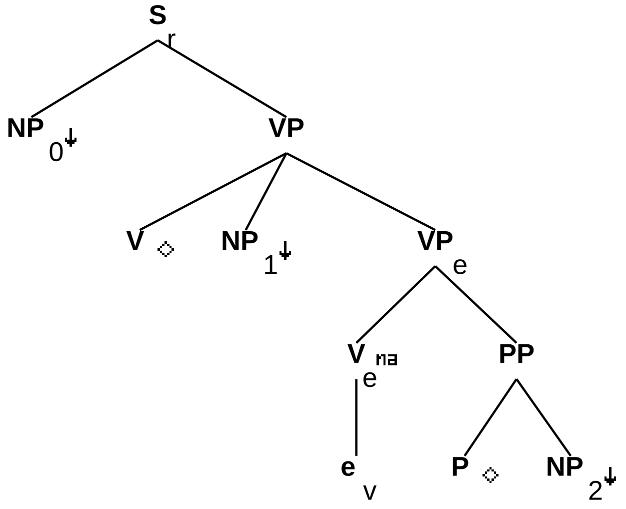
Family "Tnx0Vnx1Pnx2"

March 5, 2008

1 Tree "alphanx0Vnx1Pnx2"

1.1 graphe



1.2 comments

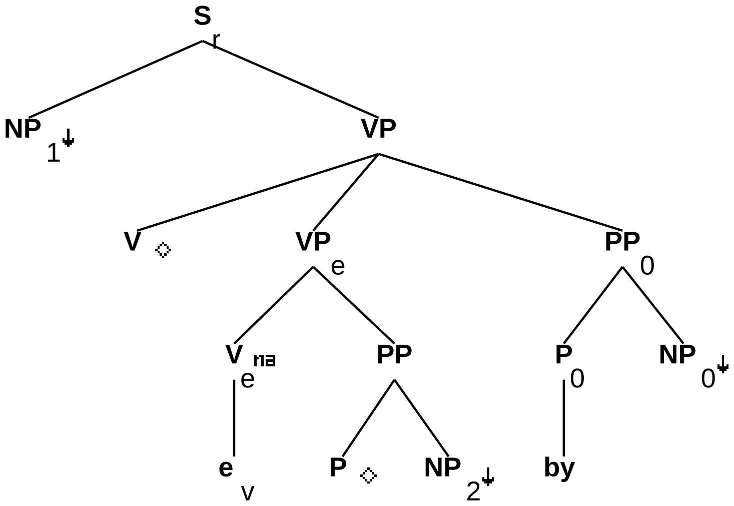
```
Ditransitive with PP

He put his reputation on the line.
```

1.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_0:<wh>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>

VP.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>
VP.b:<compar> = VP_e.t:<compar>
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

2 Tree ”alphanx1VPnx2bynx0”

2.1 graphe



2.2 comments

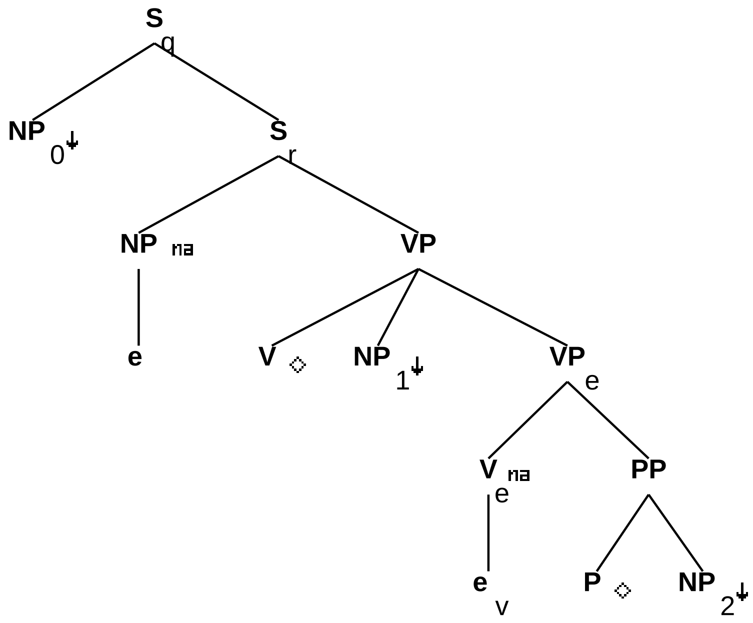
```
Passive:
the poodle was put in the oven by Max
```

2.3 features

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


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

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

3 Tree ”alphaW0nx0Vnx1Pnx2”

3.1 graphe



3.2 comments

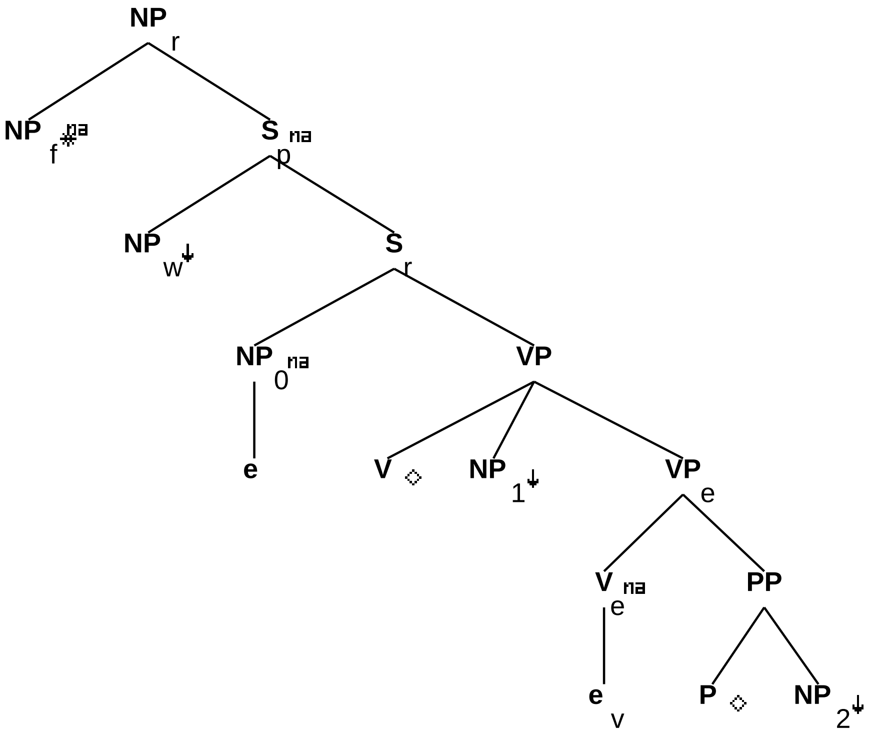
```
Need to decide what VP agrees with
```

3.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<case>
S_r.t:<conj> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm

S_q.b:<extracted> = +

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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<wh> = +
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
NP:<trace> = NP_0.t:<trace>
NP:<agr> = NP_0.t:<agr>
NP:<case> = NP_0.t:<case>
NP.t:<wh> = NP_0.t:<wh>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

4 Tree "betaN0nx0Vnx1Pnx2"

4.1 graphe



4.2 comments

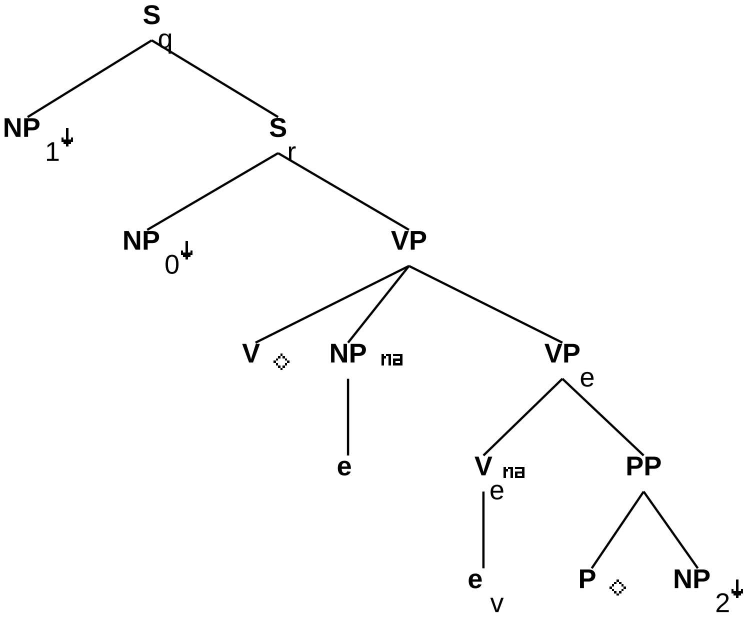
```
Need to decide what VP agrees with
```

4.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf
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:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.t:<conj> = nil
S_r.t:<comp> = nil

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
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> = +
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

5 Tree ”alphaW1nx0Vnx1Pnx2”

5.1 graphe



5.2 comments

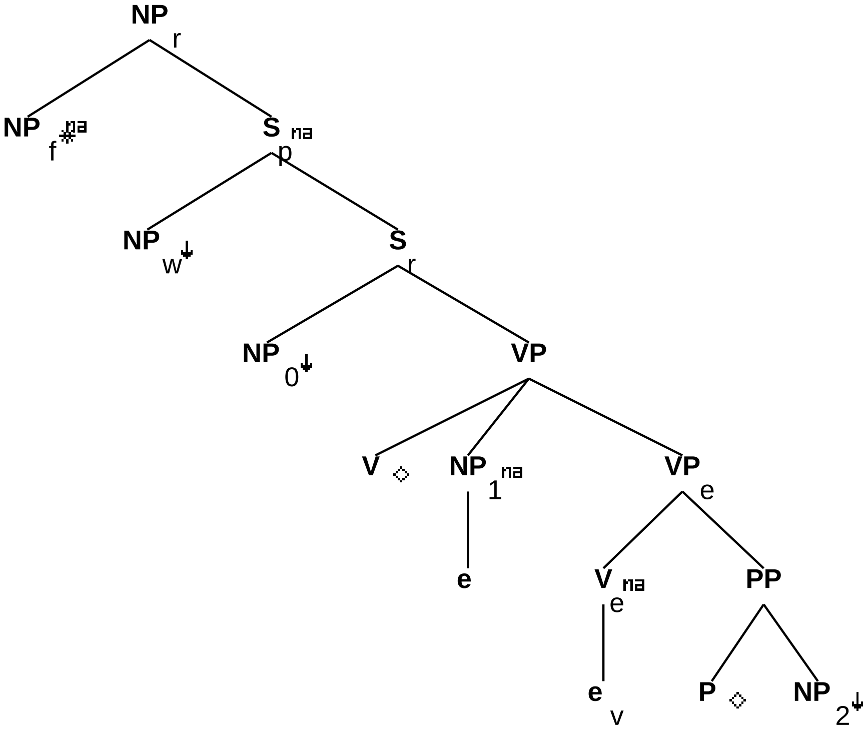
```
Need to decide what VP agrees with
```

5.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
S_q.b:<extracted> = +

S_q.b:<wh> = NP_1:<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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_2:<case> = PP.b:<assign-case>
NP.t:<case> = acc
NP:<trace> = NP_1.t:<trace>
NP:<agr> = NP_1.t:<agr>
NP:<case> = NP_1.t:<case>
NP.t:<wh> = NP_1.t:<wh>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

6 Tree "betaN1nx0Vnx1Pnx2"

6.1 graphe



6.2 comments

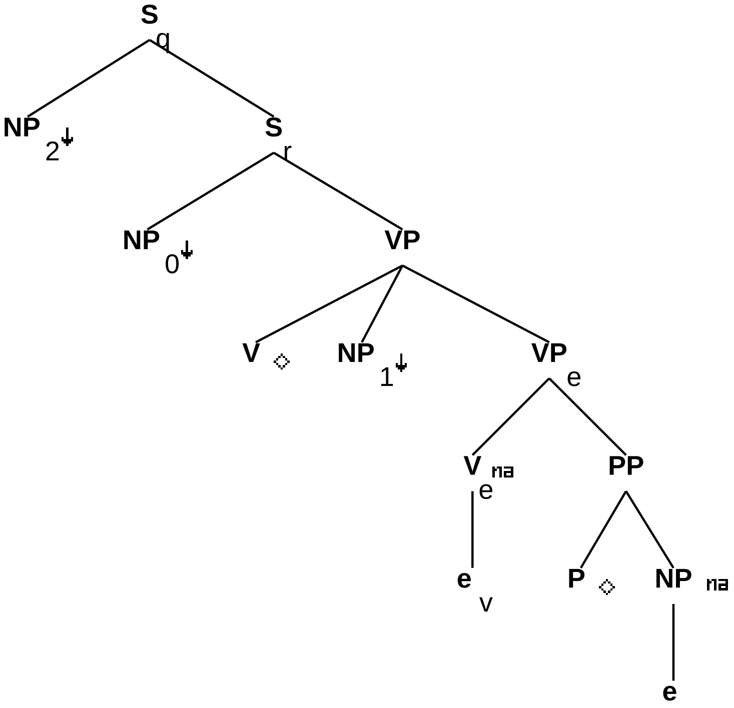
```
Need to decide what VP agrees with
```

6.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.t:<comp> = nil

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc
NP_2:<case> = PP.b:<assign-case>
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> = +
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

