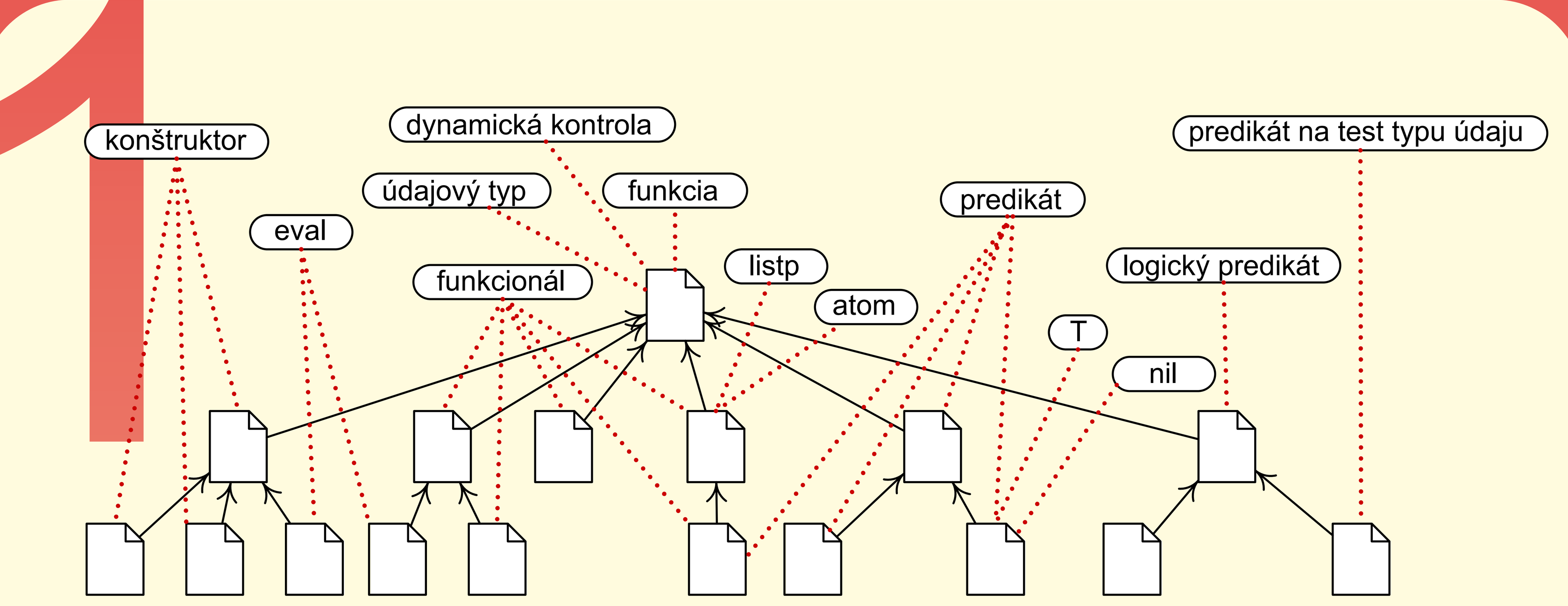
Petra Vrablecová Supervisor: Marián Šimko

