CS228 Logic, Assignment 1 Report

Nitish Shah, 160030005

September 30, 2018

1 Propositions:

1.1 Sets:

C : the set of all courses, $|C| = N_c$
. if $c_i \in C$ then n_i = the number of lectures of c_i to be held in one week
R : the set of all rooms, $|R| = N_r$
T : the set of all possible times slots of all different permitted sizes(1hr, 1.5hr, 2hr, 3hr), $|T| = N_t$

1.2 Propositions:

1.2.1 Course - Time (CT)

$$C_{im} = \begin{cases} 1 & \text{when } i^{th}\text{course is assigned a time-slot } t_m \\ 0 & \text{otherwise} \end{cases}$$

where $c_i \in C$, i=1,2...N_c and $t_m \in T$, $m = 1, 2...N_t$

1.2.2 Course - Lecture number - Time (CLT)

$$P_{ijm} = \begin{cases} 1 & \text{when } j^{th} \text{ lecture(of a week) of the } i^{th} \text{ course is held in time-slot } t_m \\ 0 & \text{otherwise} \end{cases}$$

where $c_i \in C$, i=1,2...N_c, $t_m \in T$, $m = 1, 2...N_t$ and j=1...n_i

1.2.3 Course - Room - Time (CRT)

$$Q_{irm} = \begin{cases} 1 & \text{when } i^{th} \text{ course is held in time-slot } t_m \text{ in the } r^{th} \text{ room} \\ 0 & \text{otherwise} \end{cases}$$

where$c_i \in C$, i=1,2...N_c, $t_m \in T$, $m = 1, 2...N_t$ and $r_r \in R$, r=1,...,N_r

1.3 Note

1. These propositions will be zero when the course can't be held in that room (ie if the course requires a different type of room) or if the course cannot be held in that time-slot (if a lecture has 3 1hr lectures in a week, then the propositions with 1.5hr time slots will always be zero)
2. if $t_m, t_n \in T$ and $t_m = t_n$ means that t_m and t_n overlap
3. if $t_m, t_n \in T$ and $t_m \neq t_n$ means that t_m and t_n don't overlap

2 Constraints

2.1 Course-Time-Constraints (CT)

these propositions take care of not scheduling multiple courses at the same time for professors or batches

2.1.1 No two lectures that have a common professor should be held at overlapping times:

$$\bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (C_{im} \implies \bigwedge_{\substack{n=0 \\ t_n \in T \\ t_n = t_m}}^{N_t} (\neg C_{jn})) \quad \text{if } c_i, c_j \text{ are taken by the same professor} \tag{1}$$

If professor is taking course c_i at time t_m, then no other course c_j taken by the same professor should be scheduled on an overlapping time t_n

2.1.2 Two lectures that are taken by the same batch cannot be scheduled at the same time:

$$\bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (C_{im} \implies \bigwedge_{\substack{n=0 \\ t_n \in T \\ t_n = t_m}}^{N_t} (\neg C_{jn})) \quad \text{if } c_i, c_j \text{ are taken by same batch} \tag{2}$$

If one batch is taking the course c_i at some time t_m, then no other course c_j taken by the same batch should be scheduled on an overlapping time t_n
Note: This also takes care of the fact that if two batches, say a and b are taking the same course, then no other course either a or b takes will be scheduled with the common course they take, because for batch a, no other course that a takes will be scheduled at any overlapping time to the common course, and the same logic applies for b.