7 Tree "alphaW2nx0Vnx1Pnx2"

7.1 graphe



7.2 comments

```
Need to decide what VP agrees with
```

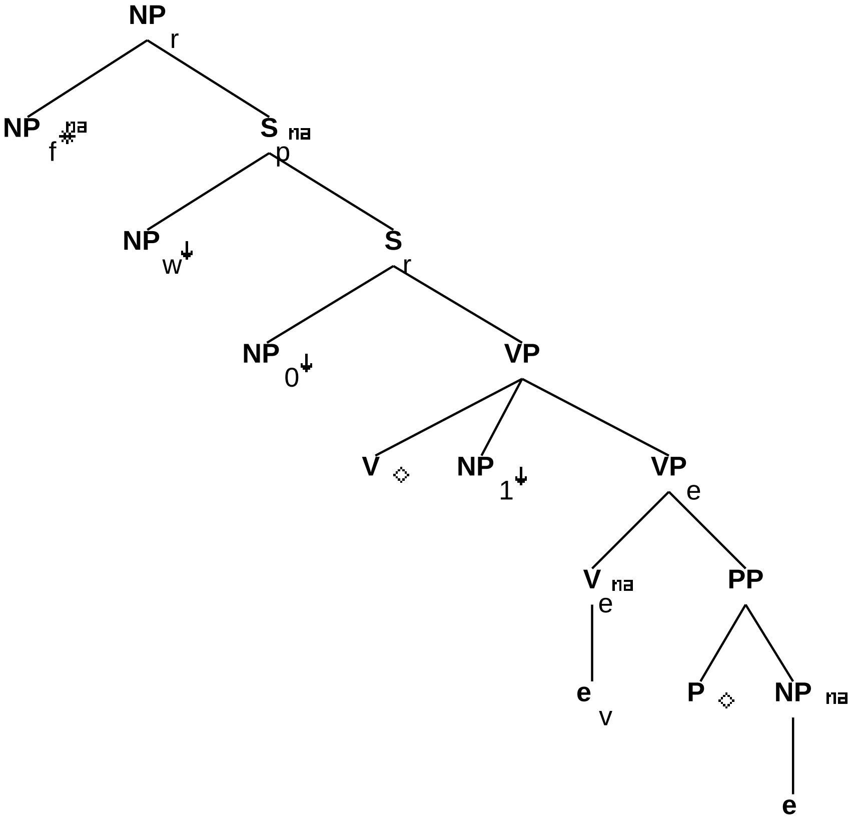
7.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

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

S_q.b:<wh> = NP_2:<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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
NP_2:<case> = NP.t:<case>
NP_2:<agr> = NP.t:<agr>
NP_2:<trace> = NP.t:<trace>
NP_2:<wh> = NP.t:<wh>
NP.t:<wh> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP.t:<wh>
V.t:<passive> = -
NP_2:<case> = NP.t:<case>
```

8 Tree "betaN2nx0Vnx1Pnx2"

8.1 graphe



8.2 comments

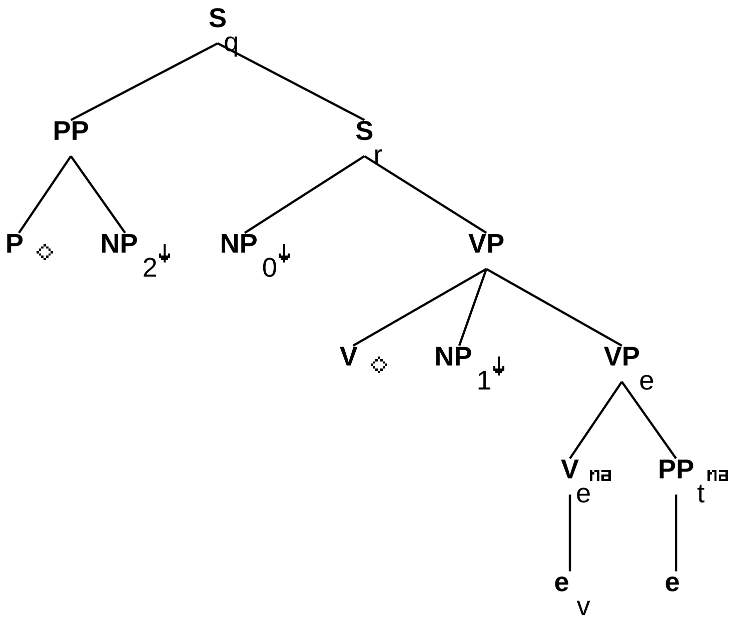
```
Need to decide what VP agrees with
```

8.3 features

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

NP_f.b:<case> = nom/acc

S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.t:<comp> = nil

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc
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> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

9 Tree ”alphapW2nx0Vnx1Pnx2”

9.1 graphe



9.2 comments

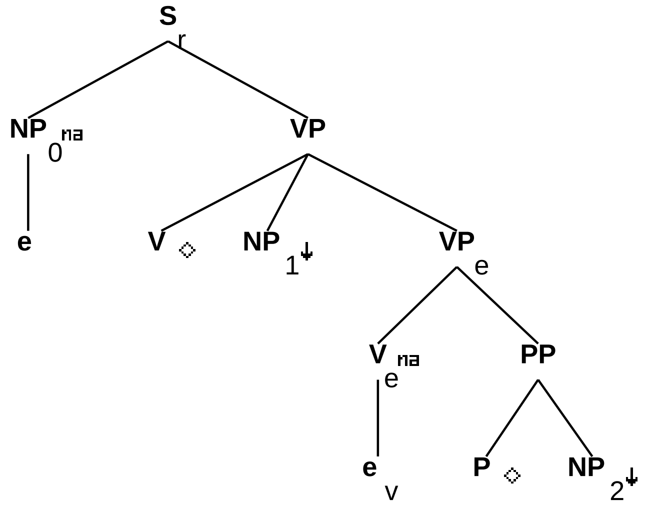
```
Need to decide what VP agrees with
```

9.3 features

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
S_q.b:<extracted> = +

S_q.b:<wh> = PP_2:<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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
PP_2:<trace> = PP.t:<trace>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

10 Tree ”alphaInx0Vnx1Pnx2”

10.1 graphe



10.2 comments

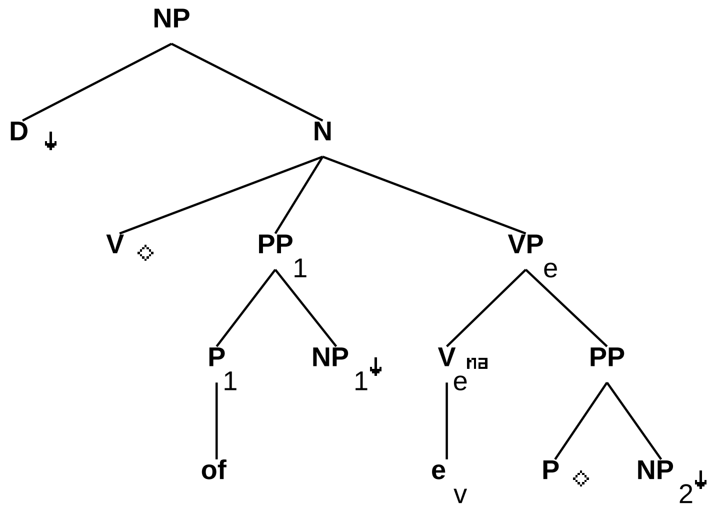
no comments

10.3 features

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


S_r.b:<mode> = imp
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_0:<wh>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>

VP.t:<neg> = -
VP.t:<mode> = base
VP.b:<mode> = V.t:<mode>
VP.t:<tense> = pres
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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
V.b:<mode> = base
```

11 Tree ”alphaDnx0Vnx1Pnx2”

11.1 graphe



11.2 comments

Ditransitive Determiner gerund with PP:

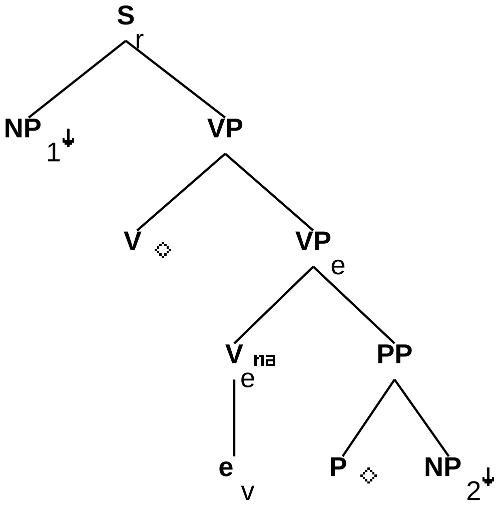
... ’the putting of his reputation on the line’

11.3 features

```
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
NP.b:<const> = D.t:<const>
NP.b:<definite> = D.t:<definite>
NP.b:<quan> = D.t:<quan>
NP.b:<card> = D.t:<card>
NP.b:<gen> = D.t:<gen>
NP.b:<decreas> = D.t:<decreas>
NP.b:<wh> = D.t:<wh>
NP.b:<case> = nom/acc
NP.b:<agr num> = sing
NP.b:<agr pers> = 3
NP.b:<agr 3rdsing> = +
P_1.b:<assign-case> = acc
PP_1.b:<assign-case> = P_1.t:<assign-case>
PP_1.b:<assign-case> = NP_1.t:<case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_1.b:<wh> = NP_1:<wh>
V.b:<mode> = ger
```

12 Tree "alphanx1VPnx2"

12.1 graphe



12.2 comments

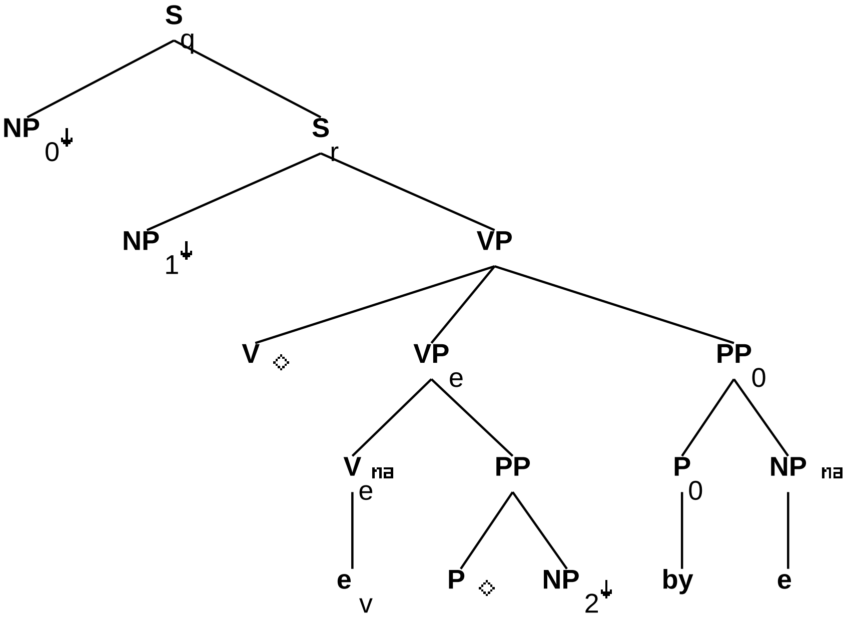
```
Passive w/out by-phrase:
the poodle was put in the oven
```

12.3 features

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


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

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

13 Tree ”alphaW0nx1VPnx2bynx0”

13.1 graphe



13.2 comments

```
passive, extraction from by-phrase:
who was the poodle put in the oven by
```

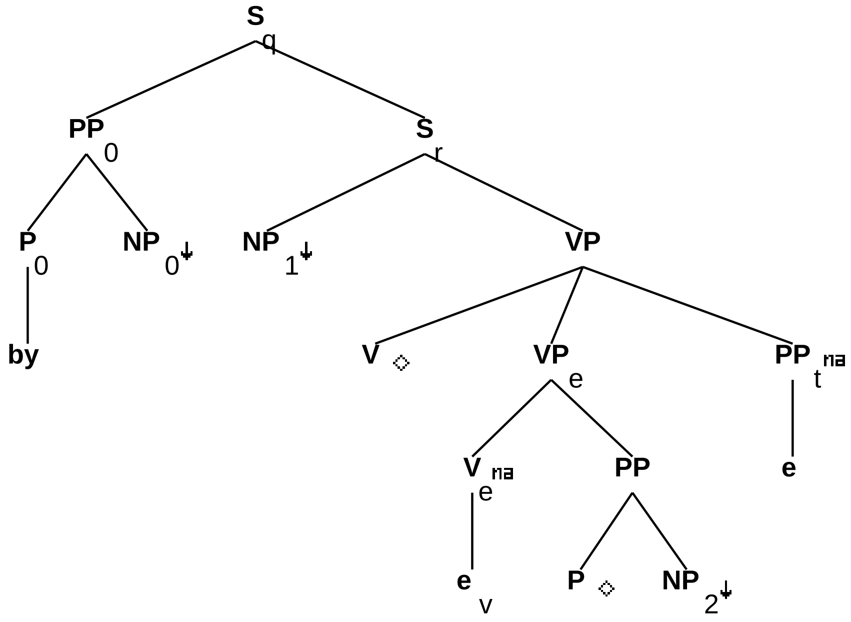
13.3 features

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


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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +

S_q.b:<wh> = NP_0:<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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_2:<case> = PP.b:<assign-case>
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>
NP:<case> = PP_0.b:<assign-case>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP:<wh>
```

