F. Experiment with LFG

F.1. Verb lexicon

```
běhat   V at @(INTRANSREFLPASSKDE běhat).
boj     V bat { @(RFLSENOPASS bát_se) | @(2RFLSENOPASS bát_se)
                | @(IRFLSENOPASSSUBJ bát_se) }.
dokázat V at @(IRFLPASSSUBJ dokázat).
dopátrat        V at @(2RFLSENOPASS dopátrat_se).
dychtit V it @(IRFLPASSSUBJ dychtit).
hatit   V it @(TRANSRFLPERPASSDAT hatit).
hnízdit V it @(INTRANSREFLPASSKDE hnízdit).
chápat  V at @(SRFLPERPASSZE chápat).
chránit V it @(2RFLSENOPASS chránit_se).
chvátat V at @(INTRANSREFLPASSKAM chvátat).
kazit   V it @(RFLSENOPASS kazit_se).
konfiskovat     V ovat { @(TRANSRFLPERPASSDAT konfiskovat)
                | @(TRANSRFLPERPASS konfiskovat) }.
křížit  V it @(TRANSRFLPERPASSDAT křížit).
lovit   V it { @(TRANSRFLPERPASS lovit) | @(INTRANSREFLPASS lovit) }.
muset   V et @(MODALRAISESUBJ muset).
nabaštit        V it @(2RFLSENOPASS nabaštit_se).
nacvičit        V it @(RFLSINOPASSACC nacvičit_si).
nachytat        V at @(2RFLPERPASS nachytat).
nalovit V it @(2RFLPERPASS nalovit).
nastěhovat      V ovat @(RFLSENOPASSKAM nastěhovat_se).
nažrat  V at @(2RFLSENOPASS nažrat_se).
plánovat        V ovat @(RFLSINOPASSACC plánovat_si).
popírat V at @(SRFLPERPASSZE popírat).
poputovat       V ovat @(INTRANSREFLPASSKAM poputovat).
pršet   V et @(METEOVERB pršet).
přikazovat      V ovat @(IRFLPERPASSOBJ přikazovat).
přinutit        V it @(IRFLPERPASSOBJ přinutit).
relaxovat       V ovat @(INTRANSREFLPASSKDE relaxovat).
schovávat       V at @(RFLSENOPASSKDE schovávat_se).
vodit   V it { @(TRANSRFLPERPASSKAM vodit) | @(TRANSRFLPERPASS vodit) }.
vyhazovat       V ovat { @(TRANSRFLPERPASSKAM vyhazovat)
                | @(TRANSRFLPERPASS vyhazovat) }.
```

```
vyšplhat        V at @(RFLSENOPASSKAM vyšplhat_se).
zabydlet        V et @(RFLSENOPASSKDE zabydlet_se).
zapomínat       V at @(IRFLPASSSUBJ zapomínat).
```

