Slovenská technická univerzita

Fakulta informatiky a informačných technológii

Ilkovičova 3, 812 19 Bratislava

**Umelá inteligencia**

**2.** **Zadanie – Prehľadávania stavového priestoru**

**Michal Ostrodický**

Cvičiaci: Ing. Ivan Kapustík

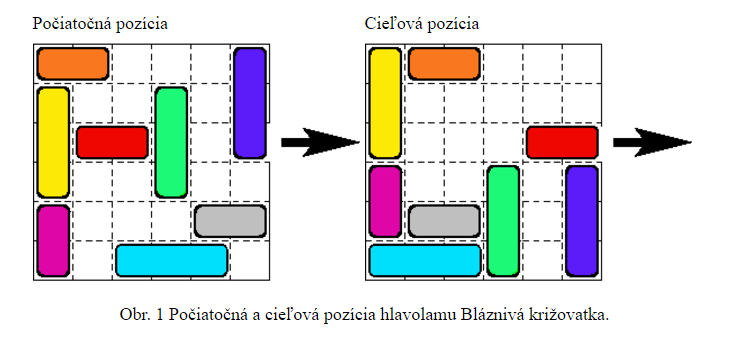
Študijný odbor: Informatika

Ročník: 2. Bc

Akademický rok: 2016/2017

# Znenie zadania:

Úlohou je nájsť riešenie hlavolamu Bláznivá križovatka. Hlavolam je reprezentovaný mriežkou, ktorá má rozmery 6 krát 6 políčok a obsahuje niekoľko vozidiel (áut a nákladiakov) rozložených na mriežke tak, aby sa neprekrývali. Všetky vozidlá majú šírku 1 políčko, autá sú dlhé 2 a nákladiaky sú dlhé 3 políčka. V prípade, že vozidlo nie je blokované iným vozidlom alebo okrajom mriežky, môže sa posúvať dopredu alebo dozadu, nie však do strany, ani sa nemôže otáčať. V jednom kroku sa môže pohybovať len jedno vozidlo. V prípade, že je pred (za) vozidlom voľných n políčok, môže sa vozidlo pohnúť o 1 až n políčok dopredu (dozadu). Ak sú napríklad pred vozidlom voľné 3 políčka (napr. oranžové vozidlo na počiatočnej pozícii, obr. 1), to sa môže posunúť buď o 1, 2, alebo 3 políčka.

Hlavolam je vyriešený, keď je červené auto (v smere jeho jazdy) na okraji križovatky a môže z nej teda dostať von. Predpokladajte, že červené auto je vždy otočené horizontálne a smeruje doprava. Je potrebné nájsť postupnosť posunov vozidiel (nie pre všetky počiatočné pozície táto postupnosť existuje) tak, aby sa červené auto dostalo von z križovatky alebo vypísať, že úloha nemá riešenie. Príklad možnej počiatočnej a cieľovej pozície je na obr. 1.

**Problém 1.** Použite algoritmus prehľadávania do šírky a do hĺbky. Porovnajte ich výsledky.

# Opis riešenia:

Zadanie som riešil algoritmami neinformovaného hľadania – prehľadávaním do šírky(Breadth first search - BFS) a do hĺbky(Depth first search - DFS). Prehľadávanie do šírky používa dátovú štruktúru radu(Queue) a princíp FIFO, čiže First in First out. Ten stav, ktorý bol vložený do frontu najskôr bude pri nasledujúcom výbere z frontu vybraný ako prvý.

Dátovú štruktúru zásobník(Stack) som použil v prehľadávaní do hĺbky. Stack je založený princípe LIFO, last in first out. Ten stav, ktorý sme pridali do zásobníka ako posledný bude pri nasledujúcom výbere vybraný ako prvý.

Obe prehľadávania generujú veľa stavov a mnohokrát sa tieto stavy opakujú. V mojom riešení som použil Hashmapu na uchovávanie stavov, ktoré som už vytvoril, aby som vedel jednoznačne povedať, či som daný stav už niekde vyššie v strome prehľadávania nevygeneroval. Ako som už spomínal vyššie, v BFS som používal štruktúru rad na uloženie ešte nespracovaných stavov, a v DFS štruktúru zásobníka.

Dôležitou časťou v riešení je použitie vhodných operátorov na nájdenie cieľového stavu. Moje riešenie používa 4 operátory a to pohyb vozidla – vpravo, vľavo, dole a hore.

Cieľová pozícia je dosiahnutá vtedy ak sa červené vozidlo(vždy prvé v poradí) dostane na x-ovú súradnicu 5 v hociktorom riadku križovatky.

Činnosť algoritmov prehľadávania sa vo všeobecnosti veľmi podobajú, oba spôsoby vyhľadávania majú rôzne výhody a nevýhody. V nasledujúcich kapitolách sa pokúsim porovnať tieto dve hľadania.

# Reprezentácia údajov:

# Algoritmus

## **Zložitosť**

# Testovanie

# Zhodnotenie

## porovnanie vlastností