Dolování z grafů pro podporu výuky

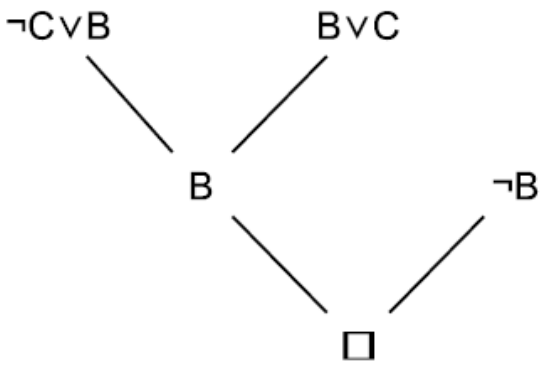
Diplomová práce
FI MU Brno 2013

Autor: Karel Vaculík
Vedoucí práce: doc. RNDr. Lubomír Popelínský, Ph.D.

Motivace

- Konstrukční úlohy řešené studenty (rezoluční důkazy, tablové důkazy, ...)
  - Velké množství
  - Stromová struktura
  - Užitečné je vystihnout zajímavé souvislosti

  ⇒ Použití metod pro dolování z grafů



Cíle práce

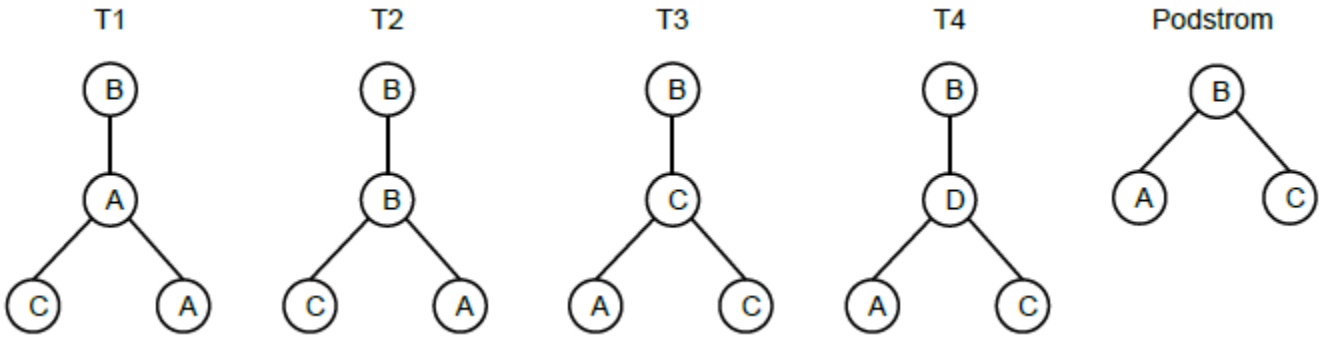
- Přehled metod pro dolování z grafů se zaměřením na stromy
- Návrh a implementace systému pro dolování ze stromů pro klasifikaci řešených úloh v logice
- Analýza rezolučních důkazů ve výrokové logice
- Ověření systému na řešených úlohách z kursů logiky na FI

Dolování ze stromů

- Především dolování častých podstromů
- Stromy: volné, (ne)uspořádané kořenové
- Podstromy (kořen. stromy): induced, embedded

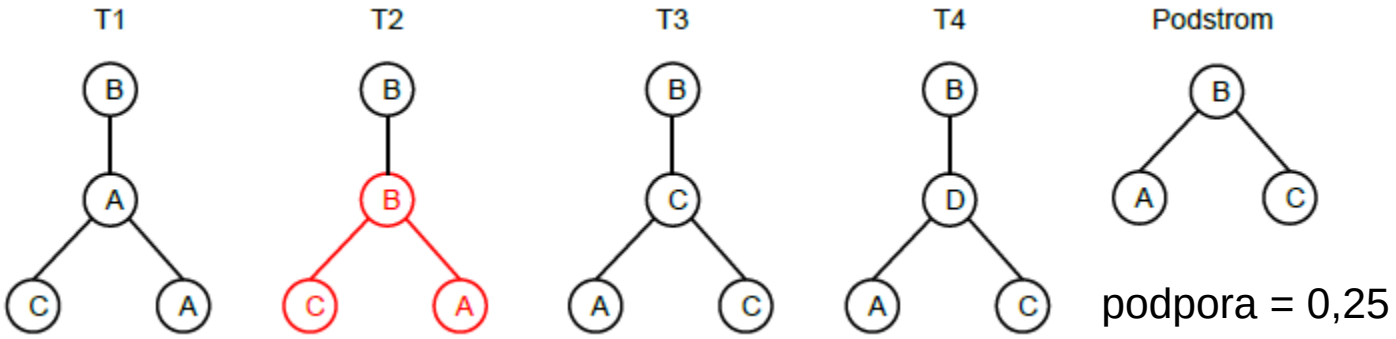
Dolování ze stromů

- Především dolování častých podstromů
- Stromy: volné, (ne)uspořádané kořenové
- Podstromy (kořen. stromy): induced, embedded
- Př: kořenové neuspořádané stromy:



