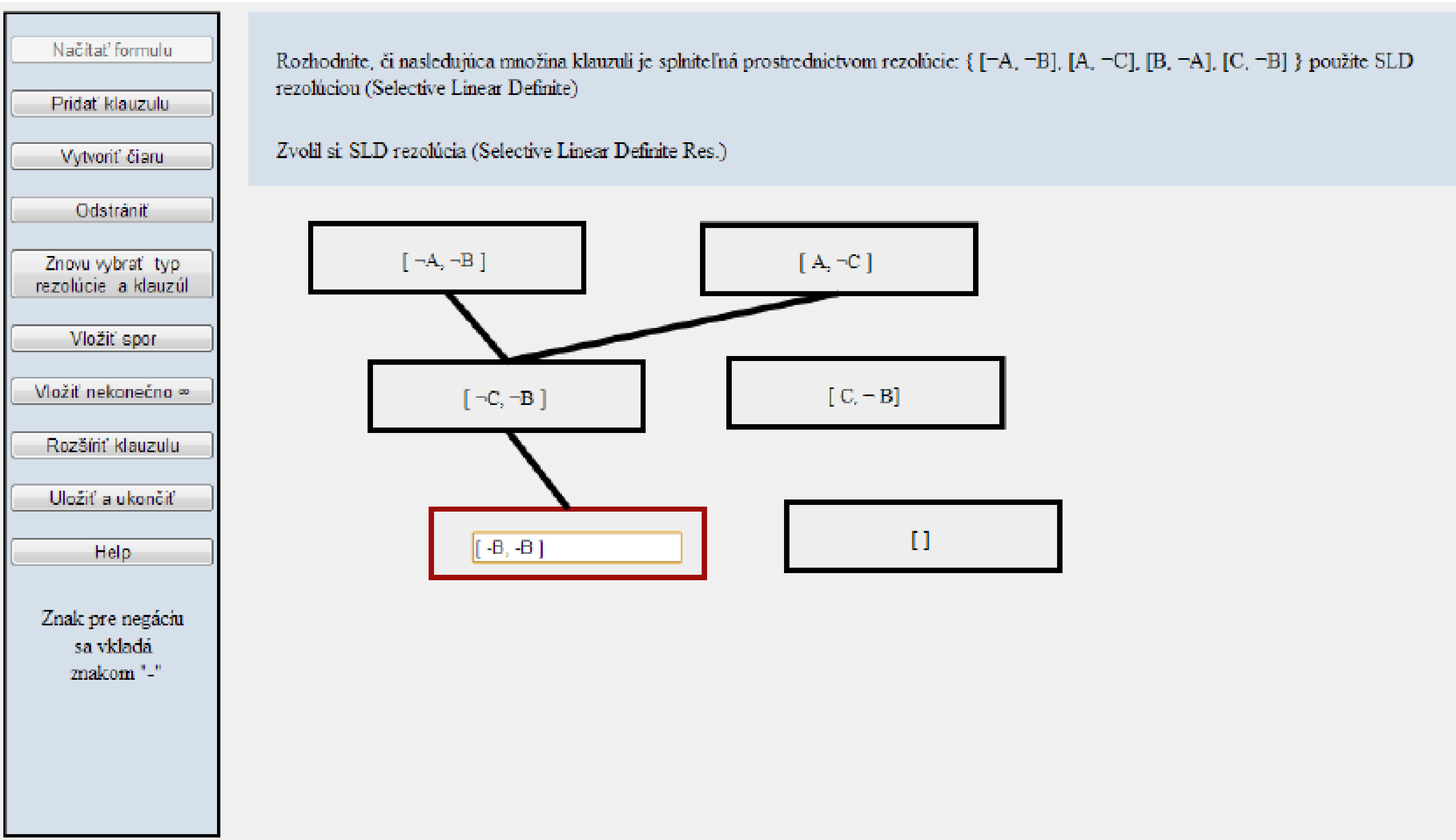
Analysis of resolution proofs (current state)

Karel Vaculík

IA008 Computational Logic

Introduction



Source: VRÁBEL, Patrik. Webové prostředí pro vstup rezolučních důkazů [online]. 2013 [cit. 2013-11-26]. Diplomová práce. Masarykova univerzita, Fakulta informatiky.

