Domáca úloha č. 3
Peter Kovács

1 Heuristiky

1. Majme konzistentné heuristiky, teda pre nich platí $h(x) \leq c(x,y) + h(y)$. Zároveň si uvedomme, že $\forall h c(x,y) + h(y) \leq max_{h \in H}(c(x,y) + h(y))$, z vlastností maxima. Rovnako aj pre ľavú stranu nerovnosti $h(x) \leq max_{h \in H}(h(x))$ Potom
$$(\forall h)(\exists f) : h(x) \leq max_{h \in H}(h(x)) = f(x) \leq c(x,y) + f(y) \leq max_{g \in H}(c(x,y) + g(y))$$
a teda je táto maximová heuristika konzistentná

2. Majme heuristiku, ktorá pre každý stav s nadobúda hodnotu $h(s) = c(s,f)$, kde f je cieľový stav. Táto heuristika je zjavne prípustná. Vyrobme h' také, že $h'(s) = c(s,f)$, ak $c(s,f) > c$ a $h'(s) = 0$, ak $c(s,f) \leq c$. c nech je najmenšia nenulová hodnota $c(s,f)$. Potom je táto heuristika stále prípustná (niekde sme znížili hodnotu), ale už nie je monotónna. Tvrdenie teda neplatí.

3. Zvoľme ako h_1 a h_2 heuristiku, ktorá presne odhaduje cenu riešenia. Zjavne takáto heuristika je prípustná a monotónna. Ak však sčítam h_1 a h_2 dostávam dvojnásobok ceny riešenia čo je v spore s prípustnosťou heuristiky.

4. Prehľadávajme graf stavov. Pre spor predpokladajme, že sme ako prvé našli riešenie G', ktoré je viac ako dvakrát horšie ako optimálne riešenie. Potom ale $f(G') = g(G') + h(G') > 2 \cdot OPT$, pretože $h(G') = 0$. Nech x je vrchol na ceste k riešeniu, ktoré je najviac 2-krát horšie ako optimálne, taký že sa nachádza na hranici prehľadávania tesne pred objavením riešenia G'. Potom $f(x) = g(x) + h(x) \leq 2 \cdot OPT$, keďže $h(x) \leq 2 \cdot minimalnacestazx$. Ale dostávame; $f(G') > 2 \cdot OPT \geq f(x)$, čo je spor pretože by x muselo by+t expandované skôr ako G'

2 Analýza prohledávacího problému

1. Vlastnosti prostredia:
   - Plne porozovateľné - po celý čas vidím, celý plánik a všetky polohy žiab
   - Deteministické - ťahy vykonávam iba ja, prostredie sa nemení samé a každý ťah má presne určený dôsledok.
   - Statické - prostredie ovplyvňujem iba ja svojimi ťahmi
   - Diskrétní - konečne mnoho stavov.
   - Nie je multiagentní - ťahy vykonávam iba ja
   - DeadEndy - nie sú, ku každému ťahu vieme urobiť inverzný, ktorý zvráti jeho účinok

2. Veľkosť stavového prostoru - Potrebujeme vypočítať počet rôznych rozložení žiab na plániku. To môžeme napríklad tak, že najprv vyberieme miesta pre svetlé žaby $\binom{2n^2-1}{n^2-1}$ a zo zvyšných n^2 políčok potrebujem vybrať to, na ktorom nebude žaba. Možností je n^2. Celkový počet možností, ako rozostaviť žaby, je teda:
$$n^2 \binom{2n^2-1}{n^2-1}$$

3. Uvažujme heuristiku, kde každá figúrka, ktorá nie je na dobrom políčku pripočíta k výslednej hodnote jedna. Heuristická hodnota je teda počet zle postavených figúriek. Na opravu každej figúrky treba aspoň jeden krok. Máme teda prípustnú heuristiku. Vzhľadom na to, že máme prípustnú heuristiku, môžeme použiť A* s touto heuristikou. V prípade veľkých instancií použijeme IDA* alebo MA*, aby sme zamedzili pretečeniu pamäte.