F.2. Templates

```
TRANSRFLPERPASS(P) =
        @(LR-TRANSRFLPERPASS (^ PRED)=’P<(^ SUBJ)(^ OBJ)>’).
TRANSRFLPERPASSDAT(P) =
        @(LR-TRANSRFLPERPASS {(^ PRED)=’P<(^ SUBJ)(^ OBJ)(^ OBJ2)>’
                        (^ OBJ2 CASE)=DAT}).
TRANSRFLPERPASSKAM(P) =
        @(LR-TRANSRFLPERPASS {(^ PRED)=’P<(^ SUBJ)(^ OBJ)(^ ACOMP)>’
                        (^ ACOMP SEM)=KAM}).

INTRANSREFLPASS(P) =
        @(LR-INTRANSREFLPASS (^ PRED)=’P<(^ SUBJ)>’).
INTRANSREFLPASSKAM(P) =
        @(LR-INTRANSREFLPASS {(^ PRED)=’P<(^ SUBJ)(^ ACOMP)>’
                        (^ ACOMP SEM)=KAM}).
INTRANSREFLPASSKDE(P) =
        @(LR-INTRANSREFLPASS {(^ PRED)=’P<(^ SUBJ)(^ ACOMP)>’
                        (^ ACOMP SEM)=KDE}).

RFLSENOPASS(P) =
        @(LR-RFLSENOPASS (^ PRED)=’P<(^ SUBJ)>’).
RFLSENOPASSKAM(P) =
        @(LR-RFLSENOPASS {(^ PRED)=’P<(^ SUBJ)(^ ACOMP)>’
                        (^ ACOMP SEM)=KAM}).
RFLSENOPASSKDE(P) =
        @(LR-RFLSENOPASS {(^ PRED)=’P<(^ SUBJ)(^ ACOMP)>’
                        (^ ACOMP SEM)=KDE}).

RFLSINOPASSACC(P) =
        @(LR-RFLSINOPASS {(^ PRED)=’P<(^ SUBJ)(^ OBJ)>’
                        (^ OBJ CASE)=ACC}).

2RFLSENOPASS(P) =
        @(LR-RFLSENOPASS {(^ PRED)=’P<(^ SUBJ)(^ OBJ)>’
                        (^ OBJ CASE)=GEN}).
2RFLPERPASS(P) =
        @(LR-RFLPERPASS {(^ PRED)=’P<(^ SUBJ)(^ OBJ)>’
                        (^ OBJ CASE)=GEN}).

IRFLPERPASSOBJ(P) =
        @(LR-TRANSRFLPERPASS {(^ PRED)='P<(^ SUBJ)(^ OBJ)(^ XCOMP)>'
                        (^ XCOMP SUBJ)=(^ OBJ)
                        (^ XCOMP INF)=c +}).
IRFLPASSSUBJ(P) =
        @(LR-RFLPASS {(^ PRED)='P<(^ SUBJ)(^ XCOMP)>'
                        (^ XCOMP SUBJ)=(^ SUBJ)
                        (^ XCOMP INF)=c +}).

IRFLSENOPASSSUBJ(P) =
        @(LR-RFLSENOPASS {(^ PRED)='P<(^ SUBJ)(^ XCOMP)>'
                        (^ XCOMP SUBJ)=(^ SUBJ)
                        (^ XCOMP INF)=c +}).

SRFLPERPASSZE(P) =
        @(LR-RFLPERPASS {(^ PRED)='P<(^ SUBJ)(^ SCOMP)>'
                        (^ SCOMP CONJ)=že}).

MODALRAISESUBJ(P) =
        { (^ PRED)='P<(^ XCOMP)>(^ SUBJ)'
                (^ XCOMP SUBJ)=(^ SUBJ)
        | (^ PRED)='P<(^ XCOMP)>'
                ~(^ XCOMP SUBJ) }
        (^ XCOMP INF)=c +
        ~(^ REFL).

METEOVERB(P) =
        { (^ PRED)='P<NULL>'
        | (^ PRED)='P<NULL (^ ACOMP)>'
                (^ ACOMP SEM)=KDE}
        ~(^ REFL).
```

F.3. Lexical rules

```
LR-TRANSRFLPERPASS(SCHEMATA) =
        { SCHEMATA
        (^ OBJ CASE)=ACC
        ~(^ REFL)
        |SCHEMATA
        (^ REFL)=c SE
        (^ OBJ)->(^ SUBJ)
        (^ OBJ CASE)=NOM
        (^ SUBJ)->NULL
        |SCHEMATA
        (^ OBJ)->(^ SUBJ)
        (^ OBJ CASE)=NOM
        (^ SUBJ)->NULL
        ~(^ REFL)
        ~(^ TENSE)
        (^ PARTICIPLE)=c PASS }.

LR-INTRANSREFLPASS(SCHEMATA) =
        { SCHEMATA
        ~(^ REFL)
        |SCHEMATA
        (^ REFL)=c SE
        (^ SUBJ)->NULL
        (^ GENDER)=Neut
        (^ NUM)=SG
        (^ PERSON)=3 }.

LR-RFLPERPASS(SCHEMATA) =
        { SCHEMATA
        ~(^ REFL)
        | { SCHEMATA
        (^ REFL)=c SE
        (^ SUBJ)->NULL
        |SCHEMATA
        (^ SUBJ)->NULL
        ~(^ REFL)
        ~(^ TENSE)
        (^ PARTICIPLE)=c PASS }
        (^ GENDER)=Neut
        (^ NUM)=SG
        (^ PERSON)=3 }.

LR-RFLPASS(SCHEMATA) =
        { SCHEMATA
        ~(^ REFL)
        |SCHEMATA
        (^ REFL)=c SE
        (^ SUBJ)->NULL
        (^ GENDER)=Neut
        (^ NUM)=SG
        (^ PERSON)=3 }.

LR-RFLSENOPASS(SCHEMATA) =
        SCHEMATA
        (^ REFL)=c SE .

LR-RFLSINOPASS(SCHEMATA) =
SCHEMATA
(^ REFL)=c SI .
```