```
V.t:<mode> = ppart
V.t:<passive> = +
```

14 Tree ”alphapW0nx1VPnx2bynx0”

14.1 graphe



14.2 comments

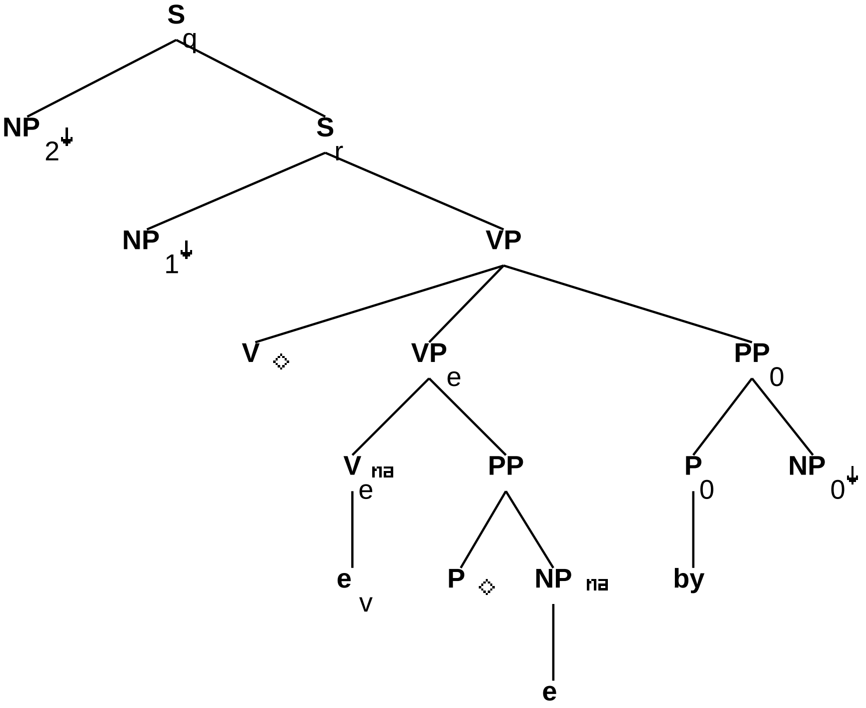
```
passive, extraction of by-phrase:
by whom was the poodle put in the oven
```

14.3 features

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


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:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = PP_0.t:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_0:<case> = PP_0.b:<assign-case>
NP_1:<agr> = VP.t:<agr>
NP_2:<case> = PP.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0:<trace> = PP.t:<trace>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

15 Tree "alphaW2nx1VPnx2bynx0"

15.1 graphe



15.2 comments

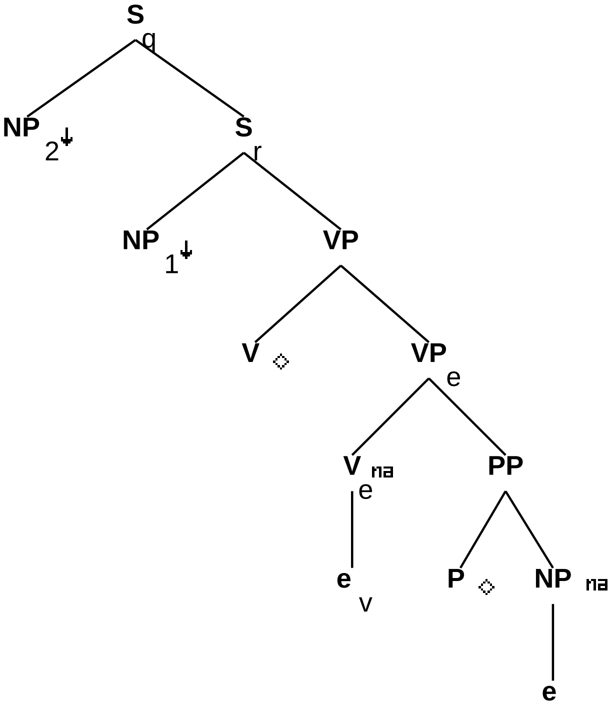
```
passive, extraction of NP2:
what was the poodle put in by Max
```

15.3 features

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


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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +

S_q.b:<wh> = NP_2:<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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_0:<case> = PP_0.b:<assign-case>
NP_2:<agr> = NP.t:<agr>
NP_2:<case> = NP.t:<case>
NP_2:<trace> = NP.t:<trace>
NP_2:<wh> = NP.t:<wh>
NP:<case> = PP.b:<assign-case>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<wh> = NP.t:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

16 Tree "alphaW2nx1VPnx2"

16.1 graphe



16.2 comments

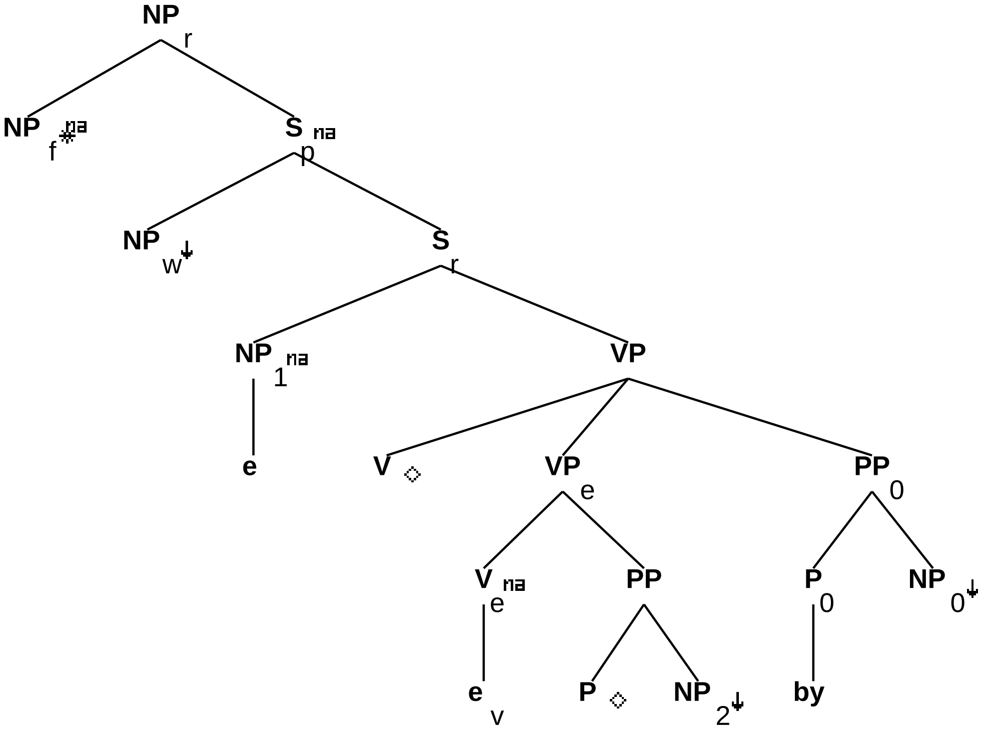
```
passive, extraction of NP2, w/O by-phrase:
what was the poodle put in
```

16.3 features

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


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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +

S_q.b:<wh> = NP_2:<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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_2:<agr> = NP.t:<agr>
NP_2:<case> = NP.t:<case>
NP_2:<trace> = NP.t:<trace>
NP_2:<wh> = NP.t:<wh>
NP:<case> = PP.b:<assign-case>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<wh> = NP.t:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

17 Tree ”betaN1nx1VPnx2bynx0”

17.1 graphe



17.2 comments

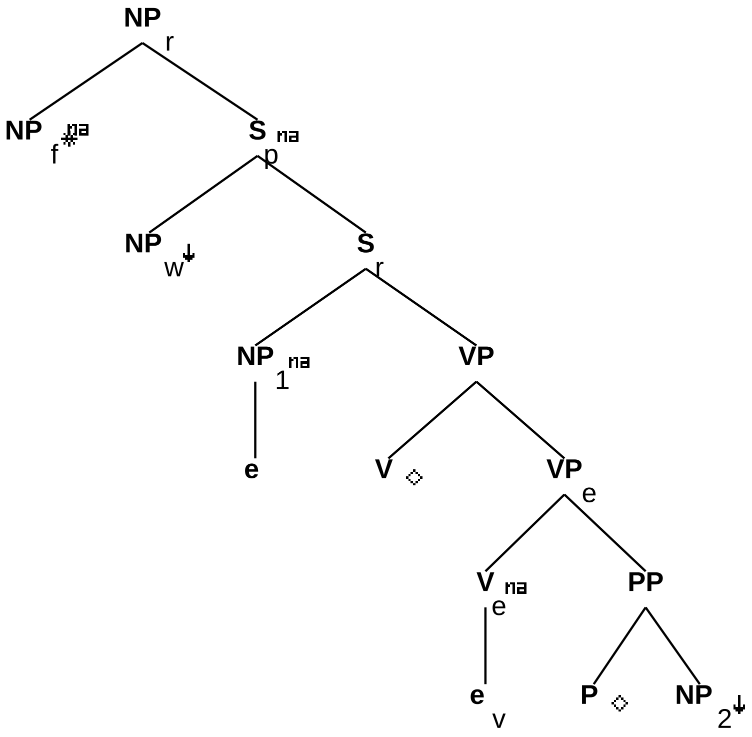
```
That relative clause, extraction from NP1:
(I saw) the dog that was put in the oven by Max
```

17.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf/ppart
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP.b:<assign-case>
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> = +
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
```

18 Tree "betaN1nx1VPnx2"

18.1 graphe



18.2 comments

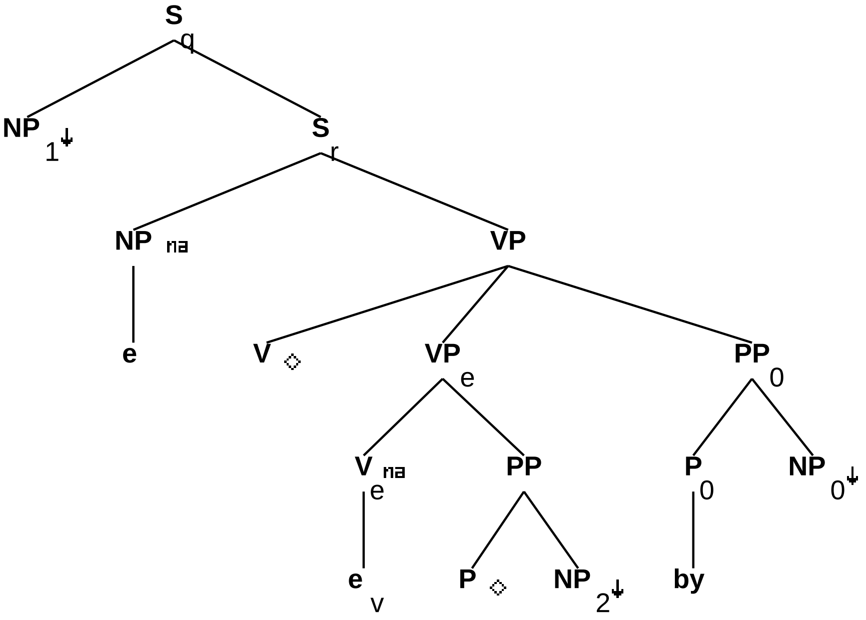
```
Passive that relative clause, extraction from NP1, w/o by-phrase:
(I saw) the dog that was put in the oven
```

