GRP 12: Toronto Daytrip Planner

Jasper N (20285349)

Yu Xuan L (20288359)

Darwin C (20267164)

Sijing Z (20243635)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

October 29, 2022

Abstract

Using data set found on transitfeeds.com[1], our project evaluates and provides the most optimal ways one can get from one point to another within Toronto, using just the public transport system. Our project takes into consideration many different variables including transfers, budget, time constraints, etc. As the user you will input your starting location, desired ending location, current time, desired arrival time, travel budget, age, extra visiting locations (eg. Toronto Zoo), and other miscellaneous options. Using this information, our program will show you a list of trains, buses, and streetcars you need to take at which times in order to reach your location within your budget and time constraints.

Propositions

List of the propositions used in the model, and their (English) interpretation.

- **Budget/Price**

  1. **Adult** = true if the user does not fall into the child or seniors category.
  2. **Youth** = true if the user falls into the youth category (13-19).
  3. **Senior** = true if the user falls into the senior category (65+).
  4. **PRESTO** = true if the user has an active PRESTO account with sufficient balance to complete the trip with.
  5. **PRESTO_Adult** = true if the user holds a PRESTO card and is an adult.
  6. **PRESTO_Youth** = true if the user holds a PRESTO card and is a youth.
  7. **PRESTO_Senior** = true if the user holds a PRESTO card and is a senior.
  8. **PRESTO_DayPass** = true if the user is a PRESTO card holder and an adult or child or senior, with his/her amount of stops throughout the trip costing more than a single Day Pass (13.50$).
  9. **Surpass_DayPass_Adult** = true if the user is an adult and their planned trip will surpass PRESTO Day Pass pricing if they were to take the normal PRESTO price.
  10. **Surpass_DayPass_Others** = true if the user is not an adult and their planned trip will surpass PRESTO Day Pass pricing if they were to take the normal PRESTO price.

- **Time**

  1. **Within_Time_Constraint** = true if the valid trips found are within the time limit specified by the user.
  2. **Rush_Hour** = true if the trip time is within the periods 7am - 10am and 4pm - 7pm.

- **Additional_Stop**$_{(a,b)}$ = true if the user defines intermediate stops that are feasible on top of going from destination a to b.
- **Valid_Trip**$_{(a,b)}$ = true if all the user defined requirements are met and a route is found from destination a to b.

Constraints

In order for the program to find the optimal trip, our propositions must all meet requirements for a valid trip plan. These requirements will be made possible using constraints.

- **Additional Stops Constraint**:
  If the additional stops that the user wants to take interferes with other trip variables and is not feasible (either within time constraint or the budget limit), then Additional_Stops = false. Logically, this can be stated as...

  $$\text{Additional\_Stops } (\neg\text{Within\_Time\_Constraint} \lor \neg \text{ Within\_Budget}) \rightarrow \neg\text{Valid\_Trip}$$

- **Budget Constraints**:
  - **Within_Budget** = true if there exists a route in which its price is below user's budget limit.
  - **Adult** = true if the user does not fall into the child or seniors category.
  - **Youth** = true if the user falls into the youth category (13-19).
  - **Senior** = true if the user falls into the senior category (65+).
  - **PRESTO** = true if the user has an active PRESTO account with sufficient balance to complete the trip with.
  - **PRESTO_Adult** = true if the user holds a PRESTO card and is an adult.
  - **PRESTO_Youth** = true if the user holds a PRESTO card and is a youth.
  - **PRESTO_Senior** = true if the user holds a PRESTO card and is a senior.
  - **PRESTO_DayPass** = true if the user is a PRESTO card holder and an adult or child or senior, with his/her amount of stops throughout the trip costing more than a single Day Pass (13.50$).
  - **Surpass_DayPass_Adult** = true if the user is an adult and their planned trip will surpass PRESTO Day Pass pricing if they were to take the normal PRESTO price.
  - **Surpass_DayPass_Others** = true if the user is not an adult and their planned trip will surpass PRESTO Day Pass pricing if they were to take the normal PRESTO price.

TTC Pricing		
Fare Type	Adult	Senior(65+) Youth (13-19)
Single Fare	$ 3.25	$2.30
PRESTO	$3.20	$2.25
PRESTO Day Pass	$13.50	$13.50