STU FIIT

Relationship Discovery in Educational Content

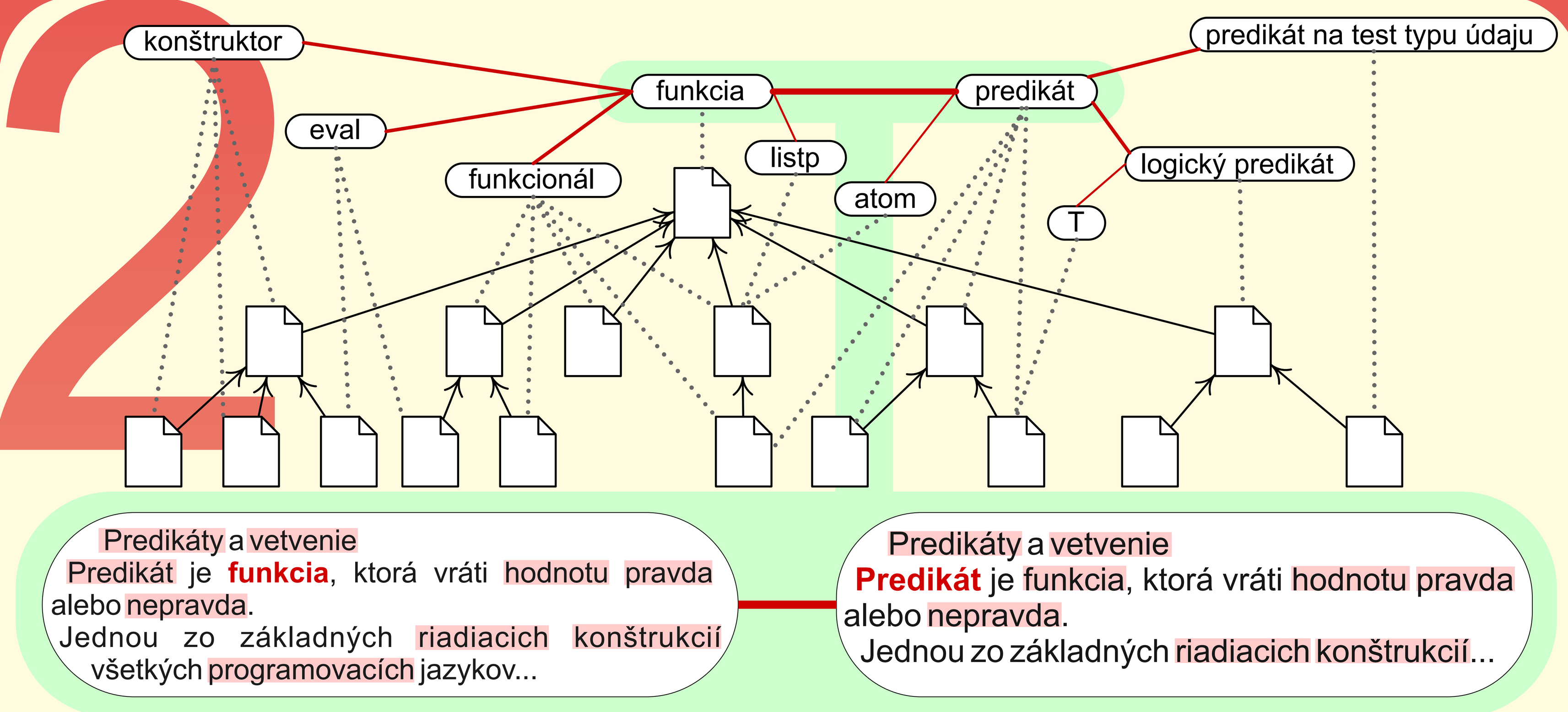
The domain model is an essential part of an **adaptive learning** system. It is needed as a basis for tracking users' progress in learning and adaption of the content accordingly. It expresses the **semantics** of educational content in the form of **metadata**. Manual construction of the complete and correct domain model is a demanding task for the teacher therefore there are attempts to automate it. It consists of relevant domain terms (**RDTs**) and **relationships** between them. The core of the model comprise **hierarchical** relationships (i.e., is-a relationship).

1



The **input** of our method is a set of learning objects (**LOs**) – text documents with **RDTs** – keywords expressing the semantics of the documents' content. The content is firstly preprocessed using methods and techniques of natural language processing.