18.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf/ppart
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
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> = +
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
```

19 Tree "alphaW1nx1VPnx2bynx0"

19.1 graphe



19.2 comments

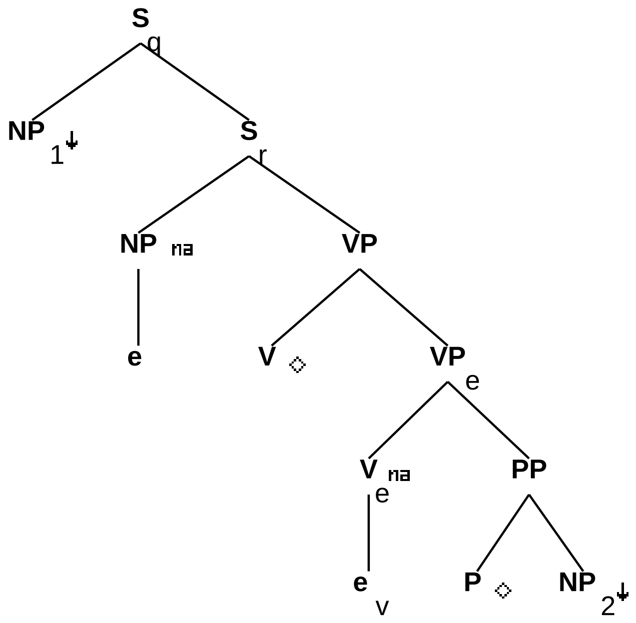
```
Wh question on NP1 in passive constructions, with by-phrase
what was put in the oven by Max
```

19.3 features

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



S_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:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<case>
S_r.t:<conj> = nil
S_q.b:<extracted> = +

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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_1.t:<wh> = +
NP_2:<case> = PP.b:<assign-case>
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>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

20 Tree ”alphaW1nx1VPnx2”

20.1 graphe



20.2 comments

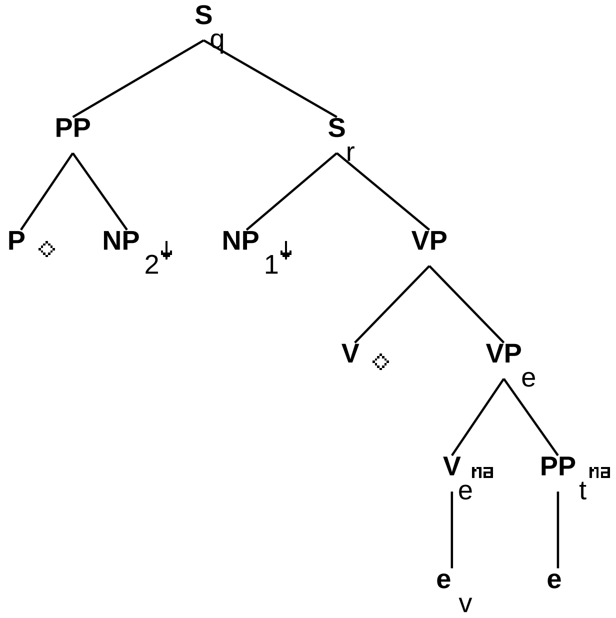
```
Wh question on NP1 in passive constructions, w/o by-phrase
what was put in the oven
```

20.3 features

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



S_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:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<case>
S_r.t:<conj> = nil
S_q.b:<extracted> = +

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_q.b:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_2:<case> = PP.b:<assign-case>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<wh> = +
NP.t:<trace> = NP_1.t:<trace>
NP.t:<wh> = NP_1.t:<wh>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

21 Tree "alphapW2nx1VPnx2"

21.1 graphe



21.2 comments

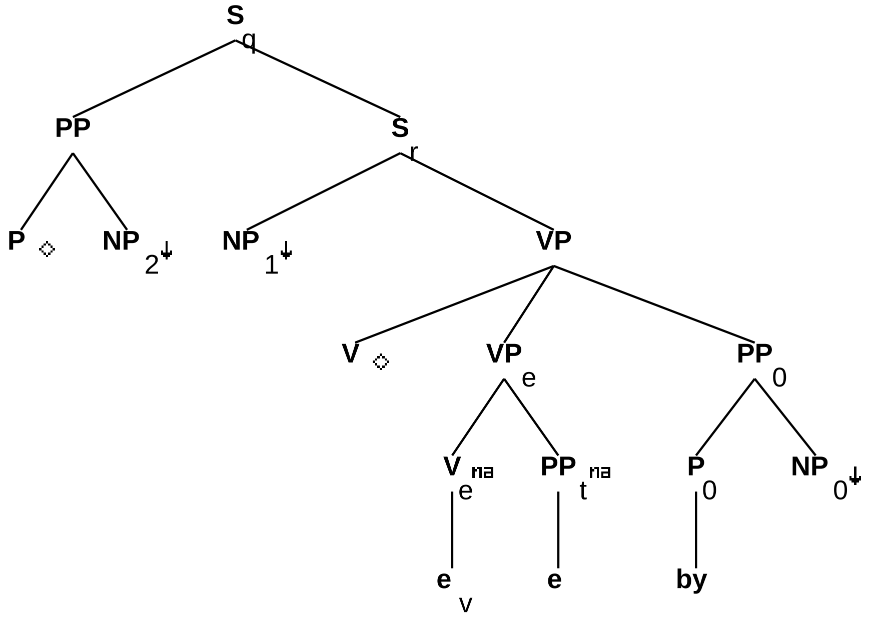
```
passive, extraction of NP2, w/o by-phrase:
where was the poodle put by Max
```

21.3 features

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


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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = PP_2:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_2:<case> = PP.b:<assign-case>
PP_2:<trace> = PP.t:<trace>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

22 Tree "alphapW2nx1VPnx2bynx0"

22.1 graphe



22.2 comments

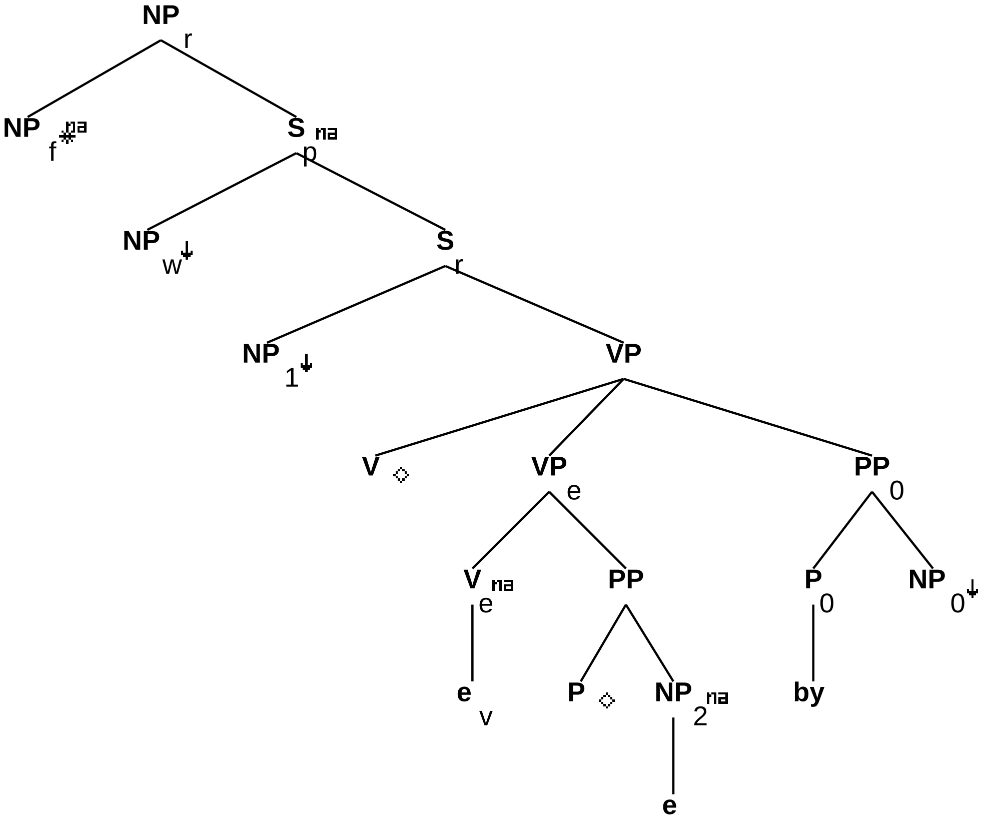
```
passive, extraction of NP2:
where was the poodle put by Max
```

22.3 features

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


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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = 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_q.b:<extracted> = +
S_q.b:<wh> = PP_2:<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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
NP_0:<case> = PP_0.b:<assign-case>
NP_2:<case> = PP.b:<assign-case>
PP_2:<trace> = PP.t:<trace>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

23 Tree "betaN2nx1VPnx2bynx0"

23.1 graphe



23.2 comments

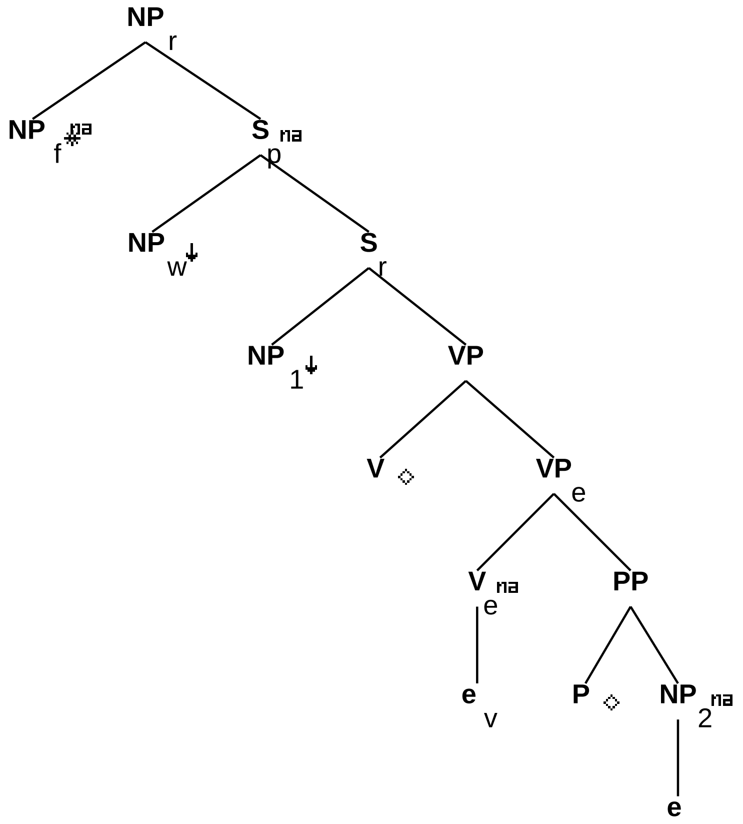
```
That relative clause, extraction of NP2 from indirect object
(I saw) the oven that the poodle was put in by Max
```

23.3 features

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

NP_f.b:<case> = nom/acc

S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.t:<mode> = ind/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
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> = +
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.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_2:<case>
PP_0.b:<wh> = NP_0:<wh>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

24 Tree "betaN2nx1VPnx2"

24.1 graphe



24.2 comments

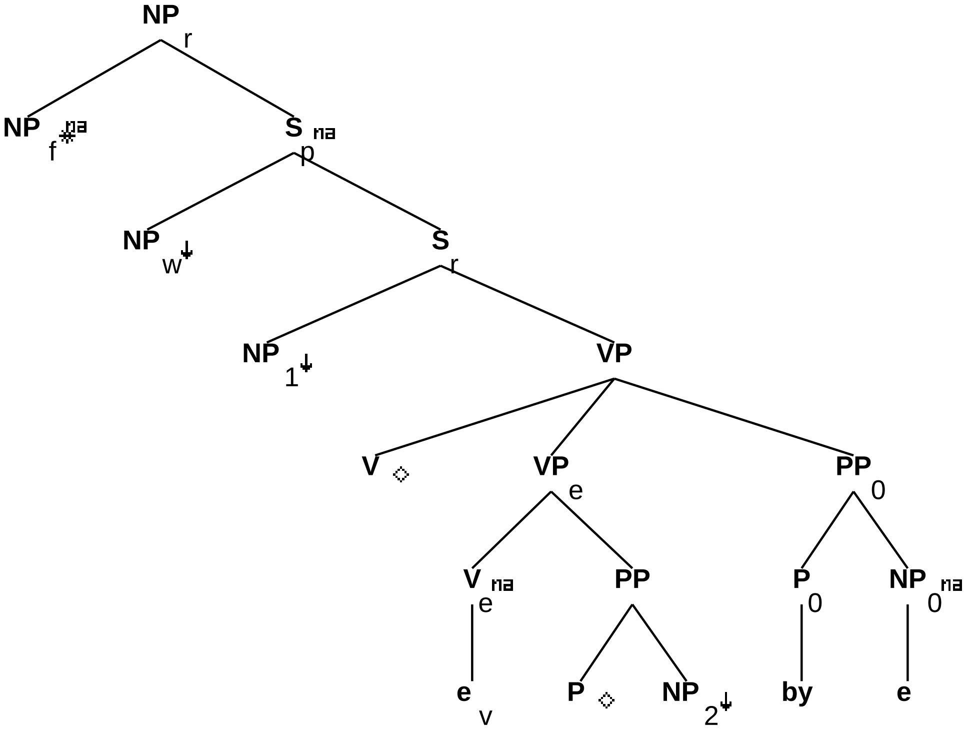
```
That relative clause, extraction of NP2 from indirect object, w/o by-phrase
(I saw) the oven that the poodle was put in
```

24.3 features

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

NP_f.b:<case> = nom/acc

S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.t:<mode> = ind/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
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> = +
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_2:<case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

25 Tree "betaN0nx1VPnx2bynx0"

25.1 graphe



25.2 comments

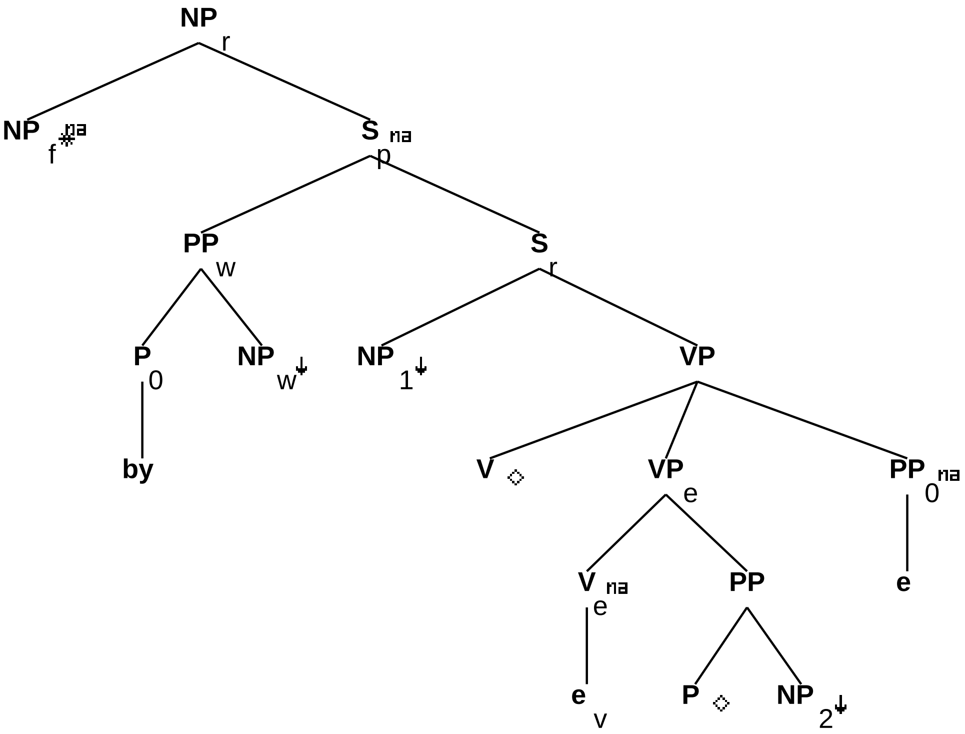
```
That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that the poodle was put in the oven by
```

25.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP.b:<assign-case>
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> = +
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

26 Tree ”betaNbynx0nx1VPnx2bynx0”

26.1 graphe



26.2 comments

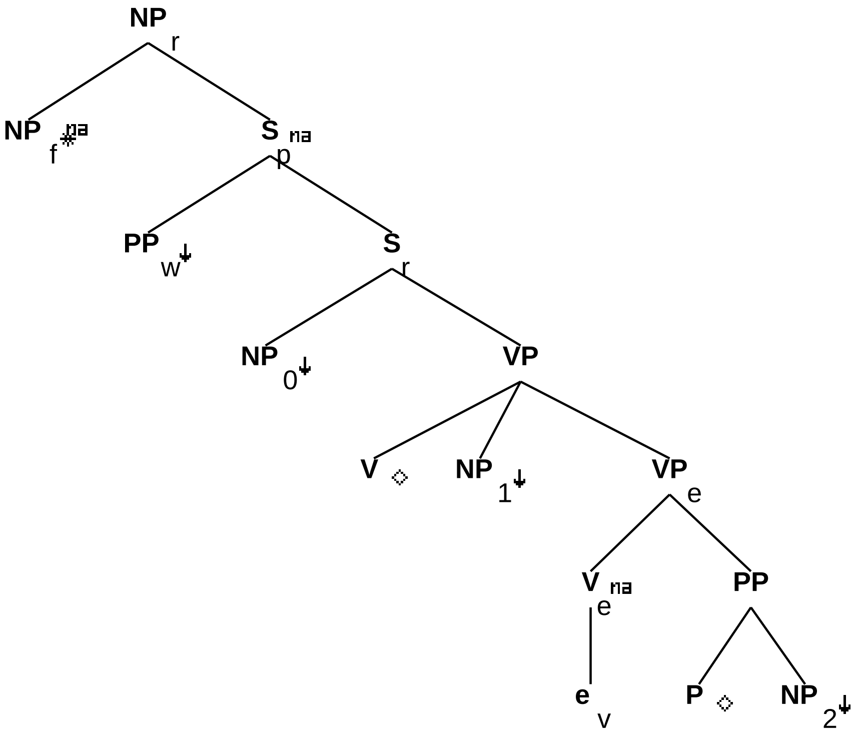
```
That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that the poodle was put in the oven by
```

26.3 features

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




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/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
