Converter

Contents

1 Data sets

The converter works with the three following datasets:

- English MRS Test Suite;
- WeScience corpus;
- PEST corpus;
- Deepwoods corpus.

But in general it should be applicable to any dataset which is in the input format (see Section 4).

2 System requirements

Python 2.6 or 2.7. This converter depends on the Python module NLTK[1].

Installation of the NLTK package is not mandatory: it is enough to download the source code and add its location to the PYTHONPATH environment variable, which makes NLTK available from anywhere. Assume, NLTK source code is located in /tmp/nltk:

```
export PYTHONPATH="$PYTHONPATH:/tmp/nltk"
```

To make sure that the package is now available, try:

[1] http://pypi.python.org/pypi/nltk

```
$ python
>>> import nltk
```

NLTK depends on some other packages. The same trick can be use if any of them are missing.

3 Run the converter

Put the files converter.py and erg.hds in the folder where you would like to have the output of the converter. The script will generate a folder “results”.

```
python converter.py --data <path_input> --grammar <grammar_folder_path> \
--dt <dt_folder_path> --dm <dm_folder_path> --dtm <dtm_folder_path> \
--cfg <cfg_folder_path> --cfg_no_unary_rules <true|false> \
--sent_tok <sent_tok_folder_path> --tok <erg|ptb> \
--tex <tex_folder_path>
```

–data a path to a folder that contains the .gz files in the format appropriate for the conversion;

–grammar a path to the folder that contains the relevant ERG grammar;

–dt a path to the folder where the DELPH-IN Syntactic Derivation Tree-derived output will be generated in CoNLLX file;

–dm a path to the folder where the DELPH-IN MRS-Derived output will be generated;

–dtm a path to the folder where the combined representation of DELPH-IN Syntactic Derivation Tree- and DELPH-IN MRS-Derived output will be generated;

–cfg a path to the folder where the phrase-structure tree derived from ERG derivation tree will be generated (can only be used with –tok ERG);

–cfg_no_unary_rules whether to collapse or keep the unary rules of the phrase-structure representation of the ERG derivation tree (can only be used with –tok ERG and specified –cfg option);

–sent_tok tokenized original sentence;

–tok tokenized style (ERG or PTB);

–tex a path to the folder where the LaTEX output will be generated (a graphical representation of the sentences).

The output files will have the name identical to the input folder name and the extensions will be “conll” for DELPH-IN Derivation Tree, “pas” for DELPH-IN MRS-Derived and “tex” for LaTEX files.

4 Input format

Each sentence should be in a separate .gz file and contain the following representations:

1. the sentence (see Section 4.1)
2. PTB-style tokenization of the sentence (see Section 4.2)
3. ERG-style tokenization of the sentence (optional)
4. Derivation Tree (see Section 4.3)
5. MRS (optional)
6. Elementary Dependency Structure (see Section 4.4)

The representations should be ordered as they are listed above and separated by empty line(s).

The script doesn’t work with ERG-style tokenization representation and with MRS representation so these representations can be omitted in the input file.

4.1 The sentence representation

The line containing the sentence should confront the following regular expression (Python syntax):

```
[^‘]+‘(.*)\’(?: \[(.*)\])?$
```

Example for English MRS Test Suite and WeScience corpus:

```
[11] (1 of 1) {1} ‘It rained.’ []
```

Example for PEST corpus:

```
[20209013] (1 of 1) {1} ‘((S (NP-SBJ (DT A) (JJ similar)
(NN technique)) (VP (VBZ is) (ADJP-PRD (RB almost)
(JJ impossible) (SBAR (WHNP-1 (-NONE- 0))
(S (NP-SBJ (-NONE- *)) (VP (TO to) (VP (VB apply)
(NP (-NONE- *T*-1)) (PP-CLR (TO to) (NP (NP (JJ other)
(NNS crops)) (, ,) (PP (JJ such) (IN as) (NP (NN cotton)
(, ,) (NNS soybeans) (CC and) (NN rice)))))))))) (. .)))’
[A similar technique is almost impossible to apply to
other crops, such as cotton, soybeans and rice.]
```

This regular expression says the following:

1. the first part of the line is some text which doesn’t contain symbol `‘`:

   ```
   [^‘]+
   ```

   Example: `[11] (1 of 1) {1}`

2. the second part of the line is a text between symbols `‘` and `’`:

```
‘(.*)\’
```

Example for English MRS Test Suite and WeScience corpus:

```
‘It rained.’
```

Example for PEST corpus:

```
‘((S (NP-SBJ (DT A) (JJ similar) (NN technique))
(VP (VBZ is) (ADJP-PRD (RB almost) (JJ impossible)
(SBAR (WHNP-1 (-NONE- 0))
(S (NP-SBJ (-NONE- *)) (VP (TO to) (VP (VB apply)
(NP (-NONE- *T*-1)) (PP-CLR (TO to)
(NP (NP (JJ other) (NNS crops)) (, ,) (PP (JJ such)
(IN as) (NP (NN cotton) (, ,) (NNS soybeans)
(CC and) (NN rice)))))))))) (. .)))’
```

3. the third part of the line is an optional text in square brackets:

   ```
   (?: \[(.*)\])?$
   ```

   Example for PEST corpus:

   ```
   [A similar technique is almost impossible to apply to other
   crops, such as cotton, soybeans and rice.]
   ```

4.2 PTB tokenization

This part starts with the line:

```
<
```

and ends with the line:

```
>
```

Example for English MRS Test Suite:

```
<
  (1, 0, 1, <0:2>, 1, "It", 0, "null")
  (2, 1, 2, <3:9>, 1, "rained", 0, "null")
  (3, 2, 3, <9:10>, 1, ".", 0, "null")
>
```

Example for WeScience and PEST corpora:

```
<
 (1, 0, 1, <9:15>, 1, "Google", 0, "null", "NNP" 0.8791 "NN" 0.0759
"JJ" 0.045)
>
```

WeScience and PEST corpora contain the column with assigned PoS tags and their probabilities.

