

TDT4136 Introduction to Artificial Intelligence

Assignment 5 – Solving Constraint Satisfaction Problems

Oppgave 1

Se vedlagt Python-fil assignment5.py

Oppgave 2

```
C:\Users\tysse>cd C:\Users\tysse\Dropbox\NTNU\Høst 2018\AI\øvinger\ov5

C:\Users\tysse\Dropbox\NTNU\Høst 2018\AI\øvinger\ov5>python assignment5.py
Assignment 5 - Solving Constraint Satisfaction Problems
Martin Tysseland and Ernst Torsgård

Sudoku:  easy
BackTrack Called:  0
Backtrack Failure: 0
7 8 4 | 9 3 2 | 1 5 6
6 1 9 | 4 8 5 | 3 2 7
2 3 5 | 1 7 6 | 4 8 9
-----+-----+-----
5 7 8 | 2 6 1 | 9 3 4
3 4 1 | 8 9 7 | 5 6 2
9 2 6 | 5 4 3 | 8 7 1
-----+-----+-----
4 5 3 | 7 2 9 | 6 1 8
8 6 2 | 3 1 4 | 7 9 5
1 9 7 | 6 5 8 | 2 4 3

Sudoku:  medium
BackTrack Called:  2
Backtrack Failure: 0
8 7 5 | 9 3 6 | 1 4 2
1 6 9 | 7 2 4 | 3 8 5
2 4 3 | 8 5 1 | 6 7 9
-----+-----+-----
4 5 2 | 6 9 7 | 8 3 1
9 8 6 | 4 1 3 | 2 5 7
7 3 1 | 5 8 2 | 9 6 4
-----+-----+-----
5 1 7 | 3 6 9 | 4 2 8
6 2 8 | 1 4 5 | 7 9 3
3 9 4 | 2 7 8 | 5 1 6
```

```
Sudoku:  hard
BackTrack Called:  25
Backtrack Failure: 21
1 5 2 | 3 4 6 | 8 9 7
4 3 7 | 1 8 9 | 6 5 2
6 8 9 | 5 7 2 | 3 1 4
-----+-----+-----
8 2 1 | 6 3 7 | 9 4 5
5 4 3 | 8 9 1 | 7 2 6
9 7 6 | 4 2 5 | 1 8 3
-----+-----+-----
7 9 8 | 2 5 3 | 4 6 1
3 6 5 | 9 1 4 | 2 7 8
2 1 4 | 7 6 8 | 5 3 9

Sudoku:  veryhard
BackTrack Called:  42
Backtrack Failure: 35
4 3 1 | 8 6 7 | 9 2 5
6 5 2 | 4 9 1 | 3 8 7
8 9 7 | 5 3 2 | 1 6 4
-----+-----+-----
3 8 4 | 9 7 6 | 5 1 2
5 1 9 | 2 8 4 | 7 3 6
2 7 6 | 3 1 5 | 8 4 9
-----+-----+-----
9 4 3 | 7 2 8 | 6 5 1
7 6 5 | 1 4 3 | 2 9 8
1 2 8 | 6 5 9 | 4 7 3

C:\Users\tysse\Dropbox\NTNU\Høst 2018\AI\Øvinger\ov5>
```

Oppgave 3

Man kan se antall ganger BACKTRACK-funksjonen ble kjørt og returnerte *failure* i oppgave 2. *Failure* blir returnert hver gang BACKTRACK velger en verdi som gir ingen løsning på problemet (en variabel i *domains* ender opp med ingen gyldige verdier).

For *easy* var det ikke nødvendig å kjøre BACKTRACK siden INFERENCE med AC-3 løste brettet på første forsøk. INFERENCE kom da frem til et *domains* der alle variablene kun hadde en gyldig verdi med en gang.

For *medium* ble BACKTRACK kjørt to ganger og returnerte *failure* i ingen av tilfellene. Dette betyr at den første verdien til variabelen som SELECT-UNASSIGNED-VARIABLE i BACKTRACK returnerte var den som gav en løsning på problemet i begge tilfellene.

For *hard* ble BACKTRACK kjørt 25 ganger og returnerte *failure* 21 ganger. Nå er brettet mye vanskeligere og INFERENCE greier ikke å eliminere så mange verdier for variablene i *domains*. Dette fører til at BACKTRACK må teste mange flere verdier enn tidligere for å finne en løsning.

For *veryhard* ble BACKTRACK kjørt 42 ganger og returnerte *failure* 35 ganger. Samme tilfelle som i *hard*, men enda vanskeligere. Hadde vi forbedret INFERENCE sånn at den brukte flere komplekse strategier ville vi kommet frem til en løsning raskere.

Finner en løsning:

BACKTRACK called > BACKTRACK returned failure

Ingen løsning:

BACKTRACK called = BACKTRACK returned failure