F.4. Grammar

```
TEST  CZECH
   NP → A*: (↑ ATTR)=↓;
        N.

   PP → PREP
        NP: (↑ OBJ)=↓.

   S → { { NP: (↑ SUBJ)=↓
               (↑ SUBJ CASE)=NOM
          |VP: (↑ SUBJ)=↓
               (↓ INF)=c +}
         { ADV: ↓∈(↑ ADJ)
          |PP: ↓∈(↑ ADJ)}*
         VP: (↑ PERSON)=(↑ SUBJ PERSON)
             (↑ NUM)=(↑ SUBJ NUM)
             (↑ GENDER)=(↑ SUBJ GENDER)
             (↑ TENSE)
        |{ ADV: { (↑ ACOMP)=↓
                 |↓∈(↑ ADJ)}
          |PP: { (↑ ACOMP)=↓
                |↓∈(↑ ADJ)}}*
         VP: (↑ PERSON)=3
             (↑ NUM)=SG
             (↑ GENDER)=Neut
             (↑ TENSE)}.

   S′ → CONJ
        S.

   VP → (PRTCL)
        V
        { (NP: (↑ OBJ)=↓)
         |(NP: (↑ OBJ2)=↓)
         |ADV: { (↑ ACOMP)=↓
                |↓∈(↑ ADJ)}
         |PP: { (↑ (↓ PCASE))=(↓ OBJ)
               |(↑ ACOMP)=↓
               |↓∈(↑ ADJ)}
         |VP: (↑ XCOMP)=↓
         |S′: (↑ SCOMP)=↓}*.

----
```

Figure F.1.: Simple grammar in LFG

F.5. Test sentences

the following picture shows sentences used for testing the lexicon. The sentences contain only limited vocabulary because a larger vocabulary would also need a large morphological module. As the main point was to show verb frames we consider this small lexicon sufficient.

The numbers in parentheses mean number of analyses, time of processing and number of steps needed for analysis.

```
S: kočky běhají na zahradě (1 0.27 13)
S: na zahradě se běhá (1 0.15 11)
S: kočky se bojí (1 0.08 8)
S: kočky se bojí na zahradě (1 0.31 15)
S: kočky se bojí v noci (1 0.55 15)
S: kočky se bojí Hanky (1 0.24 12)
S: kočky se bojí lovit myšky (1 1.1 18)
S: kočky se bojí lovit v noci (2 3.36 21)
S: kočky dychtí lovit myšky (1 0.25 14)
S: Hanka hatí kočce lov (1 0.13 14)
S: lov se hatí kočce (1 0.13 12)
S: myšky hnízdí v gauči (1 0.2 13)
S: maso se kazí (1 0.05 8)
S: kočka konfiskuje myšce maso (1 0.13 14)
S: maso se konfiskuje myšce (1 0.16 12)
S: kočky musí lovit myšky (1 0.15 16)
S: myšky se musí lovit (1 0.17 12)
S: musí pršet (1 0.06 10)
S: kočky se nabaští masa (1 0.09 12)
S: kočky naloví myšek (1 0.08 10)
S: tady se naloví myšek  (1 0.14 11)
S: kočky zapomínají lovit myšky (1 0.23 16)
S: myšky se zapomínají lovit (1 0.22 11)
S: Hanka chápe že kočky loví (1 0.18 16)
S: Hanka chápe že myšky se loví v noci (2 1.94 30)
S: Hanka chápe že myšky se musí lovit (1 0.72 27)
S: Hanka přinutí myšky chápat že kočky loví (1 0.58 27)
S: kočky se přinutí lovit myšky (1 1.03 18)
S: kočky se vyšplhají na gauč (1 0.41 15)
```

