Inteligență artificială
Laborator 4

Probleme de satisfacere a constrângerilor

O **problemă de satisfacere a constrângerilor** (*Constraint Satisfaction Problem*):
- o mulțime de variabile $X=\{X_1,..., X_n\}$
- fiecare variabilă X_i poate lua valori dintr-un domeniu D_i
- o mulțime de constrângeri $C=\{C_1 ,..., C_t\}$ care specifică combinațiile permise de valori.
O **soluție** pentru o astfel de problemă este o asignare de valori variabilelor a.î. toate constrângerile să fie satisfăcute.

În definiție nu se impune nici o condiție asupra tipului variabilelor. Acestea pot fi întregi, logice sau de orice alt tip. Nici modul de definire a constrângerilor nu este limitat. Constrângerile pot fi date atât explicit, prin specificarea tuplelor de valori permise, cât şi implicit, prin relații (ex: $X_i > 2$).

Graful restricțiilor: nodurile reprezintă variabilele, muchiile reprezintă restricțiile între variabile.

Exemplu: Problema colorării unei hărți
Considerăm o hartă cu *n* țări. Fiecare regiune/țară poate fi colorată cu o culoare dintr-o mulțime de culori asociate. Să se coloreze harta a.î. regiunile/țările vecine să fie colorate diferit.

Modelare: asignăm fiecărei regiuni/țări de pe hartă o variabilă; domeniul fiecărei variabile este o mulțime de culori specificată. Restricțiile sunt de forma: $X_i \neq X_j$ (două țări vecine au culori diferite). Obs: între două țări vecine în graful constrâns vom adăuga o muchie.

Metode de rezolvare:
- Algoritmul **Backtracking**: menține o soluție parțială (o mulțime de variabile instanțiate corect) pe care o extinde pas cu pas. Inițial, mulțimea este vidă. La fiecare pas se selectează următoarea variabilă din ordonare şi se încearcă asignarea variabilei cu o valoare consistentă cu instanțierea parțială. Dacă este găsită o astfel de valoare, algoritmul continuă procedeul cu următoarea variabilă din ordonare. În caz contrar, ne întoarcem la variabila anterioară şi îi asignăm o altă valoare consistentă.

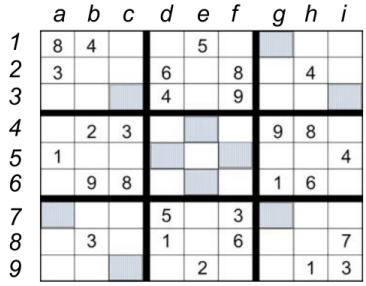
- **Propagarea constrangerilor**
Forward-checking: fiecare valoare aleasă trebuie să fie compatibilă cu cel puțin o valoare din domeniul variabilelor neasignate cu care au în comun o restricție. Se instanțiază variabila cu o valoare şi apoi se elimină valori din domeniul variabilelor viitoare care sunt în conflict cu instanțierea curentă. Dacă domeniul unei variabile viitoare devine vid, algoritmul consideră următoarea valoare posibilă pentru variabila curentă.

Ordonarea variabilelor
MRV (*Minimum remaining values*): alege variabila cu cele mai puține valori rămase în domeniu

Temă
Considerăm o tablă de dimensiune 9x9, parțial completată cu cifre de la 1 la 9. Fiecare linie, coloană sau regiune 3x3 conține toate valorile de la 1 la 9. Anumite căsuțe trebuie să conțină un număr par. Problema constă în identificarea numerelor lipsă.

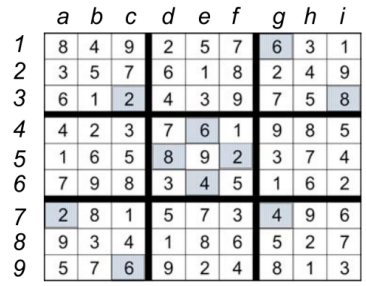
Aplicați algoritmul *Forward checking* împreună cu o metodă de ordonare a variabilelor pentru a determina soluțiile unei instanțe date.

Exemplu:



	a	b	c	d	e	f	g	h	i
1	8	4			5				
2	3			6		8		4	
3				4		9			
4		2	3				9	8	
5	1								4
6		9	8				1	6	
7				5		3			
8		3		1		6			7
9					2			1	3

Soluția instanței de mai sus:



	a	b	c	d	e	f	g	h	i
1	8	4	9	2	5	7	6	3	1
2	3	5	7	6	1	8	2	4	9
3	6	1	2	4	3	9	7	5	8
4	4	2	3	7	6	1	9	8	5
5	1	6	5	8	9	2	3	7	4
6	7	9	8	3	4	5	1	6	2
7	2	8	1	5	7	3	4	9	6
8	9	3	4	1	8	6	5	2	7
9	5	7	6	9	2	4	8	1	3

Etape

(0.3) 1. Modelarea problemei ca o problemă de satisfacere a constrângerilor
- identificarea variabilelor, domeniilor, restricțiilor
- inițializarea acestora pentru o instanță dată

(0.5) 2. Implementarea metodei *Forward Checking*

(0.2) 3. Implementarea metodei de ordonare a variabilelor (MRV)

Bonus (0.1p): Implementați și testați metoda *Arc consistency*

Instanțe de testare: http://www.sudoku-space.com/even-odd-sudoku/

Termen limită: săptămâna 5 (30 oct - 3 noiembrie)

Resurse

Artificial Intelligence: A Modern Approach, capitolul 5 http://aima.cs.berkeley.edu/newchap05.pdf

Homework

Consider a board of size 9x9, partially filled with digits from 1 to 9. Each row, column, or 3x3 sub-region contains all values from 1 to 9. Certain cells must contain an even number. The problem consists in identifying the missing numbers.

Apply the *Forward checking* algorithm together with a variable ordering method to determine the solutions of a given instance.

Example:

	a	b	c	d	e	f	g	h	i
1	8	4			5				
2	3			6		8		4	
3				4		9			
4		2	3				9	8	
5	1								4
6		9	8				1	6	
7				5		3			
8		3		1		6			7
9					2			1	3

The solution of the instance:

	a	b	c	d	e	f	g	h	i
1	8	4	9	2	5	7	6	3	1
2	3	5	7	6	1	8	2	4	9
3	6	1	2	4	3	9	7	5	8
4	4	2	3	7	6	1	9	8	5
5	1	6	5	8	9	2	3	7	4
6	7	9	8	3	4	5	1	6	2
7	2	8	1	5	7	3	4	9	6
8	9	3	4	1	8	6	5	2	7
9	5	7	6	9	2	4	8	1	3

Steps
(0.3) 1. Model the problem as a constraint satisfaction problem
- identify the variables, the domains, the constraints
- initialize them for a given instance
(0.5) 2. Implement the *Forward checking* method
(0.2) 3. Implement a variable ordering method.

Bonus(0.1p): Implement and test the *Arc consistency* method

Instances: http://www.sudoku-space.com/even-odd-sudoku/

Deadline: week 5 (30 Oct - 3 Nov)

Resources
Artificial Intelligence: A Modern Approach, Chapter 5 http://aima.cs.berkeley.edu/newchap05.pdf