2

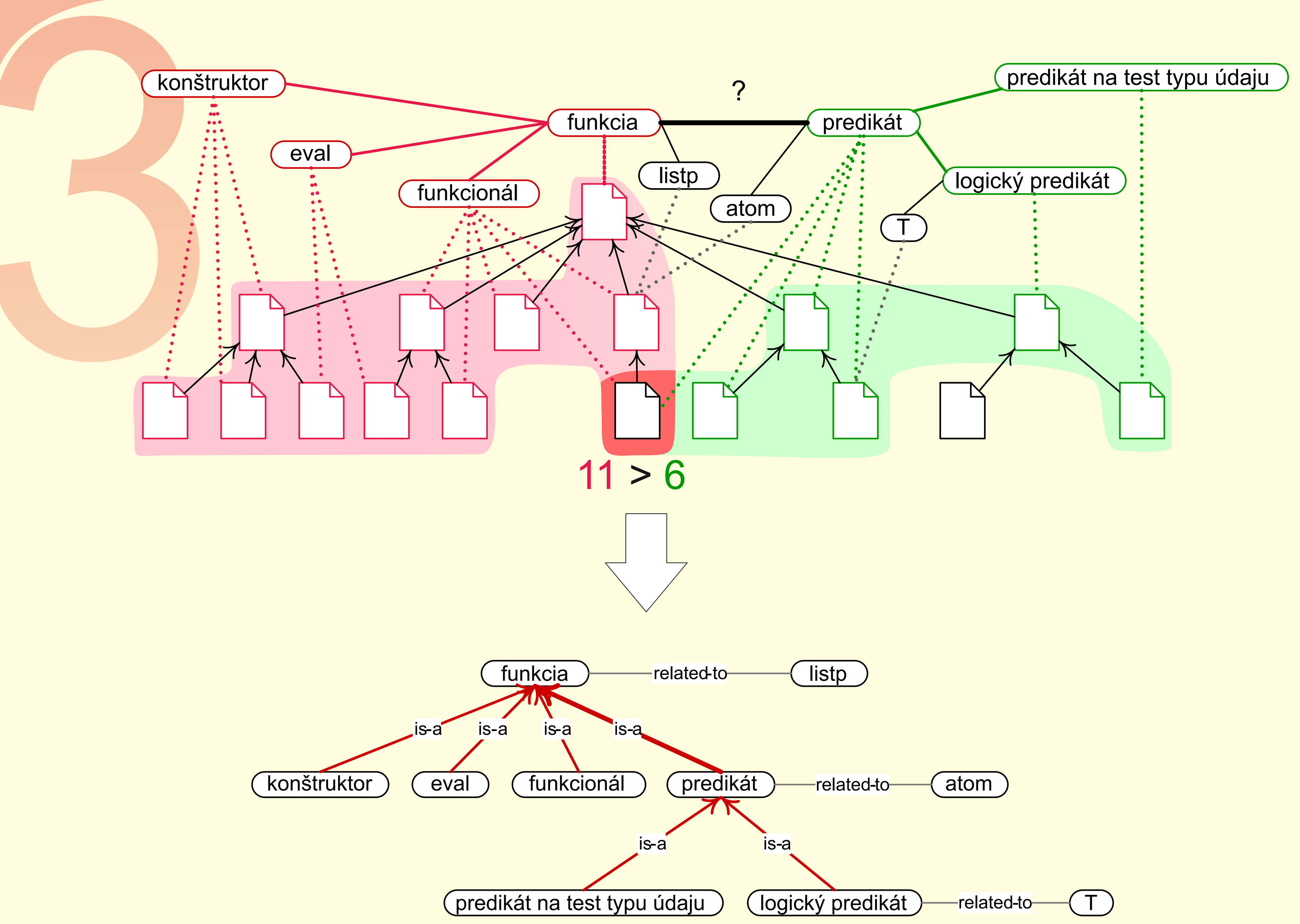


In the next step the **Latent Semantic Analysis** is applied on the content of learning objects. Relationships between RDTs are created based on the similarity of words surrounding RDTs in the text.

We created a **method for discovering relationships** between relevant domain terms in the domain model:

- **facilitates** the process of **domain model acquisition**,
- uses a less explored **statistical approach** (pros: language independent, no need for syntax knowledge),
- suitable for educational content because of its coherent vocabulary and structure,
- great potential to **supplement** methods based solely on **linguistic processing**,
- in the future integrated into the educational content management system to help the university staff.

3



Algorithm 1 The hierarchical relationship discovery algorithm

```
procedure FindHierarchical(RDT-RDTRels, RDT-LORels, RDTs)
    hierarchicalRels ← [ ]
    RDTNeighbours ← GetNeighbors(RDT-RDTRels)

    for all relationship ∈ RDT-RDTRels do
        if relationship[weight] > k then
            fromRDT ← relationship[from]
            fromSet ← GetLOSet(fromRDT, RDTNeighbors[fromRDT], RDT-LORels)

            toRDT ← relationship[to]
            toSet ← GetLOSet(toRDT, RDTNeighbors[toRDT], RDT-LORels)

            if fromSet ∩ toSet ≠ ∅ then
                if |fromSet| > |toSet| then                ▷ fromRDT is more general term
                    hierarchicalRels ← [toRDT, fromRDT, relationship[weight]]

                else if |toSet| > |fromSet| then           ▷ toRDT is more general term
                    hierarchicalRels ← [fromRDT, toRDT, relationship[weight]]

                else if |fromSet| == |toSet| then
                        ▷ if LO sets' cardinalities are equal compare occurrences of RDTs in LOs
                    if TextOccurrences(fromRDT) > TextOccurrences(toRDT) then
                        hierarchicalRels ← [toRDT, fromRDT, relationship[weight]]
                    else if TextOccurrences(toRDT) > TextOccurrences(fromRDT) then
                        hierarchicalRels ← [fromRDT, toRDT, relationship[weight]]
                    end if
                end if
            end if
        end if
    end for
    return hierarchicalRels
end procedure
```

Experiments

We conducted experiments on the course of Functional and Logic programming. The characteristics of data and the current results are shown in the table below. We used topological recall (R_T), topological precision (P_T) and F-measure (F_T) to evaluate our method against the gold standard. The graphs below show the measures for top k generated is-a relationships – Functional Programming (a), Logic Programming (b).

	Functional programming	Logic programming
#learning objects	79	42
#words	28,455	23,383
average length of learning objects	360.19	556.74
#relevant domain terms	162	138
average relevant domain term length	1.70	1.41
average number of RDTs per learning object	1.94	2.10
F_T	0.55	0.46
P_T	0.66	0.77
R_T	0.48	0.33
k	69	38
P_T (existing relationships)	0.86	0.63
R_T (existing relationships)	0.64	0.55