

Rapport d'Implémentation pour le Projet de Gestion de Trajets en Prolog

Introduction

Le projet a pour objectif de développer un programme en Prolog capable d'aider un usager d'un réseau de transport à identifier le trajet lui permettant de se rendre d'une station à une autre, en respectant certaines conditions telles que l'horaire de départ, l'horaire d'arrivée, minimiser le nombre de correspondances, ou la durée du voyage.

Représentation du Réseau de Transport

Pour représenter le réseau de transport, nous avons utilisé un prédicat **ligne/5** avec les caractéristiques suivantes :

- **Nom** : le nom de la ligne (numéro ou lettre majuscule).
- **Type** : le type de moyen de transport (métro, tram, ou bus).
- **LArret**: une liste de paires [**Arrêt**, **Temps**] représentant les arrêts desservis par la ligne et le temps nécessaire pour parcourir la distance entre eux.
- **LTrajetAller**: les horaires de départ, l'intervalle entre les départs et l'horaire du dernier départ de la ligne.
- **LTrajetRetour**: les horaires de départ, l'intervalle entre les départs et l'horaire du dernier départ de la ligne en sens inverse.

Exercice 01 : Manipulation des Horaires

Les prédicats **addh/3** et **affiche/1** ont été développés pour faciliter la manipulation des horaires. Le prédicat **addh/3** est conçu pour manipuler les horaires en ajoutant un certain nombre de minutes à un horaire donné. Son objectif est de faciliter le calcul des horaires lors de la planification d'itinéraires en tenant compte des préférences de l'utilisateur., Le prédicat **affiche/1** est responsable de l'affichage lisible par l'utilisateur des horaires. Son objectif est de présenter clairement les horaires dans un format compréhensible.

Exercice 02 : Recherche des Itinéraires

- Le prédicat **lig/3** vérifie si une ligne de transport passe d'un premier arrêt à un second arrêt .
- **Arguments** :
- **Arret1, Arret2** : Les deux arrêts entre lesquels on cherche une ligne de transport.
- **Ligne** : La variable qui sera unifiée avec le numéro ou le nom de la ligne trouvée.
- **Fonctionnement** :
- Récupère les informations de la ligne depuis la base de connaissances.
- Vérifie que les arrêts spécifiés (**Arret1** et **Arret2**) sont présents dans la liste des arrêts desservis par la ligne.
- **connectes/3** : Prédicat qui vérifie si deux arrêts sont connectés dans la liste des arrêts. Il utilise **append/3** pour diviser la liste des arrêts en deux parties, avant et après l'arrêt 1. Ensuite, il divise la partie après l'arrêt 1 en deux parties, la première étant l'arrêt 2 et la seconde étant le reste. Si cette division est possible, alors les deux arrêts sont connectés.

- Les prédicats **ligtot/4** et **ligtard/4** sont utilisés pour trouver une ligne entre deux arrêts avec un départ le plus tôt possible après un certain horaire, et de trouver une ligne entre deux arrêts avec une arrivée le plus tard possible
- Ces prédicats sont conçus pour trouver des itinéraires entre deux arrêts en prenant en compte l'horaire de départ (pour **ligtot/4**) et l'horaire d'arrivée (pour **ligtard/4**).
- **ligtot/4** : Prédicat qui recherche un itinéraire total entre deux arrêts sur une ligne donnée en optimisant le départ le plus tôt possible. Il utilise **lig/3** pour vérifier si les deux arrêts sont connectés sur la ligne, puis vérifie si l'horaire de départ spécifié est possible en utilisant le prédicat **horaireDepartPlusTard/3**. Il ajoute également la possibilité de spécifier un horaire de départ tardif.
- **horaireDepartPlusTard/3** : Prédicat qui calcule l'horaire de départ le plus tard possible après un certain horaire. Il prend une liste d'arrêts (**Arrets**), un horaire de départ spécifié (**Horaire**), et retourne l'horaire de départ le plus tard possible (**PlusTard**). Il utilise la récursion pour parcourir la liste des arrêts, en ajoutant 1 minute à chaque itération.
- **addMinutes/3** : Prédicat pour ajouter un nombre spécifié de minutes à un horaire. Il prend un horaire initial (**[H, M]**), un nombre de minutes à ajouter (**Minutes**), et retourne le nouvel horaire (**[H2, M2]**).
- **ligtard/4** : Prédicat qui recherche un itinéraire total entre deux arrêts sur une ligne donnée en optimisant l'arrivée la plus tardive possible. Il utilise **lig/3** pour vérifier si les deux arrêts sont connectés sur la ligne, puis vérifie si l'horaire d'arrivée spécifié est possible en utilisant le prédicat **horaireArriveePlusTard/3**.
- **tempsMinimal/4** : Prédicat qui trouve le temps minimal entre deux arrêts dans une liste d'arrêts. Il prend la liste des arrêts (**Arrets**), les deux arrêts spécifiés (**Arret1** et **Arret2**), et retourne le temps minimal entre ces deux arrêts (**Temps**).
- **horaireArriveePlusTard/3** : Prédicat qui calcule l'horaire d'arrivée le plus tard possible avant un certain horaire. Il prend une liste d'arrêts (**Arrets**), un horaire d'arrivée spécifié (**HorairePlusTard**), et retourne l'horaire d'arrivée le plus tard possible (**PlusTard**). Il utilise la récursion pour parcourir la liste des arrêts, en ajoutant 1 minute à chaque itération.