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
S_r.b:<control> = NP_1.t:<control>

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
P.t:<assign-case> = PP.b:<assign-case>
NP_2:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

27 Tree "betaNpxnx0Vnx1Pnx2"

27.1 graphe



27.2 comments

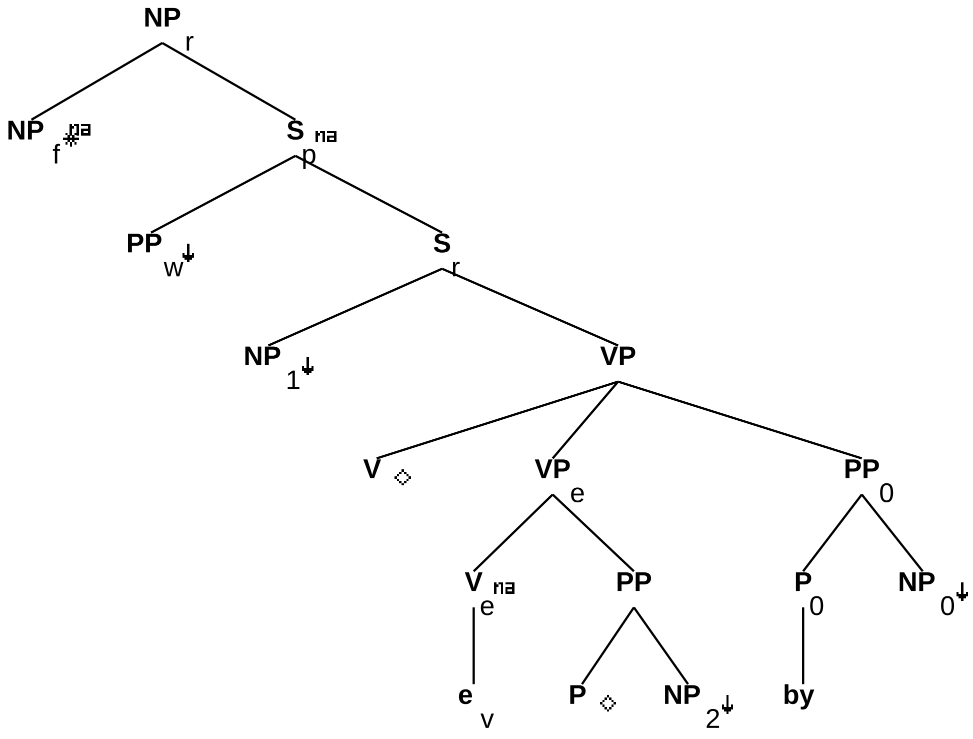
```
no comments
```

27.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
PP_w.t:<wh> = +
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

28 Tree ”betaNpxnx1VPnx2bynx0”

28.1 graphe



28.2 comments

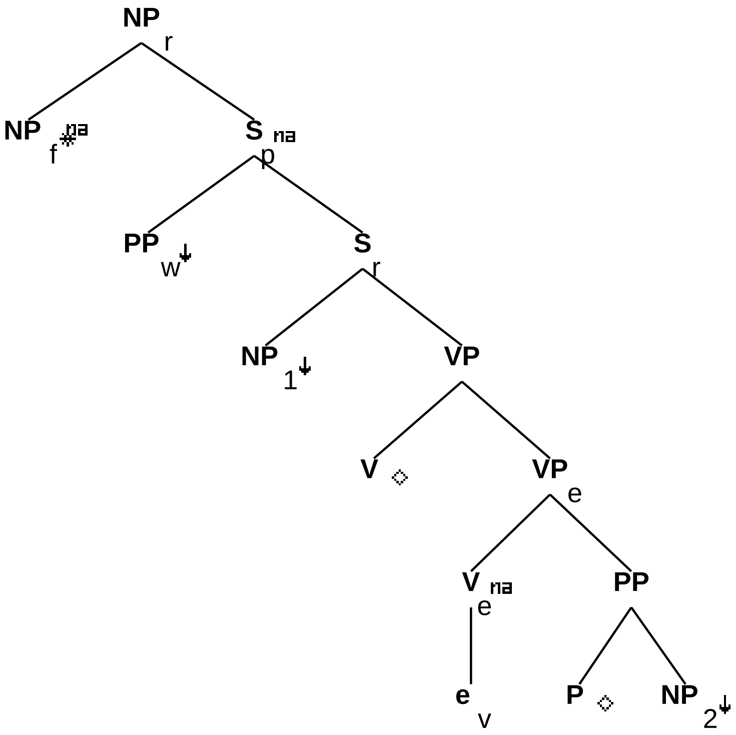
```
Passive:
the poodle was put in the oven by Max
```

28.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


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

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
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_w.t:<wh> = +
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

29 Tree "betaNpxnx1VPnx2"

29.1 graphe



29.2 comments

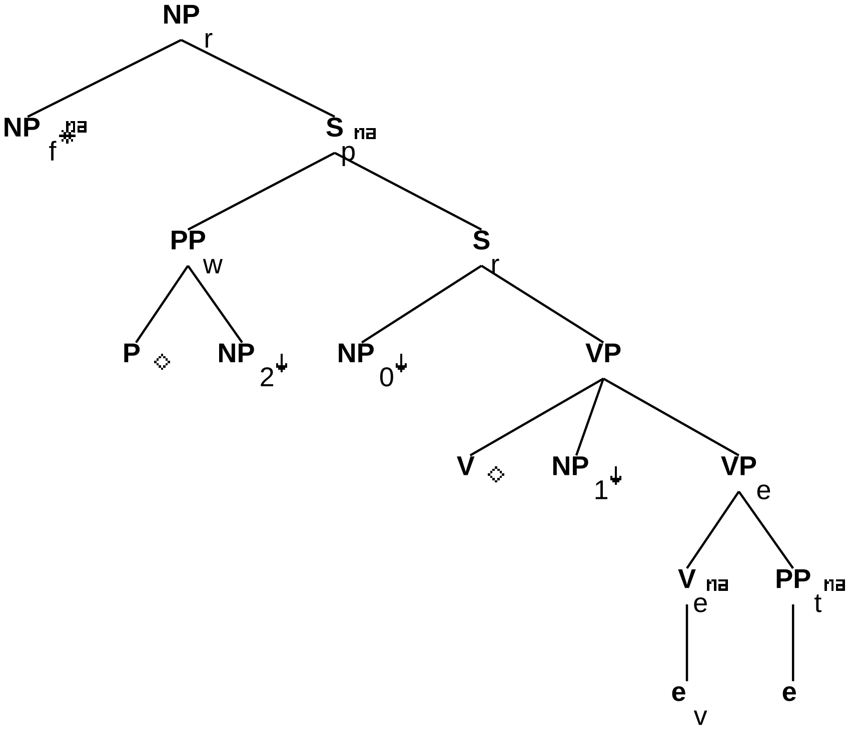
```
Passive w/out by-phrase:
the poodle was put in the oven
```

29.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


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

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
PP_w.t:<wh> = +
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

30 Tree "betaNpx2nx0Vnx1Pnx2"

30.1 graphe



30.2 comments

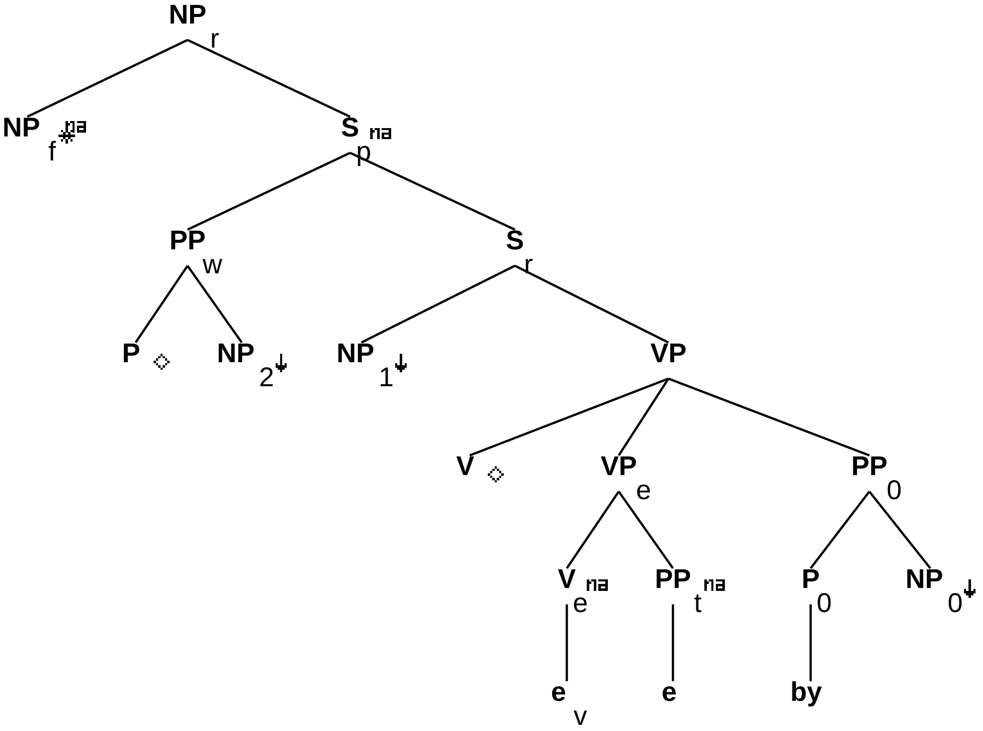
```
Need to decide what VP agrees with
```

30.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.t:<comp> = nil

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc
NP_2:<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_2:<wh>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

31 Tree ”betaNpx2nx1VPnx2bynx0”

31.1 graphe



31.2 comments

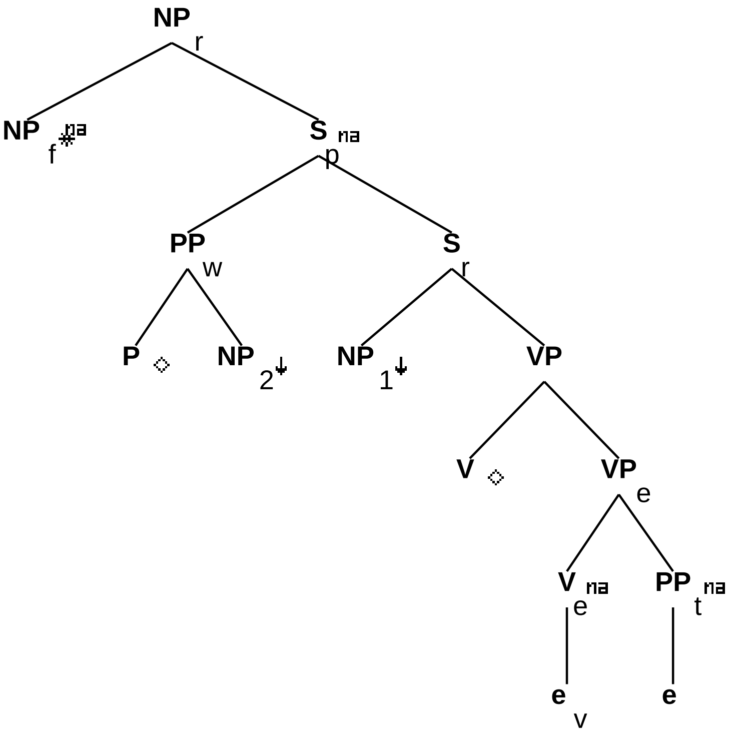
```
That relative clause, extraction of NP2 from indirect object
(I saw) the oven that the poodle was put in by Max
```

31.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP_w.b:<assign-case>
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_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_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

32 Tree ”betaNpx2nx1VPnx2”

32.1 graphe



32.2 comments

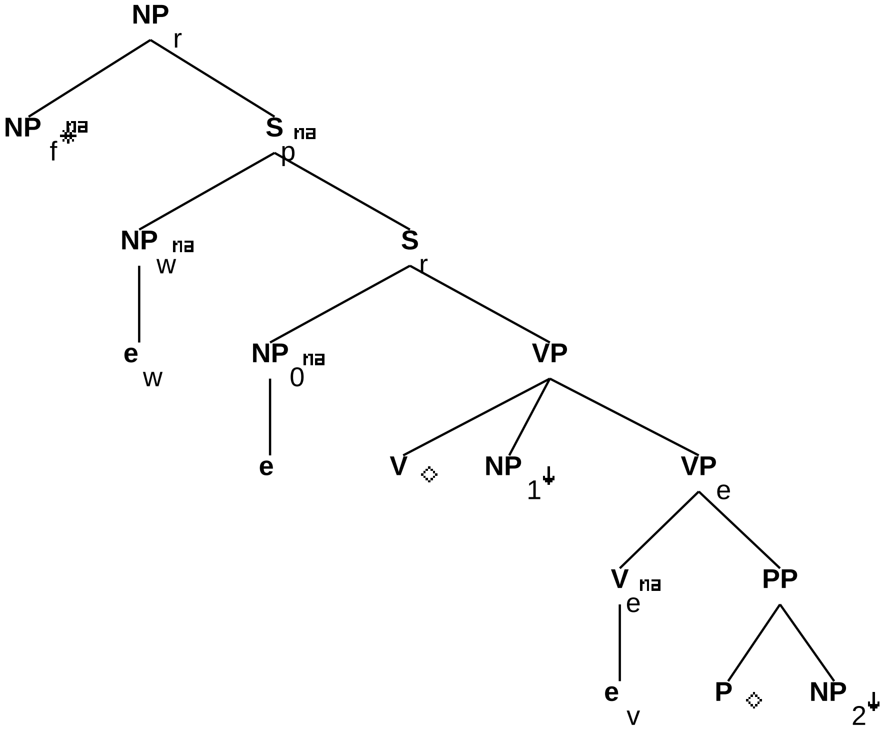
```
That relative clause, extraction of NP2 from indirect object, w/o by-phrase
(I saw) the oven that the poodle was put in
```

32.3 features

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

NP_f.b:<case> = nom/acc

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



S_r.t:<mode> = ind/inf
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.t:<comp> = nil

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<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_2:<wh>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

33 Tree "betaNc0nx0Vnx1Pnx2"

33.1 graphe



33.2 comments

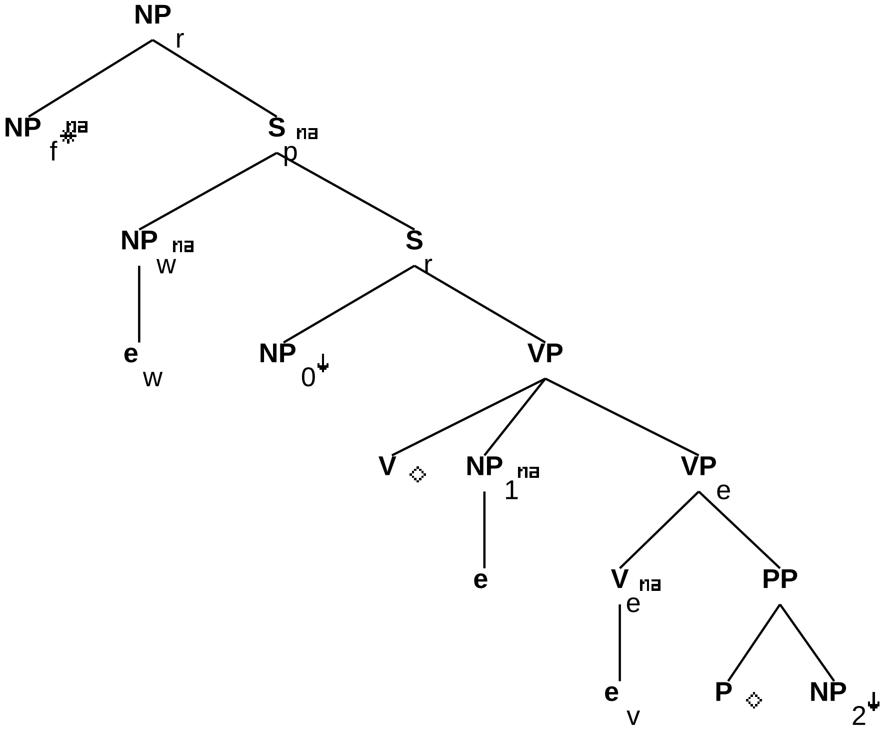
```
Need to decide what VP agrees with
```

33.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = nom/acc

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



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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
NP_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

34 Tree "betaNc1nx0Vnx1Pnx2"

34.1 graphe



34.2 comments

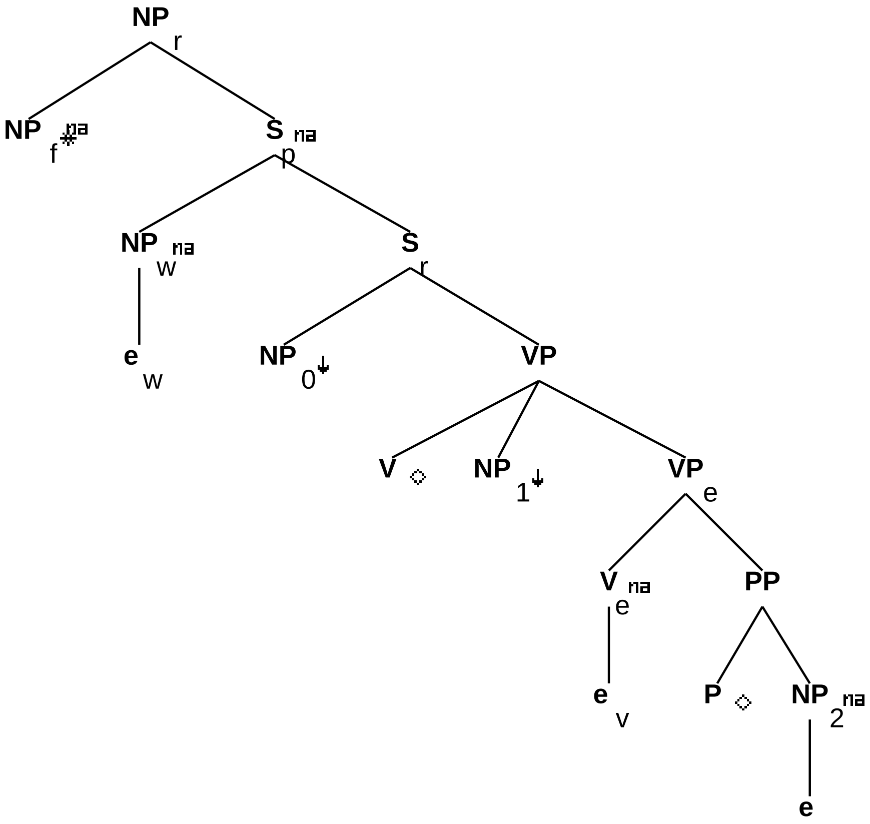
```
Need to decide what VP agrees with
```

34.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = nom/acc

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



S_r.t:<inv> = -
S_r.b:<inv> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<tense> = VP.t:<tense>
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
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>

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc
NP_2:<case> = PP.b:<assign-case>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

35 Tree "betaNc2nx0Vnx1Pnx2"

35.1 graphe



35.2 comments

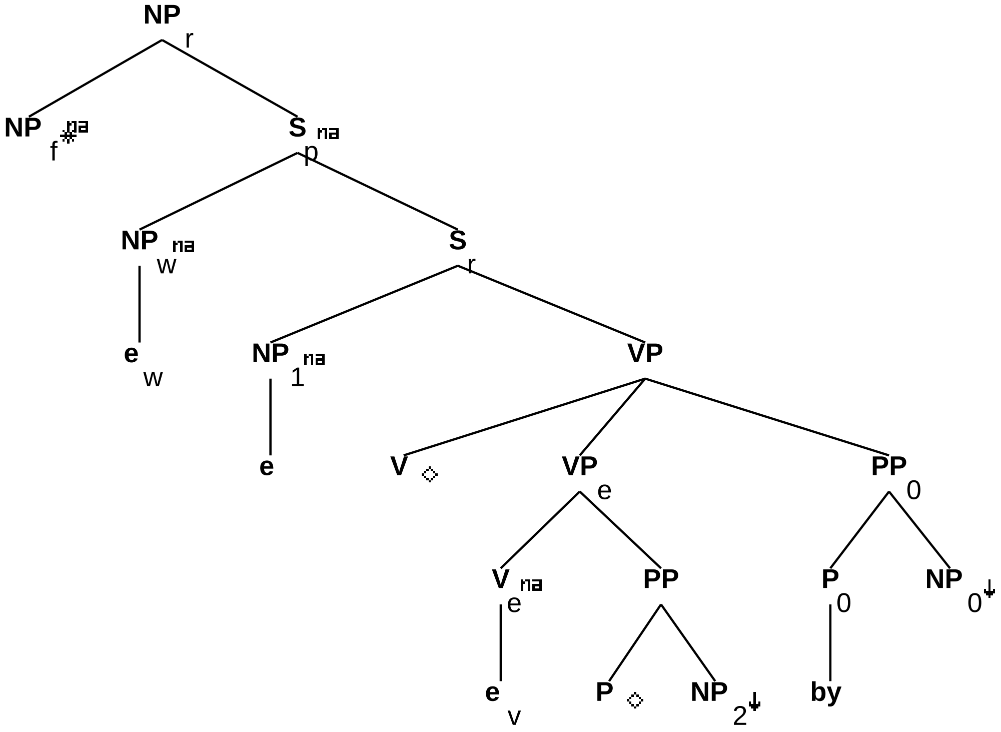
```
Need to decide what VP agrees with
```

35.3 features

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

NP_f.b:<case> = nom/acc
S_r.b:<assign-comp> = VP.t:<assign-comp>



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

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
NP_1.t:<case> = acc
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.b:<agr>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

36 Tree ”betaNc1nx1VPnx2bynx0”

36.1 graphe



36.2 comments

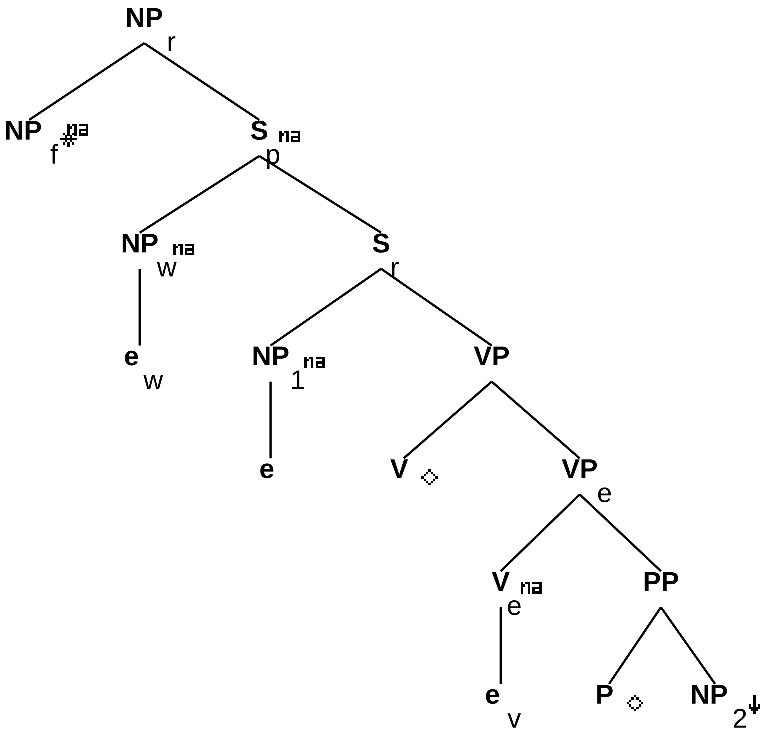
```
That relative clause, extraction from NP1:
(I saw) the dog that was put in the oven by Max
```

36.3 features

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

NP_f.b:<case> = nom/acc

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



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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP.b:<assign-case>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
```

