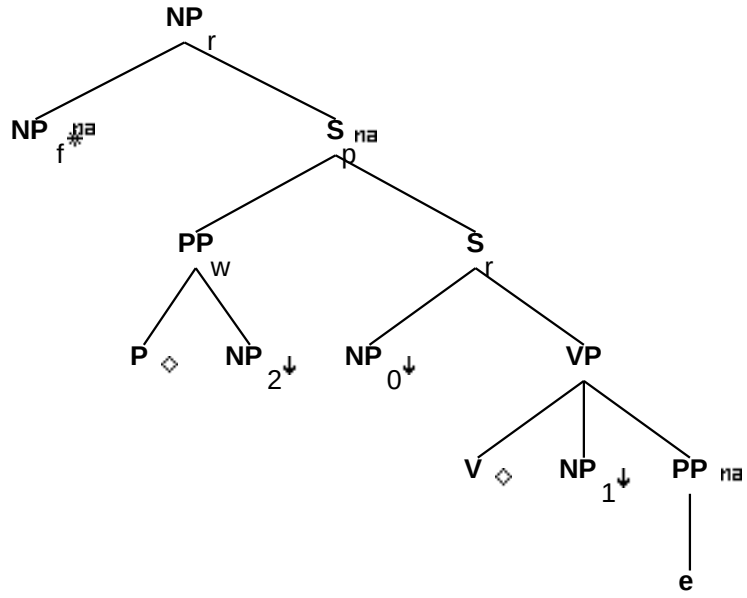
42.3 features

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind
S_r.t:<inv> = -
S_r.b:<comp> = nil
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.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.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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>
NP_r.b:<compar> = NP_f.t:<compar>

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

43 Tree "betaRNpx2nx0Vnx1Pnx2"

43.1 graphe



43.2 comments

```
Relative clause on an PP predicate (with pied-piping) in prepositional resultative.

'The shape into which Max pounded the dough into'
```

43.3 features

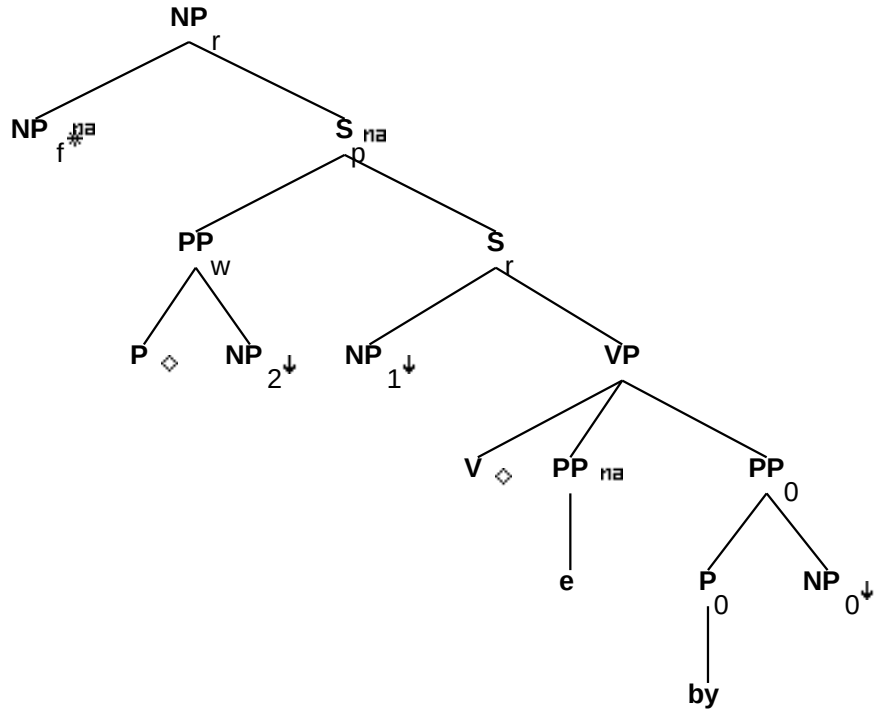
```
S_r.b:<extracted> = -
```

```
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:<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> = -
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> = -
S_r.b:<control> = NP_0.t:<control>
S_r.t:<inv> = -

PP_w.t:<wh> = +
PP_w.t:<trace> = PP.b:<trace>
PP_w.t:<case> = PP.b:<case>
PP_w.t:<agr> = PP.b:<agr>
P.t:<assign-case> = PP_w.b:<assign-case>
PP_w.b:<wh> = NP:<wh>
PP.b:<wh> = NP:<wh>
NP:<case> = PP_w.b:<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>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
```

44 Tree "betaRNpx2nx1VOnx2bynx0"

44.1 graphe



44.2 comments

```
Relative clause on (pied-piped) PP predicate
in passivized (with by-phrase) prepositional resultatives

'The shape into which the dough was pounded by Max'
```

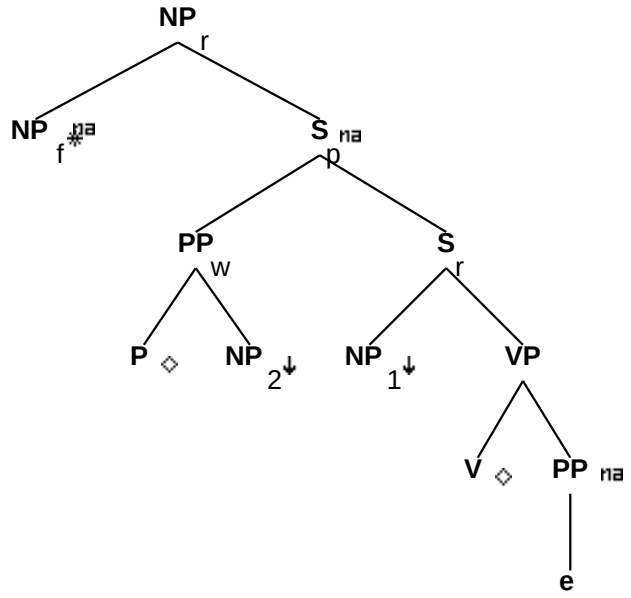
44.3 features

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<tense> = VP.t:<tense>
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> = -
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>
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_f.b:<case> = acc/nom
S_r.t:<comp> = nil
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>
NP:<case> = PP_w.b:<assign-case>
PP_w.t:<trace> = PP.b:<trace>
PP_w.t:<case> = PP.b:<case>
PP_w.t:<agr> = PP.b:<agr>
PP_w.t:<wh> = +
P.t:<assign-case> = PP_w.b:<assign-case>
PP_w.b:<wh> = NP:<wh>
PP.b:<wh> = NP:<wh>
```

45 Tree "betaRNpx2nx1VPnx2"

45.1 graphe



45.2 comments

```
Relative clause on (pied-piped) PP predicate
in passivized (w/o by-phrase) prepositional resultatives.

'The shape into which the dough was pounded'
```

45.3 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_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>
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_f.b:<case> = acc/nom
S_r.t:<comp> = nil
S_r.t:<mode> = ind/inf
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
NP_r.b:<compar> = NP_f.t:<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>

NP:<case> = PP_w.b:<assign-case>
PP_w.t:<trace> = PP.b:<trace>
PP_w.t:<case> = PP.b:<case>
PP_w.t:<agr> = PP.b:<agr>
PP_w.t:<wh> = +
P.t:<assign-case> = PP_w.b:<assign-case>
PP_w.b:<wh> = NP:<wh>
PP.b:<wh> = NP:<wh>
```