Прв дел: Проблем со пребарување на простор на состојби

Пакман на ролери

1. Минимална репрезентација би била N подредени парови каде секој подреден пар се состои од x и y координатите на секој пакман односно
    1. $State=\{(x,y)\,|\,x,y\in\{1,2,3,\ldots N\}\}$
2. Бидејќи има NxN единечни полиња и N пакмани, најголемиот број на состојби би бил $(NxN)^N$, бидејќи секој пакман може да се најде на секое поле.
3. Во случај кога не би имало никакви ограничувања за движењето на еден пакман би имало **фактор на гранење од 5.**
    1. Но поради ограничувањата дека пакманите не смеат да излезат надвор од границите на квадратната површина, и не смее да има повеќе од еден пакман на едно поле, факторот на гранење е најдобро да се изрази преку просечниот број на деца кои ќе ги има секој јазел.
    2. Да го земеме случајот кога на еден пакман му е преостаната само една акција→стоп. Тоа значи дека пакманот не би можел да ги прескокне пакманите околу него бидејќи би се нашол на поле кое веќе има пакман или на поле кое е надвор од границите.
    3. Тоа значи дека еден пакман секогаш има барем една акција која може да ја преземе, односно минималниот фактор на гранење за еден јазел е 1, а максималниот е 5.
    4. Во преостанатите случаи еден пакман може да има број на акции кои може да ги преземе меѓу 1 и 5, т.е. 2, 3 или 4 акции.
    5. Просечниот број на акции кои може да ги преземе еден пакман според тоа би бил (1+2+3+4+5)/5 = 3, па така максималниот фактор на гранење на овој проблем е 5, минималниот е 1, а просечниот е 3. Дополнително просечниот можевме да го пресметаме и како просек од максималниот и минималниот, т.е. (5+1)/2=3.
    6. Тоа не’ доведува до заклучокот дека во верзијата на проблемот во која имаме N пакмани, максималниот фактор на гранење е 5^N, бидејќи имаме N пакмани кои имаат 5 акции, минималниот е пак 1, а просечниот е 3^N

7. Доколку пак акцијата Стоп не доведува до промена во состојбата за дадениот пакман, односно не генерира ново дете, тогаш максималниот фактор на гранење за еден пакман е 4, просечниот е 2.5, заокружено на 3.
   1. Преку заклучокот од потточка 6, во оваа ситуација кај која Стоп не доведува до промена на состојба, имаме 4^N максимален фактор на гранење, и пак 3^N просечен фактор.
4. Почетна состојба: ($П_a$ → a=1,2,3,...N → П-тиот пакман)
   1. $П_1$ → (1,1)
   2. $П_2$ → (1,2)
   3. $П_3$ → (1,3)
5. Целна состојба: ($П_a$ → a=1,2,3,...N → П-тиот пакман)
   1. $П_1$ → (3,3)
   2. $П_2$ → (3,2)
   3. $П_3$ → (3,1)
6. Легални акции кога проблемот се наоѓа во произволна состојба се оние акции кои за секој пакман ги задоволуваат условите во следниот Python3 код
   1.

```
def move_all_pacmans(state):
    new_state = list()
    For pacman in state:
            new_pacman = update_pacman(pacman)
            if new_pacman not in state and new_pacman not in new_state and \
                <= new_pacman[0] < N and 0 <= new_pacman[1] < N :
                    new_state.append(new_pacman)
    return tuple(new_state)
```

7. Нетривијална евристика во која има само еден пакман може да биде менхетен растојанието од позицијата на пакманот до следната позиција.
   1. $h(n)=|(goal[0]-new\_pacman[0])|+|(goal[1]-new\_pacman[1])|$
   2. Оваа евристика е допустлива. Доказ:
      1. Нека h(n) биде цената на чинење која е добиена преку евристичката функција, и g(n) вистинското растојание.
      2. Оваа евристика е допустлива бидејќи со секоја акција од пакманот, менхетен растојанието може да се промени (зголеми или намали) за најмногу 1 (една единица), поради акциите кои му се возможни на пакманот(не може да се

движи дијагонално, односно со секоја акција или ја менува x-координатата или y-координатата. Таквите ограничувања гарантираат дека менхетен растојанието од позицијата на пакманот до целта никогаш нема да биде поголемо од вистинското растојание. Всушност во рамките на овој проблем тоа, растојанието пресметано преку евристиката е еднакво со вистинското растојание.