Dolování ze stromů

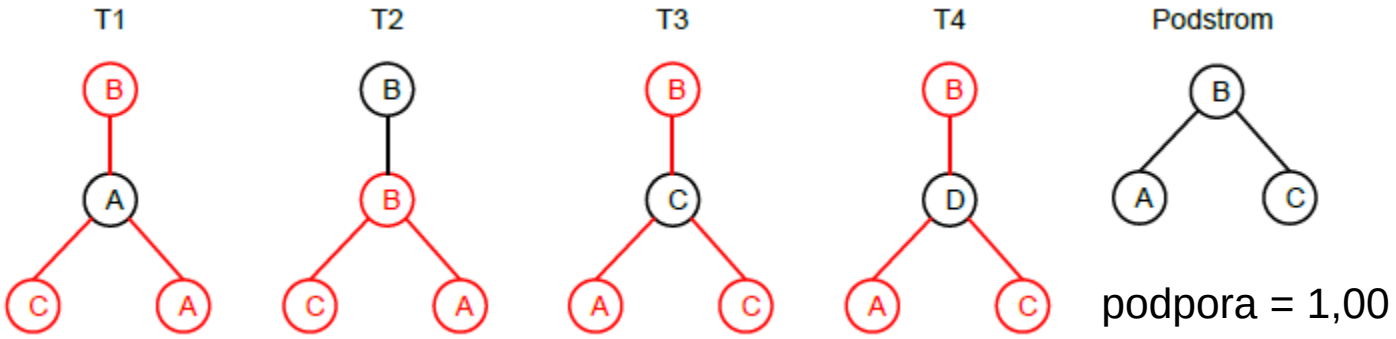
- Především dolování častých podstromů
- Stromy: volné, (ne)uspořádané kořenové
- Podstromy (kořen. stromy): induced, embedded
- Př: kořenové neuspořádané stromy:



(podpora = počet výskytů / celkový počet stromů)

Dolování ze stromů

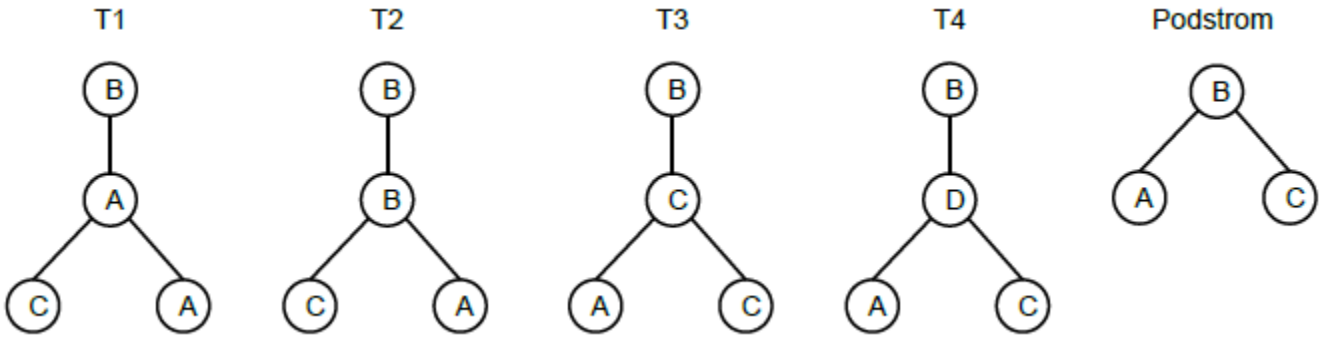
- Především dolování častých podstromů
- Stromy: volné, (ne)uspořádané kořenové
- Podstromy (kořen. stromy): induced, embedded
- Př: kořenové neuspořádané stromy:



(podpora = počet výskytů / celkový počet stromů)

Dolování ze stromů

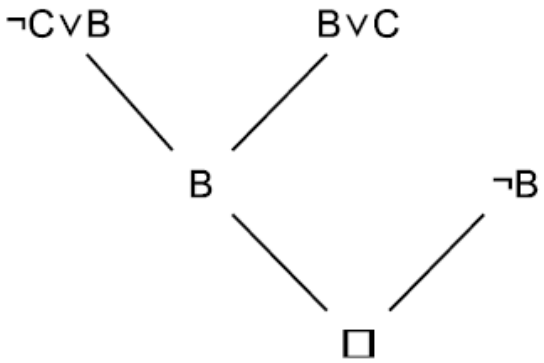
- Především dolování častých podstromů
- Stromy: volné, (ne)uspořádané kořenové
- Podstromy (kořen. stromy): induced, embedded
- Př: kořenové neuspořádané stromy:



- Úloha: nalézt všechny časté podstromy splňující zvolenou minimání podporu

Dolování ze stromů

- V rámci této práce:
  - Neuspořádané kořenové stromy
  - Induced podstromy



Algoritmy pro dolování ze stromů

- FreeTreeMiner
- TreeMiner
- Freqt
- uFreqt
- Unot
- PathJoin
- HybridTreeMiner
- Sleuth

Algoritmy pro dolování ze stromů

- FreeTreeMiner
- TreeMiner
- Freqt
- uFreqt
- Unot
- PathJoin
- HybridTreeMiner
- Sleuth

FreeTreeMiner: Pouze volné stromy

TreeMiner, Freqt: Pouze uspořádáné stromy

uFreqt, Unot, PathJoin: Nejsou k dispozici

HybridTreeMiner: Nevhodný výstup

Vytvořený systém

Nový systém obsahující moduly pro:

- Předzpracování dat
- Dolování častých podstromů (pomocí SLEUTH)
- **Klasifikaci rezolučních důkazů**
- Vizualizaci stromů s postromy a rozhodovacího stromu pro klasifikaci

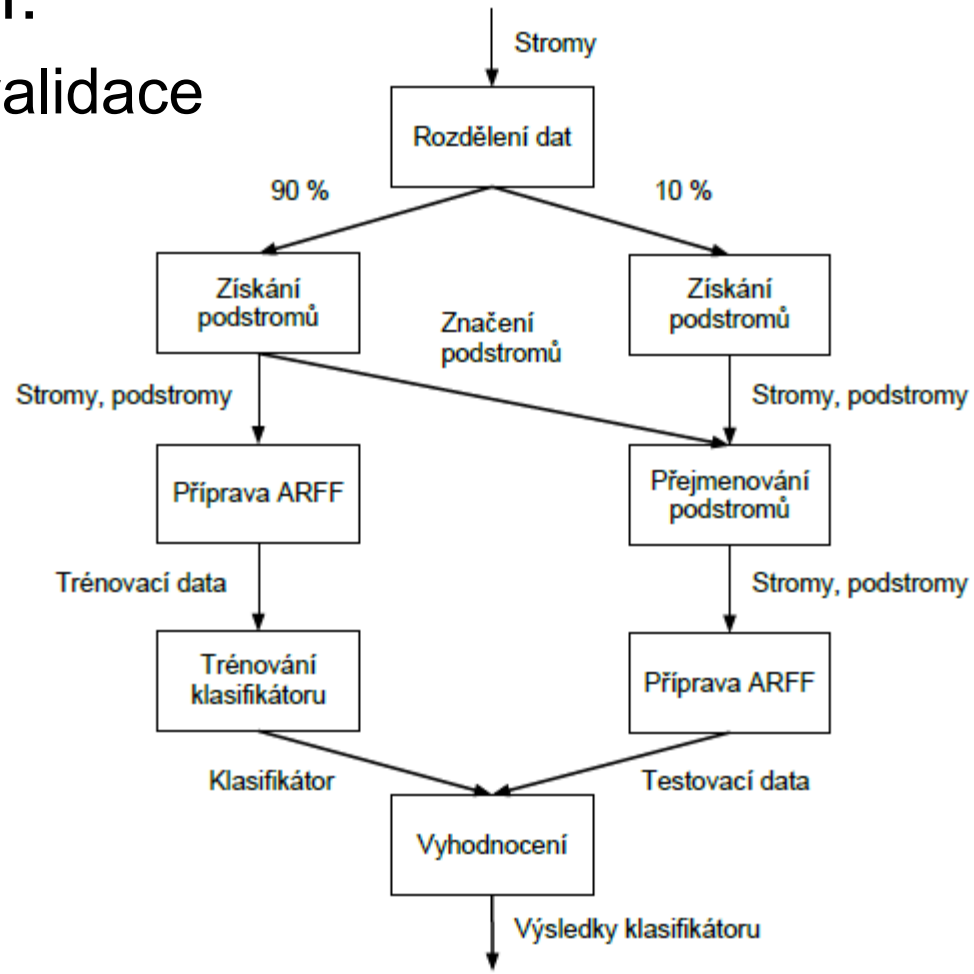
Klasifikace

- Stromy (důkazy) reprezentovány podstromy

č. stromu	$podstrom_1$	$podstrom_2$	...	$podstrom_m$	třída
1	true	false	...	false	$třída_i$
...	...	...	...	...	...
n	false	true	...	true	$třída_j$

Klasifikace – postup

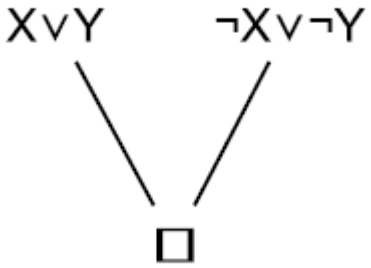
- Způsob vyhodnocení:
  - 10-složková křížová validace
- Získání podstromů:
  1. Časté podstromy (SLEUTH)
  2. Emergentní vzory
  3. Zobecnění
- Klasifikátory z Weka



Klasifikace – emergentní vzory

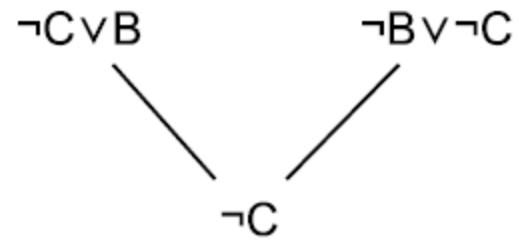
- Emergentní vzor – vzor (podstrom), jehož podpora se výrazně liší pro různé soubory dat (pro klasifikaci rozdělení podle tříd)
- Používaná metrika:
  - $\mathrm{GrowthRate}_{\mathrm{DATA1}}(\mathrm{Pattern}) = \frac{\mathrm{support}_{\mathrm{DATA1}}(\mathrm{Pattern})}{\mathrm{support}_{\mathrm{DATA2}}(\mathrm{Pattern})}$
- V systému přidáno:
  - $\mathrm{support}_{\mathrm{DATA1}}(\mathrm{Pattern})$
  - Počet uzlů v $\mathrm{Pattern}$