Figure F.2.: Testing sentences

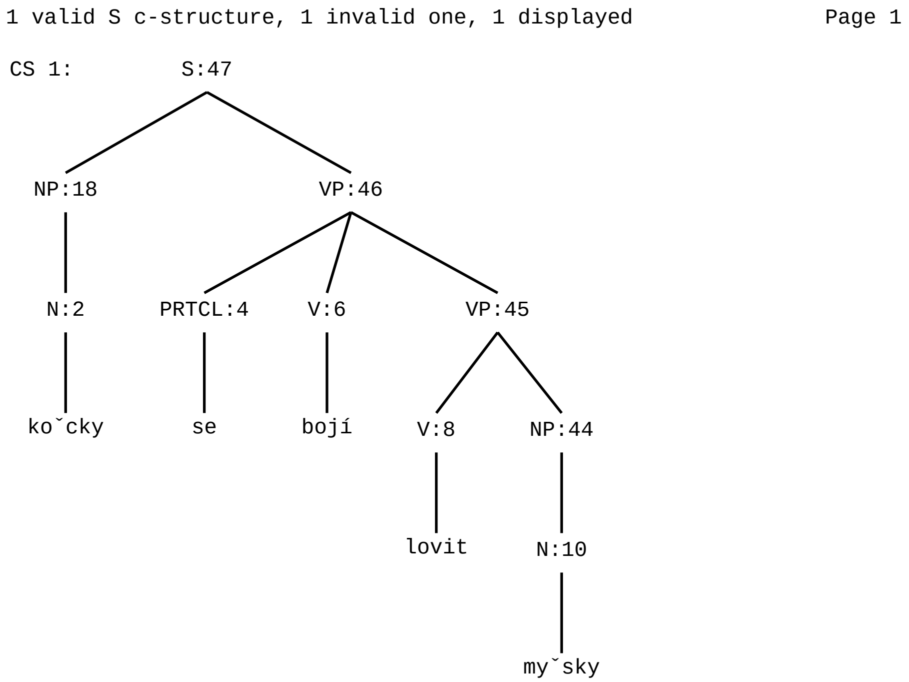
The pictures on the following pages show c-structures and f-structures of these sentences:

(140) a. *Kočky se bojí lovit myšky.*
Cats$_{Nom}$ SE fear hunt$_{Inf}$ mice$_{Acc}$.

b. *Myšky se zapomínají lovit.*
Mice$_{Nom}$ SE forget hunt$_{Inf}$.
'It's beeing forgotten to hunt mice.'

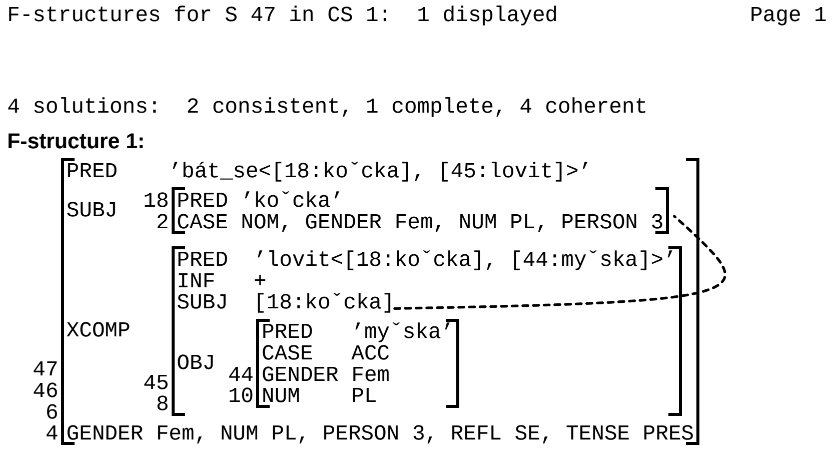
c. *Hanka chápe že myšky se musí lovit.*
Hanka$_{Nom}$ understands that mice$_{Nom}$ SE must hunt$_{Inf}$.
'Hanka understands that mice must be hunted.'

d. *Hanka přinutí myšky chápat že kočky loví.*
Hanka$_{Nom}$ makes mice$_{Acc}$ understand$_{Inf}$ that cats$_{Nom}$ hunt.