37 Tree "betaNc1nx1VPnx2"

37.1 graphe



37.2 comments

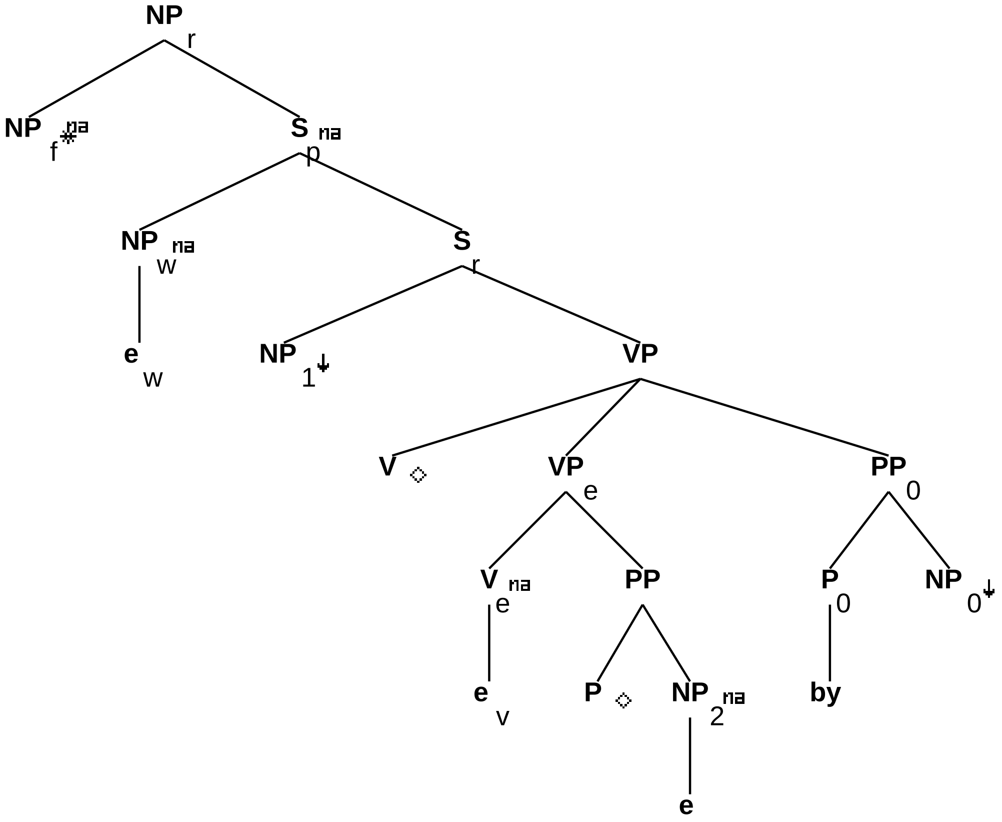
```
Passive that relative clause, extraction from NP1, w/o by-phrase:
(I saw) the dog that was put in the oven
```

