A corpus-based exercise book of Czech language

Ondřej Kučera

Abstract

We present a system designed as an electronic corpus-based exercise book of Czech morphology and syntax with the exercises directly selected from the Prague Dependency Treebank. In this way we want to make schoolchildren familiar with such an important academic product. Obviously, we do not expect that the schoolchildren (will) do grammar practicing so enthusiastically as they surf the web, chat with friends or write a web log. However, we believe that an electronic exercise book can make practicing more fun.

Two kinds of exercises are provided, morphological and syntactic, i. e. the exercises give practice in classifying parts of speech and morphological categories of words and in parsing a sentence and classifying syntactic functions of words. The Prague Dependency Treebank data cannot be used directly though, because of the differences between the academic approach and the approach taught in schools. Some of the sentences have to be completely discarded and several transformations have to be applied to the others in order to convert the original representation to the form the schoolchildren are familiar with.

1 Introduction

If we want to create an exercise book of Czech (or any language) we can choose two ways. First, we can do all the work ourselves from the scratch. We either make up the sentences or find them in books or newspapers and we process them one after another – we assign parts of speech and morphological categories; thus a sentence is parsed and the syntactic functions are assigned to words. Then the exercise book is formulated. However, this approach has many disadvantages. It is a very demanding, time-consuming task and we are almost bound to make some mistakes. We probably will not be able to collect more than a couple of tens (possibly hundreds) of sentences but more importantly the sentences will not reflect the real usage of the language – the sentences will usually be simple and short. The advantage of this approach lies in usability anytime.

Alternatively we can build the exercise book automatically (or semi-automatically, to be more precise), providing that an annotated corpus is available. This way we remove the disadvantages mentioned above – the hardest job is already done, the corpus exists and is annotated. There still remain some annotation errors in the corpus but very likely they occur in a much lesser amount. Annotation is usually done by more annotators at once so the chance that they would agree on a wrong decision is relatively low. An annotated corpus is built to reflect the present state of language and so will do the exercise book. The volume of such exercise book corresponds to the volume of the corpus, which goes beyond thousands of sentences.

We have chosen the latter way because the Prague Dependency Treebank (PDT) is available. PDT could not be taken as it is, thus a lot of adjustments had to be made. The following text will describe them in details. First we introduce the PDT itself in Section 2. Then, Section 3 is devoted to the filtering of the PDT sentences which could not be included in the exercise book and finally in Section 4 we discuss transformations needed to map the the PDT syntactic annotations into the school analyses.

We are not the first who came up with the idea to build an electronic exercise book. There exist a number of electronic exercises and educational applications. We studied some of them in order to inspire ourselves by their features and to learn from their mistakes. All of them contain different kinds of exercises, including some parts of morphology and syntax but none of them matches our idea – to take a sentence and analyse it in a complex way morphologically and syntactically. More importantly, none of the existing systems is of such a volume as our system is. This fact reflects the way we chose to build it – (semi)automatically from the Prague Dependency Treebank.

2 Prague Dependency Treebank

The Prague Dependency Treebank belongs to the top of the world corpus linguistics and its second edition is ready to be officially published (PDT 2.0, 2006). PDT has arisen from the tradition of the successful Prague School of Linguistics. The *dependency* approach to syntactic analysis with the main role of the verb has been applied. The annotations go from the morphological layer through the intermediate syntactic-analytical layer to the tectogrammatical layer (layer of underlying syntactic structure). The data (2 mil. words) have been annotated in the same direction, i. e., from a more simple layer to a more complex one. This fact corresponds to the amount of data annotated on a particular layer – 2 million words have been annotated on the lowest morphological layer, 1.5 million words on both morphological and analytical layers and 0.8 million words on all three layers.

Within the PDT conceptual framework, a sentence is represented as a rooted ordered tree with labeled nodes and edges on both syntactic (Hajičová, Kirschner, and Sgall, 1999) and tectogrammatical (Mikulová, Marie et al., 2006) layers. Thus we speak about syntactic and tectogrammatical trees, respectively. Representation on the morphological layer (Zeman et al., 2005) correspond to a list of morphological tags of the same length as the sentence is. Figure 1 illustrates the analytical and morphological annotation of the sentence *Zásadní pákou je tlak na naši peněženku.* [A strain on our purse seems to be the fundamental lever.] One token of the morphological layer is represented by exactly one node of the tree (*zásadní* [fundamental], *pákou* [lever], *je* [seems-to-be], *tlak* [strain], *na* [on], *naši* [our], *peněženku* [purse], '.') and the dependency relation between two nodes is captured by en edge between them, i. e. between the dependent and its governor. The actual type of the relation is given as a function label of the edge, for example the edge *(tlak, je)* is labeled by `Sb` (subject), i. e. the function of the node *tlak*. Together with a syntactic function, a morphological tag is displayed (*tlak*, `NNIS1---A---`). Since there is a *m:n* correspondence between the number of nodes in analytical and tectogrammatical trees, it would be rather confusing to display the annotations on those layers together in one tree, hence we provide a separate tree visualizing the tectogrammatical annotation of the sentence – see Figure 2. The tectogrammatical lemma and the functor are relevant to our task, thus we display them with each node in tectogrammatical tree, e. g. (*tlak*, `ACT`).

Analogously to the separate visualization of annotations, the inner data format – the *Prague Markup Language* – represents the particular annotations separately (Pajas and Štěpánek, 2005) as is illustrated in Tables 1, 2 and 3. The given tables correspond to the files with the morphological, analytical and tectogrammatical layers of annotation, respectively, of the sample sentence *Zásadní pákou je tlak na naši peněženku.* We provide a comment on the particular lines of files if it is relevant to our issue. One can compare Tables 1 and 2 with Figure 1 and Table 3 with Figure 2.

3 Filtering the sentences

The Prague Dependency Treebank contains a lot of sentences unsuitable for our exercises – sentences containing phenomena which different school text books either cover differently or even do not cover at all. Such sentences had to be eliminated (we cannot exercise students in something they have never been taught), preferably automatically.

FilterSentences is a procedure designed to filter sentences of the PDT data annotated on all three layers (0.8 million words in 49,442 sentences). So far we have formulated the following nine filters that decide whether a sentence should be kept or discarded.

