

## 1. Проблем со пребарување на простор на состојби

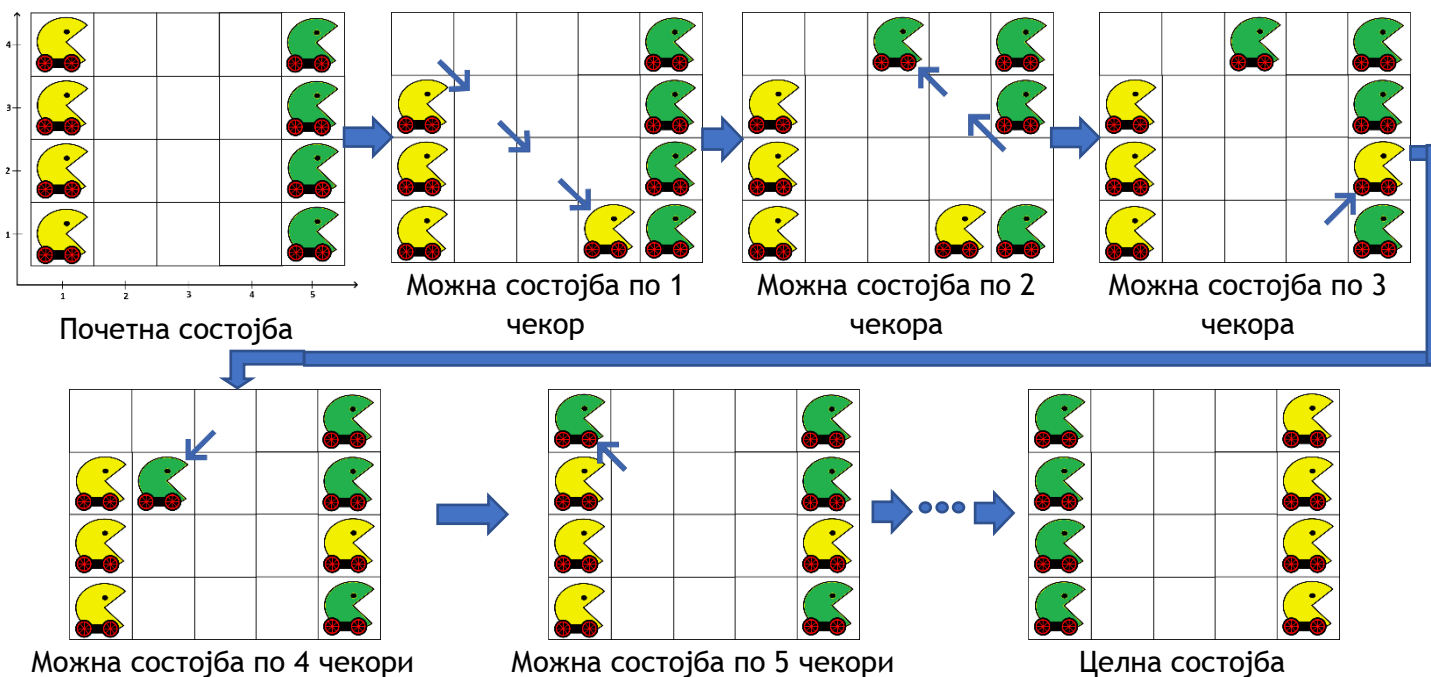
### Пакмани по дијагонала

Дефиниција на проблемот:

Во светот на Пакман се организира претстава во која има две групи на Пакмани (жолти и зелени) кои возат ролери низ правоаголна површина составена од  $M \times N$  единечни полиња. Ваша задача во овој проблем е да им помогнете на двете групи на Пакмани да си ги заменат позициите на површината.

Во проблемот има вкупно  $M$  жолти и  $M$  зелени Пакмани ( $M$  е парен број) на ролери на површина со димензии  $M \times N$  ( $N > M$ ) иницијално распоредени така што жолтите Пакмани се во првата колона (на позиции  $(1,1), (2,1), \dots, (M,1)$ ) и зелените Пакмани се во последната колона (на позиции  $(1,N), (2,N), \dots, (M,N)$ ). Крајните позиции на Пакманите треба да бидат такви што зелените Пакмани ќе дојдат на позиција на жолтите и обратно, т.е. зелените се во прва колона и жолтите се во последна колона. Во целта не е битно како жолтите и зелените Пакмани ќе бидат распоредени во соодветните колони (не треба да се внимава на редослед).

