Ahmet Burak KOÇ / N20152984
CMP 682 Artificial Intelligence Homework 1
Hacettepe University Computer Engineering

Q1) MATLAB script (Generator3SAT.M) is written to generate 3SAT problems with various number of clauses and number of variables. **Generated five 3SAT problems in SMT-LIB2 format is given (3SAT_<id>.smt2)**.

Q2) Five original form of unrestricted satisfiability problem is written in SMT-LIB2 file **(USAT_<id>-org.smt2)**. Five of them is converted to CNF. Then, clauses with more than 3 literals are transformed to 3SAT. Final 3-CNF problem with clauses containing maximum 3 literals are written in SMT-LIB2 file **(USAT_<id>-converted.smt2)**.

Q3) MATLAB script (Generator_N_Queens.m) is written to generate N-Queens SAT problem. By using this script, **4-Queens SAT is generated in SMT-LIB2 format. 4-Queens file and corresponding successful Z3 solver result is given (4-Queens.smt2, 4-Queens_Res.txt)**.

Q4) In some critical systems, real-time programming is conducted in order to execute all periodic tasks in corresponding periods and in corresponding time intervals. To do that, all tasks must be well scheduled. As number of tasks and number of processors increase, scheduling gets more complex. Through defining as CSP problem such complex scheduling problems can be solved.

There are n preemptive tasks and each task is characterized by a 4-tuple (O_i, C_i, T_i, D_i) where

- O_i: Offset of $task_i$,
- C_i: Worst-case execution time (WCET) of $task_i$,
- T_i: Period of $task_i$,
- D_i: Deadline of $task_i$. Each job k generated at time $O_i + (k-1)T_i$ must finish its execution before $D_i + O_i + (k-1)T_i$ for each $task_i$.
- $I_{i,k}$ is the k^{th} availability interval for $task_i$.
  - $I_{i,k} = [O_i + (k-1)T_i\,,\ O_i + (k-1)T_i + D_i - 1]$
- T is hyperperiod which is LCM (Least Common Multiple) of all periods. All availabilities are repeated in every hyperperiod so scheduling is going to be limited to 1 hyper period.

CSP problem definition:

Variables: $x_{i,j}(t)$ per $task_i$, $processor_j$ and time step t.

$$x_{i,j}(t) = \begin{cases} 1 & if\ task_i\ on\ processor_j\ @t \\ 0 & otherwise \end{cases}$$

Number of tasks: n

Number of processors: m

Therefore, $n * m * T$ variables exist and each can take two possible values so $Domain_{i,j}(t) = \{0,1\}$, where true (=1) or false (=0).

Constraints:

- $x_{i,j}(t) = 0$, $\forall t \notin I_{i,1} \cup \ldots \cup I_{i,\frac{T}{T_i}}$
    - Problem is defined for only one hyper period so no solution for t values out-of-scope.
- $\sum_i x_{i,j}(t) \leq 1$
    - At most 1 task can be executed at the same time slot of the same processor.
- $\sum_j x_{i,j}(t) \leq 1$
    - A task can be executed in 1 processor at most at the same time.
- $\sum_{t \in I_{i,k}} \sum_j x_{i,j}(t) = C_i, \quad \forall k \in 1..\frac{T}{T_i}$
    - Each task needs to complete its total execution time every availability period. Remember that tasks are periodic.

These constraints are defined propositional logic as below.

- $\sum_i x_{i,j}(t) \leq 1$: For all j and t values sum of $x_{i,j}(t)$ is smaller or equal to 1. Hence, for each pair of i values $(not(x_{i1,j}(t))\ or\ not(x_{i2,j}(t)))$ must be true.
- $\sum_j x_{i,j}(t) \leq 1$: For all i and t values sum of $x_{i,j}(t)$ is smaller or equal to 1. Hence, for each pair of j values $(not(x_{i,j1}(t))\ or\ not(x_{i,j2}(t)))$ must be true.
- $\sum_{t \in I_{i,k}} \sum_j x_{i,j}(t) = C_i, \quad \forall k \in 1..\frac{T}{T_i}$: For all j and $t \in\ I_{i,k}$ values sum of $x_{i,j}(t)$ equals to C_i. There are $D_i * m$ pairs of j and t values.
    - If true values larger than of C_i, it is not a solution. Hence, for each $(C_i + 1)$ items chosen combinations of j, t pairs
      $$(not(x_{i,j_1}(t_1))\ or\ not(x_{i,j_2}(t_2))\ or\ \ldots\ or\ not(x_{i,j_m}(t_{T_i})))$$
      must be true.
    - Summation result which is smaller than C_i means task execution is not completed. To avoid such solution, $D_i * m - C_i + 1$ items chosen combinations of j, t pairs,
      $$(\ x_{i,j_1}(t_1)\ or\ x_{i,j_2}(t_2)\ or\ \ldots\ or\ x_{i,j_m}(t_{D_i})\ )$$
      must be true.

To generate SAT problem in SMT-LIB2 format, MATLAB script “MyCSP.m” is written. **One sample SAT is generated for inputs (MyCSP.smt2) and successful Z3 solver result is given (MyCSP_Res.txt).**

number of tasks	: 4
number of processors	: 2
Offset	: 0 0 0 0
Worst-case execution time	: 1 1 3 3
Periods of tasks	: 4 5 5 3
Deadlines of tasks	: 4 5 5 3