2.2 Course - Lecture number - Time Constraints (CLT)

These propositions take care of scheduling required number of lectures in a week

2.2.1 No lectures of a course overlap

$$\bigwedge_{i=1}^{N_c} \bigwedge_{j=1}^{n_i} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (P_{ijm} \implies \bigwedge_{\substack{n=0 \\ t_n \in T \\ t_n = t_m}}^{N_t} \bigwedge_{\substack{j'=1 \\ j \neq j'}}^{n_i} (\neg P_{ij'n})) \tag{3}$$

if j^{th} lecture of course c_i is held at time t_m then no other lecture of that course can be held at any overlapping time t_n

2.2.2 The n^{th} lecture of a course is scheduled atleast once every week

$$\bigwedge_{i=1}^{N_c} \bigwedge_{j=1}^{n_i} \bigvee_{\substack{m=0 \\ t_m \in T}}^{N_t} (P_{ijm}) \tag{4}$$

2.2.3 The n^{th} lecture of a course is scheduled atmost once every week

$$\bigwedge_{i=1}^{N_c} \bigwedge_{j=1}^{n_i} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (P_{ijm} \implies \bigwedge_{\substack{n=0 \\ t_n \in T \\ n \neq m}}^{N_t} (\neg P_{ij'n})) \tag{5}$$

The above two constraints make sure that each lecture of a course is held only once every week

2.3 Course - Room -Time constraints (CRT)

2.3.1 No room has two lectures on overlapping times

$$\bigwedge_{i=1}^{N_c} \bigwedge_{r=1}^{N_r} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (Q_{irm} \implies \bigwedge_{\substack{n=0 \\ t_n \in T \\ t_n = t_m}}^{N_t} \bigwedge_{\substack{i'=1 \\ i' \neq i}}^{N_c} (\neg Q_{i'rn})) \tag{6}$$

if course c_i is held in the r^{th} room at time t_m then no course $c_{i'}$ can be held at any overlapping time t_n in the same room

2.3.2 A course should only be assigned one room

$$\bigwedge_{i=1}^{N_c} \bigwedge_{r=1}^{N_r} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (Q_{irm} \implies \bigwedge_{\substack{r'=0 \\ r' \neq r}}^{N_r} (\neg Q_{ir'n})) \tag{7}$$

if course c_i is held in the r^{th} room at time t_m then c_i can't be held at any other room at the same time

2.4 Proposition consistency constraints

2.4.1 Course-room-time and course-time constraints (CRT and CT)

1.

$$\bigwedge_{i=1}^{N_c} \bigwedge_{r=1}^{N_r} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (Q_{irm} \implies C_{im}) \tag{8}$$

A course assigned a room at time t_m trivially implies that a lecture should be scheduled at the same time
2.

$$\bigwedge_{i=1}^{N_c} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (C_{im} \implies \bigvee_{r=1}^{N_r} (Q_{irm})) \tag{9}$$

If a course c_i is scheduled at some time t_m then it should be assigned atleast one room

2.4.2 Course-lecture number-time and course-time constraints constraints (CLT and CT)

1.

$$\bigwedge_{i=1}^{N_c} \bigwedge_{j=1}^{n_i} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (P_{irm} \implies C_{im}) \tag{10}$$

If some j_{th} lecture of the course is scheduled at some time t_m trivially implies that a lecture is scheduled at the same time

2.

$$\bigwedge_{i=1}^{N_c} \bigwedge_{\substack{m=0 \\ t_m \in T}}^{N_t} (C_{im} \implies \bigvee_{j=1}^{n_i} (P_{ijm})) \tag{11}$$

If a course c_i is scheduled at some time t_m then some j^{th} lecture of the course should be scheduled at that time

3 Python Program Info

3.1 Steps

1. First the input file is parsed and information is stored in some lookup tables
2. Time is discretized to 30 minute intervals, so a course can start or end only at (institute start time) + $x * 30$ minutes for some natural number x e.g. if institute start time is 8:30am, then courses only start or end at 8:30, 9:00, 9:30 etc.
3. All possible time slots in the given institute time are generated
4. All the propositions mentioned above are generated and stored in lookup tables
5. All the constraints mentioned above are generated and stored
6. These constraints are passed to Z3 SAT solver to check if a timetable with the given constraints is possible
7. if it is, then the timetable is printed in the console