Data sources

- Spring 2013: IB101 Introduction to Logic
  - ~ 400 students
- Fall 2013: IA008 Computational Logic
  - ~ 65 students

Current state of analysis

- General overview
- Graph mining

(Use of time component in future work)

General overview

General overview

- Summary of entities:

	GRAPHS	NODES*	EDGES*
SPRING	1385	13259	11709
FALL	410	4213	3764
TOTAL	1795	17472	15473

*without deleted nodes and edges

General overview

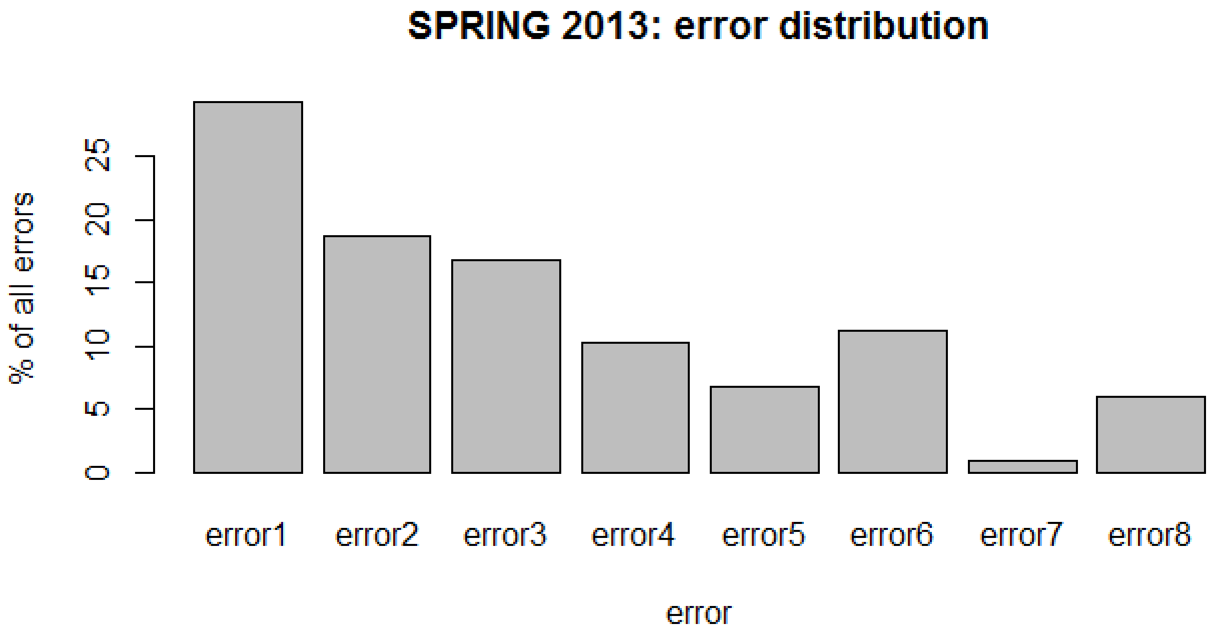
- Errors
  - Simple scripts for finding errors