4.3 Derivation Tree

The derivation tree is detected by the script if it starts with one of the following lines:

```
(ROOT_STRICT

(ROOT_INFORMAL

(ROOT_FRAG

(ROOT_INFFRAG
```

Example of the full derivation tree for the sentence *It rained.*:

```
(ROOT_STRICT
 (90 SB-HD_MC_C 0.708895 0 2
  (18 it/n_-_pr-it-x_le 0.69661 0 1
   ("it" 17
    "token [ +CARG \"It\" +CLASS alphabetic [ +CASE
capitalized+lower +INITIAL + ] +FORM \"it\"
+FROM \"0\" +ID *diff-list* [ LAST #1=*top*
LIST *cons* [ FIRST \"1\" REST #1 ] ] +PRED predsort
+TNT null_tnt [ +MAIN tnt_main +PRBS *null*
+TAGS *null* ] +TO \"2\" +TRAIT native_trait ]"))
  (89 W_PERIOD_PLR 0.0647273 1 2
   (88 V_PST_OLR 0 1 2
    (31 rain_v1/v_-_it_le 0 1 2
     ("rained." 16
      "token [ +CARG \"rained\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM \"rained.\" +FROM \"3\" +ID *diff-list*
[ LAST #1=*list* LIST *cons* [ FIRST \"2\"
REST *cons* [ FIRST \"3\" REST #1 ] ] ] +PRED predsort
+TNT tnt [ +MAIN tnt_main +PRBS *null* +TAGS *null* ]
+TO \"10\" +TRAIT token_trait ]"))))))
```

The script uses:

- words in the leaves as tokens (e.g. `rained.`), ERG tokenization;
- lexical entries as lemmas (e.g. `rain_v1`);
- lexical rules as part-of-speech tags (e.g. `v_-_it_le`);
- information contained in `+FROM` and `+TO` properties as unique indexes of the tokens that are related to the indexes in PTB tokenization section (e.g. `+FROM \"3\" +TO \"10\"` is a unique index of the token *rained.* in ERG tokenization that corresponds to the indexes `<3:9>` (*rained*) and `<9:10>` (.) in PTB tokenization section.

4.4 Elementary Dependency Structures

The script recognizes the Elementary Dependency Structure (EDS) for the sentence if the regular expression below (Python syntax) can detect it:

```
re.match('^\{.+',line) and
(line[-1] == ':' or re.search('\:\s+\(fragmented\)', line))
```

Example of EDS:

```
{e2:
 e2:_rain_v_1<3:10>[]
}
```

5 Output

The script produces the following output:

1. bilexical dependencies extracted from the Derivation Tree - **DELPH-IN Syntactic Derivation Tree** (see Section 5.1);
2. bilexical dependencies extracted from the Elementary Dependency Structure - **DELPH-IN MRS-derived Dependencies** (see Section 5.2).

   It is output in the data description format similar to the representation of PAS generated by the Enju parser. However, the analyzis is different and therefore the dependencies are not the same as the ones produced by Enju.
3. Graphical representation in LaTEX of the dependencies extracted from the Derivation Tree (in red color) and from the Elementary Dependency Structure (in blue color).

All three outputs are generated for both ERG and PTB tokenization styles.

5.1 DELPH-IN Syntactic Derivation Tree

This output is in data description format similar to CoNLLX[2], but the analyzis differs substantially therefore the dependencies between the words are not the same as they would be in CoNLL.

Table 1 describes the fields in this output format.

5.2 DELPH-IN MRS-derived Dependencies

This output is in the data description format similar to the representation of PAS generated by the Enju parser. However, the analyzis is different and therefore the dependencies are not the same as the ones produced by Enju. Table 2 provides the details.

[2] `http://ilk.uvt.nl/conll/#dataformat`

Field Number	**Field Description**
1	Token counter, starting at 1 for each new sentence.
2	Word form or punctuation symbol (e.g. *company*)
3	Lexical entry from the Derivation Tree (e.g. *company_n1*) that represents lemma
4	Lexical rule from the Derivation Tree (e.g. *n_pp_mc-of_le*) that represents gold PoS
5	For ERG tokenization it is the same lexical rule as in field 4 (e.g. *n_pp_mc-of_le*), while for PTB tokenization it is a PTB PoS tag (e.g. NN)
6	_
7	Head index
8	Dependency relation
9	_
10	_

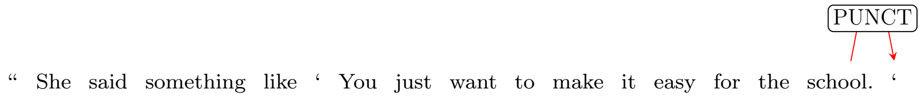
Table 1: Fields in the output format DELPH-IN Syntactic Derivation Tree

Field Number	**Field Description**
1	Word form or punctuation symbol of the head (e.g. *largest*).
2	Lexical entry from the Derivation Tree (e.g. *largest_a1*) that represents lemma of the head.
3	The field contains the lexical rule from the Derivation Tree (e.g. *aj_-_i-sup_le*) that represents gold PoS of the head.
4	For ERG tokenization the field is identical to the field 3 and contains the lexical rule for the head (e.g. *aj_-_i-sup_le*). For PTB tokenization it is a PTB PoS tag for the head (e.g. *JJS*)
5	Index of the word form of the head counting from 0 for each sentence (e.g. *10*).
6	Dependency relation (e.g. *ARG1*)
7	The same dependency relation as in the field 6 (e.g. *ARG1*)
8	Word form or punctuation symbol of the dependent (e.g. *largest*).
9	Lexical entry from the Derivation Tree (e.g. *largest_a1*) that represents lemma of the dependent.
10	Lexical rule from the Derivation Tree (e.g. *aj_-_i-sup_le*) that represents gold PoS of the dependent.
11	For ERG tokenization it is the same lexical rule for the dependent as in the field 3 (e.g. *aj_-_i-sup_le*), while for PTB tokenization it is a PTB PoS tag for the dependent (e.g. *JJS*)
12	Index of the word form of the dependent counting from 0 for each sentence (e.g. *10*).