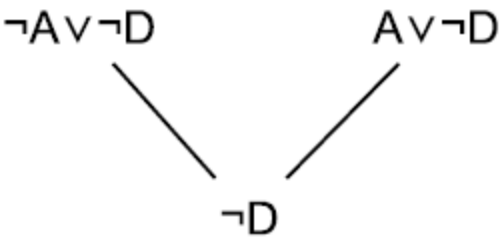
Př. Emergentního vzoru:



Klasifikace – zobecněné vzory

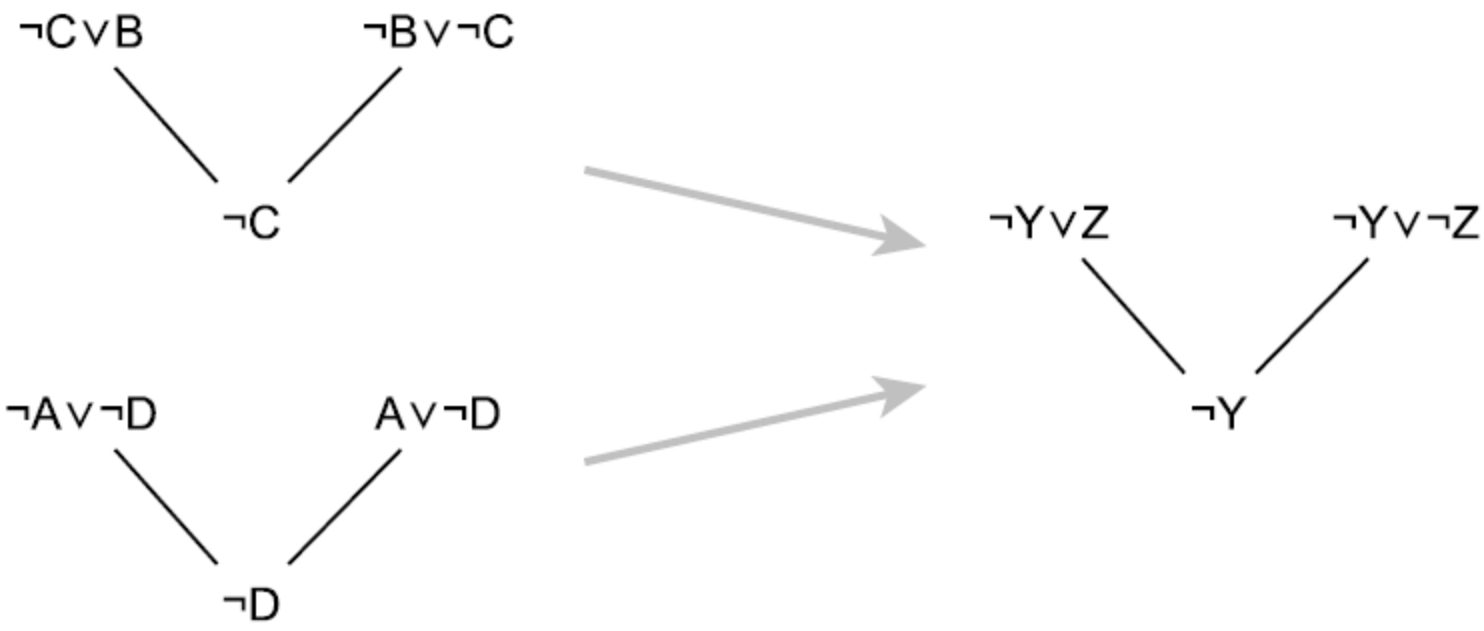
- Problém: stejné řešení zachyceno více způsoby





Klasifikace – zobecněné vzory

- Problém: stejné řešení zachyceno více způsoby
- Řešení: zobecnění podstromů pro sjednocení stejných struktur

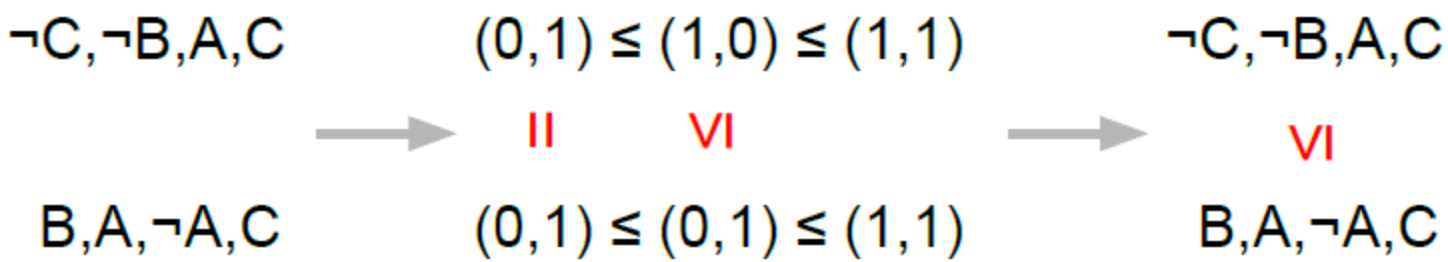


Klasifikace – zobecněné vzory

- Pouze 3-uzlové vzory (aplikace rezolučního pravidla)
- Uspořádání na znacích literálů
  - (|*negativní literály*|, |*pozitivní literály*|)

