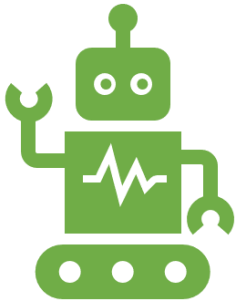
Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Проблеми кои задоволуваат услови



Аудиториски вежби по курсот Вештачка интелигенција

2023/2024

Реални проблеми

- Во секојдневието многу проблеми имаат ограничувања
- За да се реши проблемот мора сите ограничувања да се задоволени
- Ако ја разгледаме играта Sudoku така што некои полиња се иницијално пополнети со броеви. Треба да се пополнат празните полиња со броеви од 1 до 9 така што број не се повторува во ниеден ред, колона и блок

Дефиниција

- Проблеми кои задоволуваат услови се проблеми чие решение има одредени ограничувања/услови (constraints) од каде доаѓа името Constraint Satisfaction (CSP).
- Се состои од:
  - конечно множество на променливи во кои се чува решението ($V = \{V_1, V_2, V_3, \dots, V_n\}$)
  - множество од дискретни вредности – домен од каде се избираат решенијата ($D = \{D_1, D_2, D_3, \dots, D_n\}$)
  - конечно множество на ограничувања ($C = \{C_1, C_2, C_3, \dots, C_n\}$)

Дефиниција

- Елементите во доменот можат да бидат непрекинати или дискретни но во ВИ, генерално се справуваме само со дискретни вредности
- Сите множества се конечни освен доменот
- Секоја променлива во множеството може да има различни домени

Видови ограничувања

- Унарни ограничувања вклучуваат една променлива
  - пр., x ≠ 1
- Бинарни ограничувања вклучуваат парови променливи,
  - пр., x ≠ y
- Ограничувања од повисок ред вклучуваат три или повеќе променливи
  - пр., ограничувања на колона во матрица
- Преференци (soft constraints) пр. вредност 1 е подобро од вредност 2 може да се претстави со додавање на цена за секое доделување
  - се користи при оптимизација на проблеми со ограничувања

Придобивки од CSP

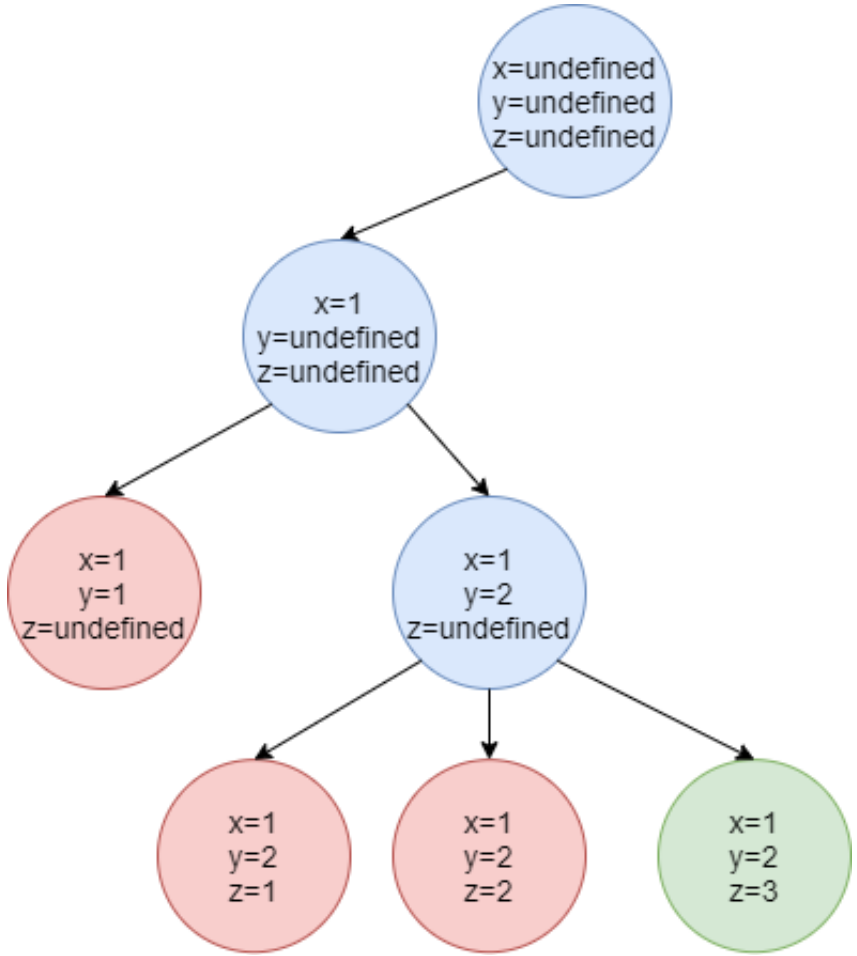
- Чиста спецификација на многу проблеми со генеричка цел
- Во CSP се знае која променлива го прекршува кое ограничување и може да се фокусира пребарувањето
- Автоматско кастрење на гранки кои прекршуваат ограничувања