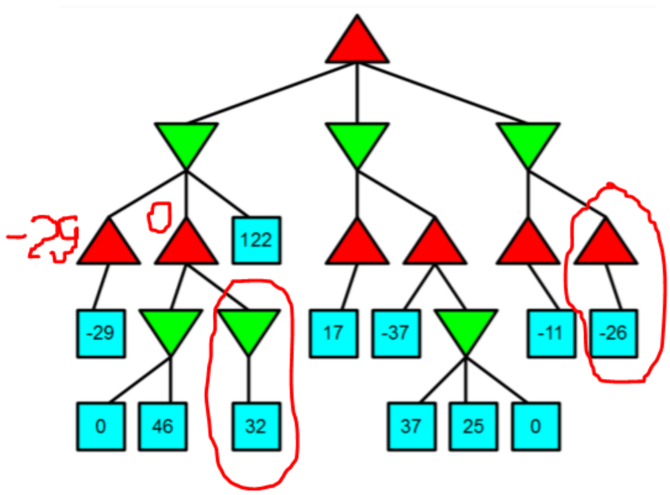
3 Hraní her

1. Zjavne musí byť práve na ťahu krúžok, pretože ak by bol na ťahu krížik, ten by mohol položiť krížik do prvého stĺpca a stredného riadku. Tým pádom by vo svojom ďalšom ťahu vyhral. V tomto momente už ku každému ťahu krúžku existuje ťah krížika, ktorý ho zablokuje a nastane remíza.

   Z toho vieme povedať, že začínal krúžok. Rozborom možných prípadov, sa dá ukázať, že položením krúžku do stredu hrany, by si 1. hráš spôsobil prehru. Preti budeme predpokladať, že prvý ťah bolo nakreslenie krúžku do rohu plánika. Vidíme, že krížik musí blokovať krúžok. Posledný ťah teda bude krížik v dolnom riadku. Toto môžeme overiť výpisom možností, kde uvidíme, že pri inom poradí ťahov by jeden z hráčou pripustil svoju prehru.

2. Prerezávanie nám dovolí neprehľadávať časti stromu, ktoré sú zakrúžkované červenou. Prvý podstrom, ten vľavo, nemusíme prehľadávať, pretože hráč MIN už vyberie iba cestu, ktorá je menšia rovná -29 z ľavého podstromu dostaneme, že MAX vyberie aspoň 0, teda zvyšok stromu nemá zmysel prehľadávať, nakoľko MIN twnto podstrom určite nevyberie.

   Pravý zakrúžkovaný podstrom môžeme prerezať, preto, že jeho sused má hodnotu -11, hráč MIN v druhej vrstve teda vyberie maximálne -11. Hráč MAX v prvej vrstve , ale tento podstrom určite nevyberie, keďže jeho stredný syn bude mať hodnotu 0.



3. Algoritmus by prehľadával aj také stromy, ktoré dávajú ohodnotenie ktoré je rovné nejakému, ktoré sme už našli. Tak by sme zabezpečili aby sa prehľadalo všetko čo vie dať rovnako dobre ťahy. V tých stromoch vybranú hodnotu v otcovi náhodný výber neovplyvní nakoľko majú hodnotu rovnakú. Potrebujem ich ale prehľadať aby som sa vedel vybrať ľubovoľným z nich.

4. Prinajhoršom môže prerezávanie prejsť celý strom, teda 121 vrcholov. Hodnoty môžu byť napríklad také, ako na obrázku vľavo Hodnoty vždy narastajú (alebo klesujú, podľa toho či hrá MAX alebo MIN) a preto treba preskúmať každý podstrom.

   Najlepší prípad vidíme na obrázku vpravo, kde najprv musí samozrejme algoritmus preliezť nejakú cestu od koreňa do listu. POtom už prelieza len nutné vrcholy tak, aby mohol čo najskôr odrezať celý podstrom. Žltou sú vyznačené navštívené vrcholy, ktorých je 37.

   V obrázkoch prvé tri čísla symbolizujú 3 listy vrcholu, ktorým je 4 té číslo v riadku. Následne má už tabuľka stromovú štruktúru.

