

INSTRUCTIONS

There is a .cpp file named dpll.cpp. It can be compiled in a Linux environment with g++ support. Once compiled, it generates an executable file that takes a .kb file as input. It can also take another argument which is a string of facts separated by space as given in the multi-agent task assignment example execution in question. Both the example.kb (for the abstract Boolean problem) and agent_Jobs.kb (for the multi-agent task-assignment problem) files have been included in the zip.

Steps to solve for multi-agent task-assignment problem:

For the jobs "painter sander gluer joiner"

```
$ g++ dpll.cpp  
$ ./a.out agent_Jobs.kb "painter sander gluer joiner"
```

For the jobs "cutter welder painter joiner recharger"

```
$ g++ dpll.cpp  
$ ./a.out agent_Jobs.kb "cutter welder painter joiner recharger"
```

Steps to solve for the abstract Boolean problem as given in the question:

```
$ g++ dpll.cpp  
$ ./a.out example.kb
```

Note that in the multi-agent task-assignment problem, when at last the program prints the three agents, it assumes that agents are represented as single characters in the props.

By default, both the unit_clause and pure_symbol heuristics are active. To disable either of them, please comment the code related to that heuristic (around line number 591 in DPLL function) in dpll.cpp. Comments are provided in the dpll.cpp file to make it easier to do so.

The knowledge base for the builder-agent team assignment problem is agent_Jobs.kb (has been converted to CNF and attached in this zip folder).

For any queries please feel free to call or message me.

Abhinandan Aryadipta (Abhi)
UIN: 925001240
Email: aryarockstar@tamu.edu
Ph: 9799857204