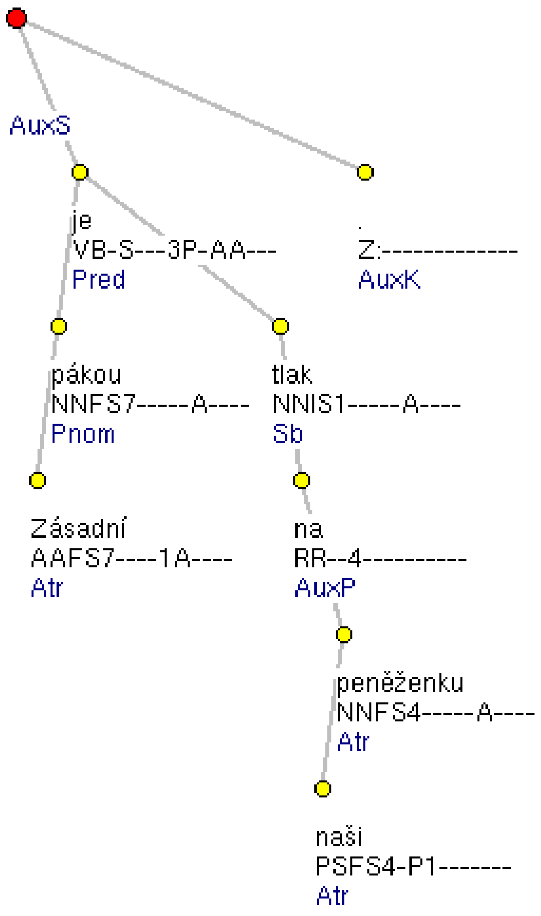


Figure 1: The analytical tree of the sentence *Zásadní pákou je tlak na naši peněženku.*

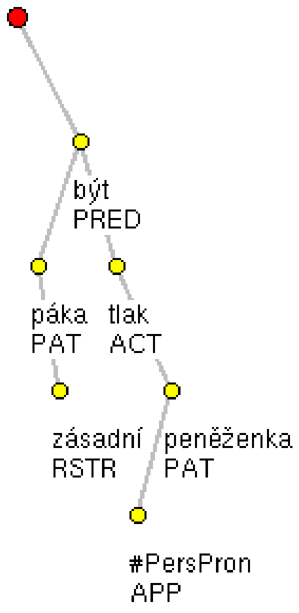


Figure 2: The tectogrammatical tree of the sentence *Zásadní pákou je tlak na naši peněženku.*

```
<s id="m-cmpr9410-049-p3s5">                  <!-- sentence identification -->
  <m id="m-cmpr9410-049-p3s5w1">              <!-- m-token identification -->
   <src.rf>manual</src.rf>                    <!-- manual annotation -->
   <w.rf>w#w-cmpr9410-049-p3s5w1</w.rf>       <!-- link to the w-layer -->
   <form>Zásadní</form>                       <!-- form -->
   <lemma>zásadní_,s</lemma>                  <!-- lemma -->
   <tag>AAFS7----1A----</tag>                 <!-- tag -->
  </m>
  <m id="m-cmpr9410-049-p3s5w2">
   <src.rf>manual</src.rf>
   <w.rf>w#w-cmpr9410-049-p3s5w2</w.rf>
   <form>pákou</form>
   <lemma>páka</lemma>
   <tag>NNFS7-----A----</tag>
  </m>
  ...
</s>
```

Table 1: Representation of the sentence *Zásadní pákou je tlak na naši peněženku.* on the morphological layer

```
<LM id="a-cmpr9410-049-p3s5">                <!-- sentence identification -->
  <s.rf>m#m-cmpr9410-049-p3s5</s.rf>          <!-- link to the m-layer -->
  <ord>0</ord>                                <!-- linear order of the node in the sentence -->
  <children>
   <LM id="a-cmpr9410-049-p3s5w3">            <!-- a-token identification -->
    <m.rf>m#m-cmpr9410-049-p3s5w3</m.rf>
    <afun>Pred</afun>                         <!-- syntactic function -->
    <ord>3</ord>
    <children>
     <LM id="a-cmpr9410-049-p3s5w2">
      <m.rf>m#m-cmpr9410-049-p3s5w2</m.rf>
      <afun>Pnom</afun>
      <ord>2</ord>
      <children>
       ...
      </children>
     </LM>
    <LM id="a-cmpr9410-049-p3s5w4">
     <m.rf>m#m-cmpr9410-049-p3s5w4</m.rf>
      <afun>Sb</afun>
      <ord>4</ord>
      <children>
       ...
      </children>
    </LM>
    </children>
   </LM>
   <LM id="a-cmpr9410-049-p3s5w8">
    <m.rf>m#m-cmpr9410-049-p3s5w8</m.rf>
    <afun>AuxK</afun>
    <ord>8</ord>
   </LM>
  </children>
</LM>
```

Table 2: Representation of the sentence *Zásadní pákou je tlak na naši peněženku.* on the analytical layer (see also Figure 1)

```
<LM id="t-cmpr9410-049-p3s5">                           <!-- sentence identification -->
  <atree.rf>a#a-cmpr9410-049-p3s5</atree.rf>            <!-- link to the a-layer -->
  <nodetype>root</nodetype>
  <deepord>0</deepord>
  <children>
   <LM id="t-cmpr9410-049-p3s5w3">                      <!-- t-token identification -->
    <a>
     <lex.rf>a#a-cmpr9410-049-p3s5w3</lex.rf>
    </a>
    <nodetype>complex</nodetype>
    <t_lemma>být</t_lemma>
    <functor>PRED</functor>                             <!-- functor -->
    <tfa>f</tfa>
    <deepord>3</deepord>
    <sentmod>enunc</sentmod>
    <gram>                                              <!-- grammatemes -->
     <sempos>v</sempos>
     <verbmod>ind</verbmod>
     <deontmod>decl</deontmod>
     <tense>sim</tense>
     <aspect>proc</aspect>
     <resultative>res0</resultative>
     <dispmod>disp0</dispmod>
     <iterativeness>it0</iterativeness>
    </gram>
    <val_frame.rf>v#v-w243f1</val_frame.rf>
    <children>
     ...
    </children>
   </LM>
  </children>
</LM>
```

Table 3: Representation of the sentence *Zásadní pákou je tlak na naši peněženku.* on the tectogrammatical layer (see also Figure 2)

3.1 SimpleSentence

This filter discards all sentences containing more than just one clause. The reason for this is that schoolchildren are not taught to parse combinations of clauses and thus to analyse such sentences. Testing whether a sentence has more than one clause is (technically) the most complicated filter because of the number of cases it must deal with. The filter has three phases:

1. On the analytical layer, if a sentence contains a coordination of which a predicate is a member, then the sentence is discarded.

2. On the analytical layer, if there is a comma with the syntactic function *AuxX* and its parent does not bear the syntactic function *Coord*, a sentence is discarded.

3. On the tectogrammatical layer, if there exists a coordination whose functor is *CONJ*, *ADVS*, *CSQ*, *DISJ*, *GRAD*, *REAS*, *CONFR*, *CONTRA*, *OPER* or *APPS* and at least one of its children's *tense* grammatemes is *post*, *ant*, *sim* or *nir*, the sentence is discarded.

For illustration we present the analytical tree of the sentence with more than one clause *Pro firmy, které se rozhodnou využít jejich služeb, to nemusí být zrovna levná záležitost.* [For companies which decide to use their services it need not be exactly a cheap affair.] in Figure 3. The sentence was discarded because of the commas bearing the *AuxX* function and having the parents with the *Atr* function (see point 2 above).

