GRP 27: Curriculum Satisfiability

Luke Gannon (22LFG1)

Kareem Yakubu (21KOY)

Will Wang (21WZW1)

Stuart Bootland (22SMB4)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

November 3, 2023

Abstract

This project focuses on the complex task of academic scheduling, considering multiple factors such as professors, courses, classrooms, and academic programs. The goal is to create a schedule that satisfies a variety of constraints, including professor qualifications, course prerequisites, and program requirements.

The scheduling process is modeled using propositional logic, with each aspect of the schedule represented as a proposition. These propositions are then combined into a logical theory, which is solved using a SAT solver. The solution to this theory represents a valid schedule.

The project also explores the impact of different constraints on the scheduling process. For example, it investigates how the qualifications of professors affect the courses they can teach, and how the prerequisites of a course determine when it can be scheduled. The project also considers the requirements of different academic programs, ensuring that all required courses are scheduled in the correct terms.

By exploring these aspects of academic scheduling, this project aims to provide insights into the challenges and complexities of this task. It also demonstrates how propositional logic and SAT solvers can be used to solve complex scheduling problems.

Propositions

In this section, we provide a list of propositions used in the model and their corresponding English interpretations.

1. **ProfessorAssigned (Term, Professor, Course, Day, Time)**: This proposition represents a professor being assigned to teach a course at a specific term, day, and time.
   - Example: "Professor: 'Dr. Jane Smith' is assigned to Course: 'CS101' at Time: '10:00' on Day: 'Monday' in Term: 'Spring 2022'."
2. **ProfessorQualified (Professor, Course)**: This proposition represents a professor being qualified to teach a course.
   - Example: "Professor: 'Dr. Jane Smith' is qualified to teach Course: 'CS101'."
3. **CourseAssigned (Course, Room, Term, Day, Time)**: This proposition represents a course being assigned to a room at a specific term, day, and time.
   - Example: "Course: 'CS101' is assigned to Room: '101' at Time: '10:00' on Day: 'Monday' in Term: 'Spring 2022'."
4. **CoursePrerequisite (Course, Prerequisite)**: This proposition represents a course having a prerequisite.

- Example: "Course: 'CS102' has Prerequisite: 'CS101'."

5. **ProgramReqCourse (Course, Program, Year)**: This proposition represents a course being required for a program in a specific year.
   - Example: "Course: 'CS101' is required for Program: 'Computer Science' in Year: '1'."
6. **ProgramSharesPreReq (Program1, Program2, Course)**: This proposition represents two programs sharing a prerequisite course.
   - Example: "Program: 'Computer Science' and Program: 'Software Engineering' share Prerequisite: 'CS101'."
7. **ProgramCanComplete (Program, Term)**: This proposition represents a program being able to be completed in a specific term.
   - Example: "Program: 'Computer Science' can be completed in Term: 'Spring 2022'."
8. **ClassroomAssigned (Room, Course, Term, Day, Time)**: This proposition represents a classroom being assigned to a course at a specific term, day, and time.
   - Example: "Room: '101' is assigned to Course: 'CS101' at Time: '10:00' on Day: 'Monday' in Term: 'Spring 2022'."

Constraints

List of constraint types used in the model and their (English) interpretation.

1. **A professor cannot be assigned to teach two courses at the same time.**
   Constraint $((\forall c_1 \forall c_2 \forall t((C_A(c_1,t) \wedge \neg C_A(c_2,t)) \vee (\neg C_A(c_1,t) \wedge C_A(c_2,t))))))$
2. **Professors can only teach courses for which they're qualified.**
   Constraint $(\forall p \forall x(\neg P_Q(p,x) \implies \neg P_A(p,x)))$
3. **Prerequisites are scheduled in a term before the course that requires them.**
   Constraint $(\forall x \forall y \forall t((CP(x,y) \wedge CA((x,t)) \implies CA(y,t+1)))$
4. **There are at least 2 lectures per course.**
   Constraint $(\forall c L(c) \leq 2)$
5. **Course may only be scheduled within year they are required.**
   Constraint $(\forall p \forall c \forall y_1(P_{RC}(c,p,y_1) \implies \neg \exists y_2(C_A(c,y_2)) \wedge (y_1 \neq y_2)))$
6. **A Program can only be "completed" during a Term if all the required Courses can be scheduled during that term.**
   Constraint $(\forall p \forall c \forall t(C_A(c,t) \wedge P_{RC}(c,p,t)) \implies P_{cc}(p,t)())$
7. **The proposition ProgramCanComplete must be true.**
   Constraint $(\forall p \forall t \quad P_{cc}(p,t))$