37.3 features

```
NP_r.b:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
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:<refl> = -
NP_f.b:<case> = nom/acc

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



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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP.b:<assign-case>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
```

38 Tree "betaNc2nx1VPnx2bynx0"

38.1 graphe



38.2 comments

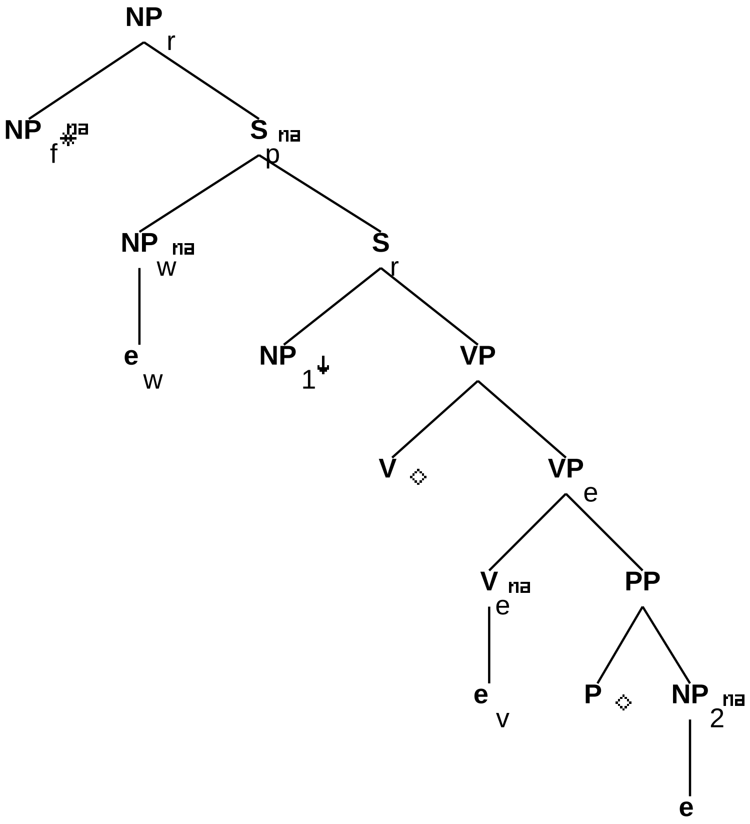
```
That relative clause, extraction of NP2 from indirect object
(I saw) the oven that the poodle was put in by Max
```

38.3 features

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

NP_f.b:<case> = nom/acc
S_r.b:<assign-comp> = VP.t:<assign-comp>
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
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>

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.b:<agr>
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.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_2:<case>
PP_0.b:<wh> = NP_0:<wh>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

39 Tree ”betaNc2nx1VPnx2”

39.1 graphe



39.2 comments

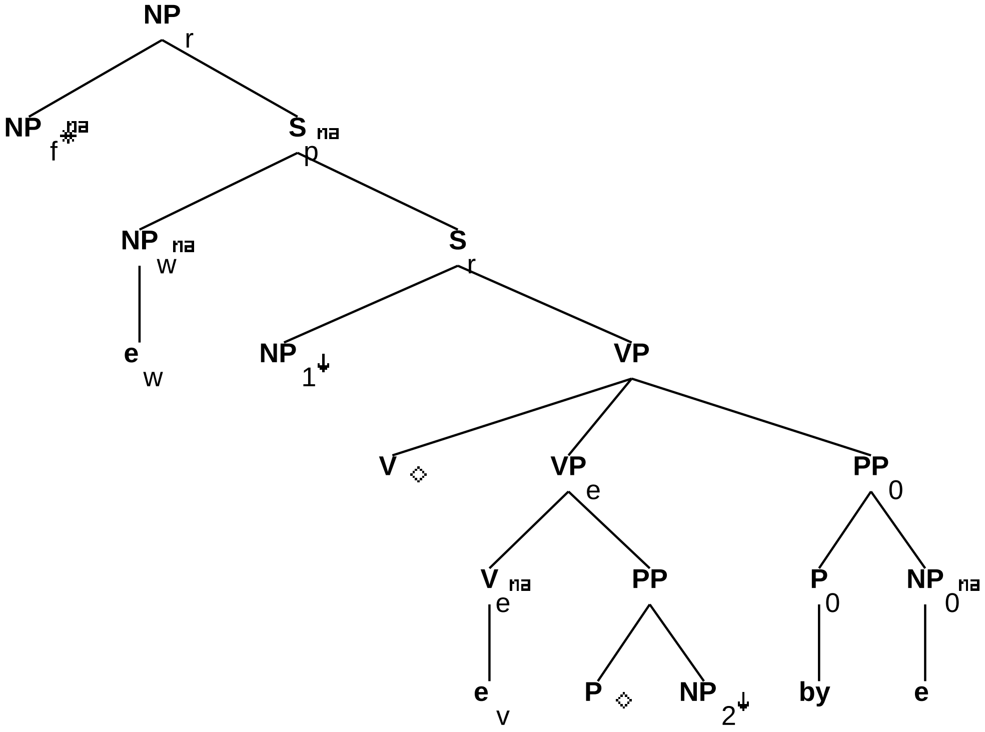
That relative clause, extraction of NP2 from indirect object, w/o by-phrase
(I saw) the oven that the poodle was put in

39.3 features

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

NP_f.b:<case> = nom/acc
S_r.b:<assign-comp> = VP.t:<assign-comp>
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:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
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>

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.b:<agr>
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_2:<case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

40 Tree ”betaNc0nx1VPnx2bynx0”

40.1 graphe



40.2 comments

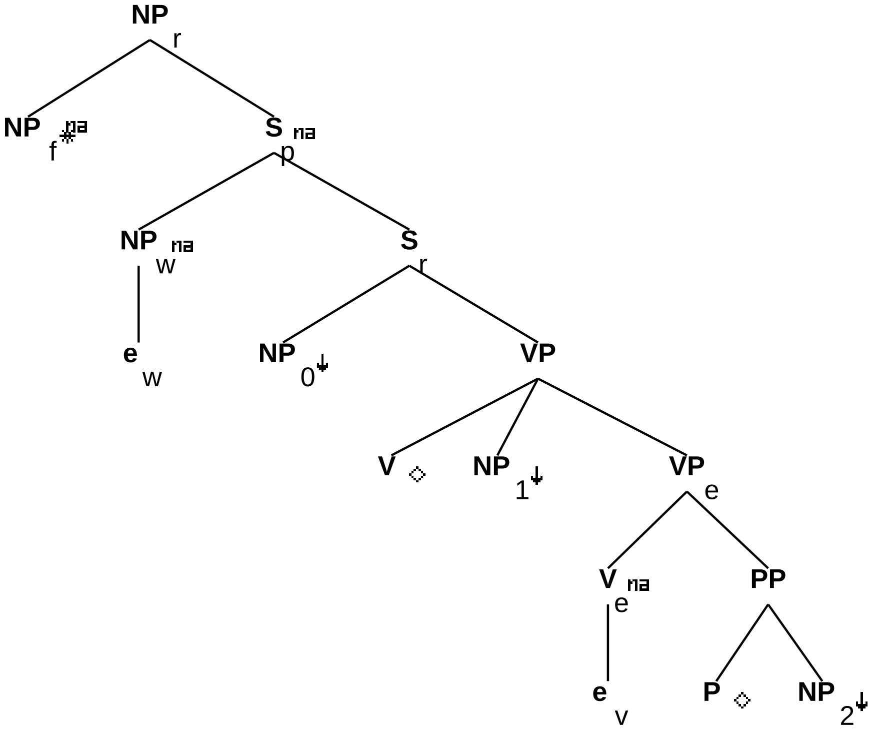
```
That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that the poodle was put in the oven by
```

40.3 features

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

NP_f.b:<case> = nom/acc
S_r.b:<assign-comp> = VP.t:<assign-comp>



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

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_2:<case> = PP.b:<assign-case>
NP_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

41 Tree ”betaNcnx0Vnx1Pnx2”

41.1 graphe



41.2 comments

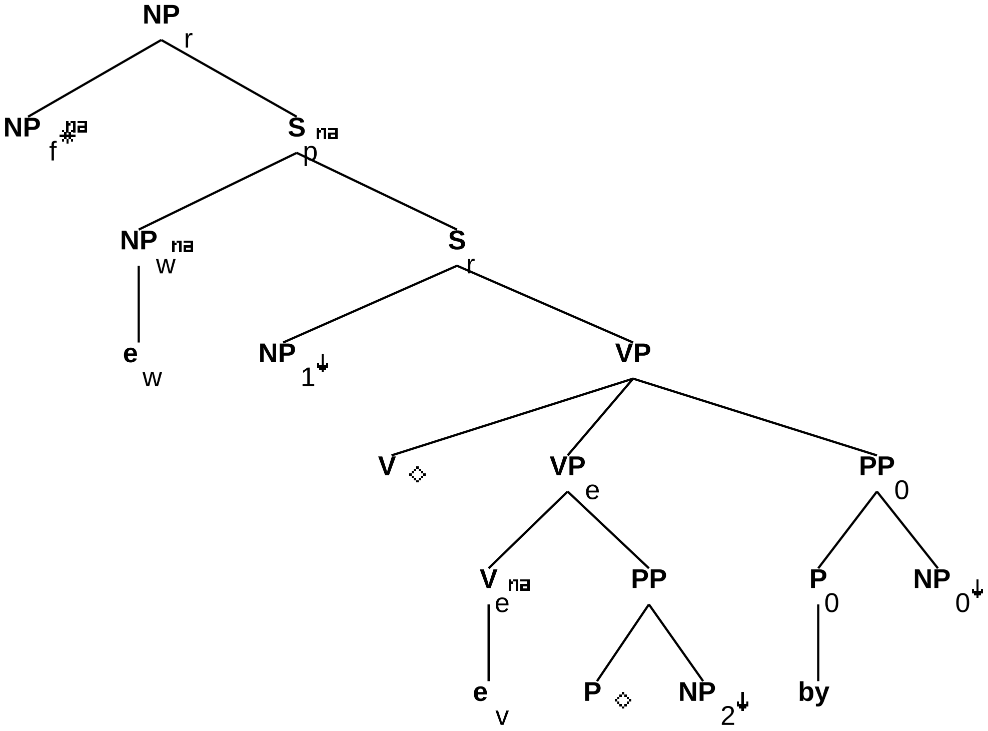
no comments

41.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<control> = NP_0.t:<control>
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>
S_r.t:<inv> = -
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>

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>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

42 Tree "betaNcnx1VPnx2bynx0"

42.1 graphe



42.2 comments

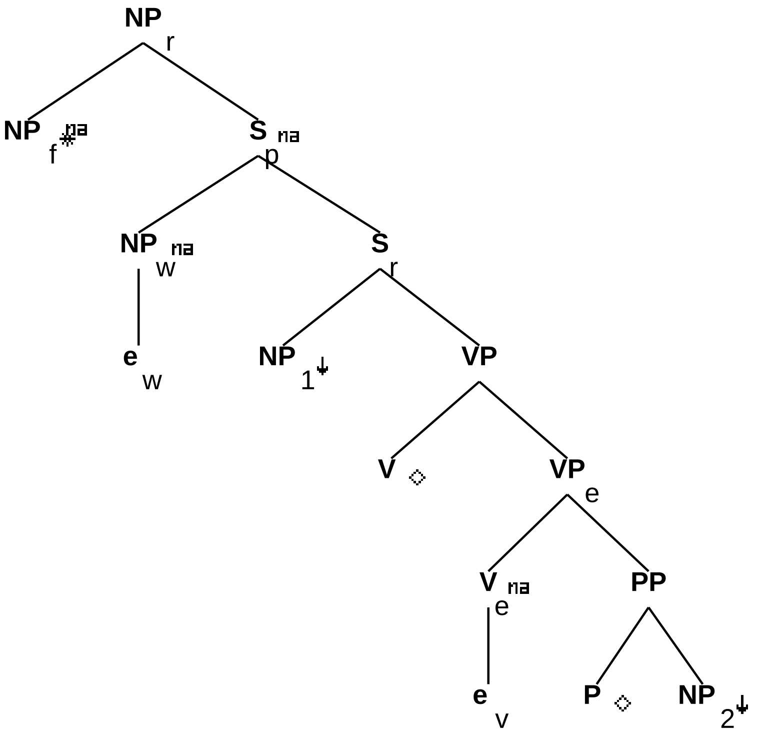
```
Passive:
the poodle was put in the oven by Max
```

42.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<control> = NP_1.t:<control>
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>
S_r.t:<inv> = -
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

43 Tree "betaNcnx1VPnx2"

43.1 graphe



43.2 comments

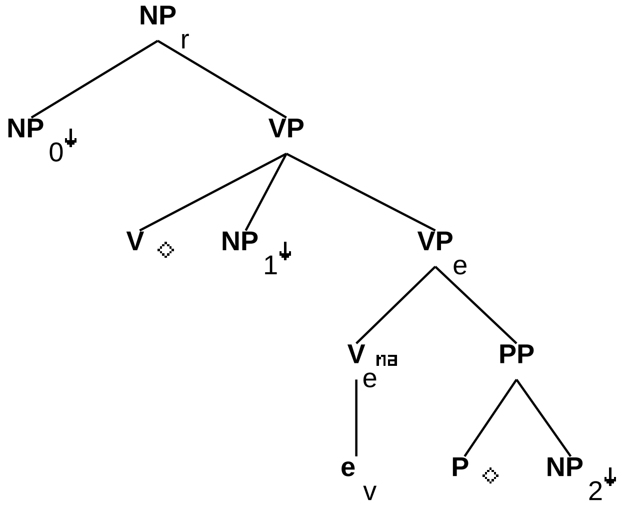
```
Passive w/out by-phrase:
the poodle was put in the oven
```