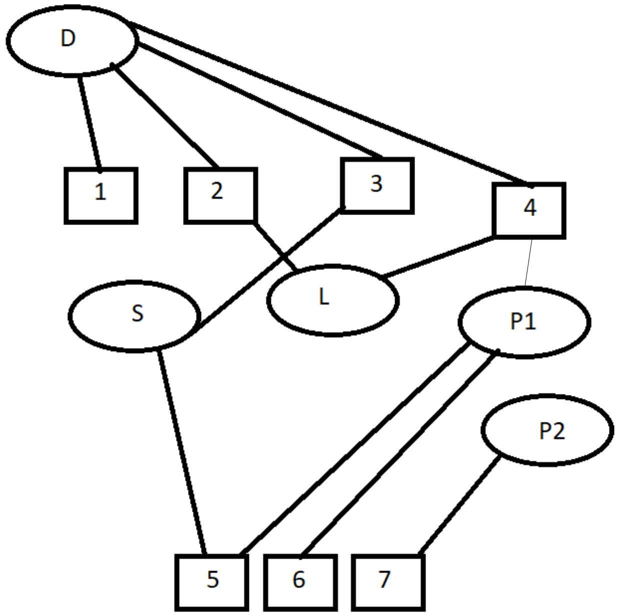
3. Математички: $h(n)=|(new_{pos}[0]-pacman[0])|+|(new_{pos}[1]-pacman[1])|\leq g(n)$
   Каде g(n) е вистинската цена на чинење

8. Допустливи евристики се:
   1. $\frac{\sum_{i=1}^{k} h_i}{k}$ е допустлива бидејќи е просечното менхетен растојание од секој пакман до својата цел е помало(односно еднакво) до вистинското растојание.
   2. $\min h_i$ допустлива е бидејќи најмалото менхетен растојание е сепак <= од вистинското растојание
   3. Исто како и $\min h_i$, $\max h_i$ е допустлива од истата причина.
9.
   1. DFS → Од една страна не би бил оптимален бидејќи може да го најде најдлабокото односно најлошото решение на проблемот. Од друга страна пак, решението го наоѓа најбрзо и користејќи најмалку меморија, па доколку проблемот има целна состојба која не премногу блиску до коренот од дрвото, **тогаш сепак DFS е најоптимален.**
   2. UCS → Го наоѓа оптималното решение, но поради тоа што сите цени на чинење за стигање од еден јазол до друг се приближно исти, се приближува до BFS, што како и BFS, **не** е оптимален
   3. BFS → наоѓа оптимално решение но троши многу меморија и време за да го најде кога целта е далеку од коренот, односно подлабоко во дрвото, па за овој проблем не би бил оптимален бидејќи целната состојба ретко ќе се наоѓа поблиску до коренот
   4. A*, користејќи евристичка функција директно се движи кон најоптималното решение, и тоа го прави на најоптимален начин гледано од мемориски и временски аспект. За овој проблем, доколку користиме информирано пребарување, овој алгоритам **е дефинитивно најоптимален.**

Втор дел: Проблеми кои исполнуваат услови

Задача со подготовка на колоквиум

1.
    1. Доменот на сите променливи се броевите од 1 до 7, кои кореспондираат со секој ден во неделата почнувајќи од понеделник и завршувајќи со недела domain = [1,2,3,4,5,6,7]
    2. Променливите се сите активности:
        1. Домашна задача (D)
        2. Лаб вежба (L)
        3. Тест за самостојна проверка (S)
        4. Повторување на теоретски дел(P1)
        5. Повторување на програмски дел(P2)
    3. Constraints
        1. Сите активности мора да имаат различна вредност
        2. Домашна задача и лаб вежба не смее да добијат вредност поголема од 4
        3. Лаб вежба мора да има вредност еднаква на 2, 4 или 6
        4. Повторувањето на теоретскиот дел мора да има помала вредност од повторувањето на програмскиот дел
        5. Разликата меѓу вредностите од двете повторувања мора да биде > 1
        6. Вредностите на D, L и S, мора да бидат помали од вредностите на P1
    4. Граф: (го нацртав имајќи ги предвид условите)