8. **A Room can only be assigned to a Course if the room is free at the time the course is offered.**
Constraint($\forall c_1.\forall c_2.\forall r.\forall te.\forall ti.\forall d.((C(r, c_1, te, d, ti) \wedge (C(r, c_2, te, d, ti)) \implies c_1 = c_2$*))*

Model Exploration

Throughout the project, we explored the model in various ways, including:

1. Initial exploration of individual student enrollment and course scheduling.
2. Transition to the program scheduling perspective as suggested in the feedback.
3. Iterative refinement of course scheduling and prerequisite handling.
4. Investigation of shared courses and their impact on scheduling constraints.
5. Examination of different course arrangement scenarios and their effects on scheduling flexibility.
6. Analysis of the influence of adding or removing prerequisites on scheduling constraints.

First-Order Extension

If we were to extend our model to a predicate logic setting, we would represent propositions and constraints in a more formalized way. For example:

Jape Proofs

1:	$(R\wedge\neg T)\vee(\neg R\wedge\neg T)$	premise
2:	$(R\wedge T)\vee(\neg R\wedge T)$	assumption
3:	$R\wedge T$	assumption
4:	T	$\wedge$ elim 3
5:	$R\wedge\neg T$	assumption
6:	$\neg T$	$\wedge$ elim 5
7:	$\perp$	$\neg$ elim 4,6
8:	$\neg R\wedge\neg T$	assumption
9:	$\neg T$	$\wedge$ elim 8
10:	$\perp$	$\neg$ elim 4,9
11:	$\perp$	$\vee$ elim 1,5–7,8–10
12:	$\neg R\wedge T$	assumption
13:	T	$\wedge$ elim 12
14:	$R\wedge\neg T$	assumption
15:	$\neg T$	$\wedge$ elim 14
16:	$\perp$	$\neg$ elim 13,15
17:	$\neg R\wedge\neg T$	assumption
18:	$\neg T$	$\wedge$ elim 17
19:	$\perp$	$\neg$ elim 13,18
20:	$\perp$	$\vee$ elim 1,14–16,17–19
21:	$\perp$	$\vee$ elim 2,3–11,12–20
22:	$\neg((R\wedge T)\vee(\neg R\wedge T))$	$\neg$ intro 2–21

R=Prerequisite course, T=Course with prerequisite R
Proof 1: If one course is a prerequisite of another, then the prerequisite must be taken before.

1:	$P,\ P\rightarrow T,\ R$	premises
2:	T	$\rightarrow$ elim 1.2,1.1
3:	$T\wedge R$	$\wedge$ intro 2,1.3

P=Program, T=Required Course, R=Elective Course Proof 2: A required course and an elective course can be taken during the same term

1:	P, P→R, P→T	premises
2:	(¬R∧T)∨(R∧¬T)	premise
3:	T	→ elim 1.3,1.1
4:	R	→ elim 1.2,1.1
5:	¬R∧T	assumption
6:	¬R	∧ elim 5
7:	⊥	¬ elim 4,6
8:	¬(R∧T)	contra (constructive) 7
9:	R∧¬T	assumption
10:	¬T	∧ elim 9
11:	⊥	¬ elim 3,10
12:	¬(R∧T)	contra (constructive) 11
13:	¬(R∧T)	∨ elim 2,5–8,9–12

P=Program, R=course1 scheduled for a time, T=course2 scheduled for the same time. Proof 3: Two required courses cannot be scheduled in the same time frame.

Notes

Need help with first order extension as allot of logic already in predicate format.