Во секој чекор се придвижува само еден од сите Пакмани (без оглед на бојата). На површината нема ниту ѕидови, ниту духови. Еден Пакман (без оглед на бојата) може да се движи само дијагонално, при што може да застане на било кое поле до кое може да стигне ако се движи по некоја од дијагоналите. Притоа не може да стигне до поле за кое треба да прескокне некој друг Пакман кој се наоѓа на површината. Недозволено е да има два Пакмани на исто поле, како што не е дозволено ни еден Пакман да прескокнува преку друг Пакман.



На сликите е визуелно прикажано како може да изгледа проблемот кога групите се составени од 4 Пакмани, а површината е со димензии 4x5. Во низата чекори се гледа еден можен начин еден жолт и еден зелен Пакман да си ги променат местата. Во приказот на целната состојба можете да забележите дека сите жолти Пакмани си ги имаат променето местата со сите зелени Пакмани.

За дадениот проблем чие решавање треба да го дефинирате како пребарување низ просторот 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 N Пакмани на ролери и површина со големина NxN. 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 на ролери на површината, на пример жолт Пакман кој се наоѓа во првата колона и треба да стигне во последната колона. 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 на ролери во оригиналниот проблем со N Пакмани на ролери. Која од следните евристики е допустлива? Образложете за секоја евристика.

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

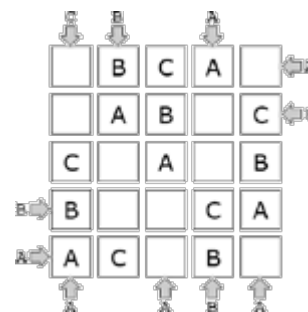
(ж) Кој од следните алгоритми за пребарување ќе гарантираат најбрзо распоредување на Пакман играчите (оптималност)? Образложете за секој алгоритам. Доколку имате избор, кој од понудените алгоритми би го избрале како најдобар за проблемот? Образложете го вашиот избор.

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

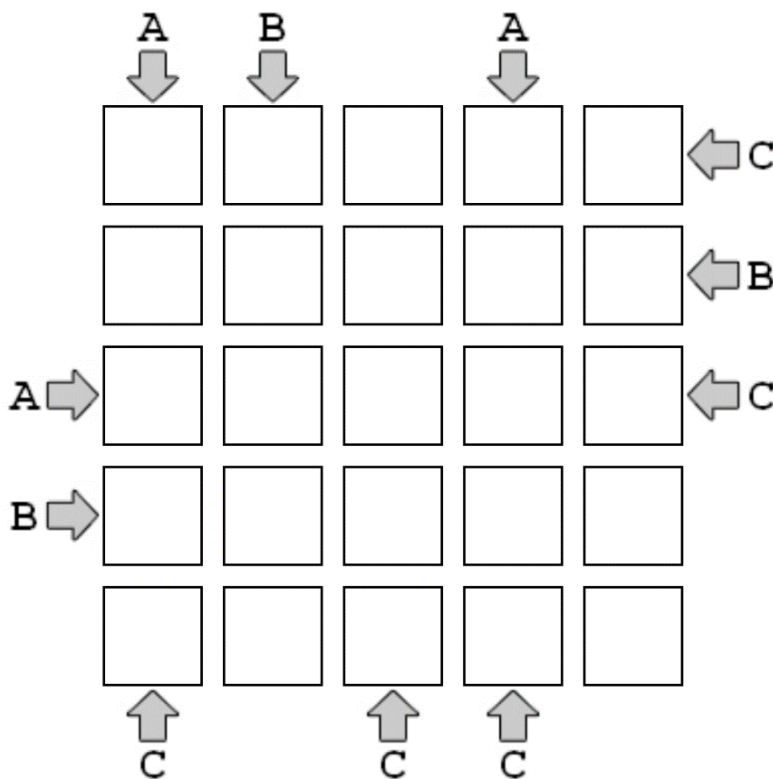
## 2. Проблема кој задоволува услови

### ABC

ABC е логичка игра во која на табла со димензии 5x5 во секоја редица и во секоја колона треба да се постави секоја од буквите A, B и C, како и две празни полиња. Дополнително за дел од редиците/колониите има дадено информација за тоа која е првата буква која се гледа во насоката дефинирана со стрелката. На пример  $A \rightarrow$  значи дека гледано надесно, првата буква која ќе се види е A.



Ваша задача е да го решавате ABC проблем на табла 5x5 даден на сликата во продолжение.



За дадениот проблем чие решавање треба да го дефинирате како исполнување на услови, потребно е да одговорите на следните прашања/задачи поделени по делови:

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

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

(в) Користејќи евристики за определување следна променлива, за која ќе се направи доделување на вредноста применувајќи проверка нанапред (forward checking), најдете едно решение за проблемот од примерот! За секое направено доделување да се образложи која евристика е искористена и како тоа доделување се одразува врз проверката нанапред.