43.3 features

```
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:<rel-clause> = +
NP_r.b:<pron> = NP_f.t:<pron>
NP_f.b:<case> = acc/nom
NP_f.b:<case> = nom/acc

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


S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<control> = NP_1.t:<control>
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>
S_r.t:<inv> = -
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

44 Tree "alphaGnx0Vnx1Pnx2"

44.1 graphe



44.2 comments

```
Multi Anchor Ditransitive NP gerund with PP:

John did not approve of [Mary('s) putting her reputation on the line].
```

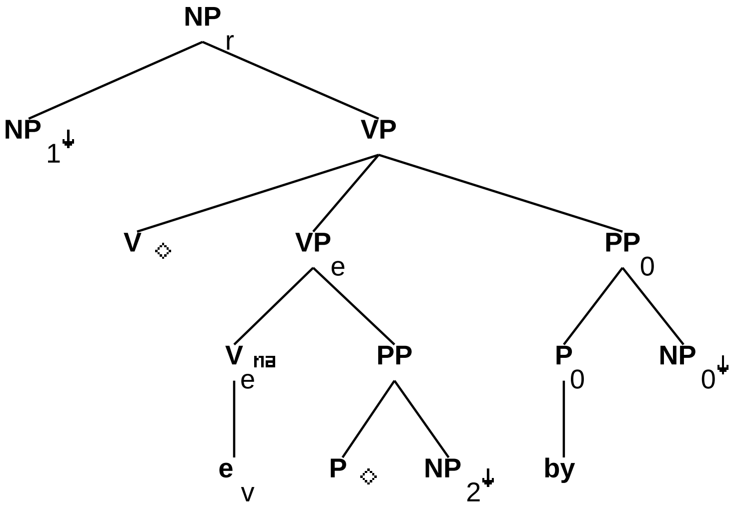
44.3 features

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


NP_r.b:<gerund> = +

VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<wh> = NP_r.b:<wh>
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
```

```
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
NP_0:<case> = acc/gen
```

45 Tree ”alphaGnx1VPnx2bynx0”

45.1 graphe



45.2 comments

Multi Anchor Ditransitive (with PP) gerund passive with the ’by’ phrase:

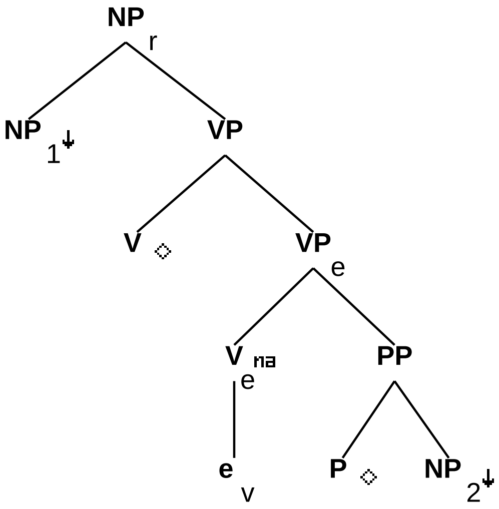
John feared [Mary(’s) reputation being put on the line by the reporters].

45.3 features

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


NP_r.b:<gerund> = +

VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<case> = PP_0.b:<assign-case>
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
V.t:<mode> = ppart
V.t:<passive> = +
NP_1:<case> = acc/gen
```

46 Tree ”alphaGnx1VPnx2”

46.1 graphe



46.2 comments

Multi Anchor Ditransitive (with PP) gerund passive without the ’by’ phrase:

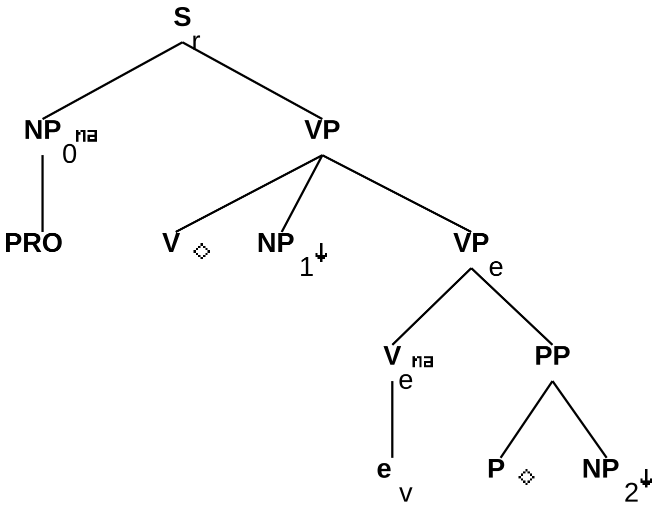
John feared [Mary(’s) reputation being put on the line].

46.3 features

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


NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<wh> = NP_r.b:<wh>
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
NP_1:<case> = acc/gen
```

47 Tree ”alphanx0Vnx1Pnx2-PRO”

47.1 graphe



47.2 comments

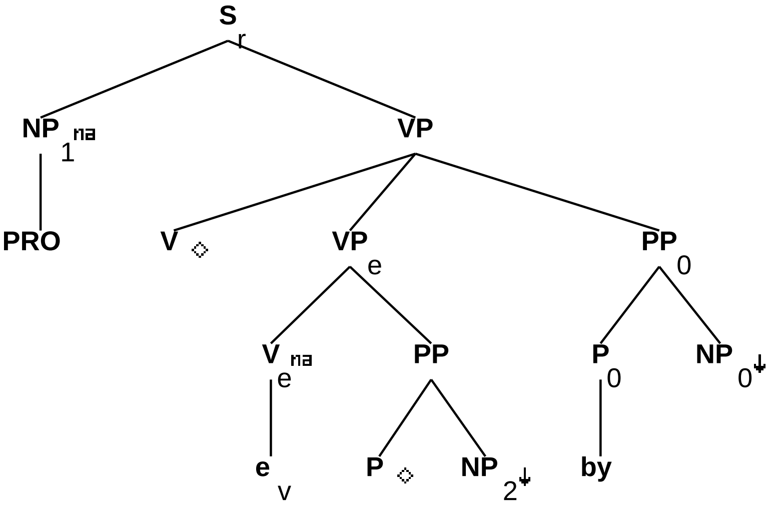
```
Multi Anchor Ditransitive with PP
w/ PRO subject

He didn’t want [PRO to put his reputation on the line].
While [PRO giving the cheese to John] Jack tripped.
```

47.3 features

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

NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = -
NP_0.t:<case> = none
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
VP.t:<mode> = inf/ger
```

48 Tree "alphanx1VPnx2bynx0-PRO"

48.1 graphe



48.2 comments

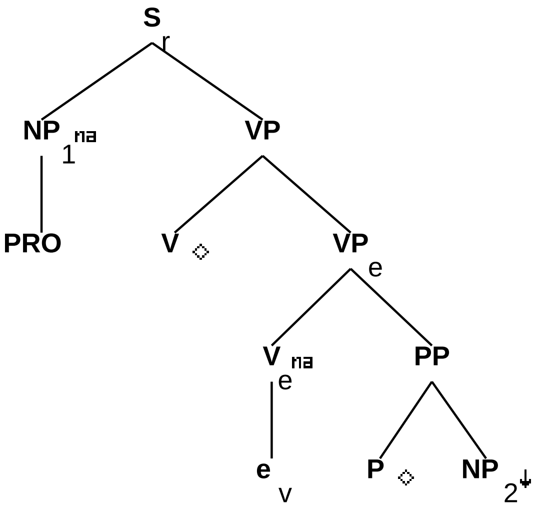
```
Multi Anchor Ditransitive with PP
Passive w/ by-phrase, w/ PRO subject

The poodle didn't want [PRO to be put in the oven by Max].
While [PRO being given to the customer by the clerk] the parrot flew away.
```

48.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<wh> = NP_1:<wh>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.b:<control> = NP_1.t:<control>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<wh> = -
NP_1.t:<case> = none
NP_2:<case> = PP.b:<assign-case>
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
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
VP.t:<mode> = inf/ger
```

49 Tree "alphanx1VPnx2-PRO"

49.1 graphe



49.2 comments

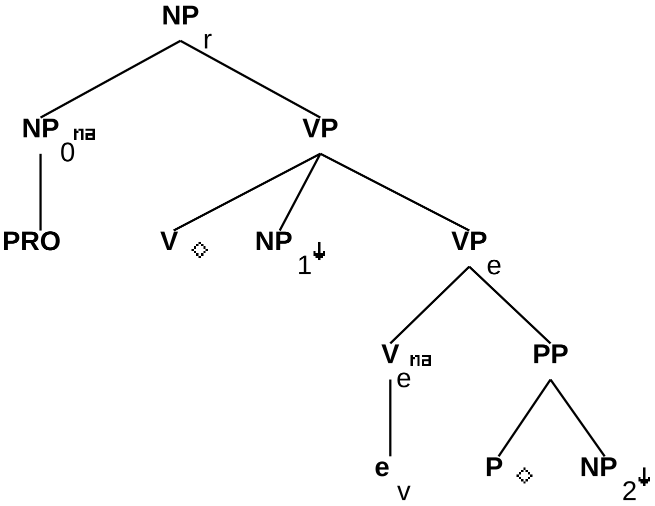
```
Multi Anchor Ditransitive with PP
Passive w/out by-phrase, w/ PRO subject

The poodle didn't want [PRO to be put in the oven].
While [PRO being given to the customer] the parrot flew away.
```

49.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<wh> = NP_1:<wh>
S_r.b:<assign-case> = NP_1.t:<case>
S_r.b:<control> = NP_1.t:<control>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<mode> = VP.t:<mode>
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>
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_1:<agr> = S_r.b:<agr>
NP_1:<wh> = -
NP_1.t:<case> = none
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
VP.t:<mode> = inf/ger
```

50 Tree "alphaGnx0Vnx1Pnx2-PRO"

50.1 graphe



50.2 comments

```
Multi Anchor Ditransitive NP gerund with PP
w/ PRO subject

John did not support [PRO putting his reputation on the line].
```

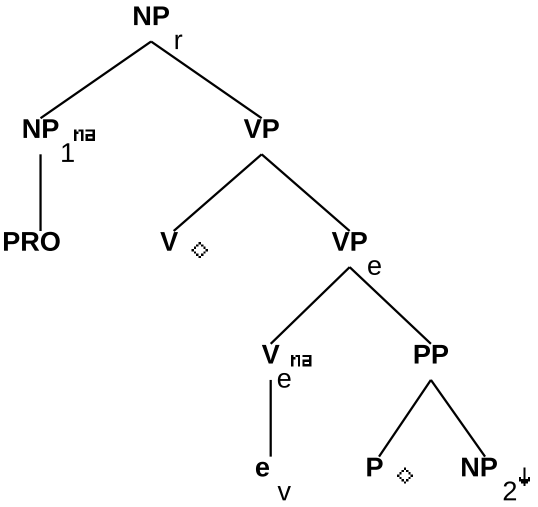
50.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

NP_0:<wh> = NP_r.b:<wh>
NP_0.t:<case> = none
NP_0.t:<wh> = -
NP_1:<case> = acc
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
```

```
PP.b:<wh> = NP_2:<wh>
V.t:<passive> = -
```

51 Tree ”alphaGnx1VPnx2-PRO”

51.1 graphe



51.2 comments

```
Multi Anchor Ditransitive with PP
Gerund passive without the by-phrase
w/ PRO subject

The poodle feared [PRO being put in the oven].
```

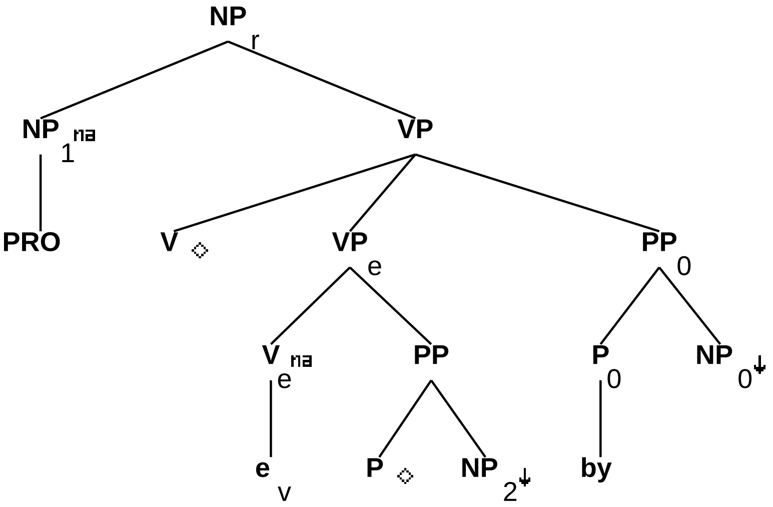
51.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none
NP_1.t:<case> = none
NP_1.t:<wh> = -
```

```
NP_1:<wh> = NP_r.b:<wh>
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
V.t:<mode> = ppart
V.t:<passive> = +
```

52 Tree ”alphaGnx1VPnx2bynx0-PRO”

52.1 graphe



52.2 comments

```
Multi Anchor Ditransitive with PP
Gerund passive with the 'by' phrase
w/ PRO subject

The poodle feared [PRO being put in the oven by the cat].
```

52.3 features

```
NP_r.b:<wh> = NP_1:<wh>
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none
NP_1.t:<wh> = -
NP_1.t:<case> = none
NP_0:<case> = PP_0.b:<assign-case>
NP_2:<case> = PP.b:<assign-case>
P.t:<assign-case> = PP.b:<assign-case>
PP.b:<wh> = NP_2:<wh>
PP_0.b:<wh> = NP_0:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
V.t:<mode> = ppart
V.t:<passive> = +
```