Table 2: Fields in the output format DELPH-IN MRS-derived Dependencies

6 Special cases

There are several sentences in the current version of Deepbank where the final punctuation of the original sentence is missing from the derivation tree when it is not part of the sentence semantically. In this case for the DELPH-IN Syntactic Derivation Tree analysis we create a dependency from the nearest word to the left of the punctuation symbol, as it is shown in the example below. The opening quote at the end of the sentence is not semantically part of the sentence and is missing from the derivation tree. It was mostlikely a typo in the original text and was meant to be a closing quote.


PUNCT
“ She said something like ‘ You just want to make it easy for the school. ‘

20044110.gz

A. Input file example from English MRS Test Suite

```
[11] (1 of 1) {1} ‘It rained.’ []

[11:0] (active)

<
  (1, 0, 1, <0:2>, 1, "It", 0, "null")
  (2, 1, 2, <3:9>, 1, "rained", 0, "null")
  (3, 2, 3, <9:10>, 1, ".", 0, "null")
>

<
  (17, 0, 1, <0:2>, 1, "it", 0, "null")
  (16, 1, 2, <3:10>, 1, "rained.", 0, "null")
>

(ROOT_STRICT
 (90 SB-HD_MC_C 0.708895 0 2
  (18 it/n_-_pr-it-x_le 0.69661 0 1
   ("it" 17
    "token [ +CARG \"It\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL + ] +FORM \"it\"
+FROM \"0\" +ID *diff-list* [ LAST #1=*top* LIST *cons*
[ FIRST \"1\" REST #1 ] ]
+PRED predsort +TNT null_tnt
[ +MAIN tnt_main +PRBS *null* +TAGS *null* ]
+TO \"2\" +TRAIT native_trait ]"))
  (89 W_PERIOD_PLR 0.0647273 1 2
   (88 V_PST_OLR 0 1 2
    (31 rain_v1/v_-_it_le 0 1 2
     ("rained." 16
      "token [ +CARG \"rained\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM \"rained.\" +FROM \"3\" +ID *diff-list*
[ LAST #1=*list* LIST *cons* [ FIRST \"2\"
REST *cons* [ FIRST \"3\" REST #1 ] ] ]
+PRED predsort +TNT tnt
[ +MAIN tnt_main +PRBS *null* +TAGS *null* ]
+TO \"10\" +TRAIT token_trait ]"))))))


 [ LTOP: h1
   INDEX: e2
[ e SF: PROP TENSE: PAST MOOD: INDICATIVE PROG: - PERF: - ]
```

```
   RELS: <
          [ "_rain_v_1_rel"<3:10>
            LBL: h3
            ARG0: e2 ] >
   HCONS: < > ]

{e2:
 e2:_rain_v_1<3:10>[]
}
```

