CISC 499 Undergraduate Project Midterm Report

Feb 16, 2024

For: Prof./Dr. Chen,

By: Jinyang Chen, Xindong Liu, Delin Yang

Course-Curriculum Map

Table of Contents

1. Abstract

After half a semester of study, discussion, and research, our group has gained a preliminary understanding of the concept of the Course-Curriculum Map. Following the advice of our mentor, Prof./Dr. Chen, we have initially designed a model based on the various plans within the Computing School, as well as the relationships between courses. In the upcoming period, we will continue to optimize and iterate the course model and use it as the basis to build an interactive visualization program. Our aim is to provide clear and direct guidance and assistance for students in the Computing School in their planning and course selection.

2. Target

In the Course-Curriculum Map project, our objective is to collect the courses required for various plans within the Computing School as datasets. We will analyze and summarize the relationships between courses and use these relationships and datasets as the basis for modeling. Ultimately, we will need to design a visualization program for our model to showcase the project's outcomes.

Overall, our project consists of five parts: data collection, data analysis, model establishment, visualization program, and final reporting. Currently, our group is in the third stage of building the model.

3. Design of model

3.1 Design concept

Before discussing the design schemes for the model, I'd like to prioritize explaining our design concept. After collecting the data, our team unanimously agreed that the number of courses required for various plans within the Computing School, as a dataset, is very small, less than 500 samples. Therefore, we believe that when designing the model, there's no need to choose overly versatile and complex model samples; instead, we should opt for simple, direct, and efficient data structures as the foundation of our design model. This is not only because a too-small dataset would blur the distinction between simple and complex models in practical operation, but also because simple data structures can significantly enhance efficiency when we optimize and iterate the model in the future. Hence, our design concept is simple, direct, and efficient.

3.2 Characteristics of plans

Before selecting a data structure to build the model, we must delve into the data analysis, study the correlations between various courses, and understand and summarize the relationships among them.

At the same time, since we are designing a model to replicate the courses required for various plans within the Computing School, the relationships between courses must align with the plans and reflect the characteristics of the plans themselves.

- We believe that the plans themselves are mutually exclusive, meaning that out of the nine major plans, each student can only choose one.
- Plans are phased, meaning each plan is divided into several parts with clear stage correlations between many of them. For example, completing the 100-level course section is a prerequisite for advancing to the 200-level course section.
- Each stage of the plan also exhibits diversity, as many plans require students to choose a sufficient number of courses from elective courses, with fewer restrictions on specific choices.

It is these characteristics that allow us to deduce the connections between various courses for the purpose of designing the model.

3.3 Relationships between courses

Based on the characteristics of the plans themselves, we categorize the relationships between courses into three types to construct a model for displaying and selecting plans. We refer to these three types of relationships as: inheritance, mutual exclusion, and selection.

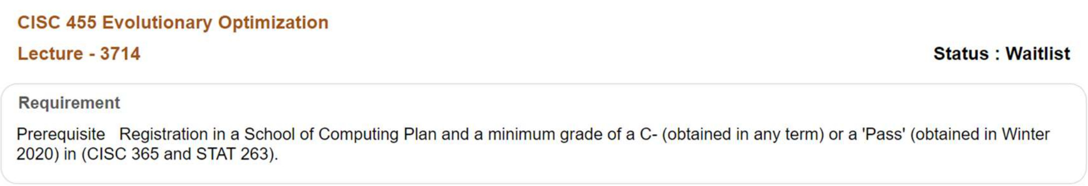
CISC 455 Evolutionary Optimization
Lecture - 3714
Status : Waitlist
Requirement
Prerequisite Registration in a School of Computing Plan and a minimum grade of a C- (obtained in any term) or a 'Pass' (obtained in Winter 2020) in (CISC 365 and STAT 263).

Figure1. Pre-course of CISC455

- The first type of relationship, inheritance, is derived from the prerequisite nature of courses themselves and the phased characteristics of plans. Specifically, it is manifested as a sequential connection between courses based on prerequisites and follow-up courses. Follow-up courses must be selected after completing prerequisite courses. For example, the course CISC121 serves as a prerequisite for the course CISC124; one cannot enroll in CISC124 without completing CISC121.

