

1. Проблем со пребарување на простор на состојби

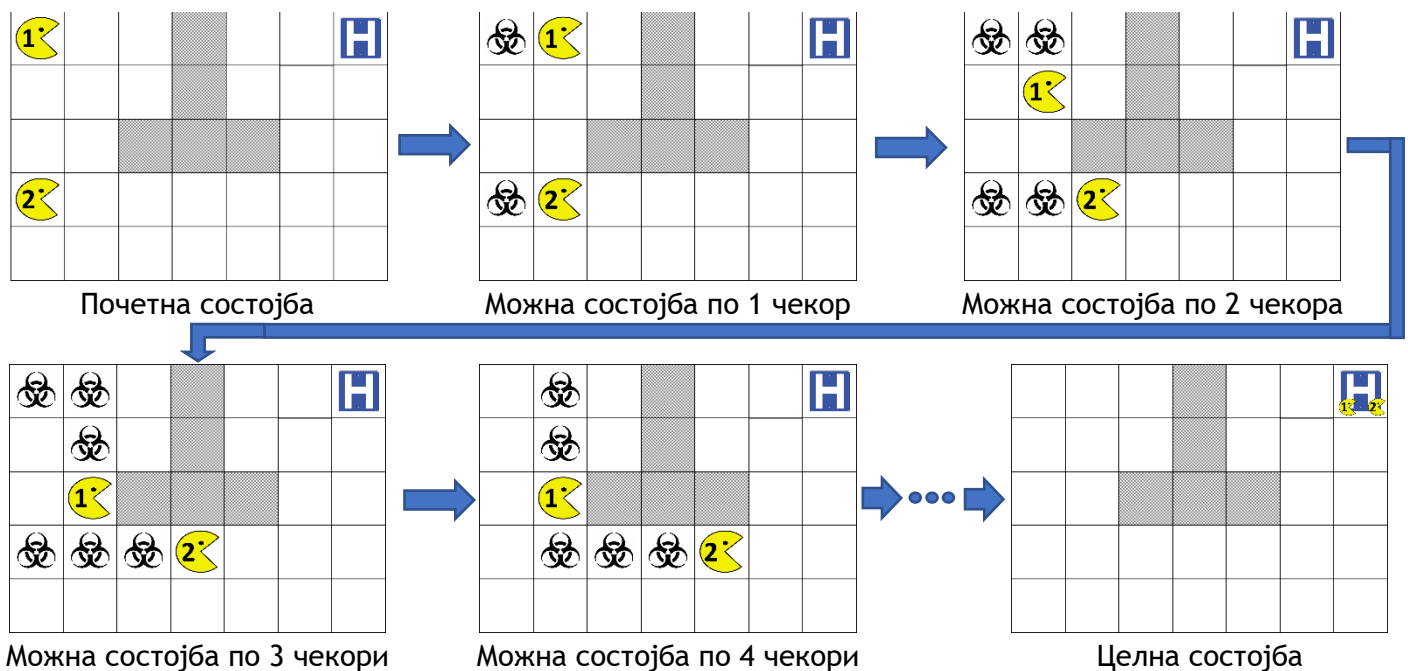
Токсичен Пакман



Дефиниција на проблемот:

Во светот на Пакман имало проблем со нуклеарна централа, па сега сите Пакман играчи се токсични и кога се движат зад себе оставаат отровна трага врз која ниту еден Пакман не смее да стапне (вклучително и Пакман играчот кој ја оставил трагата). Ваша задача во овој проблем е да им помогнете на токсичните Пакман играчи да сигнат до болницата и да се излекуваат.

Во проблемот има вкупно k токсични Пакман играчи и секој од нив е на почетна позиција s_i во рамки на голем лавиринт со димензии $M \times N$ и треба да стигне до позицијата H на болницата. На патот кон болницата, секое поле на кое поминува било кој од Пакман играчите станува отровно. Пакман играчите никогаш не смее да застанат на отровно поле. Едно поле останува отровно 2 чекори (на пример, ако полето станало отровно по првиот чекор, истото ќе остане отровно за време на вториот и третиот чекор; во четвртиот чекор полето повеќе не е отровно). Целта е сите Пакман играчи да бидат излекувани (да пристигнат во болницата).

Во секој чекор секој од k -те Пакман играчи може да се придвижи во некое слободно соседно поле во лавиринтот. Единствените дозволени акции се **Горе**, **Долу**, **Лево**, **Десно** и **Стоп**, при што акцијата Стоп значи дека Пакман играчот ќе остане на полето на кое што се наоѓа (нема да се придвижи). Не е дозволено било кој од Пакман играчите да застане на отровно поле, да се придвижи на поле на кое има сид, или да се придвижи на исто поле како некој друг Пакман играч (единствениот исклучок е полето на кое се наоѓа болницата).



На сликите е визуелно прикажано како може да изгледа проблемот. Полињата кои се сидови се претставени со сива боја, Пакман играчите се претставени со жолта боја, полето на болницата е претставено со знакот , а отровните полиња со знакот . Дадена е некоја можна почетна состојба за лавиринт со димензии 5x7 и двајца Пакман играчи означени на сликите со 1 и 2. Низата акции кои ги вршат Пакман играчите се Пакман1: Десно, Долу, Долу, Стоп и Пакман2: Десно, Десно, Десно, Десно. Соодветно на чекорите се менуваат и отровните полиња. Во приказот на целната состојба можете да забележите и дека и двата Пакман играчи се на полето на болницата (отровните полиња не се прикажани иако постојат).