B. Input file example from WeScience corpus

```
[10320030] (1 of 1) {1} ‘Google’s headquarters, the [[Googleplex]],
is located in [[Mountain View, California]].’

[10320030:0] (active)

<
  (1, 0, 1, <0:6>, 1, "Google", 0, "null", "NNP" 1.0)
  (2, 1, 2, <6:8>, 1, "’s", 0, "null", "POS" 1.0)
  (3, 2, 3, <9:21>, 1, "headquarters", 0, "null",
"NN" 0.8271 "NNS" 0.1729)
  (4, 3, 4, <21:22>, 1, ",", 0, "null", "," 1.0)
  (5, 4, 5, <23:26>, 1, "the", 0, "null", "DT" 1.0)
  (6, 5, 6, <29:39>, 1, "Googleplex", 0, "null",
"NNP" 0.6082 "JJ" 0.3053 "NN" 0.0865)
  (7, 6, 7, <41:42>, 1, ",", 0, "null", "," 1.0)
  (8, 7, 8, <43:45>, 1, "is", 0, "null", "VBZ" 1.0)
  (9, 8, 9, <46:53>, 1, "located", 0, "null", "VBN" 1.0)
  (10, 9, 10, <54:56>, 1, "in", 0, "null", "IN" 1.0)
  (11, 10, 11, <59:67>, 1, "Mountain", 0, "null", "NNP" 1.0)
  (12, 11, 12, <68:72>, 1, "View", 0, "null", "NNP" 1.0)
  (13, 12, 13, <72:73>, 1, ",", 0, "null", "," 1.0)
  (14, 13, 14, <74:84>, 1, "California", 0, "null", "NNP" 1.0)
  (15, 14, 15, <86:87>, 1, ".", 0, "null", "." 1.0)
>

<
  (134, 0, 1, <0:6>, 1, "Google", 0, "null")
  (145, 0, 1, <0:6>, 1, "google", 0, "null")
  (125, 1, 2, <6:8>, 1, "’s", 0, "null")
  (131, 2, 3, <9:22>, 1, "headquarters,", 0, "null")
  (135, 2, 3, <9:22>, 1, "headquarters,", 0, "null", "NN" 0.8271)
  (143, 2, 3, <9:22>, 1, "headquarters,", 0, "null", "NNS" 0.1729)
  (117, 3, 4, <23:26>, 1, "the", 0, "null")
  (101, 4, 5, <29:42>, 1, "Googleplex,", 0, "null")
  (136, 4, 5, <29:42>, 1, "Googleplex,", 0, "null", "NNP" 0.6082)
  (137, 4, 5, <29:42>, 1, "Googleplex,", 0, "null", "JJ" 0.3053)
  (148, 4, 5, <29:42>, 1, "googleplex,", 0, "null")
  (119, 5, 6, <43:45>, 1, "is", 0, "null")
  (139, 5, 6, <43:45>, 1, "is", 0, "null", "VBZ" 1.0)
  (121, 6, 7, <46:53>, 1, "located", 0, "null")
  (140, 6, 7, <46:53>, 1, "located", 0, "null", "VBN" 1.0)
  (123, 7, 8, <54:56>, 1, "in", 0, "null")
  (96, 8, 9, <59:67>, 1, "Mountain", 0, "null")
  (138, 8, 9, <59:67>, 1, "Mountain", 0, "null", "NNP" 1.0)
  (144, 8, 9, <59:67>, 1, "mountain", 0, "null")
  (103, 9, 10, <68:73>, 1, "View,", 0, "null")
  (141, 9, 10, <68:73>, 1, "View,", 0, "null", "NNP" 1.0)
  (146, 9, 10, <68:73>, 1, "view,", 0, "null")
  (105, 10, 11, <74:87>, 1, "California.", 0, "null")
  (142, 10, 11, <74:87>, 1, "California.", 0, "null", "NNP" 1.0)
  (147, 10, 11, <74:87>, 1, "california.", 0, "null")
>

(ROOT_STRICT
 (3657 SB-HD_MC_C 14.3486 0 11
  (3645 HDN-NP_APP-PR_C 3.86603 0 5
   (3641 SP-HD_N_C 5.80677 0 3
    (3638 SP-HD_HC_C 4.51746 0 2
     (3637 HDN_BNP-PN_C 1.40456 0 1
      (3636 N_SG_ILR 0.751608 0 1
       (149 generic_proper_ne/n_-_pn-gen_le 2.34773 0 1
        ("Google" 134
         "token [ +CARG #1=\"Google\" +CLASS proper_ne
+FORM #1 +FROM \"0\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"1\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"NNP\" ]  +PRBS *null* +TAGS *null* ]
+TO \"6\" +TRAIT generic_trait ]"))))
     (151 apostrophe_s_2_lex/n_-_cl-poss_le 1.74971 1 2
      ("’s" 125
       "token [ +CARG #1=\"’s\" +CLASS non_alphanumeric
[ +INITIAL - ] +FORM #1 +FROM \"6\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"2\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"POS\" ] +PRBS *null* +TAGS *null* ]
+TO \"8\" +TRAIT native_trait ]")))
    (3640 W_COMMA_PLR 1.40802 2 3
     (3639 N_SG_ILR 1.27536 2 3
      (206 headquarters_n1/n_-_c_le 0 2 3
       ("headquarters," 131
        "token [ +CARG \"headquarters\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ] +FORM \"headquarters,\"
+FROM \"9\" +ID *diff-list* [ LAST #1=*list* LIST *cons*
[ FIRST \"3\" REST *cons* [ FIRST \"4\" REST #1 ] ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"0.82709109999999997\" +TAG \"NN\" ]
+PRBS *null* +TAGS *null* ] +TO \"22\" +TRAIT native_trait ]")))))
   (3644 SP-HD_N_C -2.13495 3 5
    (238 the_1/d_-_the_le -0.477139 3 4
     ("the" 117
      "token [ +CARG #1=\"the\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"23\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"5\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"DT\" ] +PRBS *null* +TAGS *null* ]
+TO \"26\" +TRAIT native_trait ]"))
    (3643 W_COMMA_PLR -2.20731 4 5
     (3642 N_SG_ILR -2.33997 4 5
      (245 generic_proper_ne/n_-_pn-gen_le 0 4 5
       ("Googleplex," 101
        "token [ +CARG \"Googleplex\" +CLASS proper_ne
+FORM \"Googleplex,\" +FROM \"29\" +ID *diff-list*
[ LAST #1=*list* LIST *cons* [ FIRST \"6\" REST *cons*
[ FIRST \"7\" REST #1 ] ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"0.60819509999999999\" +TAG \"NNP\" ]
+PRBS *null* +TAGS *null* ] +TO \"42\" +TRAIT generic_trait ]"))))))
  (3656 HD-CMP_U_C 9.91443 5 11
   (279 be_c_is/v_prd_is_le 1.66757 5 6
    ("is" 119
     "token [ +CARG #1=\"is\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ] +FORM #1
+FROM \"43\" +ID *diff-list* [ LAST #2=*top* LIST *cons*
[ FIRST \"8\" REST #2 ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"VBZ\" ] +PRBS *null*
+TAGS *null* ] +TO \"45\" +TRAIT native_trait ]"))
   (3655 HD-CMP_U_C 6.07302 6 11
    (3647 HD_OPTCMP_C 0.465143 6 7
     (3646 V_PAS_ODLR 0.0015257 6 7
      (320 locate_v2/v_np-pp_prop_le -0.140226 6 7