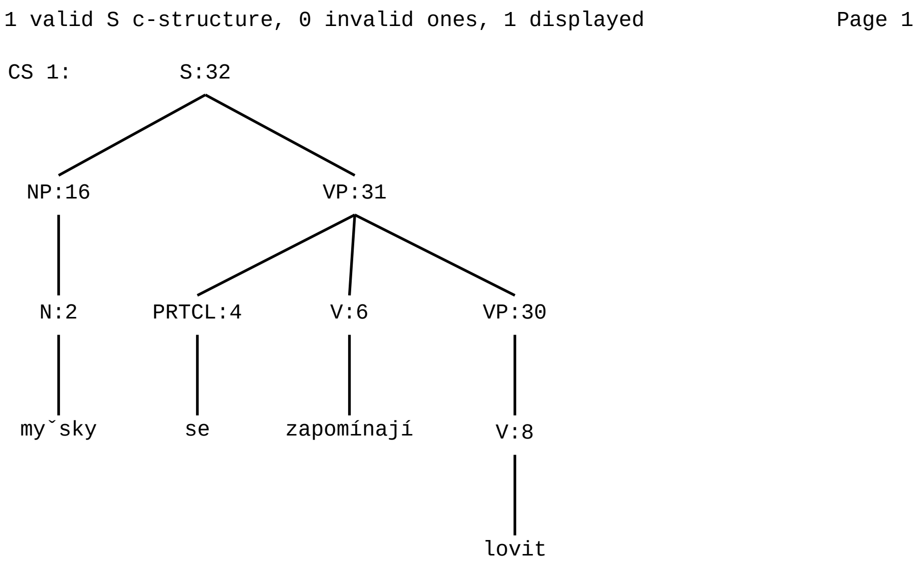
```
1 valid S c-structure, 1 invalid one, 1 displayed              Page 1
```

Figure F.3.: C-structure of sentence 140a



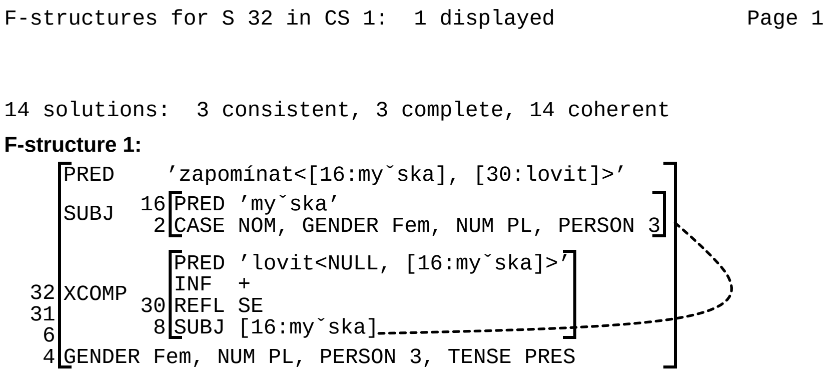
```
F-structures for S 47 in CS 1:  1 displayed              Page 1


4 solutions:  2 consistent, 1 complete, 4 coherent
F-structure 1:
    ┌PRED    'bát_se<[18:kočka], [45:lovit]>'              ┐
    │SUBJ  18┌PRED 'kočka'                                ┐│
    │       2└CASE NOM, GENDER Fem, NUM PL, PERSON 3      ┘│
    │        ┌PRED  'lovit<[18:kočka], [44:myška]>'       ┐│
    │        │INF   +                                     ││
    │        │SUBJ  [18:kočka]                            ││
    │XCOMP   │      ┌PRED   'myška'┐                      ││
    │        │OBJ   │CASE   ACC    │                      ││
  47│        │    44│GENDER Fem    │                      ││
  46│      45│    10└NUM    PL     ┘                      ││
   6│       8└                                            ┘│
   4└GENDER Fem, NUM PL, PERSON 3, REFL SE, TENSE PRES     ┘
```