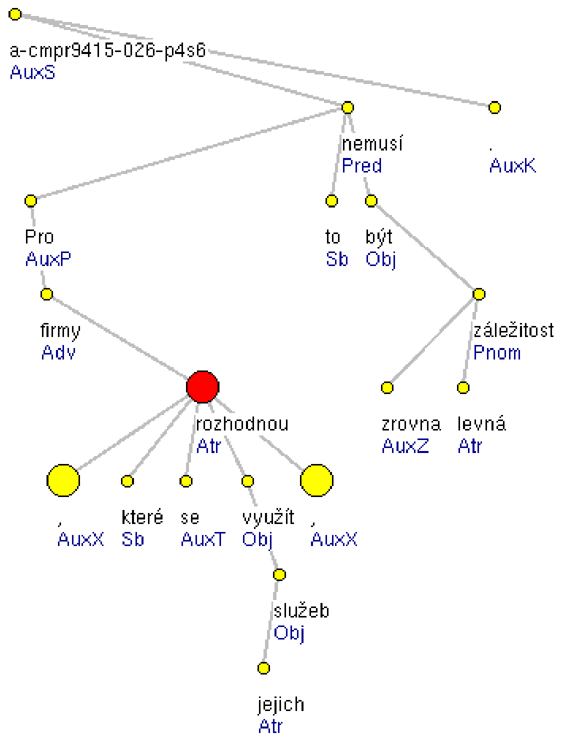


Figure 3: Example of a sentence discarded by the filter *SimpleSentence*

3.2 GraphicalSymbols

This filter discards all sentences that contain various graphical symbols – words with the syntactic function *AuxG* – e. g. colons, quotes, asterisk. The filter needs information from the morphological and analytical layers – if there exists a node with the function *AuxG* and its word form is not a dot ("."), a sentence is discarded.

The example of the discarded sentence *Dodavatel účtoval 229422 Kč/m*^{2}. [The supplier was charging 229422 Kč/m^2.] is visible in Figure 4.

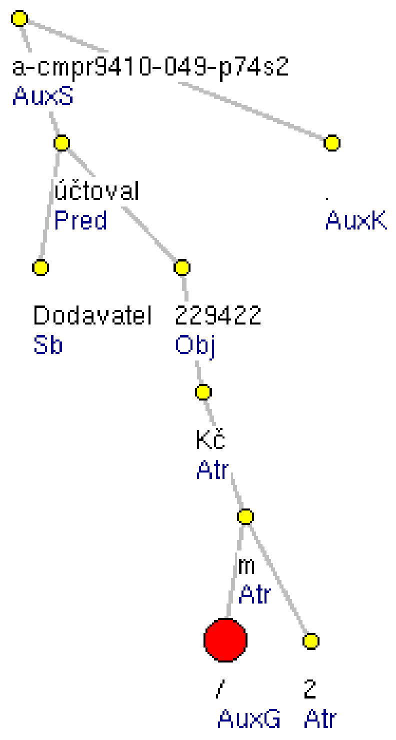


Figure 4: Example of a sentence discarded by the filter *GraphicalSymbols*

3.3 EllipsisApposition

This filter discards all sentences containing ellipses (*ExD*) or appositions (*Apos*). It needs information from the analytical layer, whenever a node with the function *ExD* or *Apos* is found, a sentence is discarded.

The example of the discarded sentence *8) Stejné požadavky jako na dovážené výrobky bude SPPI uplatňovat i u výrobků tuzemských.* [8) SPPI will apply the requirements for the imported products for the national products as well.] is visible in Figure 5.

3.4 OnePredicate

This filter preserves sentences containing one predicate (*Pred*), particularly it discards all sentences without predicate (because sentences with more than one predicate are already discarded by the *SimpleSentence* filter). This filter needs information from the analytical layer as well. It simply counts the number of predicates present in the sentence.

The example of the discarded sentence *Nová striktní omezení vlády SR proti českým exportérům* [New strict restrictions of the government of SR against the Czech exporters] is visible in Figure 6.

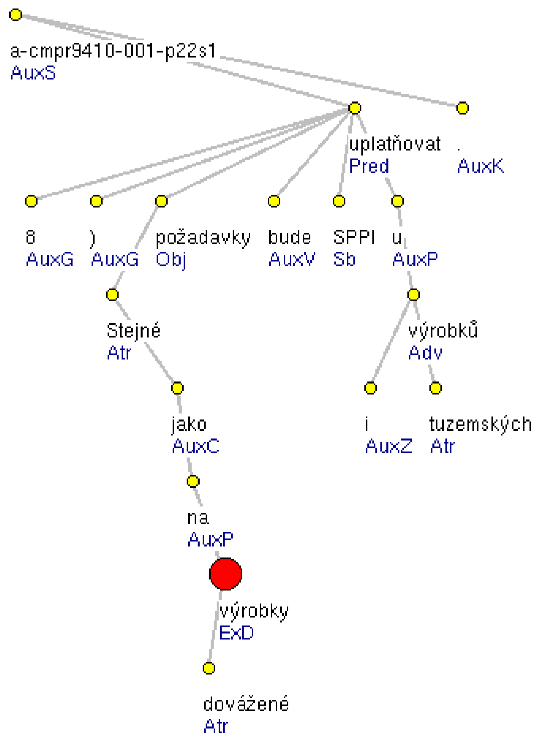


Figure 5: Example of a sentence discarded by the filter *EllipsisApposition*

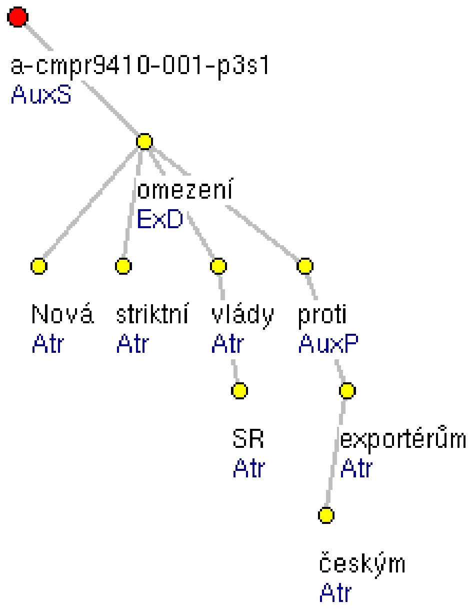


Figure 6: Example of a sentence discarded by the filter *OnePredicate*

3.5 LessThanNWords

This filter preserves sentences having not more than *MAX_COUNT* words (a threshold which we decided empirically to set to 19). The reason to use this criterion is that too long sentences can be very complicated in their syntax and they could unnecessarily confuse the students. Usually, the greatest part of these sentences contains long chains of attributes which makes them “uninteresting” to exercise.