За дадениот проблем чие решавање треба да го дефинирате како пребарување низ простор на состојби потребно е да одговорите на следните прашања/задачи поделени по делови:

(а) Дефинирајте минимална репрезентација на состојбата на проблемот. Образложете го секој елемент на состојбата.

(б) Колку најмногу состојби може да има во просторот на состојби за вашата дефиниција? Образложете го секој дел од вашето решение.

Забелешка: Прашањето треба да се одговори за општ случај, односно за k Пакман играчи и лавиринт со големина $M \times N$. Доколку има потреба, дефинирајте и дополнителни големина.

(в) Која е максималната вредност на факторот на разгранување (branching factor) за овој проблем? Образложете.

(г) Напишете ги почетната и целната состојба дадени на сликите според вашата дефиниција.

(д) Ако проблемот се наоѓа во некоја произволна состојба, кои акции се дозволени (легални)? Образложете користејќи ја вашата дефиниција за состојбата.

(ѓ) Дефинирајте самите некоја можна нетривијална евристика за поедноставена верзија на проблемот во која имате само еден Пакман играч во лавиринтот. Докажете дека вашата евристика е допустлива.

(е) Нека со h_i е означена вашата евристика дефинирана под (ѓ) која се однесува на i -тиот Пакман играч во оригиналниот проблем со k Пакман играчи. Која од следните евристики е допустлива? Образложете за секоја евристика.

$$h_a: \frac{\sum_{i=1}^k h_i}{k} \quad h_b: \sum_{i=1}^k h_i \quad h_c: \max_{1 \leq i \leq k} h_i \quad h_d: k * \max_{1 \leq i \leq k} h_i \quad h_e: \min_{1 \leq i \leq k} h_i \quad h_f: k * \min_{1 \leq i \leq k} h_i$$

(ж) Кој од следните алгоритми за пребарување ќе гарантира најбрзо излекување на Пакман играчите (оптималност)? Образложете за секој алгоритам. Доколку имате избор, кој од понудените алгоритми би го избрале како најдобар за проблемот? Образложете го вашиот избор.

- | | |
|---------------------------------|-------------------------------|
| i. DFS (Depth First Search) | ii. UCS (Uniform Cost Search) |
| iii. BFS (Breadth First Search) | iv. A* |

2. Проблем кој задоволува услови

Какурасу

Какурасу е логичка игра во која на табла со димензии $N \times N$ треба да се најдат сите сини полиња така што се исполнети сите услови. Вредноста на едно сино поле зависи од редицата/колоната во која се наоѓа. Сино поле во првата редица има вредност 1 во збирот за колоната во која се наоѓа, сино поле во втора редица има вредност 2 во збирот за колоната во која се наоѓа и така натаму. Соодветно на претходното, сино поле во првата колона има вредност 1 во збирот на редицата во која се наоѓа, сино поле во втора колона има вредност 2 во збирот на редицата во која се наоѓа и така натаму.

	1	2	3	4	5	
1			x	x		8
2	x	x		x		8
3	x	x	x			9
4		x	x	x	x	1
5						15
	10	6	7	8	11	

Десно од последната колона се дадени вредности кои се однесуваат на збирите по редици (тоа мора да биде задоволено). Под последната редица се дадени вредности кои се однесуваат на збирите по колони (тоа мора да биде задоволено).

Ваша задача е да решавате Какурасу проблем со табла 4×4 даден на сликата во продолжение.

	1	2	3	4	
1					7
2					7
3					4
4					5
	3	4	10	3	

На решението на проблемот му пристапувате постепено, следејќи ги промените во домените на променливите кои се одраз на исполнување на услов, група услови и/или додела на вредност на променливите. За дадениот проблем чие решавање треба да го дефинирате како проблем кој исполнува на услови потребно е да одговорите на следните прашања/задачи поделени по делови:

(а) Формално дефинирајте го проблемот како проблем на исполнување услови (Constraint Satisfaction Problem - CSP).

(б) Нацртајте го графот на ограничувања (услови) за проблемот.

(в) Започнувате да го решавате проблемот со исполнување на унарните услови. За домените на секоја променлива да се провери кои вредности преостануваат по примена на сите **унарни услови** од проблемот. Да се прикажат вредностите кои преостануваат во доменот на секоја променлива!

(г) Пред да се направи било каква додела на променливите, да се направи пропагација на условите со методот за проверка на конзистентноста на целиот проблем т.е. проверка на конзистентност на сите ребра од графот (**arc consistency enforcing**). За секоја направена проверка да се идентификува реброто чија конзистентност се проверува и прикажат промените во доменот на засегнатите променливи! (Не е

потребно да се прикаже секој чекор во постапката, т.е. прикажете ги само проверките за конзистентност на ребра каде се случуваат промени. Доколку дискусијата на постапката за повеќе ребра е иста/слична, образложението може да се однесува на група ребра.)

(д) Да се најде едно решение на проблемот од примерот, применувајќи ги: евристиката за определување на следна променлива за додела на вредност (Minimum Constraining Values - MRV) , евристиката за избор на вредност (Least Constraining Value - LCV) и алгоритмот за враќање наназад со проверка нанапред (backtracking algorithm with forward checking)! За секое направено доделување, да се наведе евристиката која е искористена и образложението како даден избор/додела се одразува на домените на останатите променливи при проверка нанапред.