18	17	16	16	18	12	14
19	18	17	17			
20	19	18	18			
15	6	5	5	15		
16	15	14	14			
17	16	15	15			
12	11	10	10	12		
13	12	11	11			
14	13	12	12			
19	18	17	17	19	13	
20	19	18	18			
21	20	19	19			
16	7	6	6	16		
17	16	15	15			
18	17	16	16			
13	12	11	11	13		
14	13	12	12			
15	14	13	13			
20	19	18	18	20	14	
21	20	19	19			
22	21	20	20			
17	8	7	7	17		
18	17	16	16			
19	18	17	17			
14	13	12	12	14		
15	14	13	13			
16	15	14	14			

10	10	10	10	10	10	10
0	0	0	0			
0	0	0	0			
20	20	20	20	20		
1	2	3	1			
1	2	3	1			
20	20	20	20	20		
1	2	3	1			
1	2	3	1			
20	20	0	0	0	0	
20	20	0	0			
20	20	0	0			
0	0	0	0	0		
0	0	0	0			
0	0	0	0			
0	0	0	0	0		
0	0	0	0			
0	0	0	0			
0	0	0	0	0	0	
0	0	0	0			
0	0	0	0			
0	0	0	0	0		
0	0	0	0			
0	0	0	0			
0	0	0	0	0		
0	0	0	0			
0	0	0	0			

4 Logika

4.1 Výroková logika

Zavedieme 5 premenných A,B,C,D,E. Každá z nich indikuje, či osoba začínajúca na dané písmeno patrí do gangu. Potom vieme ľahko preložiť výroky, do výrokovej logiky s týmito 5 premennými. Preložené výroky:

1. $A \rightarrow B$
2. $D \vee E$
3. $(B \wedge \neg C) \vee (\neg B \wedge C)$
4. $D \leftrightarrow C$
5. $E \rightarrow A \wedge D$

Tieto logické formule preložíme do tvaru CNF, aby sme mohli použiť rezolúciu

1. $\neg A \vee B$
2. $D \vee E$
3. $(B \vee C) \wedge (\neg B \vee \neg C)$
4. $(D \vee \neg C) \wedge (\neg D \vee C)$
5. $(\neg E \vee A) \wedge (\neg E \vee D)$

Dostávame teda klauzule pripravené na rezoluci

$$\{\neg A, B\}, \{D, E\}, \{B, C\}, \{\neg B, \neg C\}, \{D, \neg C\}, \{\neg D, C\}, \{\neg E, A\}, \{\neg E, D\}$$

Teraz prevedieme rezoluci tak, že do vyššie uvedenej knowledge base pridáme výrok $\{B\}$, teda že Berta je v gangu. A rezolúciou odvodíme spor.

Klauzule	Rezolventa
$\{D, E\}, \{D, \neg E\}$	$\{D\}$
$\{D\}, \{\neg D, C\}$	$\{C\}$
$\{C\}, \{\neg C, \neg B\}$	$\{\neg B\}$
$\{\neg B\}, \{B\}$	SPOR

Rezolúciou sme dospeli do sporu a teda Berta nie je súčasťou gangu.

Dotaz na Evu vyriešime rovnako, teda do KB pridáme $\{E\}$ a spravíme rezolúciu.

Klauzule	Rezolventa
$\{E\}, \{D, \neg E\}$	$\{D\}$
$\{E\}, \{A, \neg E\}$	$\{A\}$
$\{D\}, \{\neg D, C\}$	$\{C\}$
$\{\neg C, \neg B\}, \{C\}$	$\{\neg B\}$
$\{A\}, \{\neg A, B\}$	$\{B\}$
$\{\neg B\}, \{B\}$	SPOR

Opäť sme dospeli k sporu a teda ani Eva nie je členka gangu.

4.2 Prediktátová logika

Zavedieme prediktáty $KLUK(x), HOLKA(x), DOSTANE(x, y), DITE(x), HODNE(x)$ a konštanty $Panenka, Vlacek, Uhli$. Teraz môžeme výroky zo zadania prepísať v jazyku prediktátovej logiky.

