1 Induktivní inference v predikátové logice

Stále pracujeme s jazykem Hornových klauzulí jak pro data, tak pro naučené hypotézy.

Příklad 1.1: Pro příklad z přednášky týkající se kategorizace trojciferných čísel

př.	139	319	854	468	349	561	756	789	987	256	189	354
klas.	+	−	−	+	+	−	−	+	−	+	+	−

a) vytvořte doménovou znalost

b) vytvořte specializační graf

c) porovnejte s rozhodovacím stromem pro původní zadání

d) pro korektní a úplné řešení: jaký (minimální) počet příkladů potřebujeme pro naučení rozhodovacího stromu a jaký počet pro pravidlo v predikátové logice?

Řešení 1.1:

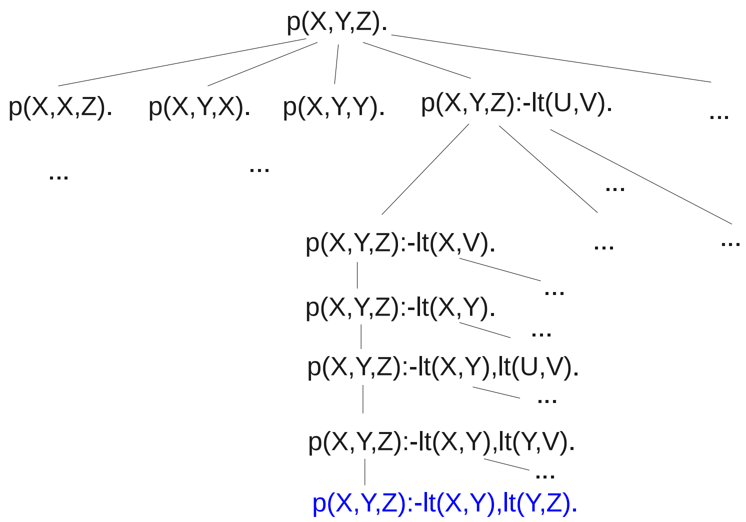
a) Víme, že kategorizace souvisí s uspořádáním cifer v číslech, pro jednoduchost dále předpokládáme, že není použita nula (v příkladech se nevyskytuje). Čísla budeme reprezentovat po jednotlivých cifrách.
Jako doménovou znalost bychom mohli použít například vestavěný operátor `</2`, ale použijeme jiné řešení, kdy si predikát představující doménovou znalost zadefinujeme:
```
lt(1,2). lt(2,3). ... lt(8,9).
lt(1,3). lt(2,4). ...
...
```

b) Běžně budeme používat čtyři specializační operátory, jejich použití si nejprve ukážeme na specializaci klauzule `p(X,Y).`

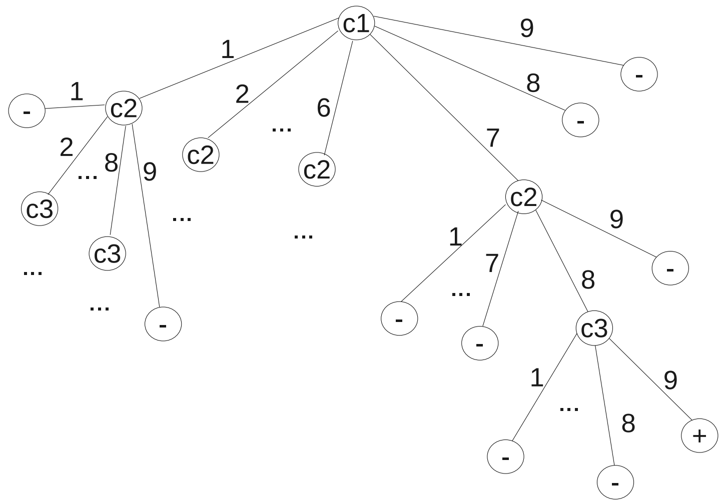
- ztotožnění dvou proměnných
  `p(X,X).`
- přidání podcíle z doménové znalosti (nebo aktuálně odvozovaného predikátu) s dosud nepoužitými proměnnými do těla klauzule
  `p(X,Y) :- p(U,V).`
  nebo
  `p(X,Y) :- r(Z). % pokud je v domenove znalosti predikat r/1`
- konzistentní nahrazení proměnné konstantou
  `p(X,[]).`
  Typicky jen pokud víme, které proměnné a jakými konstantami má smysl nahrazovat (to vypozorujeme v učicích příkladech). V tomto případě nevyužijeme.
- konzistentní nahrazení proměnné nejobecnějším termem
  `p(X,[H|T]).`
  Platí obdobné podmínky, jako pro předchozí specializační operátor.

Specializační graf má podobnou strukturu jako ve výrokové logice, jenže je typicky mnohem rozsáhlejší. Proto se často zakresluje jen jeho relevantní část, tj. ta část v podobě stromu, která ukazuje, jakými postupnými specializacemi byla nalezena výsledná teorie.