Exercice 03 : Recherche d'Itinéraires Complets

Les prédicats **itinTot/4** et **itinTard/4** trouvent un itinéraire entre deux arrêts avec un départ le plus tôt possible après un certain horaire, et une arrivée le plus tard possible avant un certain horaire, respectivement.

- **itinTot/4** : Prédicat principal pour trouver un itinéraire total de **Arret1** à **Arret2** à partir d'un certain horaire (**Horaire**). Il utilise le prédicat **ligtot/5** pour récupérer le prochain arrêt sur un itinéraire en optimisant le départ le plus tôt possible. Ensuite, il appelle le prédicat récursif **itinRec/7** pour explorer les itinéraires possibles.
- **itinRec/7** : Prédicat récursif pour explorer les itinéraires possibles. Il prend l'arrêt actuel (**ArretActuel**), l'arrêt final (**ArretFinal**), la ligne de transport

(**Ligne**), le type de transport (**TypeTransport**), l'horaire de départ (**HoraireDepart**), l'horaire actuel (**Horaire**), et la liste des arrêts déjà parcourus (**Parcours**). Le prédicat utilise **ligtot/5** pour trouver le prochain arrêt (**ArretSuivant**) sur la même ligne à partir de l'arrêt actuel et à l'horaire spécifié. Il récurse ensuite avec l'arrêt suivant comme nouvel arrêt actuel.

- La condition **HoraireActuel = HoraireDepart** permet de spécifier une condition supplémentaire sur l'horaire actuel, que vous pouvez ajuster selon vos besoins.
- La récursion se termine lorsque l'arrêt actuel est égal à l'arrêt final, et la liste de parcours est alors un itinéraire complet.
- **ligtot/5** : Prédicat pour récupérer le prochain arrêt dans un itinéraire. Il prend l'arrêt de départ (**Arret1**), l'arrêt de destination (**Arret2**), la ligne de transport (**Ligne**), l'horaire de départ (**Horaire**), et le type de transport (**TypeTransport**). Ce prédicat utilise **lig/4** pour déterminer les connexions directes entre les arrêts, puis calcule l'horaire de départ pour le prochain arrêt en ajoutant 1 à l'horaire actuel.
- **itinTard/4** : Prédicat pour trouver un itinéraire total de **Arret1** à **Arret2** qui arrive le plus tard possible avant un certain horaire (**Horaire**). Il utilise le prédicat récursif **itinRecTard/5** pour explorer les itinéraires possibles en optimisant l'arrivée la plus tardive possible.
- **itinRecTard/5** : Prédicat récursif pour explorer les itinéraires possibles en arrivant le plus tard possible. Il est similaire à **itinRec/7** mais utilise le prédicat **ligtard/6** pour trouver le prochain arrêt en optimisant l'arrivée la plus tardive possible.
- **Cas de base** : Si l'arrêt actuel est égal à l'arrêt final, l'itinéraire est terminé, et la liste **Parcours** est construite.
- **Cas récursif** : Si ce n'est pas le cas, le prédicat utilise le prédicat **ligtard/4** pour trouver le prochain arrêt (**ArretSuivant**), l'horaire de départ pour atteindre cet arrêt (**HoraireDepart**), et s'assure que cet horaire de départ est postérieur ou égal à l'horaire d'arrivée souhaité (**HoraireArrivee**). Ensuite, il rappelle récursivement le prédicat avec le nouvel arrêt comme point de départ.