Figure F.4.: F-structure of sentence 140a



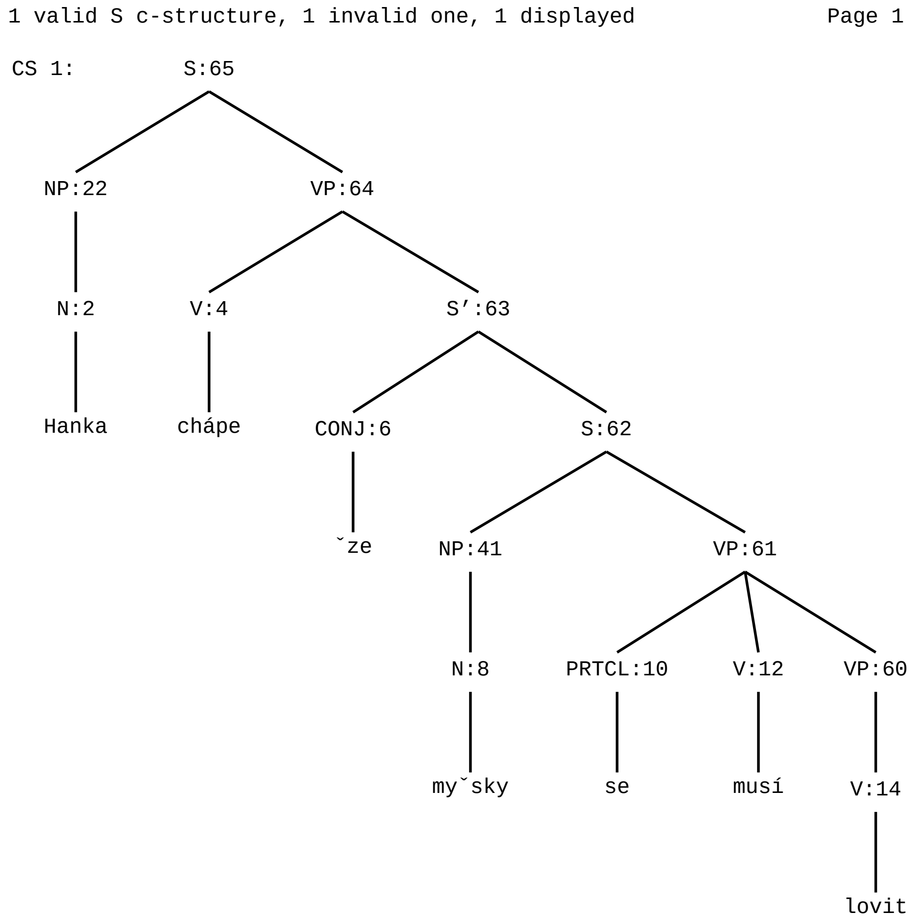
```
1 valid S c-structure, 0 invalid ones, 1 displayed              Page 1
```

Figure F.5.: C-structure of sentence 140b



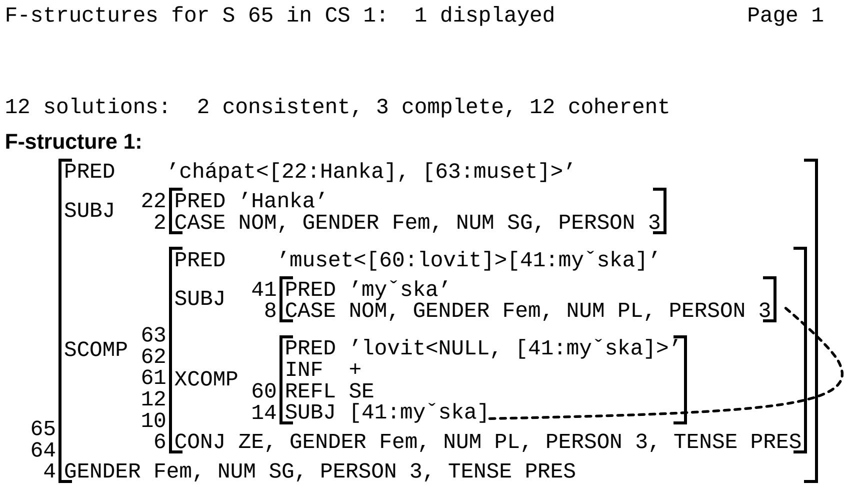
```
F-structures for S 32 in CS 1:  1 displayed              Page 1


14 solutions:  3 consistent, 3 complete, 14 coherent
F-structure 1:
    PRED    'zapomínat<[16:myˇska], [30:lovit]>'
    SUBJ  16 PRED 'myˇska'
           2 CASE NOM, GENDER Fem, NUM PL, PERSON 3

             PRED 'lovit<NULL, [16:myˇska]>'
  32 XCOMP   INF  +
  31      30 REFL SE
   6       8 SUBJ [16:myˇska]
   4 GENDER Fem, NUM PL, PERSON 3, TENSE PRES
```