If the user doesn't fall into the youth or senior category, then the user must be an adult:

$$\text{Adult} \rightarrow \neg\text{Youth} \wedge \neg\text{Senior}$$

If the user falls into the youth category, then the user is not an adult or senior:

$$\text{Youth} \rightarrow \neg\text{Adult} \wedge \neg\text{Senior}$$

If the user has an active PRESTO card with sufficient funds for the trip and also an adult, then the trip price will be calculated using PRESTO pricing:

$$\text{PRESTO} \wedge \text{Adult} \rightarrow \text{PRESTO\_Adult}$$

If the user has an active PRESTO card with sufficient funds for the trip and also a youth, then the trip price will be calculated using PRESTO pricing:

$$\text{PRESTO} \wedge \text{Youth} \rightarrow \text{PRESTO\_Youth}$$

If the user has an active PRESTO card with sufficient funds for the trip and also a senior, then the trip price will be calculated using PRESTO pricing:

$$\text{PRESTO} \wedge \text{Senior} \rightarrow \text{PRESTO\_Senior}$$

If the user is a PRESTO card holder and an adult, with his/her amount of stops throughout the trip costing more than a single Day Pass (13.50$), then the price will be just PRESTO DayPass. (PRESTO_DayPass will be true if the user needs to take more than 5 transfers over a 10 hour period):

$$\text{PRESTO} \wedge \text{Adult} \wedge \text{Surpass\_DayPass\_Adult} \rightarrow \text{PRESTO\_DayPass}$$

If the user is a PRESTO card holder and a youth or senior, with his/her amount of stops throughout the trip costing more than a single Day Pass (13.50$), then the price will be just PRESTO DayPass. (PRESTO_DayPass will be true if the user needs to take more than 6 transfers over a 12 hour period

$$\text{PRESTO} \wedge (\text{Youth} \vee \text{Senior}) \wedge \text{Surpass\_DayPass\_Adult} \rightarrow \text{PRESTO\_DayPass}$$

- **Time Constraints**:
  - **Within_Time_Constraint** = true if the valid trips found are within the time limit specified by the user.
  - **Rush_Hour** = true if the trip time is within the periods 7am - 10am and 4pm - 7pm.
- **Variable**:
  - **On_Time** = this is the variable that will be used to calculate the total trip time if the trip is during rush hour.
  - **Rush_Hour_Time** = this is the variable that will be used to calculate the total trip time if the trip is during rush hour.

  If a user travels during rush hours, use a function to calculate (still developing) the amount of extra time it would take to reach the destination, if it surpasses the time constraint defined by the user, Within_Time_Constraint = false.
  With the introduction of the rush hour variable, we need to examine the amount of time that the rush hour will add to the original trip time.
  This will give us 2 cases...

  $$\text{Rush\_Hour} \wedge \neg \text{Within\_Time\_Constraint} \rightarrow \neg \text{Valid\_Trip}$$
  $$\text{Rush\_Hour} \wedge \text{Within\_Time\_Constraint} \rightarrow \text{Valid\_Trip}$$

Model Exploration

- First create some databases:
  - Database of all the trips that happen throughout the day.
  - Database of all the different routes in the city.
  - Database of all the stops and stations in the city
- Next we need to create an algorithm to find a path from stop A to stop B using the database. Check if any transfers are needed and make sure the trips are within the given time constraints. We are currently at this step.
- Next we add all the above mentioned budget constraints to the algorithm and choose the cheapest option for the traveler.
- Finally we make any final adjustments and possibly add a user interface.

First-Order Extension

Predicate logic: For each proposition, we can represent it with a predicate that holds the current time p and the arrival time q, The universal discourse A would be the different possible stops and locations in Toronto. An example of which would be the following:

EAST - 10 VAN HORNE towards VICTORIA PARK STATION$_{(p,q)}$

WEST - 10 VAN HORNE towards DON MILLS STATION$_{(p,q)}$

SOUTH - 102 MARKHAM RD towards WARDEN STATION$_{(p,q)}$

The constraints above that use these prepositions can be replaced with their predicate counterparts using all possible times for current time and all possible times for the desired arrival time. Here are some examples:

- B = Budget Limit
- R = Rush Hour
- L = Time Limit
- G = Successful Trip

1. ∀p ∃q (T(p,q) ∧ B):
   For all current time p there exists an arrival time q where you can go on the trip "EAST - 10 VAN HORNE towards VICTORIA PARK" within the budget limit.
2. ∀p ¬∃q (T(p,q) ∧ ¬B):
   For all current time p there does not exist an arrival time q where you can go on the trip without exceeding the budget limit.
3. ∀p ∃q ((T(p,q) ∧ G) → (B ∧ L)):
   For all current time p and all arrival time q, if you go on a trip and it is a successful trip, then the trip will be within the Budget Limit and the Time Limit.
4. ∀p ∃q ((T(p,q) ∧ (¬B|¬L)) → ¬G):
   For all current time p and all arrival time q, if you go on a trip and you are not within Budget Limit or Time Limit, then you will not have a successful trip.

Requested Feedback & Questions

1. What are some other constraints types (or specific constraints) that may be added which could provide help in developing the logical propositions later on and improve our algorithm? To be specific, the constraints regarding time are not apparent, and we wish to expand upon what we have as the foundation but are having trouble doing so.
2. What are some propositions regarding the TTC that we may have missed out on or could be added to further improve the practicality of our algorithm? Any tips on using the TTC are also welcome to help us better reflect real world scenarios.
3. Is there anything specifically regarding the documentations, whether it be from over complication or the lack of explanation, that makes it difficult to understand the logic and intentions behind our project/algorithm.