Programing Assignment #7

CSCE 625 - Artificial Intelligence

Spring 2015

Name: Xiaoshu Zhang

Email: kallen5208@gmail.com

1. 1. Run and compile the code

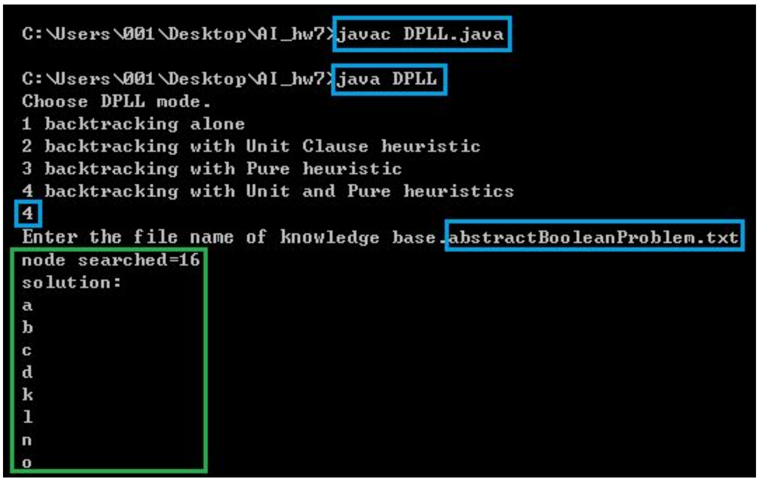
1.1.

The file I turned in are "DPLL.java", "KBGenerator.java", "KBSeed.txt", "KB.txt", "abstractBooleanProblem.txt".

1.2.

Compile with command "javac DPLL.java", then run with command "java DPLL".

Choose DPLL mode, 1 or 2 or 3 or 4. Then enter the file name of knowledge base, "abstractBooleanProblem.txt", shown in Figure 1. It will generate a transcript named "transcript_abstractBooleanProblem.txt_4.txt".



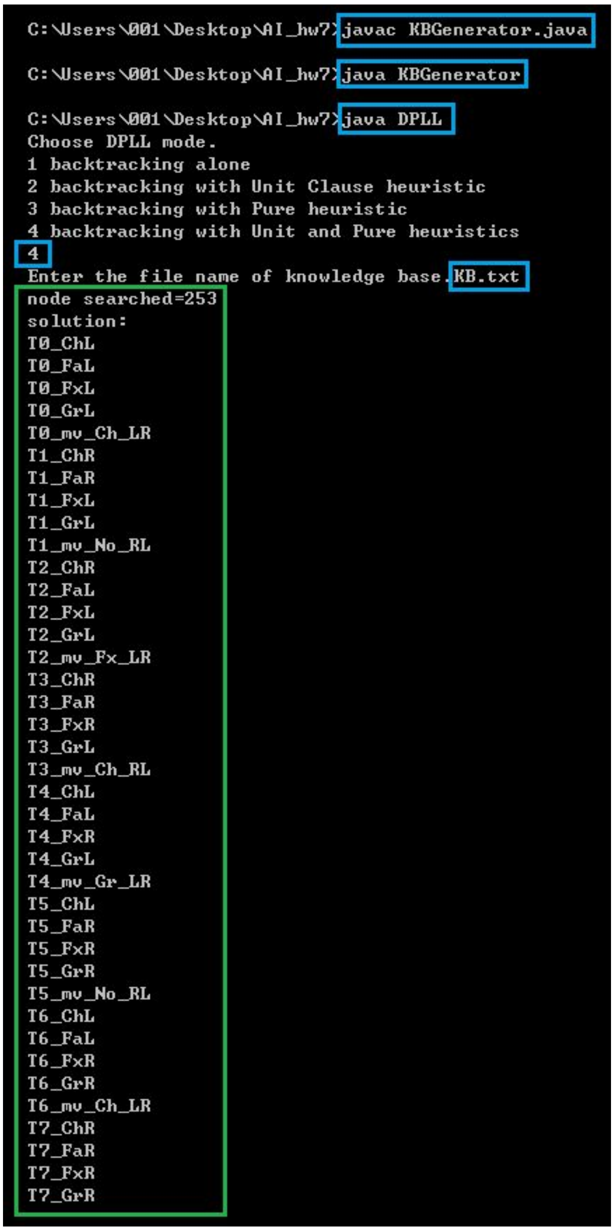
```
C:\Users\001\Desktop\AI_hw7>javac DPLL.java

C:\Users\001\Desktop\AI_hw7>java DPLL
Choose DPLL mode.
1 backtracking alone
2 backtracking with Unit Clause heuristic
3 backtracking with Pure heuristic
4 backtracking with Unit and Pure heuristics
4
Enter the file name of knowledge base.abstractBooleanProblem.txt
node searched=16
solution:
a
b
c
d
k
l
n
o
```

Figure 1

1.3.

Compile with command "javac KBGenerator.java", then run with command "java KBGenerator". It will use "KBSeed.txt" to generate "KB.txt", the knowledge base for Farmer-Fox-Chicken-Grain problem.