Figure F.6.: F-structure of sentence 140b



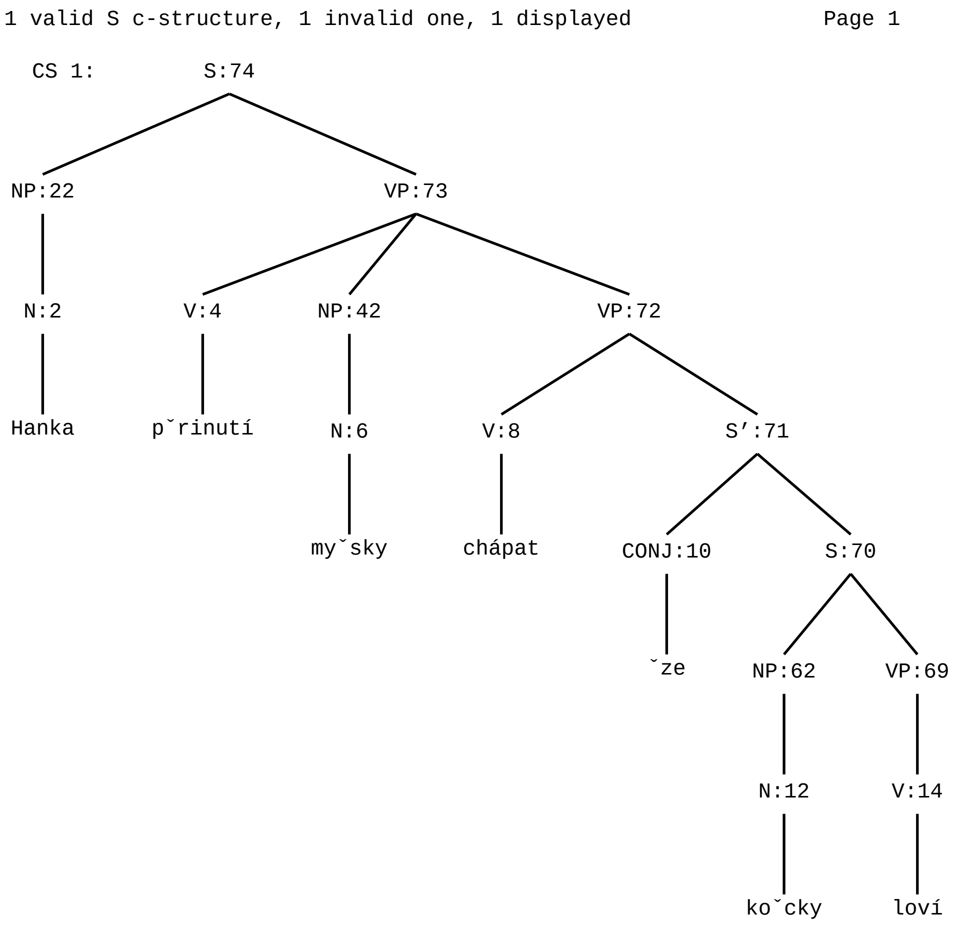
```
1 valid S c-structure, 1 invalid one, 1 displayed            Page 1
```