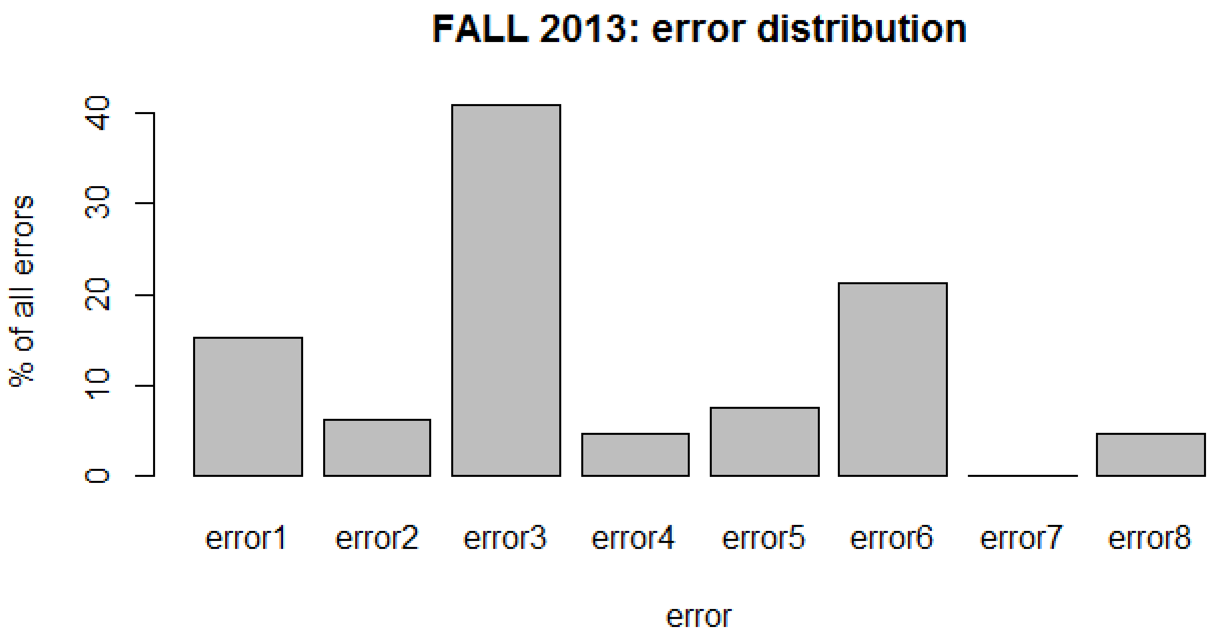
1	Zle priradený typ klauzúl k typu rezolúcie
2	Chýba druhý rodič
3	Rezolvovanie na dvoch literáloch súčasne (nebo i na více literálech)
4	Opakovanie rovnakého literálu v množine
5	Rezolvovanie na rovnakom literály
6	Preklep, alebo vlastné literály
7	Rezolvovanie v rámci jednej klauzuly
8	Žiadna rezolúcia iba spojenie množín

General overview

- Errors
  - Simple scripts for finding errors

	COUNT	% OF GRAPHS
SPRING	321	16.10
FALL	66	23.18
TOTAL	387	21.56





1	Zle priradený typ klauzúl k typu rezolúcie
2	Chýba druhý rodič
3	Rezolvovanie na dvoch literáloch súčasne (nebo i na více literálech)
4	Opakovanie rovnakého literálu v množine
5	Rezolvovanie na rovnakom literály
6	Preklep, alebo vlastné literály
7	Rezolvovanie v rámci jednej klauzuly
8	Žiadna rezolúcia iba spojenie množín

*Each error type counted in graph only once

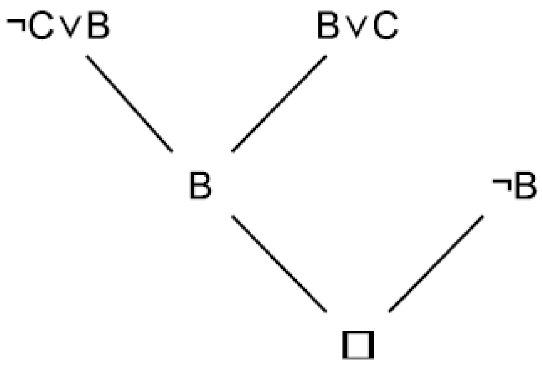
Graph mining

Graph mining – introduction

- Typical data structure for learning algorithms:

attribute 1	**Attribute 2**	**…**	**Attribute n**	**Class**
val 11	val 12	…	val 1n	class i1
…	…	…	…	…
val m1	val m2	…	val mn	class im