En résumé, ces prédicats travaillent ensemble pour trouver un itinéraire optimal en termes d'arrivée tardive, en explorant de manière récursive les différentes options d'arrêts et d'horaires de départ pour atteindre la destination.

Exercice 04 : Options pour les Itinéraires

Modification de l'exercice 02 :

-Le prédicat **lig/4** vérifie si une ligne de transport passe d'un premier arrêt à un second arrêt tout en respectant la préférence de l'utilisateur en matière de longueur du trajet.

- **Arguments** :
- **Arret1, Arret2** : Les deux arrêts entre lesquels on cherche une ligne de transport.
- **Ligne** : La variable qui sera unifiée avec le numéro ou le nom de la ligne trouvée.

- **TypeTransport** : Le type de moyen de transport que l'utilisateur recherche (metro, tram, bus).
- **LongueurPreference** : La longueur maximale du trajet que l'utilisateur préfère.
- **Fonctionnement** :
- Récupère les informations de la ligne depuis la base de connaissances.
- Vérifie que les arrêts spécifiés (**Arret1** et **Arret2**) sont présents dans la liste des arrêts desservis par la ligne.
- Vérifie que la longueur du trajet entre **Arret1** et **Arret2** est inférieure ou égale à la préférence de l'utilisateur.

-Le prédicat **ligtot/5** recherche une ligne entre deux arrêts avec un départ le plus tôt possible après une certaine heure, tout en respectant la préférence de l'utilisateur pour la longueur du trajet.

Les prédicats **lig/3**, **ligtot/4**, et **ligtard/4** ont été modifiés pour prendre en compte le type de transport. Ils permettent de vérifier si une ligne passe par deux arrêts, de trouver une ligne entre deux arrêts avec un départ le plus tôt possible après un certain horaire, et de trouver une ligne entre deux arrêts avec une arrivée le plus tard possible avant un certain horaire, respectivement.

Les prédicats existants ont été modifiés pour inclure des options telles que le choix du réseau (ferroviaire ou bus), la préférence par rapport à la longueur du trajet, et le nombre de correspondances. Ces options sont passées en tant qu'arguments supplémentaires aux prédicats concernés.

Modification de l'exercice 03 :

Les prédicats **itinTot/4** et **itinTard/4** ont été adaptés pour prendre en compte le type de transport. Ils trouvent un itinéraire entre deux arrêts avec un départ le plus tôt possible après un certain horaire, et une arrivée le plus tard possible avant un certain horaire, respectivement.

Exercice 05 : Interface Utilisateur

Une interface utilisateur simple a été implémentée où l'utilisateur peut choisir une station de départ, une station d'arrivée, ainsi que des options spécifiques comme le type de transport et ses préférences. Les résultats sont affichés à l'utilisateur de manière conviviale.

Choix d'Implémentation

- **Représentation des Horaires** : Les horaires sont représentés sous forme de paires [**Heures**, **Minutes**], facilitant les manipulations et calculs ultérieurs.
- **Représentation du Réseau de Transport** : La représentation en tant que prédicat **ligne/5** offre une structure claire pour décrire les lignes, les arrêts, les horaires, et les types de transport.

- **Adaptation des Prédicats** : Chaque prédicat a été adapté pour prendre en compte le type de transport et d'autres options. Ceci garantit une flexibilité pour l'extension future du programme.
- **Interface Utilisateur** : L'interface utilisateur est conçue de manière simple pour favoriser l'interaction avec l'utilisateur. Des instructions claires sont fournies pour guider l'utilisateur dans ses choix.

Conclusion

Le code développé répond aux exigences spécifiées dans l'énoncé du projet. Les prédicats ont été structurés de manière à être modulaires et extensibles, facilitant l'ajout de fonctionnalités supplémentaires à l'avenir. L'interface utilisateur offre une expérience conviviale tout en prenant en compte les préférences de l'utilisateur lors de la recherche d'itinéraires.