3. Supporting

A. Complete 3.00 units from the following:		**3.00**
CISC 102	Discrete Mathematics for Computing I	
MATH 110	Linear Algebra	

Figure2. Mutual exclusion between CISC102 and Math110

- The second type of relationship, mutual exclusion, is based on the exclusivity and diversity characteristics among plans and within certain stages of plans. Not only do plans exhibit mutual exclusivity among each other, but also within certain stages of a plan where the required number of courses is one, while the available choices are more than one. We consider this also as an expression of mutual exclusion. Therefore, the mutual exclusion between courses is specifically manifested as, within a set of courses, selecting any one course renders the selection of the remaining courses in that stage impossible. For example, STAT263 and STAT268 exhibit mutual exclusivity in many plans.

c. Complete 6.00 units from the following:	6.00
CISC at the 300-level or above	
CISC_Subs at the 300-level or above	
SOFT at the 300-level or above	

Figure3. Choice of part C

- The third type of relationship, selection, is derived from the diversity characteristics within certain stages of plans. It is specifically characterized by the presence of multiple distinct course choices available to students, meeting the objectives of the stage of study. In this scenario, there exists a selection relationship among these alternative courses.

These three types of relationships are one of the key foundations for designing our model.

3.4 Implementation of Course Relationships

Based on the aforementioned categorization of the three types of relationships between courses, our group of three members, after thorough discussion and consideration, decided to use a tree map as the foundational data structure for constructing the model. We chose a tree structure because it inherently provides a mechanism for child nodes, making it easy to implement the mutual exclusion relationship between courses.

For the selection relationship between courses, we believe that in the case of a relatively small dataset, we can use an exhaustive method to enumerate all possible choices that elective courses can offer and list them as child nodes. This way, we can utilize mutual exclusion in reverse to implement the selection relationship between courses.

Finally, we considered how to implement the inheritance relationship between courses. Initially, we thought about utilizing the features of child-parent nodes to implement the inheritance relationship. However, after discussion, we realized that within a plan, the sequence of stages is not necessarily clear-cut due to the diversity of course selections. In fact, it's possible that the sequence between stages might even be reversed based on the choices of different students. Therefore, we decided to design an additional new data structure to facilitate the implementation of the inheritance relationship between courses.