5. Додела на вредности:
   Првично ќе ги обележам за сите променливи кои вредности им се во доменот пред било какво доделување на вредности или пред Arc Consistency Enforcing(со „Да“ дека го има во доменот, со „Не“ дека го нема)

Променлива	1	2	3	4	5	6	7
D	Да	Да	Да	Да	Не	Не	Не
L	Не	Да	Не	Да	Не	Не	Не
S	Да	Да	Да	Да	Да	Да	Да
P1	Да	Да	Да	Да	Да	Да	Да
P2	Да	Да	Да	Да	Да	Да	Да

Потоа со Arc Consistency Enforcing ќе ги ажурирам домените на S и P1 со тоа што D, L и S мора да бидат пред P1, односно да имаат помала вредност. Тоа значи дека максималната дозволена вредност за S е 4, а од друга страна минималната вредност на P1 е 4.

Променлива	1	2	3	4	5	6	7
D	Да	Да	Да	Да	Не	Не	Не
L	Не	Да	Не	Да	Не	Не	Не
S	Да	Да	Да	Да	Не	Не	Не
P1	Не	Не	Не	Да	Да	Да	Да
P2	Да	Да	Да	Да	Да	Да	Да

Потоа пак со ACE ги ажурирам вредностите овој пат на P2, поради тоа што мора да има поголема вредност од P1, а исто така не смеат да бидат во последователни денови, се добива дека P2 може да ги добие вредностите 6 или 7.

Променлива	1	2	3	4	5	6	7
D	Да	Да	Да	Да	Не	Не	Не
L	Не	Да	Не	Да	Не	Не	Не
S	Да	Да	Да	Да	Не	Не	Не
P1	Не	Не	Не	Да	Да	Да	Да
P2	Не	Не	Не	Не	Не	Да	Да

Потоа доделување на вредности започнува со доделување на вредност за L според евристиката MRV, и tiebreaker е растечки алфабетски редослед. Доделени се вредностите кај кои пишува „Доделено“, па со Forward Checking се ажурираат домените на преостанатите променливи, бележам со „Да“ дека ја има/останува во доменот, со „Не“ дека ја нема.

Променлива	1	2	3	4	5	6	7
D	Да	Не	Да	Да	Не	Не	Не
L	//	Доделено	//	//	//	//	//
S	Да	Не	Да	Да	Не	Не	Не
P1	Не	Не	Не	Да	Да	Да	Да
P2	Не	Не	Не	Не	Не	Да	Да

Следно со MRV се доделува вредност на P1 па се прави Forward Checking

Променлива	1	2	3	4	5	6	7
D	Да	Не	Да	Да	Не	Не	Не
L	//	Доделено	//	//	//	//	//
S	Да	Не	Да	Да	Не	Не	Не
P1	Не	Не	Не	Да	Да	Не	Да
P2	//	//	//	//	//	Доделено	//

Доделување на вреднсот на D користејќи алфабетски редослед за tiebreaker

Променлива	1	2	3	4	5	6	7
D	Доделено	//	//	//	//	//	//
L	//	Доделено	//	//	//	//	//
S	Не	Не	Да	Да	Не	Не	Не
P1	Не	Не	Не	Да	Да	Не	Да
P2	//	//	//	//	//	Доделено	//

Потоа вредност на S според MRV

Променлива	1	2	3	4	5	6	7
D	Доделено	//	//	//	//	//	//
L	//	Доделено	//	//	//	//	//
S	//	//	Доделено	//	//	//	//
P1	Не	Не	Не	Да	Да	Не	Да
P2	//	//	//	//	//	Доделено	//

И за крај вредност добива P1 според LCV евристика. Вредноста е 4 бидејќи таа најдобро ги задоволува условите

Променлива	1	2	3	4	5	6	7
D	Доделено	//	//	//	//	//	//
L	//	Доделено	//	//	//	//	//
S	//	//	Доделено	//	//	//	//
P1	//	//	//	Доделено	//	//	//
P2	//	//	//	//	//	Доделено	//

Со тоа вредностите доделени на D, L, S, P1, P2 се 1, 2, 3, 4, 6, соодветно