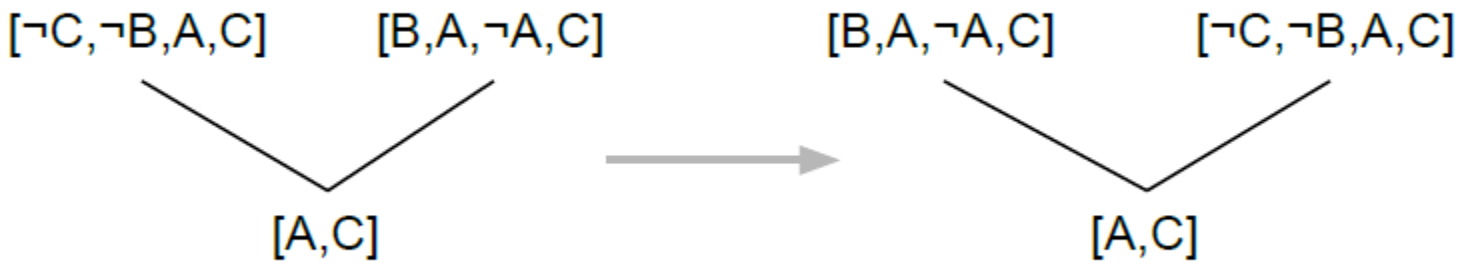
$$\neg C,\neg B,A,C \longrightarrow (0,1)_A \leq (1,0)_B \leq (1,1)_C \longrightarrow A \leq B \leq C$$

- Uspořádání na uzlech



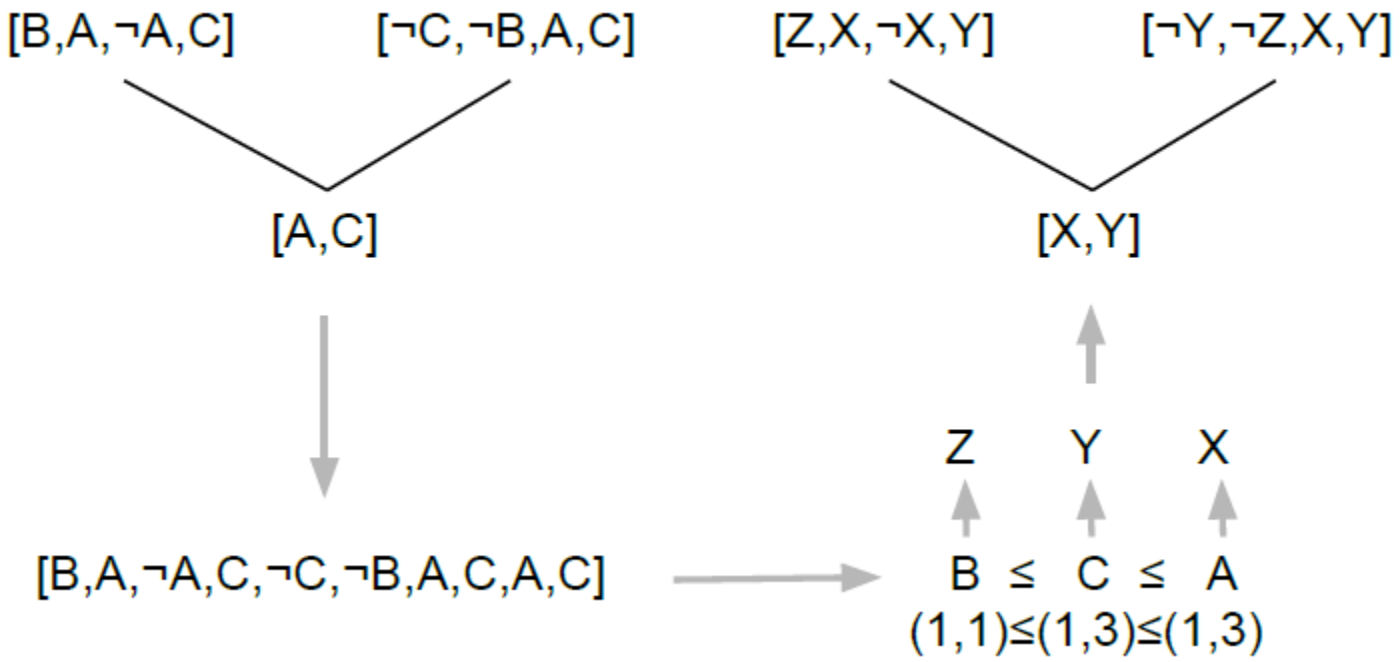
Klasifikace – zobecněné vzory

- Postup:
  1. Uspořádej rodiče



Klasifikace – zobecněné vzory

- Postup:
  1. Uspořádej rodiče
  2. Nahraď znaky literálů novými proměnnými



Klasifikace – zobecněné vzory

- Postup:
  1. Uspořádej rodiče
  2. Nahraď znaky literálů novými proměnnými
  3. Uspořádej literály v uzlech ($Z, \neg Y \sim \neg Y, Z$).