V reálných úlohách se graf konstruuje na základě učicích příkladů tak, že se pokračuje pouze větvemi, které pokrývají alespoň jeden pozitivní příklad. Jakmile uzel pokrývá pouze pozitivní příklady, přidá se do teorie a dále se nespecializuje. Proces končí, když teorie pokrývá všechny pozitivní příklady. Konkrétní algoritmy používají pro určování "perspektivních" větví řadu různých optimalizačních technik.



c) Pomocí rozhodovacích stromů se rovněž řeší kategorizační úlohy, zde se snažíme ukázat, že v tomto případě je řešení pomocí induktivní inference výrazně efektivnější.



d) Pro rozhodovací strom budeme potřebovat alespoň všechny pozitivní příklady (každý z nich má ve stromu exkluzivní cestu). Těchto příkladů je $\binom{9}{3} = 84$.
Pro induktivní inferenci budeme potřebovat jeden pozitivní příklad, třeba
`p(1,2,3).`
a dále takové negativní příklady, které zabrání průchodu všemi neperspektivními cestami, například
`p(1,1,1). p(1,3,2). p(2,1,3). p(2,3,1). p(3,2,1). p(3,1,2).`

□

Příklad 1.2: Následující úlohu dříve řešenou v rámci induktivní inference ve výrokové logice zformulujte pomocí predikátové logiky: máme tři atributy s danými doménami a příklady rozdělené do dvou tříd

`Size` $\in$ {`small, medium, large`},
`Color` $\in$ {`red, blue, green`},
`Shape` $\in$ {`square, circle, triangle`}

small	red	triangle	true
small	green	triangle	true
large	red	triangle	false
small	blue	circle	false

Najděte specializaci (všechny specializace) a generalizaci (všechny generalizace) následujících klauzulí. Vezměte v úvahu, že nemá smysl specializovat proměnnou `Id`.

a) `p(Id) :- size(Id,large), color(Id,red).`

b) `p(Id) :- color(Id,red).`

c) `p(Id) :- size(Id,large), color(Id,red), shape(Id,circle).`

d) `p(Id).`

e) Najděte všechny specializace klauzule `p(Id) :- color(Id,red).`, které splňují příklad `<large,red,square>`.

f) Určete, zda a jak jsou formule z předchozích bodů vzájemně v relaci generalizace/specializace.

Řešení 1.2: V doménové znalosti použijeme predikáty `size/2`, `color/2`, `shape/2`, trénovací příklady reprezentujeme stylem

```
size(1,small). color(1,red). shape(1,triangle).
size(2,small). color(2,green). shape(2,triangle).
...
```

a dále budeme počítat s tím, že `p(1).` a `p(2).` jsou pozitivní příklady, zatímco `p(3).` a `p(4).` negativní.

a) všech specializací je výrazně více než ve výrokové logice, uvedeny jsou jen některé, zde konkrétně jedna specializační větev:

```
p(Id) :- size(Id,large), color(Id,red), shape(X,Y).
p(Id) :- size(Id,large), color(Id,red), shape(Id,Y).
p(Id) :- size(Id,large), color(Id,red), shape(Id,circle).
...
p(Id) :- false.
```

b) např. `p(Id) :- color(Id,red), size(Id,small).`

c) např. `p(Id) :- false.`

d) např. `p(Id) :- color(Id,green).`

e) některé vybrané:

```
p(Id) :- color(Id,red), size(Id,large).
p(Id) :- color(Id,red), shape(Id,square).
p(Id) :- color(Id,red), size(Id,large), shape(Id,square).
```

f) Žádná dvojice není v relaci nejmenší generalizace nebo největší specializace. Jinak je tam relací specializace/generalizace mnoho, třeba d) je generalizací všech předchozích.

□

Příklad 1.3:

a) Vytvořte specializační graf pro predikát `member/2`.

b) Popište, jak se tento graf změní pro predikát `last/2`.

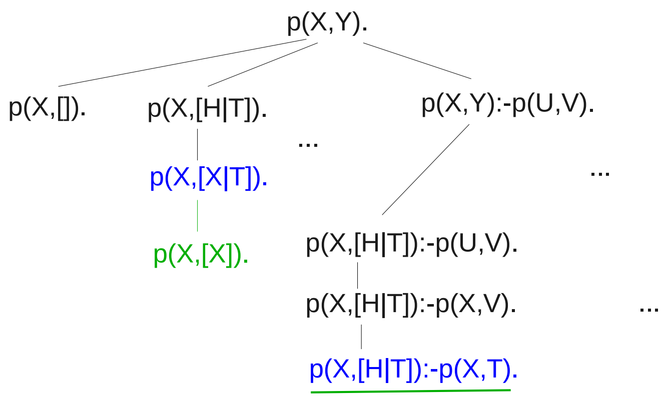
c) Na základě tohoto grafu vytvořte seznam trénovacích příkladů tak, aby učící algoritmus vždy vybral správnou cestu grafem.

Řešení 1.3:

```
member(X,[X|_]).
member(X,[_|T]):- member(X,T).


last(X,[X]).
last(X,[_|T]):- last(X,T).
```

Výsledná teorie pro `member` je vyznačena modře. Změny a výsledná teorie pro `last` jsou vyznačeny zeleně.
Pozitivní příklady pro `member`: `p(a,[a]). p(a,[b,a]). p(a,[a,b]).`
Negativní příklady pro `member`: `p(a,[]). p(a,[b,c]).`
Pro `last` jsou příklady stejné, až na to, že `p(a,[a,b]).` se přesune z pozitivních do negativních.



□

Příklad 1.4: Popište tvorbu specializačního grafu pro `reverse/2` s doménovou znalostí `append/3`.

Řešení 1.4:

```
append([],L,L).
append([H|T],L,[H|T2]):-append(T,L,T2).

reverse([],[]).
reverse([H|T],R):-reverse(T,Y),append(Y,[H],R).
```

Postup při konstrukci relevantní části specializačního grafu je obdobný jako v předchozí úloze.

□