       ("located" 121
        "token [ +CARG #1=\"located\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ] +FORM #1
+FROM \"46\" +ID *diff-list* [ LAST #2=*top* LIST *cons*
[ FIRST \"9\" REST #2 ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"VBN\" ] +PRBS *null*
+TAGS *null* ] +TO \"53\" +TRAIT native_trait ]"))))
    (3654 HD-CMP_U_C 4.2633 7 11
     (377 in/p_np_i-reg_le -0.666474 7 8
      ("in" 123
       "token [ +CARG #1=\"in\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"54\" +ID *diff-list* [ LAST #2=*top*
LIST *cons* [ FIRST \"10\" REST #2 ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"IN\" ] +PRBS *null* +TAGS *null* ]
+TO \"56\" +TRAIT native_trait ]"))
```

```
      (3653 HDN_BNP-PN_C 3.86835 8 11
       (3652 NP-HDN_CTY-CPD_C 3.47365 8 11
        (3649 W_COMMA_PLR 0.381275 8 10
         (3648 N_SG_ILR 0.562086 8 10
          (422 mtn_view_n1/n_-_pn-city_le 0 8 10
           ("mountain view," 144
            "token [ +CARG \"Mountain\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL - ] +FORM \"mountain\"
+FROM \"59\" +ID *diff-list* [ LAST #1=*top* LIST *cons*
[ FIRST \"11\" REST #1 ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"NNP\" ] +PRBS *null*
+TAGS *null* ] +TO \"67\" +TRAIT native_trait ]"
            146
            "token [ +CARG \"View\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL - ] +FORM \"view,\"
+FROM \"68\" +ID *diff-list* [ LAST #1=*list* LIST *cons*
[ FIRST \"12\" REST *cons* [ FIRST \"13\" REST #1 ] ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"NNP\" ] +PRBS *null* +TAGS *null* ]
+TO \"73\" +TRAIT native_trait ]"))))
        (3651 W_PERIOD_PLR 1.69983 10 11
         (3650 N_SG_ILR 0.107247 10 11
          (421 california_n1/n_-_pn_le 0 10 11
           ("california." 147
            "token [ +CARG \"California\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL - ] +FORM \"california.\"
+FROM \"74\" +ID *diff-list* [ LAST #1=*list* LIST *cons*
[ FIRST \"14\" REST *cons* [ FIRST \"15\" REST #1 ] ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main [ +PRB \"1\" +TAG \"NNP\" ]
+PRBS *null* +TAGS *null* ] +TO \"87\" +TRAIT native_trait ]")))))))))))


 [ LTOP: h1
   INDEX: e2
[ e SF: PROP TENSE: PRES MOOD: INDICATIVE PROG: - PERF: - ]
   RELS: <
          [ appos_rel<0:42>
            LBL: h3
            ARG0: e6
[ e SF: PROP TENSE: UNTENSED MOOD: INDICATIVE PROG: - PERF: - ]
            ARG1: x5 [ x PERS: 3 NUM: SG IND: + ]
            ARG2: x4 [ x PERS: 3 NUM: SG IND: + ] ]
          [ proper_q_rel<0:6>
            LBL: h7
            ARG0: x9 [ x PERS: 3 NUM: SG IND: + ]
            RSTR: h8
            BODY: h10 ]
          [ named_rel<0:6>
```

```
            LBL: h11
            ARG0: x9
            CARG: "Google" ]
          [ def_explicit_q_rel<6:8>
            LBL: h12
            ARG0: x5
            RSTR: h13
            BODY: h14 ]
          [ poss_rel<6:8>
            LBL: h15
            ARG0: e16
[ e SF: PROP TENSE: UNTENSED MOOD: INDICATIVE PROG: - PERF: - ]
            ARG1: x5
            ARG2: x9 ]
          [ "_headquarters_n_1_rel"<9:22>
            LBL: h15
            ARG0: x5 ]
          [ _the_q_rel<23:26>
            LBL: h17
            ARG0: x4
            RSTR: h19
            BODY: h18 ]
          [ named_rel<29:42>
            LBL: h20
            ARG0: x4
            CARG: "Googleplex" ]
          [ "_locate_v_1_rel"<46:53>
            LBL: h21
            ARG0: e2
            ARG1: i22
            ARG2: x5
            ARG3: h23 ]
          [ parg_d_rel<46:53>
            LBL: h21
            ARG0: e24 [ e SF: PROP ]
            ARG1: e2
            ARG2: x5 ]
          [ _in_p_rel<54:56>
            LBL: h25
            ARG0: e26
[ e SF: PROP TENSE: UNTENSED MOOD: INDICATIVE PROG: - PERF: - ]
            ARG1: x5
            ARG2: x27 [ x PERS: 3 NUM: SG IND: + ] ]
          [ proper_q_rel<59:87>
            LBL: h28
            ARG0: x27
            RSTR: h29
            BODY: h30 ]
          [ compound_name_rel<59:87>
            LBL: h31
            ARG0: e33
[ e SF: PROP TENSE: UNTENSED MOOD: INDICATIVE PROG: - PERF: - ]
            ARG1: x27
            ARG2: x32 [ x PERS: 3 NUM: SG IND: + ] ]
          [ proper_q_rel<59:73>
            LBL: h34
            ARG0: x32
            RSTR: h35
            BODY: h36 ]
          [ named_rel<59:73>
            LBL: h37
            ARG0: x32
            CARG: "Mountain_View" ]
          [ named_rel<74:87>
            LBL: h38
            ARG0: x27
            CARG: "California" ] >
   HCONS: < h35 QEQ h37 h29 QEQ h38 h23 QEQ h25
h19 QEQ h20 h13 QEQ h15 h8 QEQ h11 > ]

{e2:
 e6:appos<0:42>[ARG1 x5, ARG2 x4]
 _1:proper_q<0:6>[BV x9]
 x9:named<0:6>("Google")[]
 _2:def_explicit_q<6:8>[BV x5]
 e16:poss<6:8>[ARG1 x5, ARG2 x9]
 x5:_headquarters_n_1<9:22>[]
 _3:_the_q<23:26>[BV x4]
 x4:named<29:42>("Googleplex")[]
 e2:_locate_v_1<46:53>[ARG2 x5, ARG3 e26]
 e24:parg_d<46:53>[ARG1 e2, ARG2 x5]
 e26:_in_p<54:56>[ARG1 x5, ARG2 x27]
 _4:proper_q<59:87>[BV x27]
 e33:compound_name<59:87>[ARG1 x27, ARG2 x32]
 _5:proper_q<59:73>[BV x32]
 x32:named<59:73>("Mountain_View")[]
 x27:named<74:87>("California")[]
}
```