Experimenty – použitá data

- 393 řešených rezolučních důkazů ve formátu GraphML
- Zdroj: písemné testy z IB101 – Úvod do logiky a logického programování
- 2 zadání (183 + 210 stromů)
- Stromům přiřazena třída:
  - Positive – správně provedený důkaz (322 případů)
  - Negative – špatně provedený důkaz (71 případů)
- Další atributy: počet získaných bodů, typ provedené rezoluce, počty výskytů různých druhů chyb, ...

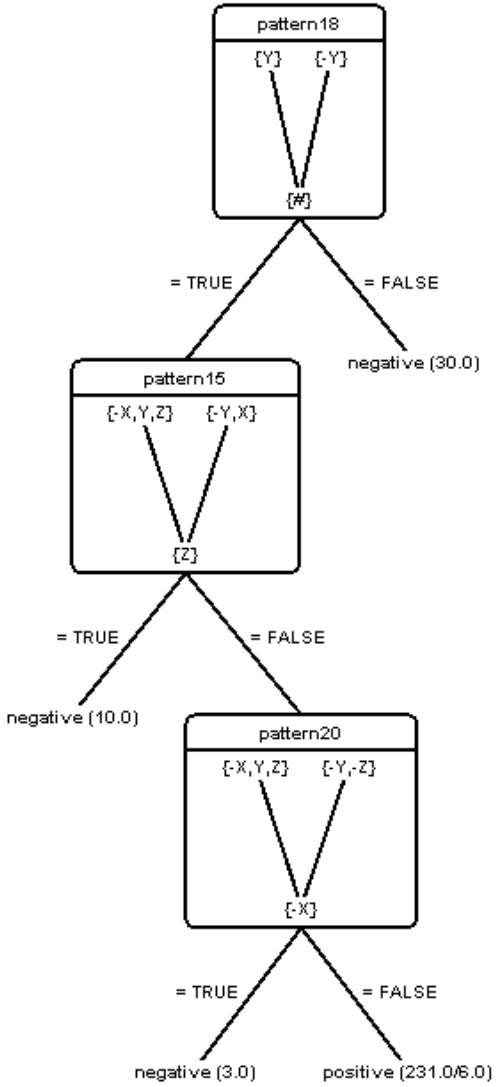
Experimenty – výsledky

- 10-složková křížová validace, baseline: 81,9 %
- Zobecněné vzory:

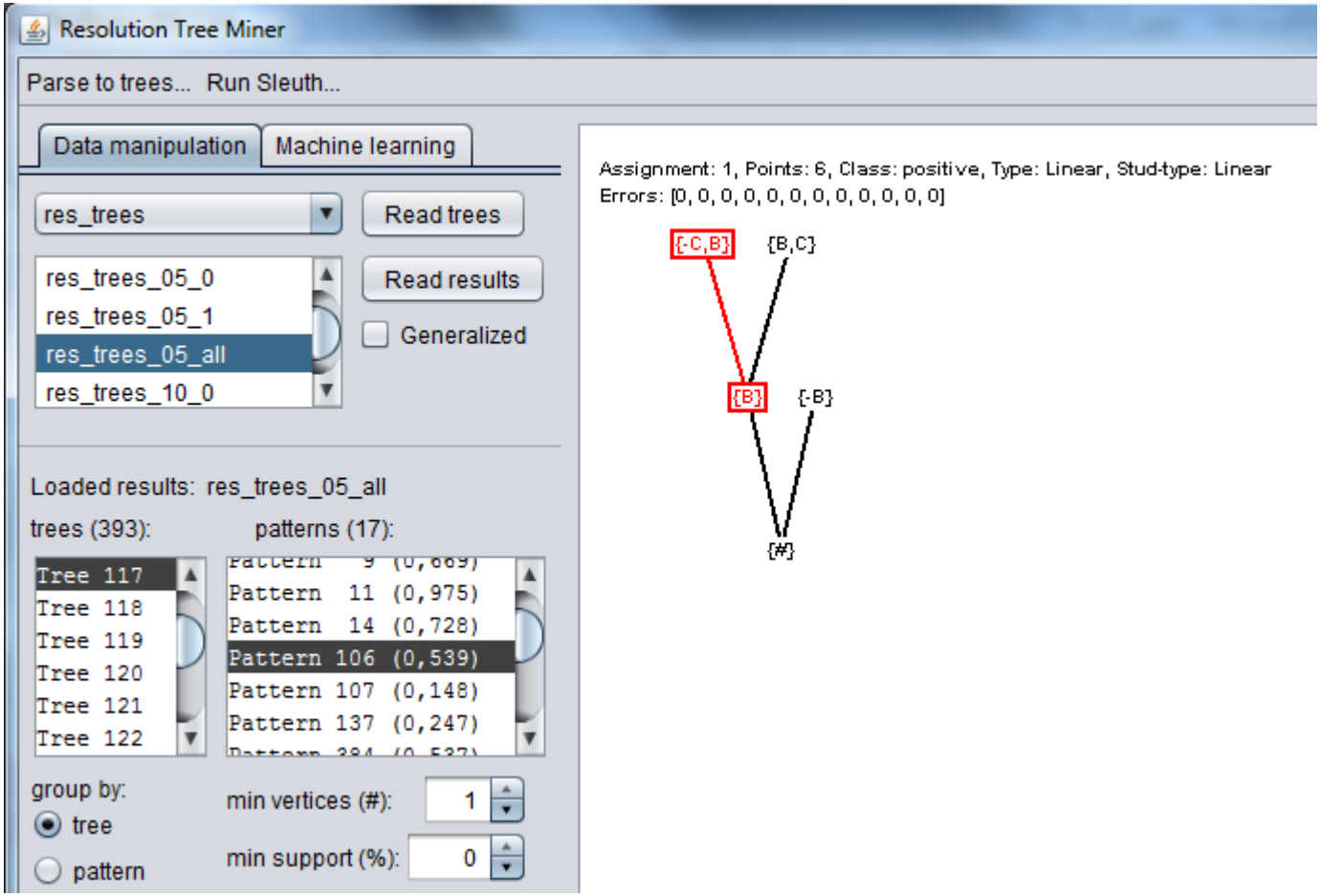
Alg.	Min. podpora (%)	Správnost (%)	Přesnost (positive)	Úplnost (positive)	Přesnost (negative)	Úplnost (negative)
J48	0	97,2	0,970	0,997	0,986	0,862
Naive Bayes	1	96,7	0,965	0,997	0,986	0,832
SMO	0	97,5	0,973	0,997	0,988	0,873
IBk	5	96,7	0,970	0,991	0,955	0,862