3.6 MoreThanNWords

This filter preserves the sentences having at least *MIN_COUNT* words (a threshold which we decided empirically to set to 5). This way we eliminated too simple sentences, often only short headings.

3.7 AuxO

This filter discards all sentences containing a word with the syntactic function *AuxO* – redundant or emotional item, “coreferential” pronoun – with one exception, which is the reflexive particle *si*.

The example of the discarded sentence *To byste se divil.* [You would be surprised.] is visible in Figure 7.

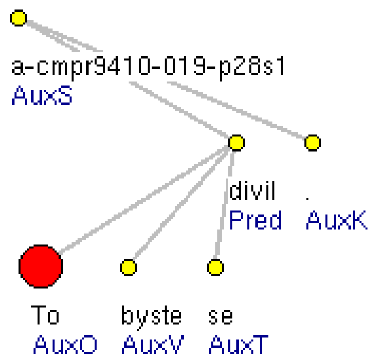


Figure 7: Example of a sentence discarded by the filter *AuxO*

3.8 IndividualSentences

This filter searches a configuration file (which is maintained manually and contains a list of PDT sentence identifiers) to select sentences that should be discarded. Its purpose is to discard sentences which should not be included in the exercise book but for which we have not so far found broader linguistically-motivated characteristics to formulate a regular filter criterion.

3.9 KeepAll

This filter exists only for technical purposes, it selects all input sentences.

3.10 Filtering results

It often happened that a sentence was discarded by more than one filter but that does not matter. The important fact is that none of the criteria is a special case of another one.

From 49,442 sentences which entered the filtering process 11,705 were kept, that is approximately 24 % (see Table 4 for detailed numbers). It is a number slightly lower than what we had hoped for but it is still a lot of sentences to make exercises from. Figure 8 shows an example of two sentences that were kept after the complete filtering: *Zahrnul jste mne mnoha profesemi.* [You have loaded me with many professions.] and *Tento postup si vyžádá v praxi zhotovování ověřených kopií.* [This procedure will require preparation of verified copies in practice.]

Filter	# input sentences	# preserved sentence (%)
SimpleSentence	49,442	22,552 (45.6)
GraphicalSymbols	22,552	20,384 (90.4)
EllipsisApposition	20,384	13,633 (66.9)
OnePredicate	13,633	13,617 (99.9)
LessThanNWords	13,617	13,010 (95.5)
MoreThanNWords	13,010	11,718 (90.1)
AuxO	11,718	11,705 (99.9)
run-them-all	**49,442**	**11,718 (23.7)**

Table 4: Quantitive review of sentence filtering

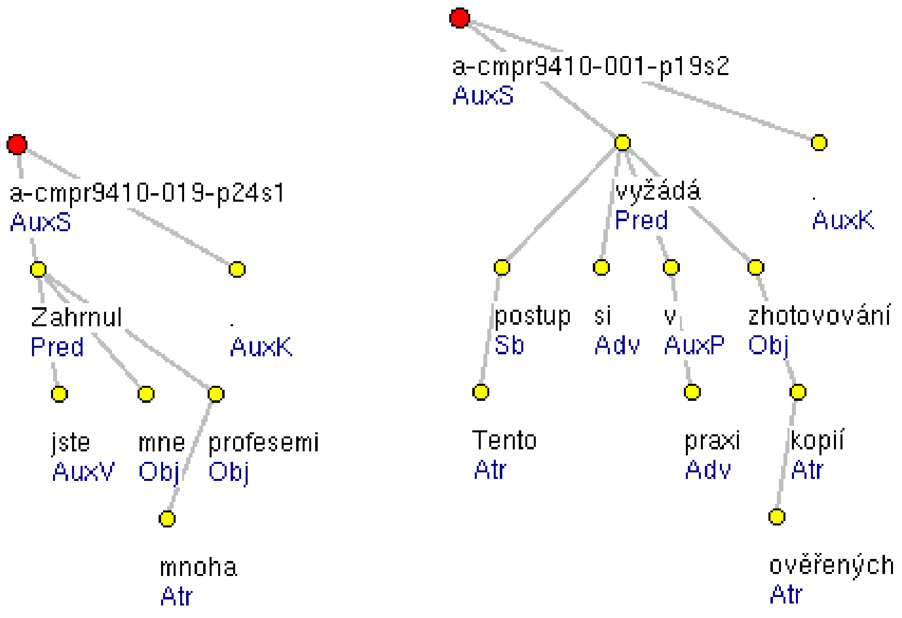


Figure 8: Example of two sentences which were kept in the data

4 Transformation the syntactic trees

Next step after filtering the unsuitable sentences was the transformation of their annotations into a form the students are familiar with. While this task was quite straightforward on the morphological layer, a lot of adjustments had to be done on the analytical layer. The analytical layer of the PDT differs in many aspects from the syntax taught in schools. First, it contains syntactic functions which do not have their counterparts in the school analysis as presented in Table 5. Second, it has 1:1 mapping of syntactic nodes and words (and punctuation marks) of the sentence, while in the school representation a node may contain (and often contains) more than just one word. In this case the whole node has only one syntactic function.

PDT syntactic functions	School syntactic functions	Description
Pred	Přs	predicate
Pnom	Přj	predicate nominal
Sb	Po	subject
Obj	Pt	object
Atr, AtrAdv, AdvAtr, AtrAtr, AtrObj, ObjAtr	Pk	adverbial
Adv	Pu	adverbial
Atv, AtvV, Obj	D	complement
Coord	–	coordination
AuxC, AuxP, AuxZ, AuxO, AuxV, AuxR, AuxY, AuxK, AuxX, AuxG	–	auxiliary sentence members

Table 5: PDT syntactic functions vs. school syntactic functions

In the next, we will refer to the PDT annotation concept as to the *PDT analysis* and to the concept taught in schools as to the *school analysis*, so Figures 9 and 10 illustrate the PDT analysis and the school analysis of the sentence *Zásadní pákou je tlak na naši peněženku.*, respectively.

With regards to the discussed differences, we went through the PDT annotation guidelines (Hajičová, Kirschner, and Sgall, 1999) systematically, analyzed all specified phenomena and designed their transformation into the school analysis schemes.

The initial school analysis scheme is a tree as it is in the PDT data – each word has its own node and each node (except the technical root node) has one parent. The resulting scheme is more general, a node may contain more than one word. During transformations we used the following three elementary operations:

- *Join the parent node.* Words at the current node are moved into the label of its parent node, children of the current node become children of its parent node, the current node is removed afterwards. Figure 11 displays the PDT analysis of the node *pákou* (on the left) and its school analysis after the *Join the parent node* transformation (on the right).
- *Absorb the child nodes.* All words at all children nodes are moved into the label of the current node, the children nodes are removed, the former children nodes of children nodes become new children nodes of the current node. Figure 12 displays the PDT analysis of the node *k* (on the left) and its school analysis after the *Absorb the child nodes* transformation (on the right).
- *Remove the node.* The node is removed from the tree. This operation is permitted only on leaves.