Backtracking пребарување

- Доделување на променливи е комутативно
  - Пример [step 1: x = 1; step 2: y = 2] е еквивалентно на [step 1: y = 2; step 2: x = 1]
- Се земаат во предвид само доделувања на една променлива во еден јазел
- Depth-first search за CSP со доделување само на една променлива во еден јазел се нарекува backtracking пребарување
- Backtracking пребарување е основниот неинформиран алгоритам за CSP

Backtracking пребарување - пример

- Треба да се доделат три вредности (1,2,3) на променливите x, y и z така што нивната вредност ќе биде различна.



Подобрување на ефикасноста на backtracking

- Со генерални методи и хевристики може да се добие многу поголема брзина
- Хевристики:
  - Q: На која променлива наредно да и доделам вредност?
    1. Најограничената променлива (Most constrained variable)
    2. Променливата која најмногу ми ги ограничува наредните променливи (Most constraining variable)
  - Q: Во кој редослед да се пробуваат вредности
    1. Вредноста која најмалку ги ограничува наредните чекори (Least constraining value)
  - Q: Дали можам да предвидам неуспех во рана фаза?
    1. Однапред проверување на вредности (Forward checking)

Хевристики

Хевристика	Како работи
Most constrained variable	Избери ја променливата која има најмал број на дозволени вредности или позната како minimum remaining values (MRV) heuristic
Most constraining variable	Ако имаме две или повеќе променливи со ист број на дозволени вредности, дополнително може да ја користиме оваа хевристика за да ја избереме онаа променлива која најмногу ги ограничува наредните променливи
Least constraining value	Избери ја вредноста која ограничува најмал број на вредности за наредните променливи
Forward checking	Чувај ги преостанатите дозволени вредности за недоделените променливи. Прекини кога некоја недоделена променлива нема дозволени вредности

Популарни CS проблеми

- Crypt Arithmetic (кодирање на букви во цифри)
- N-Queen (N кралици треба да бидат поставени во nxn матрица така што ниедна кралица нема ист ред, колона и дијагонала)
- Map Coloring (боење на различни региони на мапа така што ниеден соседен регион нема иста боја)
- Crossword (секојдневни крстозбори во весници)
- Sudoku

Конвертирање на проблеми во CSP

- За проблем да се конвертира во CSP потребни се следните чекори:
  - Чекор 1: Креирај множество на променливи
  - Чекор 2: Креирај множество на домени
  - Чекор 3: Креирај множество на ограничувања за променливите и домените
  - Чекор 4: Најди оптимално решение

Модул за решавање

- Инсталација:

```
>>pip install python-constraint
```

- Овозможува апстракција над проблемите кои треба да задоволуваат услови со едноставен интерфејс
- Работи над конечни домени и користи основен Python

