

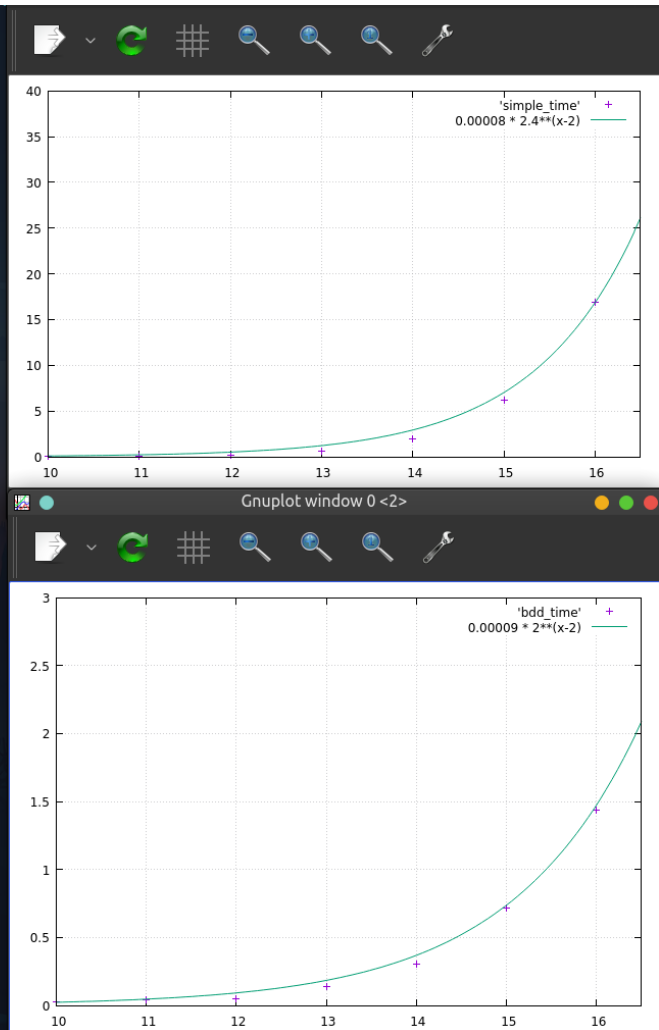
Buzatu-Pahontu Mihnea-Nicolae
324CB
HR username: mihnea_buzatu

Tema 3 AA – SAT Solver

```
mihnea@IdeaPad:SAT$ python3 checker.py simple
-- Tema 3 AA checker --

Testing the simple SAT solver:
[Test 00]: Passed [10p] Time: [0.040] Size: [7 32]
[Test 01]: Passed [10p] Time: [0.031] Size: [8 64]
[Test 02]: Passed [10p] Time: [0.027] Size: [9 128]
[Test 03]: Passed [10p] Time: [0.039] Size: [10 256]
[Test 04]: Passed [10p] Time: [0.039] Size: [11 512]
[Test 05]: Passed [10p] Time: [0.205] Size: [12 1024]
[Test 06]: Passed [10p] Time: [0.648] Size: [13 2048]
[Test 07]: Passed [10p] Time: [1.938] Size: [14 4096]
[Test 08]: Passed [10p] Time: [6.094] Size: [15 8192]
[Test 09]: Passed [10p] Time: [16.272] Size: [16 16384]
Total: 100
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$
mihnea@IdeaPad:SAT$ python3 checker.py bdd
-- Tema 3 AA checker --

Testing the BDD-SAT solver:
[Test 00]: Passed [10p] Time: [0.035] Size: [7 32]
[Test 01]: Passed [10p] Time: [0.029] Size: [8 64]
[Test 02]: Passed [10p] Time: [0.031] Size: [9 128]
[Test 03]: Passed [10p] Time: [0.032] Size: [10 256]
[Test 04]: Passed [10p] Time: [0.049] Size: [11 512]
[Test 05]: Passed [10p] Time: [0.054] Size: [12 1024]
[Test 06]: Passed [10p] Time: [0.145] Size: [13 2048]
[Test 07]: Passed [10p] Time: [0.307] Size: [14 4096]
[Test 08]: Passed [10p] Time: [0.711] Size: [15 8192]
[Test 09]: Passed [10p] Time: [1.394] Size: [16 16384]
Total: 100
mihnea@IdeaPad:SAT$
```



Atat in teorie cat si in practica, ambii algoritmi se executa in timp exponential, insa algoritmul BDD este considerabil mai rapid. Conform graficelor, folosind testele de proba, pare sa aiba o complexitate de $O(2^n)$, pe cand algoritmul Simple SAT are o complexitate experimentală de $O(2.5^n)$.

Desi complexitatea este in principal data de generarea tuturor interpretarilor, si evaluarea formulei este o operatie costisitoare, avand o complexitate de $O(n_{var} * n_{clauze})$ (in testele de proba, n_{clauze} creste exponential fata de n_{var}). Aici intervine cea mai mare optimizare a BDD, variabilele sunt eliminate pe rand, iar clauzele unde este posibil. Formula devine mai simpla pe masura ce arborele creste in adancime, iar cand o clauza devine 0, acea formula nu mai genereaza alti copii.

De cate ori este mai rapid BDD decat Simple SAT pentru testele de proba:

Test6 4.3x
Test7 6.2x
Test8 8.6x
Test9 11.6x