We will describe transformations of nodes according to the syntactic function they bear. For every discussed function we specify its citation directly into the guidelines for the syntactic annotation of PDT (Hajičová, Kirschner, and Sgall, 1999).

4.1 Syntactic function *Pred*

Predicate, a node depending on the technical root node – see Section 3.3.1.

The syntactic function *Přs* is assigned to this node and no other transformation is performed (the node remains a child of the technical root node of the whole sentence).

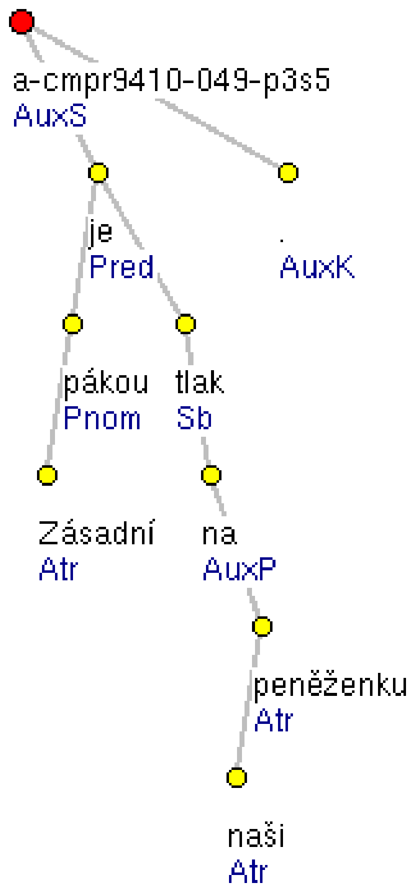


Figure 9: The PDT analysis of the sentence *Zásadní pákou je tlak na naši peněženku.*

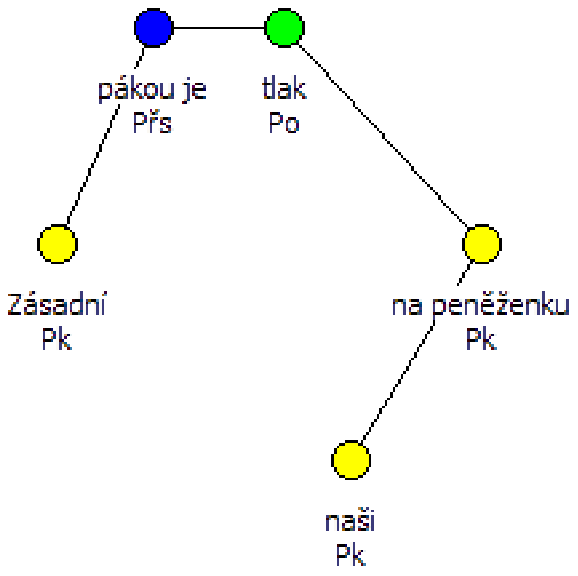


Figure 10: The school analysis of the sentence *Zásadní pákou je tlak na naši peněženku.*

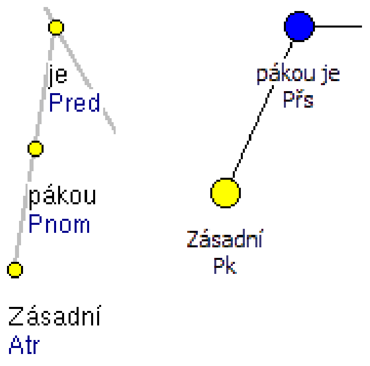


Figure 11: The operation *Join the parent node*

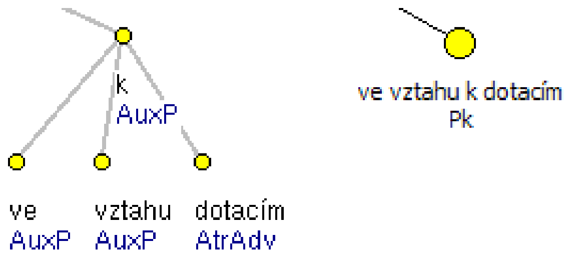


Figure 12: The operation *Absorb the child nodes*

4.2 Syntactic function *Pnom*

Nominal predicate, or nominal part of predicate with copula – see Section 3.3.1.

In the school analysis this node is a part of the predicate, so the operation *Join the parent node* is being performed as long as its parent node bears the syntactic function *Coord* or *Pred*. Then

- if the parent node bears the syntactic function *Coord*, then the operation *Join the parent node* is performed once more and it is remembered that the parent coordinates a nominal part of predicate (see also 4.22).
- Otherwise (the syntactic function of the parent node is *Pred*) the operation *Join the parent node* is performed once more and the syntactic function of the parent node is changed to *Přj*.

4.3 Syntactic function *AuxV*

Auxiliary verb be – see Section 3.3.1.

This node is also a part of predicate, the operation *Join the parent node* is to be performed. This transformation does not change the syntactic function of the parent node. The parent node's syntactic function does not have to be *Pred*, it may also be *AuxV*. In this case, the parent node is to be transformed in the same way as the current node.

4.4 Syntactic function *Sb*

Subject – see Section 3.3.2.

The most important concept of the school syntax is a *basic syntax pair*. It consists of subject and predicate standing on the same level, so the school analysis scheme does not correspond to a tree structure with a single root. It corresponds to a tree-like structure having two root nodes. The following transformation has to be performed:

- If the parent node's syntactic function is *Coord*, the operation *Join the parent node* is being performed once more and it is remembered that the parent coordinates the subject (see also 4.22).
- Otherwise (the syntactic function of the parent node is *Pred*) the dependency between the current node and its parent is removed and the technical root node of the sentence becomes the new parent node of the current node. The syntactic function of the current node is *Po*.

4.5 Syntactic function *Atr*

Attribute – see Section 3.3.3.

The syntactic function *Atr* directly corresponds to the the syntactic function *Pk*, the node does not need to be transformed.

4.6 Syntactic functions *AtrAdv, AdvAtr, AtrAtr, AtrObj, ObjAtr*

All of these syntactic functions are special cases of *Atr*, so the rules for *Atr* are applied – see section 4.5.

4.7 Syntactic function *Obj*

Object – see Section 3.3.4.

Unfortunately, object (*Obj*) in the PDT analysis is more general than object (*Pt*) in the school analysis so the following transformation has to be performed:

- If object is an infinitive verb and its parent is a modal verb, the operation *Join the parent node* is performed, so the current node becomes a part of the predicate.
- If the object's functor on the tectogrammatical layer is *EFF* and if it is a noun either in nominative, or instrumental, or accusative following the preposition *za*, then the syntactic function *D* is assigned to the current node.
- Otherwise the syntactic function *Pt* is assigned to the current node.