- Zobecněné emergentní vzory – srovnatelné výsledky

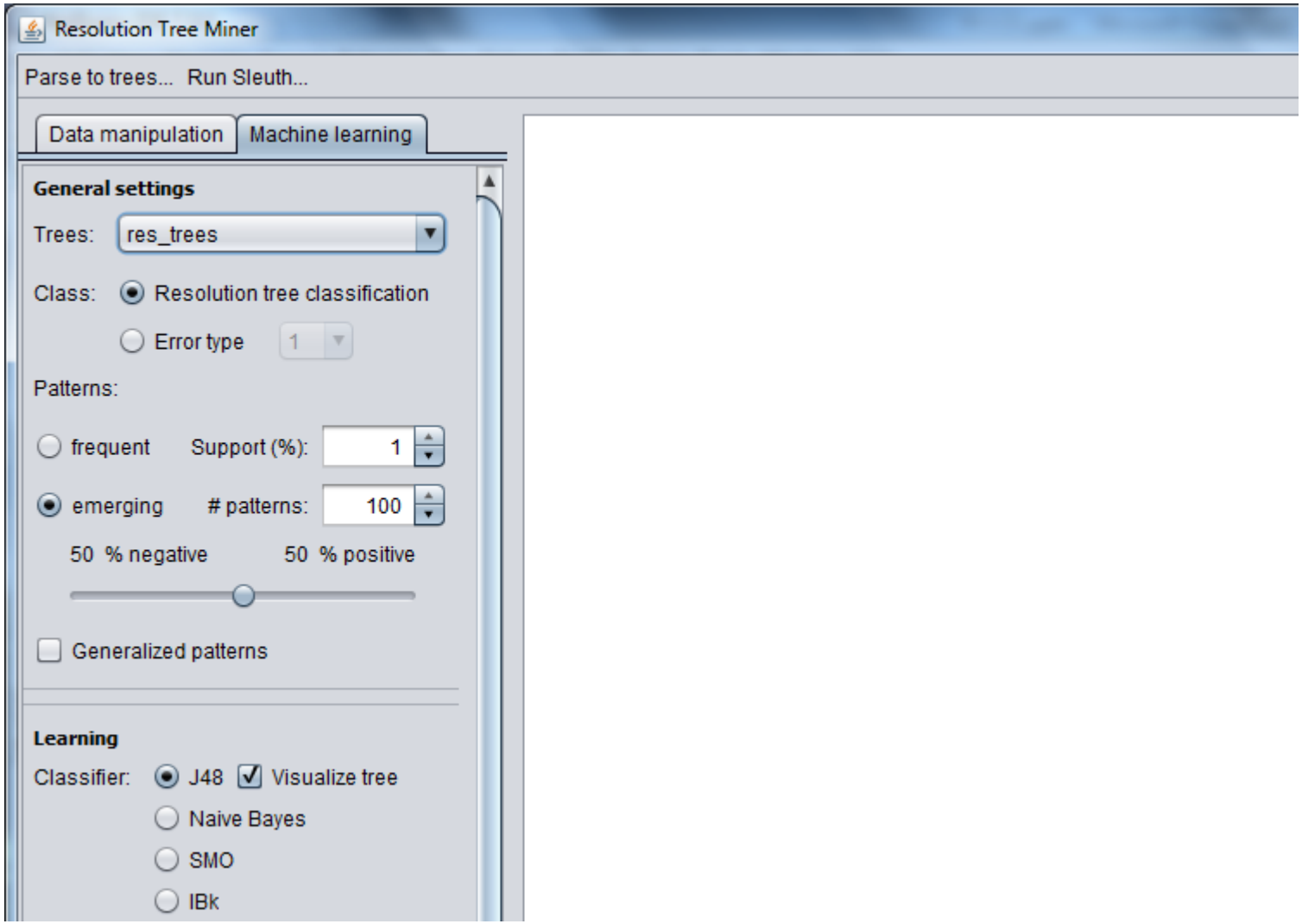
Experimenty – ukázka rozhodovacího stromu