C. Input file example from PEST corpus and Deepwoods corpus

```
[20202004] (1 of 1) {1} ‘((S (NP-SBJ (NP (NNP Bell)) (, ,)
(VP (VBN based) (NP (-NONE- *)) (PP-LOC-CLR (IN in)
(NP (NNP Los) (NNP Angeles)))) (, ,)) (VP (VBZ makes)
(CC and) (VBZ distributes) (NP (UCP (JJ electronic) (, ,)
(NN computer) (CC and) (NN building))
 (NNS products))) (. .)))’ [Bell, based in Los Angeles,
 makes and distributes electronic, computer and
building products.]

[20202004:2] (active)

<
  (42, 0, 1, <0:1>, 1, "Bell" "Bell", 0, "null", "NNP" 1.0)
  (43, 1, 2, <1:2>, 1, "," ",", 0, "null", "," 1.0)
  (44, 2, 3, <2:3>, 1, "based" "based", 0, "null", "VBN" 1.0)
  (45, 3, 4, <3:4>, 1, "in" "in", 0, "null", "IN" 1.0)
  (46, 4, 5, <4:5>, 1, "Los" "Los", 0, "null", "NNP" 1.0)
  (47, 5, 6, <5:6>, 1, "Angeles" "Angeles", 0, "null", "NNP" 1.0)
  (48, 6, 7, <6:7>, 1, "," ",", 0, "null", "," 1.0)
  (49, 7, 8, <7:8>, 1, "makes" "makes", 0, "null", "VBZ" 1.0)
  (50, 8, 9, <8:9>, 1, "and" "and", 0, "null", "CC" 1.0)
  (51, 9, 10, <9:10>, 1, "distributes" "distributes", 0,
 "null", "VBZ" 1.0)
  (52, 10, 11, <10:11>, 1, "electronic" "electronic", 0,
 "null", "JJ" 1.0)
  (53, 11, 12, <11:12>, 1, "," ",", 0, "null", "," 1.0)
  (54, 12, 13, <12:13>, 1, "computer" "computer", 0, "null", "NN" 1.0)
  (55, 13, 14, <13:14>, 1, "and" "and", 0, "null", "CC" 1.0)
  (56, 14, 15, <14:15>, 1, "building" "building", 0, "null", "NN" 1.0)
  (57, 15, 16, <15:16>, 1, "products" "products", 0, "null", "NNS" 1.0)
  (58, 16, 17, <16:17>, 1, "." ".", 0, "null", "." 1.0)
>

<
  (138, 0, 1, <0:2>, 1, "Bell,", 0, "null")
  (149, 0, 1, <0:2>, 1, "bell,", 0, "null")
  (115, 1, 2, <2:3>, 1, "based", 0, "null")
  (140, 1, 2, <2:3>, 1, "based", 0, "null", "VBN" 1.0)
  (117, 2, 3, <3:4>, 1, "in", 0, "null")
  (105, 3, 4, <4:5>, 1, "Los", 0, "null")
  (139, 3, 4, <4:5>, 1, "Los", 0, "null", "NNP" 1.0)
  (148, 3, 4, <4:5>, 1, "los", 0, "null")
  (109, 4, 5, <5:7>, 1, "Angeles,", 0, "null")
  (145, 4, 5, <5:7>, 1, "Angeles,", 0, "null", "NNP" 1.0)
  (150, 4, 5, <5:7>, 1, "angeles,", 0, "null")
  (119, 5, 6, <7:8>, 1, "makes", 0, "null")
  (141, 5, 6, <7:8>, 1, "makes", 0, "null", "VBZ" 1.0)
  (121, 6, 7, <8:9>, 1, "and", 0, "null")
  (123, 7, 8, <9:10>, 1, "distributes", 0, "null")
  (142, 7, 8, <9:10>, 1, "distributes", 0, "null", "VBZ" 1.0)
  (135, 8, 9, <10:12>, 1, "electronic,", 0, "null")
  (146, 8, 9, <10:12>, 1, "electronic,", 0, "null", "JJ" 1.0)
  (125, 9, 10, <12:13>, 1, "computer", 0, "null")
  (143, 9, 10, <12:13>, 1, "computer", 0, "null", "NN" 1.0)
  (127, 10, 11, <13:14>, 1, "and", 0, "null")
  (129, 11, 12, <14:15>, 1, "building", 0, "null")
  (144, 11, 12, <14:15>, 1, "building", 0, "null", "NN" 1.0)
  (137, 12, 13, <15:17>, 1, "products.", 0, "null")
  (147, 12, 13, <15:17>, 1, "products.", 0, "null", "NNS" 1.0)
>

(ROOT_INFORMAL
 (4223 SB-HD_MC_C 2.96671 0 13
  (4182 HDN_BNP-PN_C 3.36817 0 5
   (4181 HDN-AJ_RC-PR_C 2.75363 0 5
    (4172 W_COMMA_PLR 0.409888 0 1
     (4171 N_SG_ILR -0.223785 0 1
      (155 bell_n2/n_-_pn_le 0 0 1
       ("bell," 149
        "token [ +CARG \"Bell\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL + ]
+FORM \"bell,\" +FROM \"0\" +ID *diff-list*
 [ LAST #1=*list* LIST *cons* [ FIRST \"42\" REST *cons*
[ FIRST \"43\" REST #1 ] ] ] +PRED predsort +TNT null_tnt
 [ +MAIN tnt_main [ +PRB \"1\" +TAG \"NNP\" ]
+PRBS *null* +TAGS *null* ] +TO \"2\" +TRAIT native_trait ]"))))
    (4180 VP_RC-REDREL_C 2.72404 1 5
     (4179 HD-CMP_U_C 3.38403 1 5
      (4174 HD_OPTCMP_C 2.11616 1 2
       (4173 V_PAS_ODLR 2.26341 1 2
        (169 base_v1/v_np-pp_prop_le 2.0129 1 2
         ("based" 115
          "token [ +CARG #1=\"based\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"2\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"44\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"VBN\" ] +PRBS *null* +TAGS *null* ]
+TO \"3\" +TRAIT native_trait ]"))))
      (4178 HD-CMP_U_C 0.214338 2 5
       (232 in/p_np_i-reg_le -1.36904 2 3
        ("in" 117
         "token [ +CARG #1=\"in\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"3\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"45\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
 [ +PRB \"1\" +TAG \"IN\" ] +PRBS *null* +TAGS *null* ]