Run with command "java DPLL". Choose DPLL mode, 1 or 2 or 3 or 4. Then enter the file name of knowledge base, "KB.txt", shown in Figure 2. It will generate a transcript named "transcript_KB.txt_4.txt".



```
C:\Users\001\Desktop\AI_hw7>javac KBGenerator.java

C:\Users\001\Desktop\AI_hw7>java KBGenerator

C:\Users\001\Desktop\AI_hw7>java DPLL
Choose DPLL mode.
1 backtracking alone
2 backtracking with Unit Clause heuristic
3 backtracking with Pure heuristic
4 backtracking with Unit and Pure heuristics
4
Enter the file name of knowledge base:KB.txt
node searched=253
solution:
T0_ChL
T0_FaL
T0_FxL
T0_GrL
T0_mv_Ch_LR
T1_ChR
T1_FaR
T1_FxL
T1_GrL
T1_mv_No_RL
T2_ChR
T2_FaL
T2_FxL
T2_GrL
T2_mv_Fx_LR
T3_ChR
T3_FaR
T3_FxR
T3_GrL
T3_mv_Ch_RL
T4_ChL
T4_FaL
T4_FxR
T4_GrL
T4_mv_Gr_LR
T5_ChL
T5_FaR
T5_FxR
T5_GrR
T5_mv_No_RL
T6_ChL
T6_FaL
T6_FxR
T6_GrR
T6_mv_Ch_LR
T7_ChR
T7_FaR
T7_FxR
T7_GrR
```

Figure 2

2. Performance

Table 1 Performance

DPLL mode	Abstract Boolean Problem	Farmer-Fox-Chicken-Grain problem
backtracking alone	56	1826
backtracking with Unit Clause heuristic	34	249
backtracking with Pure heuristic	16	1532
backtracking with Unit and Pure heuristics	16	253

In Abstract Boolean Problem, when used separately, both Unit and Pure heuristic can increase the search efficiency, and Pure heuristic is more effective.

In Farmer-Fox-Chicken-Grain problem, when used separately, both Unit and Pure heuristic can increase the search efficiency, and Unit heuristic is more effective.

To sum up, in general cases both Unit and Pure heuristic can increase the search efficiency. However, their impacts may be different, depending on the characteristic of the knowledge base used.

---------------- There are appendixes in the next pages. ----------------

Appendix A: Knowledge base for Farmer-Fox-Chicken-Grain

1.1. States. Each object should have a unique location in a given time point.

T0_FaL <=> -T0_FaR

T0_FxL <=> -T0_FxR

T0_ChL <=> -T0_ChR

T0_GrL <=> -T0_GrR

1.2. Actions.

(1). Effect axioms

T0_mv_Fx_LR => T0_FaL ∧ T0_FxL ∧ T1_FaR ∧ T1_FxR

T0_mv_Fx_RL => T0_FaR ∧ T0_FxR ∧ T1_FaL ∧ T1_FxL

T0_mv_Ch_LR => T0_FaL ∧ T0_ChL ∧ T1_FaR ∧ T1_ChR

T0_mv_Ch_RL => T0_FaR ∧ T0_ChR ∧ T1_FaL ∧ T1_ChL

T0_mv_Gr_LR => T0_FaL ∧ T0_GrL ∧ T1_FaR ∧ T1_GrR

T0_mv_Gr_RL => T0_FaR ∧ T0_GrR ∧ T1_FaL ∧ T1_GrL

T0_mv_No_LR => T0_FaL ∧ T1_FaR

T0_mv_No_RL => T0_FaR ∧ T1_FaL

(2). At time *i*, there should be at least one action:

T0_mv_Fx_LR ∨ T0_mv_Fx_RL ∨ T0_mv_Ch_LR ∨ T0_mv_Ch_RL

∨ T0_mv_Gr_LR ∨ T0_mv_Gr_RL ∨ T0_mv_No_LR ∨ T0_mv_No_RL

(3). At time *i*, there should be at most one action

There are too many clauses to write here.

(4). Successor-state axioms

T1_FxL <=> T0_mv_Fx_RL ∨ (T0_FxL ∧ -T0_mv_Fx_LR)

T1_FxR <=> T0_mv_Fx_LR ∨ (T0_FxR ∧ -T0_mv_Fx_RL)

T1_ChL <=> T0_mv_Ch_RL ∨ (T0_ChL ∧ -T0_mv_Ch_LR)

T1_ChR <=> T0_mv_Ch_LR ∨ (T0_ChR ∧ -T0_mv_Ch_RL)

T1_GrL <=> T0_mv_Gr_RL ∨ (T0_GrL ∧ -T0_mv_Gr_LR)

T1_GrR <=> T0_mv_Gr_LR ∨ (T0_GrR ∧ -T0_mv_Gr_RL)

T0_FaL <=> -T1_FaL

1.3. Eat relations. If the fox and the chicken stay together, they should be with the farmer. The same with the chicken and the grain.

T0_FxL ∧ T0_ChL => T0_FaL

T0_FxR ∧ T0_ChR => T0_FaR

T0_GrL ∧ T0_ChL => T0_FaL

T0_GrR ∧ T0_ChR => T0_FaR

1.4 Facts.

T0_FaL

T0_FxL

T0_ChL

T0_GrL

1.5. Goal.

T7_FaR

T7_FxR

T7_ChR

T7_GrR