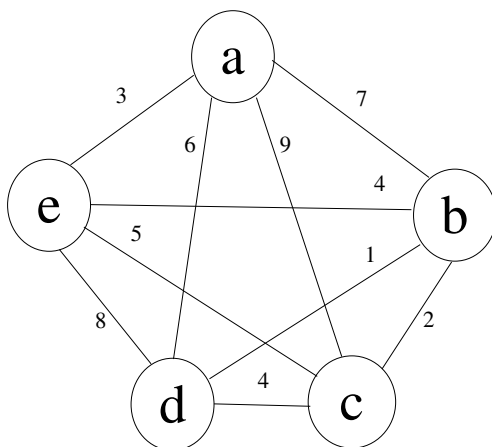


Algoritmy na ohodnocených grafech

Zápočtová práce pro rok 2002/2003

Lukáš Zapletal, INF-01

Uvažujme neorientovaný hranově ohodnocený úplný graf o pěti uzlech, který je znázorněn na obrázku. Na tomto grafu představím dva jednoduché algoritmy. První slouží k nalezení nejkratší cesty mezi dvěma uzly a druhý představuje notoricky známé řešení úlohy obchodního cestujícího.



Reprezentace grafu

Graf je v Prologu tvořen vlastnostmi $\text{edge}(X, Y)$, které reprezentují incidenci, mezi uzly X a Y . Jelikož se jedná o graf neorientovaný, je nutné takto vložit obě incidence, případně vytvořit mechanismus, který bude zajišťovat obousměrnou incidenci. Zvolil jsem druhou možnost a vytvořil predikát $\text{connected}(X, Y)$, který zjistí, zda uzly X a Y incidují bez ohledu na orientaci hrany. Výše uvedený graf je v mojí reprezentaci zapsán takto:

```
edge(a,b,7).  
edge(b,c,2).  
edge(c,d,4).  
edge(d,e,8).  
edge(e,a,3).  
edge(a,c,9).  
edge(a,d,6).  
edge(b,e,4).  
edge(b,d,1).  
edge(c,e,5).
```

Nalezení nejkratší cesty

K nalezení nejkratší cesty mezi dvěma uzly (vzhledem k ohodnocení hran a ne počtu uzlů) jsem implementoval predikát `mpath/3`. Tento triviální problém lze řešit hrubou silou, tj. vygenerováním seznamu všech možných cest z uzlu A do uzlu B a výběrem té cesty, která má nejmenší ohodnocení. Moje implementace je však zvláštní v tom, že si přímo během rekurze, ve které probíhá sestavování všech možných cest, vypočítává tato ohodnocení a ukládá je spolu s

vygenerovanými cestami do seznamu. Z něj je pak vybrán ten prvek, který má nejmenší ohodnocení.

Následuje ukázka použití predikátu `mpath/3`:

```
?- mpath(a,c,Path).  
Path = [a, e, c] ;  
No
```

Úloha obchodního cestujícího

Tato úloha spočívá v tom, že je nutno najít co nejkratší cestu v grafu takovou, která pokrývá celý graf. Jinými slovy je nutno „navštívit“ všechny uzly grafu tak, aby byla tato cesta co nejkratší.

Heuristická metoda řešení této jednoduché úlohy vychází z úplného grafu. Na začátku zvolme kružnici o třech uzlech s nízkým ohodnocením (v našem případě by to byly uzly *b*, *c*, *d*). Postupně přidáváme nejvzdálenější uzly do kružnice tak, aby bylo ohodnocení kružnice co možná nejmenší. Tento krok postupně opakujeme až do hamiltonovy kružnice. Díky předpokladu, že je graf úplný, máme zajištěno, že půjdou všechny uzly přidat.

Vstupem algoritmu (`solve/3`) v Prologu je **kružnice o libovolném počtu uzlů** (např. 3 uzly) reprezentovaná seznamem a **zbývající uzly**, které jsou taktéž reprezentovány seznamem. Výstupem je seznam, který představuje kružnici. K dispozici je též predikát `eval/2`, který vrátí ohodnocení této kružnice. Příklad použití:

```
?- solve([b,c,d],[a,e],V).  
V = [b, c, e, a, d] ;  
No  
?- eval([b,c,e,a,d],E).  
E = 17 ;  
No
```

Implementace je následující:

```
%% ULOHA OBCHODNIHO CESTUJICIHO  
solve(Kruznice,ZbUzly,Vysledek) :-  
    %% vrat nejvzdalenejsi uzel (U)  
    mdist(ZbUzly,Kruznice,U),  
    %% vloz ho tak, aby bylo oh. kruznice nejmensi  
    insert(U,Kruznice,NovaKruznice),  
    %% tento uzel vyjmi ze seznamu zbylych uzlu  
    remove(U,ZbUzly,NoveZbUzly),  
    %% a rekurzivne pokračuj pro dalsi uzel  
    solve(NovaKruznice,NoveZbUzly,Vysledek).  
  
%% limitni podminka rekurze  
solve(Kruznice,[],Kruznice).
```

V kódu se vyskytují pomocné predikáty `mdist / 3`, který vrací nejvzdálenější uzel. Jeho implementace byla nejsložitější částí zápočtové úlohy. Dále predikát `insert / 3`, který vloží uzel do kružnice na takové místo, aby bylo ohodnocení co možná nejmenší. Tento predikát používá `eval / 2` vyhodnocující jednotlivé varianty. Nakonec jsem musel implementovat jednoduchý predikát `remove / 3`, jenž odebere ze seznamu všechny dané atomy.

Závěr

Logický programovací jazyk Prolog je velice mocný a je až s podivem, jak lze některé problémy v něm řešit. Ukázkou nechť jsou tyto dvě vypracované úlohy, jejichž zdrojový kód je v příloze.

Vypracování obou úloh bylo provedeno v operačním systému GNU/Linux (Mandrake 9.1) s distribucí SWI-Prolog. Tento dokument byl vytvořen v programu Open Office (1.0.2).

Lukáš Zapletal

13. VI 2003, Olomouc