Ukázka systému – průzkum dat



Ukázka systému – modul klasifikace



Závěr

- Vytvořen nový systém pro dolování ze stromů využívající častých podstromů
- Navržena a ověřena nová metoda pro generalizaci podstromů
- Implementován vlastní způsob křížové validace
- Vytvořen modul pro předzpracování, vizualizaci stromů, podstromů a rozhodovacího stromu
- Dosaženo správnosti 97 % na reálných datech
- V blízké době: použití pro dolování z rozšířených dat a dolování z tablových důkazů

Děkuji za pozornost

Otázky – složitost klasifikace

1. Rozdělení dat podle tříd: ***O*(N)**
2. Pro každou třídu *shuffle* (lineární), celkem tedy: ***O*(N)**
3. Rozdělení do 10 složek – pro jednotlivé třídy postupně plněny složky, celkem: ***O*(N)**
4. **10x:**
    1. Naplnění training a test množin (projde všechny a postupně je tam vloží): ***O*(N)**
    2. Výstup pro Sleuth (zakódování): ***O*(V)**
    3. Použití SLEUTH: ***O*(X)**
    4. Načtení výsledků (lineární k délce souboru): ***O*(Y)** (přitom M-krát je sestaven podstrom v lineárním čase vzhledem k počtu uzlů)
    5. Extrakce emergentních vzorů:
        1. Evaluace vzorů (pos / neg stromy): ***O*(N*P + M)**
        2. Uspořádání pro 2 třídy: ***O*(2*M*log(M))**
        3. Výběr určitého počtu vzorů: ***O*(M)**
    6. Zobecnění vzorů:
        1. Výběr pouze 3-uzlových (porovnávání kódů vzorů): ***O*(M)**
        2. Přejmenování vzorů: ***O*(M*A)**
        3. Odstranění duplicit: ***O*(M*M)**
    7. Indexy vzorů ke stromům: ***O*(N*M)**
    8. Přejmenování v test data: ***O*(N*P*M)**
    9. Výstup do ARFF: ***O*(N*M)**
    10. Sestavení a otestování klasifikátoru: ***O*(C)**

N – počet stromů,
M – počet nalezených vzorů,
P – max. počet vzorů ve stromu,
V – počet vrcholů ze všech stromů,
X – složitost Sleuth,
Y – načtení výsledků Sleuth,
A – max. složitost přejmenování,
C – složitost klasifikátoru

Celkem: $O(3N + 10*(N + V + X + Y + N*P + 3M + 2M\log(M) + M*A + M*M + 3*N*M + N*P*M + C))$

Otázky – složitost klasifikace

1. Rozdělení dat podle tříd: ***O*(N)**
2. Pro každou třídu *shuffle* (lineární), celkem tedy: ***O*(N)**
3. Rozdělení do 10 složek – pro jednotlivé třídy postupně plněny složky, celkem: ***O*(N)**
4. **10x:**
    1. Naplnění training a test množin (projde všechny a postupně je tam vloží): ***O*(N)**
    2. Výstup pro Sleuth (zakódování): ***O*(V)**
    3. Použití SLEUTH: ***O*(X)**
    4. Načtení výsledků (lineární k délce souboru): ***O*(Y)** (přitom M-krát je sestaven podstrom v lineárním čase vzhledem k počtu uzlů)
    5. Extrakce emergentních vzorů:
        1. Evaluace vzorů (pos / neg stromy): ***O*(N*P + M)**
        2. Uspořádání pro 2 třídy: ***O*(2*M*log(M))**
        3. Výběr určitého počtu vzorů: ***O*(M)**
    6. Zobecnění vzorů:
        1. Výběr pouze 3-uzlových (porovnávání kódů vzorů): ***O*(M)**
        2. Přejmenování vzorů: ***O*(M*A)**
        3. Odstranění duplicit: ***O*(M*M)**
    7. Indexy vzorů ke stromům: ***O*(N*M)**
    8. Přejmenování v test data: ***O*(N*P*M)**
    9. Výstup do ARFF: ***O*(N*M)**
    10. Sestavení a otestování klasifikátoru: ***O*(C)**

N – počet stromů,
M – počet nalezených vzorů,
P – max. počet vzorů ve stromu,
V – počet vrcholů ze všech stromů,
X – složitost Sleuth,
Y – načtení výsledků Sleuth,
A – max. složitost přejmenování,
C – složitost klasifikátoru

Pro M >> N a M >> P
Celkem: *O*(V + X + Y + C + M*A + M*M + N*P*M)

Otázky – složitost přejmenování

1. Uspořádání rodičovských klauzulí:
   1. Zjištění počtu negativních a pozitivních literálů: ***O*(L)**
   2. Uspořádání literálů v klazulích: ***O*(L*log(L))**
   3. Porovnání dvou uspořádání: ***O*(L)**
2. Nahrazení proměnnými:
   1. Seskupení do jednoho seznamu: ***O*(L)**
   2. Uspořádání v seznamu: ***O*(L*log(L))**
   3. Přejmenování: ***O*(L)**
3. Lexikografické uspořádání: ***O*(L*log(L))**

L – počet literálů pro všechny 3 uzly

Celkem: $O(4L + 3L*\log(L))$