4.8 Syntactic function *Adv*

Adverbial – see Section 3.3.5.

The syntactic function *Adv* directly corresponds to the the syntactic function *Pu*, the node does not need to be transformed.

4.9 Syntactic function *Atv*

Complement (so-called determining) technically hung on a non-verb. element – see Section 3.3.6.

The syntactic function *Atv* directly corresponds to the syntactic function *D*, the node does not need to be transformed.

4.10 Syntactic function *AtvV*

Complement (so-called determining) hung on a verb, no second gov. node – see Section 3.3.6.

The syntactic function *AtvV* directly corresponds to the syntactic function *D*, the node does not need to be transformed.

4.11 Syntactic function *AuxC*

Conjunction (subord.) – see Section 3.3.7.1.

In most cases a node with the syntactic function *AuxC* stands at the beginning of a subordinate clause which was already discarded using the filter *SimpleSentence* (see Section 3.1). In the rest of the cases the operation *Absorb the child nodes* is performed. One of the former child nodes had the PDT syntactic function usually directly corresponding to some school syntactic function – this function will be assigned to the current node.

4.12 Syntactic function *AuxP*

Primary prepositions, parts of a secondary preposition – see Section 3.3.7.2.

The operation *Consume the child nodes* needs to be performed. If the node had any children (i. e. it was not a child of another node with the syntactic function *AuxP*), one of them had the syntactic function usually directly corresponding to some syntactic function – this function will be assigned to the current node.

4.13 Syntactic function *AuxZ*

Emphasizing word – see Section 3.3.7.3.

The operation *Join the parent node* has to be performed. If the parent node's syntactic function is *Coord*, it is remembered that the parent coordinates an emphasizing word (see also 4.22).

4.14 Syntactic function *AuxO*

Redundant or emotional item, "coreferential" pronoun – see Section 3.3.7.4.

As we stated in Section 3.7, sentences containing nodes with the syntactic function *AuxO* are discarded while filtering, with the exception of particle *si*. So during the transformations we know that we're dealing with the particle *si* and the operation *Join the parent node* can be performed.

4.15 Syntactic function *AuxT*

Reflex. tantum – see Section 3.3.7.5.

The operation *Join the parent node* needs to be performed.

4.16 Syntactic function *AuxR*

Ref., neither Obj *nor* AuxT*, Pass. refl.* – see Section 3.3.7.6.

The operation *Join the parent node* needs to be performed.

4.17 Syntactic function *AuxY*

Adverbs, particles not classed elsewhere – see Section 3.3.7.7.

The operation *Join the parent node* needs to be performed.

4.18 Syntactic function *AuxK*

Terminal punctuation of a sentence – see Section 3.3.7.8.

Terminal punctuation is not displayed in the school analysis, the operation *Remove the node* is performed.

4.19 Syntactic function *AuxX*

Comma (not serving as a coordinating conj.) – see Section 3.3.8.3.

Nodes with the syntactic function *AuxX* are processed during the transformation of the *Coord* nodes, see 4.22.

4.20 Syntactic function *AuxG*

Other graphic symbols, not terminal – see Section 3.3.8.4.

Because of the filter *GraphicalSymbols* (see Section 3.2) we deal only with dots (".") bearing the syntactic function *AuxG*, the operation *Join the parent node* is performed.

4.21 Syntactic function *ExD*

A technical value for a deleted item; also for the main element of a sentence without predicate (Externally-Dependent) – see Section 3.4.1.

Sentences containing ellipses were removed from data by the filter *EllipsisApposition* (see Section 3.3).

4.22 Syntactic function *Coord*

Coord. node – see Section 3.5.1.

The filter *SimpleSentence* (see Section 3.1) removes all sentences with coordinated clauses, so we do not have to worry about this case. We have to handle the coordination of words. We perform the operation *Absorb the child nodes*. Two kinds of nodes are absorbed: commas with the syntactic function *AuxX* and nodes whose syntactic function determines what kinds of nodes are actually coordinated. Next, the following cases have to be considered:

- *Sb*s (subjects) are coordinated. We treat the current node as if it actually was subject (see 4.4), the syntactic function *Po* is assigned to it and its new parent node will be the technical root node.

- *Obj*s (objects) are coordinated, they are infinitive verbs and parent node of the current node is a modal verb. The operation *Join the parent node* is performed.

- *Pnom*s (nominal predicates) are coordinated. The operation *Join the parent node* is being performed as long as its parent node's syntactic function is either *Coord* or *Pred*. Then

  – if the parent node bears the syntactic function *Coord*, then the operation *Join the parent node* is performed once more and the parent coordinates nominal parts of the predicate.

  – Otherwise (the syntactic function of the parent node is *Pred*) the operation *Join the parent node* is performed once more and the syntactic function of the parent node is changed to *Přj*.

- *AuxZ*s (emphasising words) are coordinated. The operation *Join the parent node* is performed.

- Otherwise a syntactic function is assigned to the current node according to the syntactic function of the coordinated nodes.

The situation is illustrated on the sentence *Rozdíl obou cen vyjadřuje náklady*$_{\text{Obj}}$,$_{\text{AuxX}}$ *režie*$_{\text{Obj}}$ *a*$_{\text{Coord}}$ *poplatky*$_{\text{Obj}}$ *distributora.* [The variance of prices covers the expenses, the overheads and the distribution charges.] The node *a* absorbs the nodes *náklady*, ",", *režie* and *poplatky* and syntactic function *Pt* is assigned to it. See Figures 13 and 14.