Класи

- *Problem:* класа со која се дефинира проблем и се земаат решенија на дефинираниот проблем
- *Variable:* класа со која се дефинира променлива
- *Domain:* класа со која се дефинираат можни вредности на променлива

Различни начини на решавање (Solvers)

- *Solver:* апстрактна базна класа
- *BacktrackingSolver:* решавање со backtracking
- *RecursiveBacktrackingSolver:* решавање со рекурзивен backtracking
- *MinConflicsSolver:* решавање базирано на теоријата на минимални конфликти

Поставување ограничувања

- *Constraint:* апстрактна базна класа за ограничувања
- *FunctionConstraint:* ограничување претставено со функција
- *AllDifferentConstraint:* ограничување кое наложува сите дадени променливи да имаат различна вредност
- *AllEqualConstraint:* ограничување кое наложува сите дадени променливи да имаат иста вредност
- *MaxSumConstraint:* ограничување кое наложува сите дадени променливи да имаат максимална сума како наведената
- *MinSumConstraint:* ограничување кое наложува сите дадени променливи да имаат минимална сума како наведената

Поставување ограничувања

- *InSetConstraint*: ограничување кое наложува сите дадени променливи да имаат вредност присутна во множеството вредности
- *NotInSetConstraint*: ограничување кое наложува сите дадени променливи да немаат вредност присутна во множеството вредности
- *SomeInSetConstraint*: ограничување кое наложува некои од дадените променливи да имаат вредност присутна во множеството вредности
- *SomeNotInSetConstraint*: ограничување кое наложува некои од дадените променливи да немаат вредност присутна во множеството вредности

Пример 1

- Дадена ни е мапата на Австралија и треба да ја обоиме со три бои: сина, зелена и црвена. Соседните региони не смее да имаат иста боја. Секој регион може да има една од трите бои.

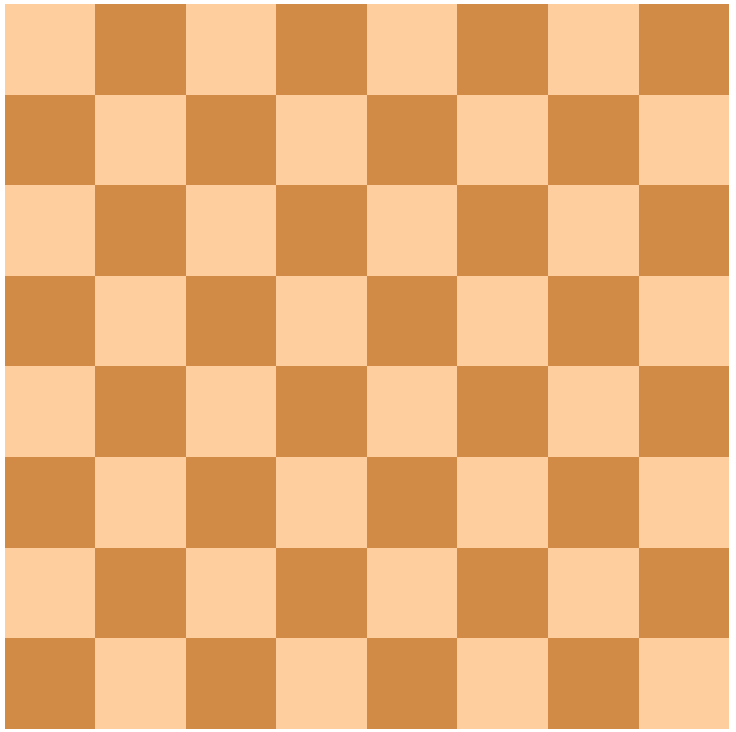
Пример 1: Декларирај множество променливи, домен и ограничувања

- Променливи: WA, NT, Q, NSW, V, SA, T
- Домени: $D_i = \{red, green, blue\}$
- Ограничувања: соседните региони мора да имаат различни бои
  - пр., WA ≠ NT