- Resolution proofs:

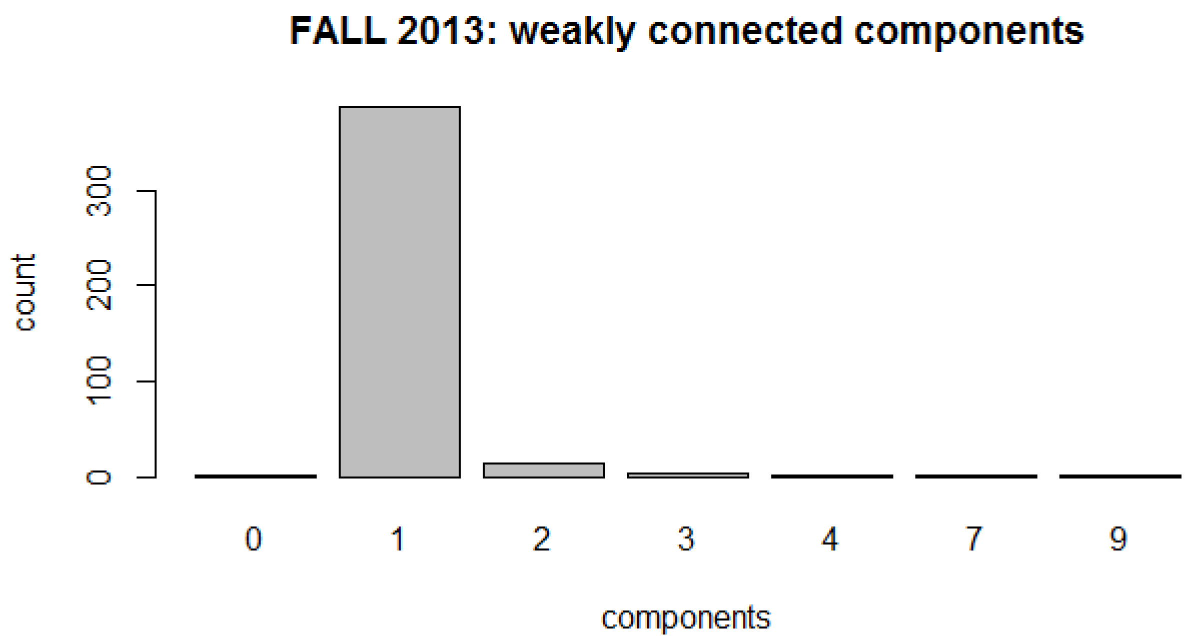


Preprocessing

- Create graphs from tables with nodes and edges
- Replace malformed nodes with “unallowed”

PREPROCESSING

- Weakly connected components:



Preprocessing

- Keep graphs with 1 component
- Keep only binary trees
- New summary:

	BINARY TREES	ORIG. GRAPHS
SPRING	1130	1385
FALL	336	410
TOTAL	1466	1795

Preprocessing

- Append to each tree:
  - List of errors
  - Correctness flag (no error)
  - Quetion type (SLD, linear, general, ...)
  - Used clause type (set, ordered list)

Classification

- Using updated application from VACULÍK, Karel. Dolování z grafů pro podporu výuky [online]. 2013 [cit. 2013-11-27]. Diplomová práce. Masarykova univerzita, Fakulta informatiky.
- New data structure for learning algorithms:

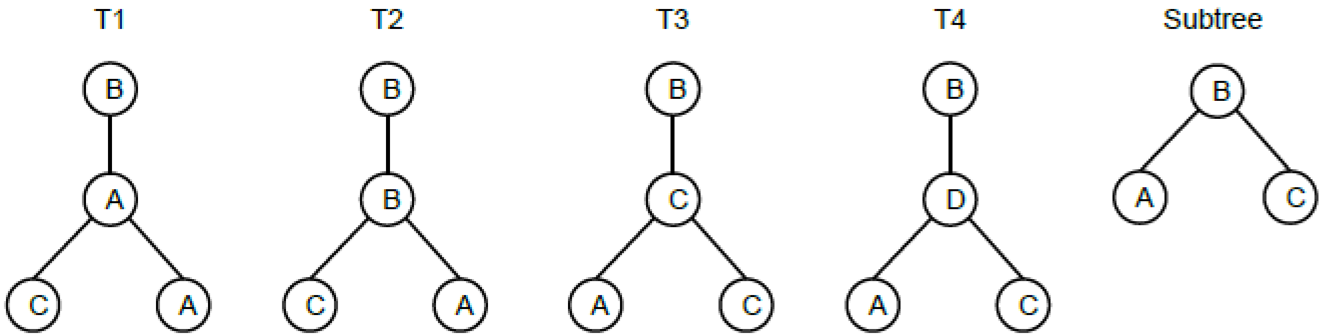
pattern 1	**pattern 2**	**...**	**pattern m**	**Addit. attr. 1**	**...**	**Addit. attr. k**	**Class**
true	false	...	false	Val 11	...	Val 1k	class i1
...	...	...	...	...	...	...	...
false	true	...	true	Val mn	...	Val mk	class im

Classification

- Patterns – two approaches:
  - Frequent subgraphs
  - Generalized subgraphs

Frequent subgraphs

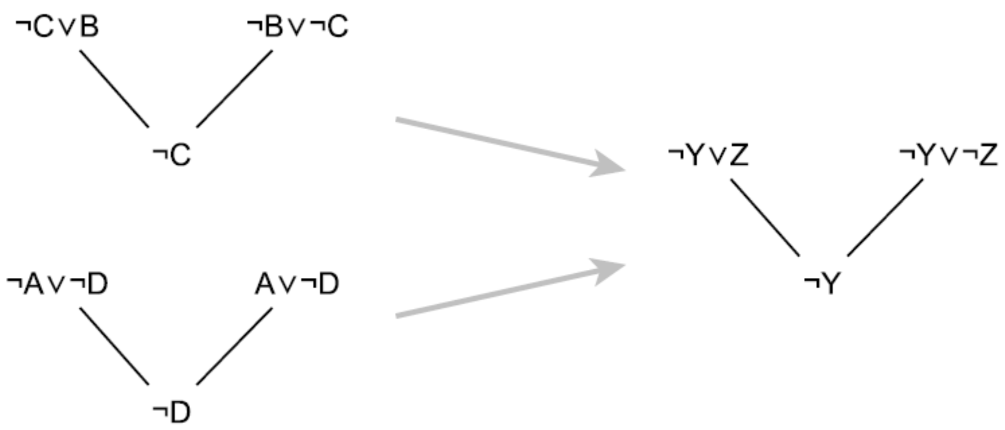
- Algorithm for frequent subgraph mining
- Specifically, mining from unordered rooted trees
- Given the minimum support, find all frequent subtrees in a set of trees
- Example (min. sup. 25%):



- Application employs Sleuth algorithm (Mohammed J. Zaki, Efficiently Mining Frequent Embedded Unordered Trees. Fundamenta Informaticae, 66(1-2):33-52. Mar/Apr 2005.)

GENERALIZED SUBGRAPHS

- Find all 3-node subgraphs representing the resolution step
- Generalize found subtrees and merge them



- Filter out infrequent subtrees

Experiments

- All binary trees (1466)
- Classes:
  - Incorrect proof (187)
  - Correct proof (1279)
- Additional attributes:
  - Clause type (set, ordered list)
  - Resolution type (SLD, linear, general, ...)
- Generalized patterns (min support 0%)
- Oversampling

Experiments

- Classification algorithms
  - J48
  - Naive Bayes
  - SMO
  - IBk
- Evaluation: 10-fold cross validation
- Best result so far:
  - J48
  - Accuracy: 96.9%

Future work

- Additional analyses
- Other graph representations
- Sequence mining