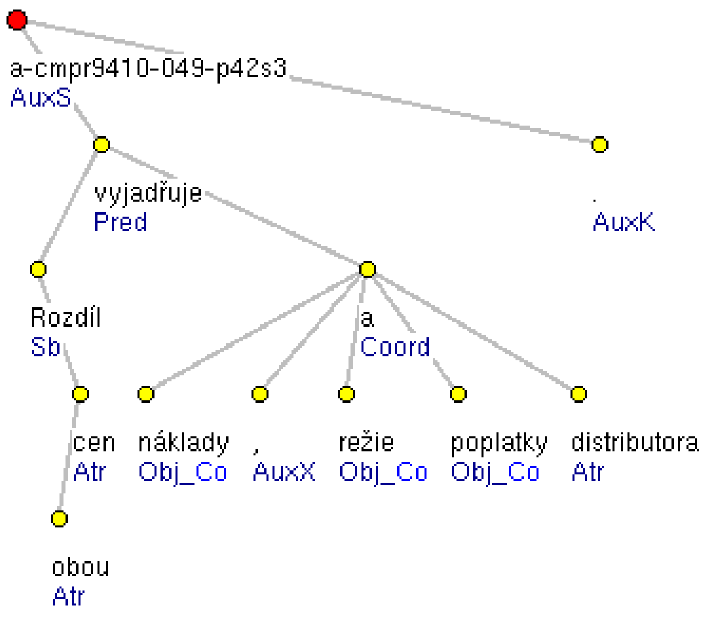


Figure 13: The syntactic function *Coord* in the PDT analysis

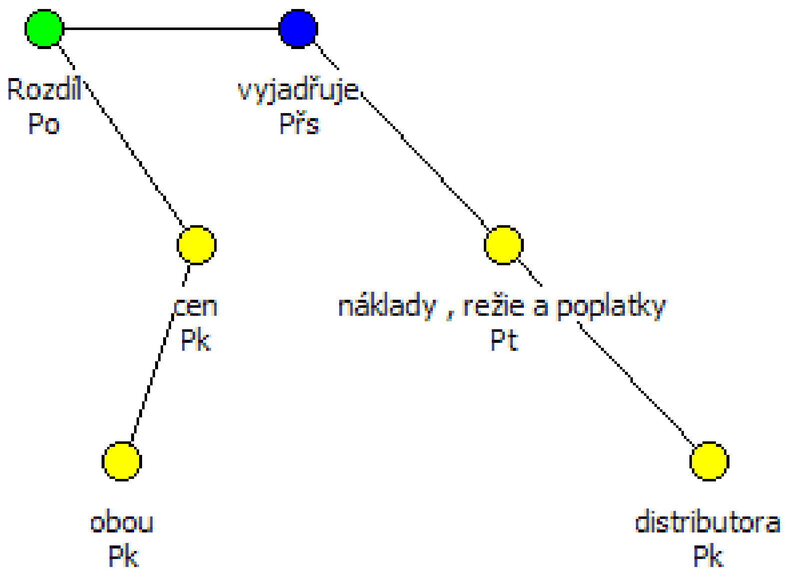


Figure 14: Transformation of the syntactic function *Coord*

4.23 Syntactic function *Apos*

Apposition (main node) – see Section 3.5.2.

Sentences containing appositions were removed from data by the filter *EllipsisApposition* (see section 3.3).

5 Implementation

For an implementation of the exercise book (and helper utilities) we decided to use the *Java* programming language. We had several reasons for this decision. First, Java is a high level language with a lot of mechanisms for making the work of a developer easier and protecting them "against themselves". Furthermore it made it simple to port the application to other operating systems than MS Windows (although they are our main target platform because of their wide spread in schools).

However we decided not to use the standard part of Java for creating graphical user interface, Swing. Instead we used *Standard Widget Toolkit*, a library from the Eclipse[1] project. This has two important advantages. The graphical user interface is implemented using native widgets of the platform, which means that the application looks naturally like other applications of the platform. It also leads to faster responses of the GUI.

The results of our work are the following three software components which constitute the whole system of the electronic exercise book. The system is provided under the *General Public License*[2], however the programs are useless without the PDT data which are covered by their own license.

5.1 FilterSentences

This component was used to prepare data suitable for usage in the exercise book. The end user will never have to use it.

5.2 Charon

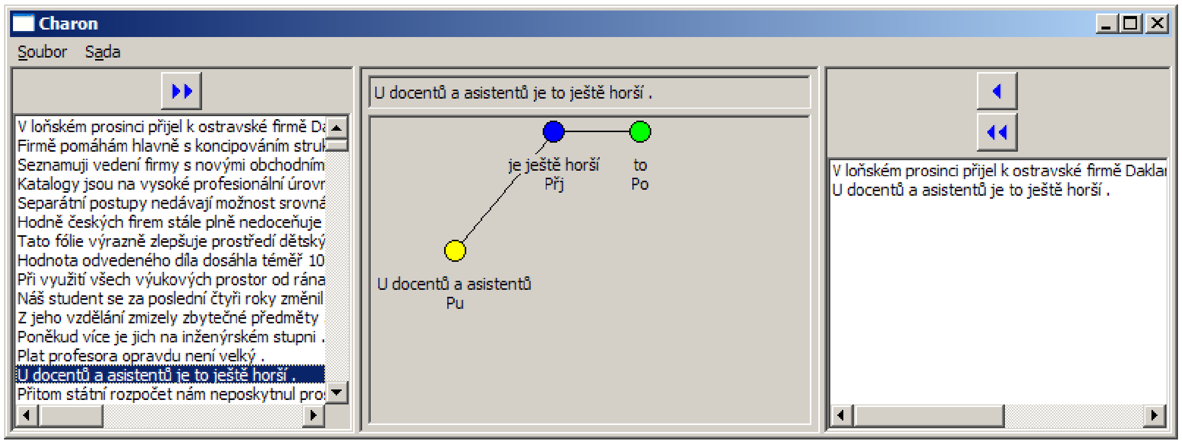
An administrative tool, used for viewing all of the available sentences and for composing the exercises. We assume that it will be used by teachers mostly. The *Charon* application consists of three components as it is shown in Figure 15. The left-hand component displays all the sentences in the exercise book. An active sentence (with dark background) is displayed in the upper part of the middle component as well (and its syntactic analysis in the lower part). When one holds the mouse cursor over any word of the sentence its morphological analysis is displayed. An active sentence is put into an exercise by clicking on the double right-arrow icon. In our demonstration, two sentences were selected.

5.3 Styx

The electronic exercise book itself. It uses the exercises composed with *Charon*. An active sentence is analyzed both morphologically and syntactically as shown in Figure 16. During the morphological analysis, the user moves on word by word and for each word selects its part of speech. According to the selected part of speech, the combo boxes for the relevant morphological categories appear and let the user choose one of several choices they consider the proper one. During the syntactic analysis, the user moves nodes using the traditional drag and drop method to catch the dependent-governor relation. Afterwards, the syntactic functions are assigned, technically via pop-up menus. Once the analyses are finished, the correct answers are provided separately for morphology and syntax – see Figure 17. "OK"

[1] For more see http://www.eclipse.org/

[2] See http://www.gnu.org/licenses/licenses.html



V loňském prosinci přijel k ostravské firmě Da
Firmě pomáhám hlavně s koncipováním struk
Seznamuji vedení firmy s novými obchodním
Katalogy jsou na vysoké profesionální úrovn
Separátní postupy nedávají možnost srovná
Hodně českých firem stále plně nedoceňuje
Tato fólie výrazně zlepšuje prostředí dětský
Hodnota odvedeného díla dosáhla téměř 10
Při využití všech výukových prostor od rána
Náš student se za poslední čtyři roky změnil
Z jeho vzdělání zmizely zbytečné předměty
Poněkud více je jich na inženýrském stupni .
Plat profesora opravdu není velký .
U docentů a asistentů je to ještě horší .
Přitom státní rozpočet nám neposkytnul pro