+TO \"4\" +TRAIT native_trait ]"))
       (4177 HDN_BNP-PN_C 0.907031 3 5
        (4176 W_COMMA_PLR 0.83469 3 5
         (4175 N_SG_ILR -0.209154 3 5
          (381 los_angeles_n1/n_-_pn-city_le 0 3 5
           ("los angeles," 148
            "token [ +CARG \"Los\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL - ]
+FORM \"los\" +FROM \"4\" +ID *diff-list*
[ LAST #1=*top* LIST *cons* [ FIRST \"46\" REST #1 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"NNP\" ] +PRBS *null* +TAGS *null* ]
+TO \"5\" +TRAIT native_trait ]"
            150
            "token [ +CARG \"Angeles\" +CLASS alphabetic
[ +CASE capitalized+lower +INITIAL - ] +FORM \"angeles,\"
+FROM \"5\" +ID *diff-list* [ LAST #1=*list* LIST *cons*
[ FIRST \"47\" REST *cons* [ FIRST \"48\" REST #1 ] ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"NNP\" ] +PRBS *null* +TAGS *null* ]
+TO \"7\" +TRAIT native_trait ]"))))))))))
  (4222 HD-HD_RNR_C -0.919733 5 13
   (4207 VP-VP_CRD-FIN-T_C 1.10544 5 8
    (4184 HD_XCMP_C 0.0352432 5 6
     (4183 V_3S-FIN_OLR 0.26696 5 6
      (300 make_v1/v_np_noarg3_le 0.314461 5 6
       ("makes" 119
        "token [ +CARG #1=\"makes\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"7\" +ID *diff-list*
 [ LAST #2=*top* LIST *cons* [ FIRST \"49\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"VBZ\" ] +PRBS *null* +TAGS *null* ]
+TO \"8\" +TRAIT native_trait ]"))))
    (4206 MRK-NH_EVNT_C 0.0157717 6 8
     (314 and_conj/c_xp_and_le -0.643034 6 7
      ("and" 121
       "token [ +CARG #1=\"and\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"8\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"50\" REST #2 ] ]
```

```
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"CC\" ] +PRBS *null* +TAGS *null* ]
+TO \"9\" +TRAIT native_trait ]"))
     (4186 HD_XCMP_C -0.202393 7 8
      (4185 V_3S-FIN_OLR -0.120609 7 8
       (327 distribute_v2/v_np_le 0 7 8
        ("distributes" 123
         "token [ +CARG #1=\"distributes\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"9\" +ID *diff-list*
[ LAST #2=*top* LIST *cons* [ FIRST \"51\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"VBZ\" ] +PRBS *null* +TAGS *null* ]
+TO \"10\" +TRAIT native_trait ]"))))))
   (4221 HDN_BNP_C -1.92298 8 13
    (4220 AJ-HDN_NORM_C -1.80577 8 13
     (4219 J-N_CRD-T_C -1.85933 8 12
      (4187 W_COMMA_PLR 0.328018 8 9
       (331 electronic_a1/aj_-_i_le 0 8 9
        ("electronic," 135
         "token [ +CARG \"electronic\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM \"electronic,\" +FROM \"10\" +ID *diff-list*
[ LAST #1=*list* LIST *cons*
[ FIRST \"52\" REST *cons* [ FIRST \"53\" REST #1 ] ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"JJ\" ] +PRBS *null* +TAGS *null* ]
+TO \"12\" +TRAIT native_trait ]")))
      (4218 N-N_CRD-IM_C -2.72077 9 12
       (337 computer_n1/n_-_c_le -0.267376 9 10
        ("computer" 125
         "token [ +CARG #1=\"computer\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"12\" +ID *diff-list*
 [ LAST #2=*top* LIST *cons* [ FIRST \"54\" REST #2 ] ]
+PRED predsort +TNT null_tnt [ +MAIN tnt_main
[ +PRB \"1\" +TAG \"NN\" ] +PRBS *null* +TAGS *null* ]
+TO \"13\" +TRAIT native_trait ]"))
       (4192 MRK-NH_N_C -0.806947 10 12
        (341 and_conj/c_xp_and_le -2.12559 10 11
         ("and" 127
          "token [ +CARG #1=\"and\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ] +FORM #1
+FROM \"13\" +ID *diff-list* [ LAST #2=*top* LIST *cons*
 [ FIRST \"55\" REST #2 ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"CC\" ]
+PRBS *null* +TAGS *null* ] +TO \"14\" +TRAIT native_trait ]"))
        (361 building_n1/n_-_c_le -0.276384 11 12
         ("building" 129
          "token [ +CARG #1=\"building\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM #1 +FROM \"14\" +ID *diff-list*
 [ LAST #2=*top* LIST *cons* [ FIRST \"56\" REST #2 ] ]
+PRED predsort +TNT null_tnt
 [ +MAIN tnt_main [ +PRB \"1\" +TAG \"NN\" ]
+PRBS *null* +TAGS *null* ] +TO \"15\" +TRAIT native_trait ]")))))
     (4195 W_PERIOD_PLR -0.554101 12 13
      (4194 N_PL_OLR 0.256482 12 13
       (368 product_n1/n_-_mc_le 0.0273073 12 13
        ("products." 137
         "token [ +CARG \"products\" +CLASS alphabetic
[ +CASE non_capitalized+lower +INITIAL - ]