- Променливи: WA, NT, Q, NSW, V, SA, T
- Домени: $D_i = \{red, green, blue\}$
- Ограничувања: соседните региони мора да имаат различни бои
  - пр., WA ≠ NT

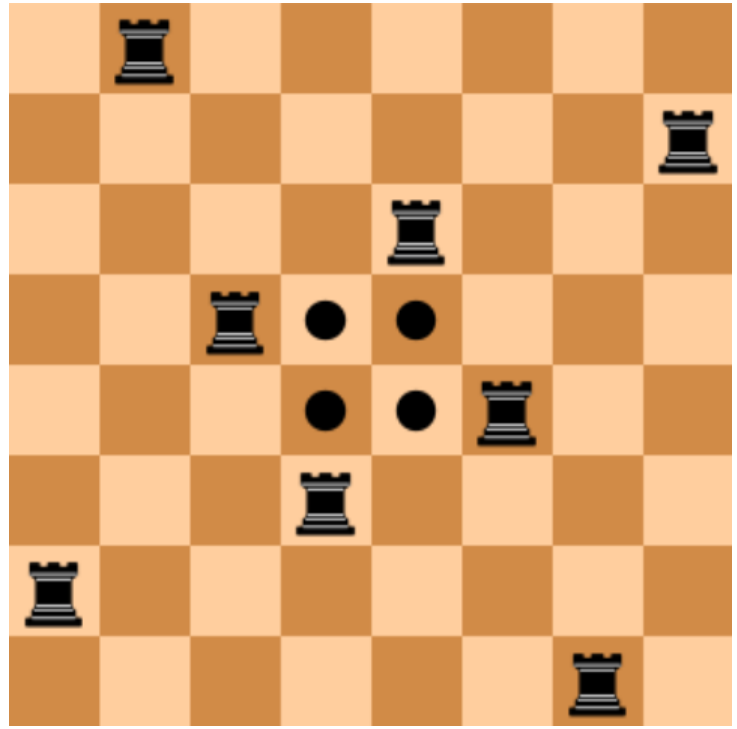
Пример 2

- Дадена ни е 8x8 табла за шах. Треба да се постават 8 топови на таблата така што ниеден топ да не се напаѓа. Топовите може да се постават на било која позиција која сметаме дека е најсоодветна. Единственото ограничување е дека не треба да се напаѓаат.



Пример 2: Декларирај множество променливи, домен и ограничувања

- Ако не сакаме топовите да се напаѓаат треба нивната редица и колона да биде различна
- секој топ постави го во различна колона и стави ги редиците како променливи. Треба да одлучиме во која редица ќе го поставиме секој топ. (може и обратно – постави ги во различни редици и стави колони како променливи)
- Променливи: rook1, rook2,…rook8
- Домени: $D_i = \{0,1,2, \ldots 7\}$



Пример 3

- Магичен квадрат е nxn квадратна матрица (каде *n* е бројот на ќелии на секоја страна) пополнет со различни природни броеви во ранг 1,2,….,n^2 така што секоја ќелија содржи различен број и сумата на секој ред, колона и дијагонала е иста. Сумата се нарекува магична константа или магична сума на магичниот квадрат.
- Даден ни е 4x4 магичен квадрат. Треба да ги пополниме ќелиите со различни природни броеви во ранг 1,2,…,16 така што секоја ќелија ќе содржи различен број и сумата на секој ред, колона и дијагонала ќе биде 34.

2	16	13	3
11	5	8	10
7	9	12	6
14	4	1	15

Пример 3: Декларирај множество променливи, домен и ограничувања

- 16 позиции кои треба да бидат пополнети со 16 различни вредности
- Променливи: position1,position2,….,position16
- Домени: $D_i = \{1,2,\ldots.16\}$ за секоја променлива

2	16	13	3
11	5	8	10
7	9	12	6
14	4	1	15