3.5 Overview of Model Design

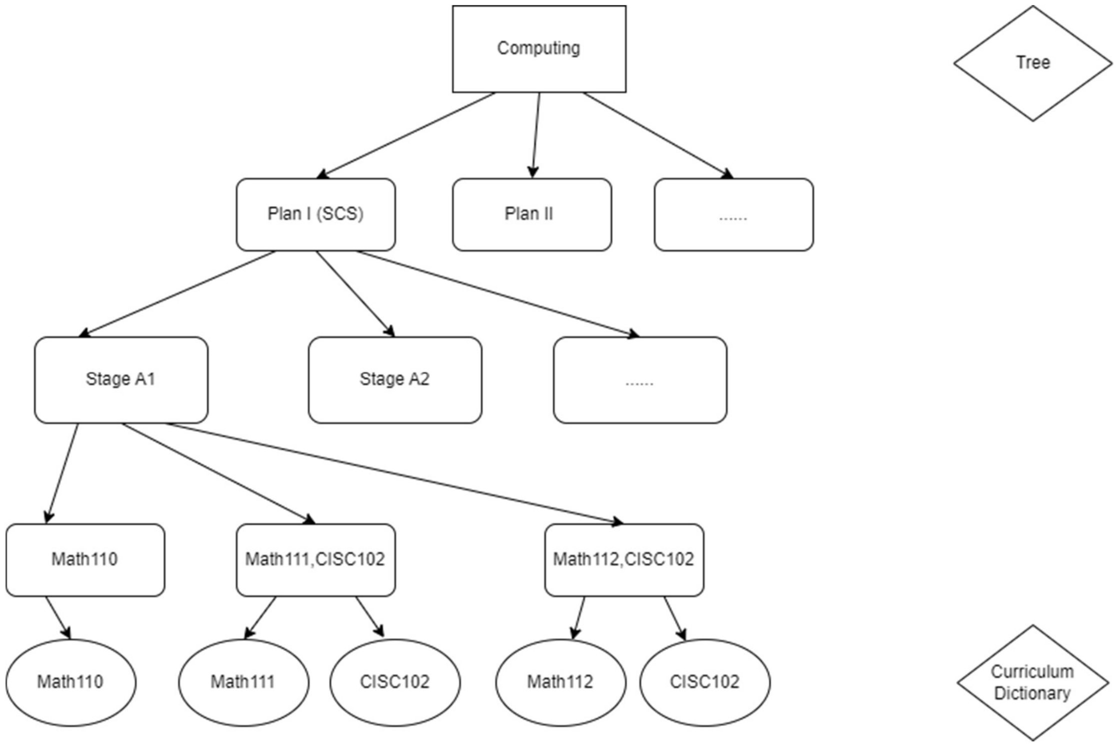

Figure4. Graph of Tree model

The model we have designed consists of two main parts. The first part is the tree map described in Section 3.4. We envision the root of the tree to represent the Computing School, with the second-level child nodes representing the nine main plans. The third-level child nodes represent the stages divided within each plan, the fourth-level child nodes represent the various ways of selecting courses that fulfill the learning objectives of each stage, and the fifth-level leaf nodes represent the specific courses within each course selection method.

Within this tree map, apart from the leaf nodes, we anticipate storing strings for all other nodes. The leaf nodes will store another type of data structure that we have designed, which constitutes the second part of the model.

{} course.json ×

D: > Wechat_File > WeChat Files > wxid_truvil1aj5xe22 > FileStorage > File > 2024-02 > {} course.json > [ ] courses > {} 0 > pre

```
{
  "courses": [
    {
      "name": "WRIT125",
      "url": "https://www.queensu.ca/academic-calendar/search/?P=WRIT%20125",
      "pre": " "
    },
```

Figure5. Implementation of Curriculum Dictionary

The second data structure we have designed, which we currently refer to as the "Course Dictionary," is akin to a dictionary in both structure and functionality. It uses the course name as the key and stores all relevant information about the course within this data structure. For instance, the most crucial prerequisite courses are stored within this Course Dictionary. The content stored in the Course Dictionary can vary greatly; it could be a list or a URL. Currently, our Course Dictionary stores a URL for a course introduction website and a list of prerequisite courses. However, in future iterations and optimizations, we may add various types of information to the Course Dictionary to facilitate the addition of other functionalities to the model.

3.6 Functionality Implementation

Our model primarily features three functionalities:

1. Displaying the current available paths, including the selection of plans and the specific course selections within each stage of the plan.
2. Assessing whether the currently selected courses meet the prerequisite

requirements (i.e., whether they satisfy the inheritance relationships between courses). If they meet the requirements, the current courses will be chosen; otherwise, alternative courses must be selected.

3. Providing the selected course paths.

We plan to implement these functionalities as follows:

1. Displaying available paths: This can be achieved by traversing the current child nodes and returning a list. Since the dataset is very small, even traversing will not significantly impact the program's runtime speed. Therefore, we won't consider complexity issues for this and subsequent functionalities.

2. Checking if courses meet prerequisites: Firstly, we'll use a list to store the selected courses. Then, we'll design a function to find the prerequisite course list from the Course Dictionary within the leaf nodes after selecting a specific course. We'll use a double loop to traverse all prerequisite courses. If all prerequisite courses are found in the selected list, the course is considered to meet the requirements and will be added to the selected list. Otherwise, an error message will be returned, prompting the user to select other courses.

3. Displaying selected paths: This will be achieved by simply returning the list of selected courses.

3.7 Choice of Tools

When building the data structure, after discussion, we unanimously decided to use PyCharm as our primary tool. PyCharm, as a well-known Python IDE, is familiar to all members of our group, and we have experience learning and using PyCharm. Additionally, when encountering difficulties, there are plenty of related learning materials available to help clarify any issues.

4. Visualization

After completing the model, we plan to develop an interactive and simple program to visualize it. We anticipate that this program will be written in Python, as it will be better suited to leverage the model created in PyCharm.

Our program will visualize all three main functionalities of the model. We envision the interface to be a tree diagram or a diagram depicting the relationships between plans as outlined previously. Additionally, users will be able to interact with our designed model through this program, selecting their own course paths, thus allowing the model to truly serve as a guide for students.

We also plan to add many additional features to the program, such as providing access to introduction websites for each course when a student selects any course. This will make our Course-Curriculum Map project more user-friendly and diverse.

5. Summary

In summary, through our collaborative efforts over the past half semester and with guidance and assistance from our mentor, we have gained a clear understanding and plan for the Course-Curriculum Map project. We are steadily progressing and striving to achieve our most important goals. We believe that in the next half semester, we can complete our project and deliver an excellent presentation.