+FORM \"products.\" +FROM \"15\" +ID *diff-list*
[ LAST #1=*list* LIST *cons* [ FIRST \"57\" REST *cons*
[ FIRST \"58\" REST #1 ] ] ] +PRED predsort +TNT null_tnt
[ +MAIN tnt_main [ +PRB \"1\" +TAG \"NNS\" ]
+PRBS *null* +TAGS *null* ] +TO \"17\" +TRAIT native_trait ]")))))))))
```

```
 [ LTOP: h1
   INDEX: e3 [ e SF: PROP TENSE: PRES MOOD:
 INDICATIVE PROG: - PERF: - ]
   RELS: <
          [ proper_q_rel<0:7>
            LBL: h4
            ARG0: x6 [ x PERS: 3 NUM: SG IND: + ]
            RSTR: h5
            BODY: h7 ]
          [ named_rel<0:2>
            LBL: h8
            ARG0: x6
            CARG: "Bell" ]
          [ "_base_v_1_rel"<2:3>
            LBL: h8
            ARG0: e9 [ e SF: PROP TENSE: UNTENSED MOOD:
INDICATIVE PROG: - PERF: - ]
            ARG1: i10
            ARG2: x6
            ARG3: h11 ]
          [ parg_d_rel<2:3>
            LBL: h8
            ARG0: e12 [ e SF: PROP ]
            ARG1: e9
            ARG2: x6 ]
          [ _in_p_rel<3:4>
            LBL: h13
            ARG0: e14 [ e SF: PROP TENSE: UNTENSED MOOD:
INDICATIVE PROG: - PERF: - ]
            ARG1: x6
            ARG2: x15 [ x PERS: 3 NUM: SG IND: + ] ]
          [ proper_q_rel<4:7>
            LBL: h16
            ARG0: x15
            RSTR: h17
            BODY: h18 ]
          [ named_rel<4:7>
            LBL: h19
            ARG0: x15
            CARG: "Los_Angeles" ]
          [ "_make_v_1_rel"<7:8>
            LBL: h20
            ARG0: e22 [ e SF: PROP TENSE: PRES MOOD:
INDICATIVE PROG: - PERF: - ]
            ARG1: x6
            ARG2: x23 [ x PERS: 3 NUM: PL ]
            ARG3: u21 ]
          [ _and_c_rel<8:9>
            LBL: h2
            ARG0: e3
            L-HNDL: h20
            L-INDEX: e22
            R-HNDL: h20
            R-INDEX: e24 [ e SF: PROP TENSE: PRES MOOD:
INDICATIVE PROG: - PERF: - ] ]
          [ "_distribute_v_to_rel"<9:10>
            LBL: h20
            ARG0: e24
            ARG1: x6
            ARG2: x23 ]
          [ udef_q_rel<10:17>
            LBL: h25
            ARG0: x23
            RSTR: h26
            BODY: h27 ]
          [ compound_rel<10:15>
            LBL: h28
            ARG0: e30 [ e SF: PROP TENSE: UNTENSED MOOD:
INDICATIVE PROG: - PERF: - ]
            ARG1: x23
            ARG2: i29 [ i PERS: 3 ] ]
          [ "_electronic_a_1_rel"<10:12>
            LBL: h31
            ARG0: e32 [ e SF: PROP ]
            ARG1: x23 ]
          [ udef_q_rel<12:13>
            LBL: h33
            ARG0: x35 [ x IND: + ]
            RSTR: h34
            BODY: h36 ]
          [ implicit_conj_rel<12:15>
            LBL: h37
            ARG0: i38
            L-HNDL: h31
            L-INDEX: e32
            R-HNDL: h28
            R-INDEX: i29 ]
          [ udef_q_rel<13:15>
            LBL: h39
            ARG0: i29
            RSTR: h40
            BODY: h41 ]
          [ "_computer_n_1_rel"<12:13>
            LBL: h42
            ARG0: x35 ]
          [ udef_q_rel<13:15>
            LBL: h43
            ARG0: x44 [ x IND: + ]
            RSTR: h45
            BODY: h46 ]
          [ _and_c_rel<13:14>
            LBL: h47
            ARG0: i29
            L-INDEX: x35
            R-INDEX: x44 ]
          [ "_building_n_1_rel"<14:15>
            LBL: h48
            ARG0: x44 ]
          [ "_product_n_1_rel"<15:17>
            LBL: h37
            ARG0: x23 ] >
   HCONS: < h45 QEQ h48 h40 QEQ h47 h34 QEQ h42
h26 QEQ h37 h17 QEQ h19 h11 QEQ h13 h5 QEQ h8 h1 QEQ h2 > ]

{e3:
 _1:proper_q<0:7>[BV x6]
 x6:named<0:2>("Bell")[]
 e9:_base_v_1<2:3>[ARG2 x6, ARG3 e14]
 e12:parg_d<2:3>[ARG1 e9, ARG2 x6]
 e14:_in_p<3:4>[ARG1 x6, ARG2 x15]
 _2:proper_q<4:7>[BV x15]
 x15:named<4:7>("Los_Angeles")[]
 e22:_make_v_1<7:8>[ARG1 x6, ARG2 x23]
 e3:_and_c<8:9>[L-INDEX e22, R-INDEX e24,
L-HNDL e22, R-HNDL e22]
 e24:_distribute_v_to<9:10>[ARG1 x6, ARG2 x23]
 _3:udef_q<10:17>[BV x23]
 e30:compound<10:15>[ARG1 x23, ARG2 i29]
 e32:_electronic_a_1<10:12>[ARG1 x23]
 _4:udef_q<12:13>[BV x35]
 i38:implicit_conj<12:15>[L-INDEX e32, R-INDEX i29,
L-HNDL e32, R-HNDL e30]
 _5:udef_q<13:15>[BV i29]
 x35:_computer_n_1<12:13>[]
 _6:udef_q<13:15>[BV x44]
 i29:_and_c<13:14>[L-INDEX x35, R-INDEX x44]
 x44:_building_n_1<14:15>[]
 x23:_product_n_1<15:17>[]
}
```