1. $\forall x((KLUK(x) \vee HOLKA(x)) \rightarrow DITE(x))$
2. $\forall x(DOSTANE(x, Panenka) \vee DOSTANE(x, Vlacek) \vee DOSTANE(x, Uhli))$
3. $\forall x(KLUK(x) \rightarrow \neg DOSTANE(x, Panenka))$
4. $\forall x(DITE(x) \wedge HODNE(x) \rightarrow \neg DOSTANE(x, Uhli))$

Všetky výroky obsahujú iba jeden obecný kvantifikátor s platnosťou na celý výrok. Jediné čo teda treba urobiť je previesť výroky do CNF a pripraiť ich tak na rezoluci:

1. $(\neg KLUK(x) \vee DITE(x)) \wedge (\neg HOLKA(x) \vee DITE(x))$
2. $(DOSTANE(x, Panenka) \vee DOSTANE(x, Vlacek) \vee DOSTANE(x, Uhli))$
3. $(\neg KLUK(x) \vee \neg DOSTANE(x, Panenka))$
4. $(\neg DITE(x) \vee \neg HODNE(x) \vee \neg DOSTANE(x, Uhli))$

Do tejto knowledge base pridáme dotaz(konkrétne jeho negáciu, abz sme mohli odovodiť spor) a vykonáme rezolúciu. Dotaz preložíme do jazyka prediktátovej logiky:

$$(\forall x(DITE(x) \rightarrow \neg DOSTANE(x, Vlacek))) \rightarrow (\forall y(KLUK(y) \rightarrow \neg HODNY(y)))$$

Negácia výroku je

$$\forall x \exists y(\neg DITE(x) \vee \neg DOSTANE(x, Vlacek) \wedge (KLUK(y) \wedge HODNY(y)))$$

Urobíme skolemizaci:

$$(\neg DITE(x) \vee \neg DOSTANE(x, Vlacek)) \wedge (KLUK(F(x))) \wedge (HODNY(F(x)))$$

Po pridaní negácie dotazu dostávame klauzule:

$$\{\neg KLUK(x_1), DITE(x_1)\},$$
$$\{\neg HOLKA(x_1), DITE(x_1)\},$$
$$\{DOSTANE(x_3, Panenka), DOSTANE(x_3, Vlacek), DOSTANE(x_3, Uhli)\},$$
$$\{\neg KLUK(x_4), \neg DOSTANE(x_4, Panenka)\},$$
$$\{\neg DITE(x_5), \neg HODNE(x_5), \neg DOSTANE(x_5, Uhli)\}$$
$$\{\neg DITE(x_6), \neg DOSTANE(x_6, Vlacek)\}$$
$$\{KLUK(F(x_7))\}$$
$$\{HODNY(F(x_8))\}$$

V tabuľke nižšie je priebeh rezolúcie.

$$\{KLUK(F(x_7))\}, \{\neg KLUK(x_1), DITE(x1)\}$$

Po substitúcií $F(x_7) = x_1$

$$\{DITE(F(X_7))\}$$

$$\{DITE(F(x_7))\}, \{\neg DITE(x_6), \neg DOSTANE(x_6, Vlacek)\}$$

Po substitúcií $F(x_7) = x_6$:

$$\{\neg DOSTANE(F(x_7), Vlacek)\}$$

$$\{\neg DOSTANE(F(x_7), Vlacek)\}, \{DOSTANE(x_3, Panenka), DOSTANE(x_3, Vlacek), DOSTANE(x_3, Uhli)\}$$

Po substitucií $F(x_7) = x_3$:

$$\{DOSTANE(F(x_7), Panenka), DOSTANE(F(x_3), Uhli)\}$$

$$\{DOSTANE(F(x_7), Panenka), DOSTANE(F(x_3), Uhli)\}, \{\neg KLUK(F(x_4)), \neg DOSTANE(x_4, Panenka)$$

Po substitúcií $F(x_7) = x_4$ dostávame:

$$\{\neg KLUK(F(x_6))\}$$

$$\{\neg KLUK(F(x_6))\}, \{KLUK(F(x_6))\}$$

z čoho dostávame SPOR