Figure F.7.: C-structure of sentence 140c



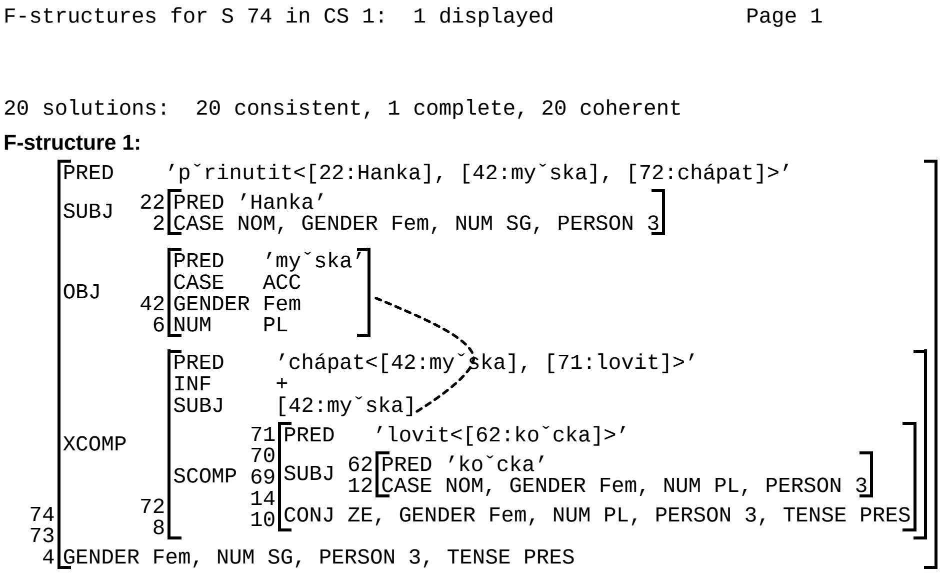
```
F-structures for S 65 in CS 1:  1 displayed            Page 1


12 solutions:  2 consistent, 3 complete, 12 coherent
F-structure 1:
    PRED    'chápat<[22:Hanka], [63:muset]>'
    SUBJ  22 PRED 'Hanka'
           2 CASE NOM, GENDER Fem, NUM SG, PERSON 3
             PRED    'muset<[60:lovit]>[41:myˇska]'
             SUBJ   41 PRED 'myˇska'
                     8 CASE NOM, GENDER Fem, NUM PL, PERSON 3
          63
    SCOMP 62            PRED 'lovit<NULL, [41:myˇska]>'
          61 XCOMP      INF  +
          12         60 REFL SE
  65      10         14 SUBJ [41:myˇska]
  64       6 CONJ ZE, GENDER Fem, NUM PL, PERSON 3, TENSE PRES
   4 GENDER Fem, NUM SG, PERSON 3, TENSE PRES
```

Figure F.8.: F-structure of sentence 140c



```
1 valid S c-structure, 1 invalid one, 1 displayed              Page 1
```

Figure F.9.: C-structure of sentence 140d



```
F-structures for S 74 in CS 1:  1 displayed              Page 1


20 solutions:  20 consistent, 1 complete, 20 coherent
F-structure 1:
      PRED    ’pˇrinutit<[22:Hanka], [42:myˇska], [72:chápat]>’
      SUBJ    22 PRED ’Hanka’
               2 CASE NOM, GENDER Fem, NUM SG, PERSON 3
                 PRED   ’myˇska’
                 CASE   ACC
      OBJ     42 GENDER Fem
               6 NUM    PL
                 PRED    ’chápat<[42:myˇska], [71:lovit]>’
                 INF     +
                 SUBJ    [42:myˇska]
      XCOMP             71 PRED   ’lovit<[62:koˇcka]>’
                        70
                 SCOMP  69 SUBJ  62 PRED ’koˇcka’
                        14       12 CASE NOM, GENDER Fem, NUM PL, PERSON 3
  74          72        10 CONJ ZE, GENDER Fem, NUM PL, PERSON 3, TENSE PRES
  73           8
   4  GENDER Fem, NUM SG, PERSON 3, TENSE PRES
```

Figure F.10.: F-structure of sentence 140d