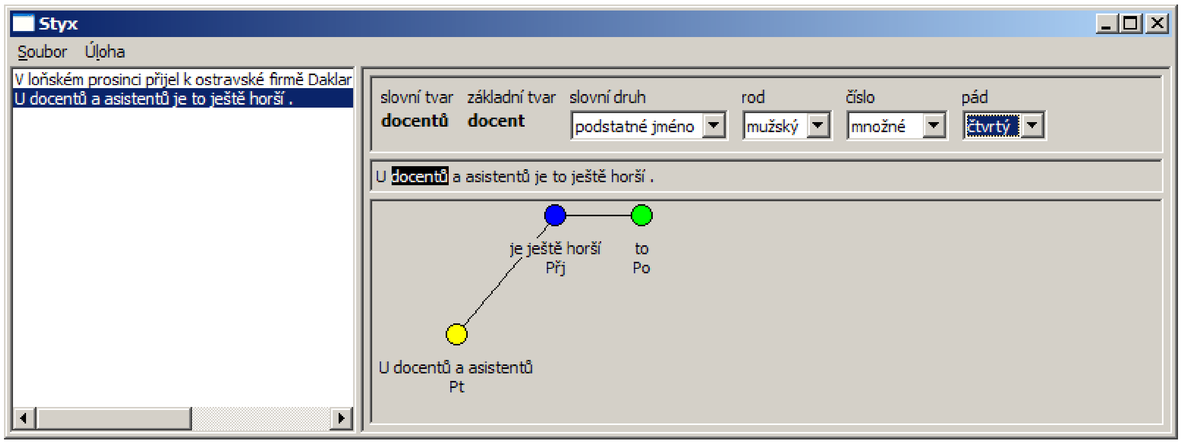
U docentů a asistentů je to ještě horší .

je ještě horší	Přj
to	Po
U docentů a asistentů	Pu

V loňském prosinci přijel k ostravské firmě Daklar
U docentů a asistentů je to ještě horší .

Figure 15: *Charon* – selecting sentences

marks those morphological categories that are assigned correctly. In reverse, “#” marks categories assigned incorrectly. Regarding an evaluation of syntactic analysis, the colors are used to distinguish discrepancies between the correct answer and the user analysis. The same coloring scheme is applied for evaluation of syntactic functions.



V loňském prosinci přijel k ostravské firmě Daklar
U docentů a asistentů je to ještě horší .

slovní tvar	základní tvar	slovní druh	rod	číslo	pád
docentů	**docent**	podstatné jméno	mužský	množné	čtvrtý

U docentů a asistentů je to ještě horší .

je ještě horší	Přj
to	Po
U docentů a asistentů	Pt

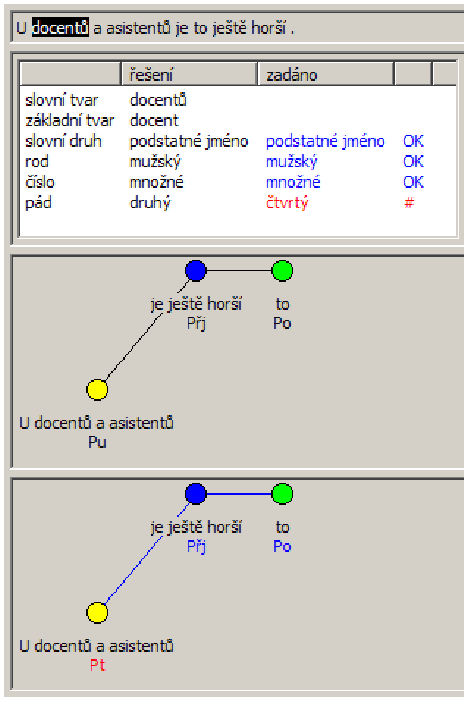
Figure 16: *Styx* – practicing

6 Conclusion

The PDT-based exercise book has completed the first step. The theoretical aspects have been analysed in details and the first version of *Styx* (STYX, 2006) has been implemented. However, there are still at least two steps to go. At the beginning of each step, the current version of *Styx* will be demonstrated to school children and their teachers. We expect that their feedback will stimulate us to improve the system in such a way that it will become a real educational toy.

Though having no feedback yet, we are already aware of some issues that need to be addressed:

- During the morphological analysis, the user selects only the part of speech for the given word and *Styx* itself provides the relevant morphological categories to analyse. In this fashion, the exercises are too simplified. To master the morphological analysis, the user has to know what categories are relevant to the given part of speech.



U docentů a asistentů je to ještě horší .

	řešení	zadáno	
slovní tvar	docentů		
základní tvar	docent		
slovní druh	podstatné jméno	podstatné jméno	OK
rod	mužský	mužský	OK
číslo	množné	množné	OK
pád	druhý	čtvrtý	#

Figure 17: *Styx* – exercise evaluation

- Revision must be made for the syntactic phenomena such as the complement and the coordination. There are the issues in which the PDT and the school approach differ the most.
- *Charon* should give a possibility of selecting sentences which contain some specific phenomenon. Currently, an administrator goes through all the sentences manually and if they fulfill his or her selection criteria he or she includes them into the exercises.

To our knowledge, at least, there is no such system for any language corpus that makes the school-children familiar with an academic product. At the same time, our system represents a challenge and an opportunity for the academicians to popularize a field devoted to natural language processing with promising future.

References

Hajičová, Eva, Zdeněk Kirschner, and Petr Sgall. 1999. A manual for analytic layer annotation of the prague dependency treebank. Technical report, ÚFAL MFF UK, Prague, Czech Republic. English translation.

Hladká, Barbora and Ondřej Kučera. 2005. Prague Dependency Treebank as an exercise book of Czech. *Proceedings of HTL/EMNLP 2005 Interactive Demonstrations, Vancouver, BC, Canada*.

Jeřábek, Jaroslav and Jan Tupý et al. 2005. *The official framework educational programme for general secondary education*. Research pedagogical institute, Prague.

Kučera, Ondřej. 2006. Pražský závislostní korpus jako cvičebnice jazyka českého. Master's thesis, Charles University, Prague, Czech Republic.

Mikulová, Marie et al. 2006. Anotace pražského závislostního korpusu na tektogramatické rovině: pokyny pro anotátory. Technical Report 28, ÚFAL MFF UK, Prague, Czech Republic.

Pajas, Petr and Jan Štěpánek. 2005. A generic xml-based format for structured linguistic annotation and its application to prague dependency treebank 2.0. Technical Report 29, ÚFAL MFF UK, Prague, Czech Republic.

PDT 2.0. 2006. http://www.ufal.mff.cuni.cz/pdt2.0. Prague Dependency Treebank, 2nd edition.

STYX. 2006. http://www.ufal.mff.cuni.cz/styx. The STYX system: an electronic exercise book of Czech.

Zeman, Dan, Jiří Hana, Hana Hanová, Jan Hajič, Barbora Hladká, and Emil Jeřábek. 2005. A manual for morphological annotation, 2nd edition. Technical Report 27, ÚFAL MFF UK, Prague, Czech Republic.