Rapport

Calcule d’un trajet de transports publics

Nom : nouri

Prénom : tarek

module : intelligence artificielle

I. Introduction :

Le présent rapport examine la mise en œuvre d'itinéraires à l'aide du langage de programmation Prolog. L'objectif principal est de résoudre le problème complexe de la recherche d'itinéraires entre deux arrêts dans un système de transport en prenant en compte les horaires de départ et d'arrivée.

La demande croissante d'optimiser les déplacements dans les réseaux de transport en commun entraîne le besoin de créer un tel système. Afin d'assurer une transition fluide d'un point à un autre, les voyageurs ont souvent besoin de planifier leurs itinéraires en fonction des horaires de départ et d'arrivée.

Ce défi est difficile en raison de la variété des trajets possibles, des contraintes temporelles et de la nécessité de fournir des résultats pertinents et précis. Une solution efficace dans le langage de programmation logique Prolog fournit une approche structurée pour résoudre ce problème en utilisant la logique formelle.

Ce rapport examine les étapes de conception et d'implémentation du système tout en mettant en lumière les choix de modélisation et les décisions prises pour garantir la précision et l'efficacité de la recherche d'itinéraires. De plus, nous parlerons des problèmes rencontrés et des solutions trouvées pour les surmonter. Cela nous aidera à comprendre comment Prolog peut être utilisé pour résoudre des problèmes de planification complexes dans le domaine des transports.

II. Description du Problème

En utilisant Prolog, ce projet vise à résoudre le problème de la recherche d'itinéraires dans un réseau de transport. L'objectif est d'aider les utilisateurs à trouver le meilleur itinéraire possible entre deux arrêts en prenant en compte les horaires de départ et d'arrivée. La modélisation des arrêts, les horaires précis, la recherche d'itinéraires optimaux et la prise en compte des contraintes temporelles de l'utilisateur font partie des spécifications.

Contraintes :

1. Réalisme temporel : Les horaires doivent être réalistes et inclure les retards potentiels.

2. Réalisme temporel : Les horaires doivent être réalistes et prendre en compte tout retard possible.

3. Adaptabilité : Le système doit s'adapter aux différents réseaux de transport et horaires.

La résolution nécessite une modélisation des données précise, une gestion des contraintes de temps et une mise en œuvre logique en Prolog. Pour répondre à ces exigences, les détails de la conception et de l'implémentation seront examinés dans les sections suivantes.

III. Approche de Solution

Le problème de recherche d'itinéraires en Prolog est résolu en modélisant les arrêts, les liaisons et les horaires à l'aide de faits. Les liaisons entre les arrêts sont étendues pour fournir des informations temporelles. La recherche d'itinéraires est effectuée à l'aide de règles qui définissent les exigences pour un itinéraire valide. Les règles incluent des contraintes temporelles spécifiées par l'utilisateur pour garantir des résultats conformes. En somme, cette méthode permet une représentation logique du réseau de transport et facilite une recherche efficace d'itinéraires respectant les horaires et les contraintes temporelles.

IV. Prédicats et Fonctionnement

	Rôle	Implémentation
lig	Il vérifie si Arret1 est sur la première ligne, puis vérifie si Arret2 est sur la première ligne et après Arret1.	recherche_arret(Arret, [[Arret|_]|LArret], LArret). recherche_arret(Arret, [[_|_]|LArret], LArret1):- recherche_arret(Arret, LArret, LArret1). lig(Arret1, Arret2, Ligne):- ligne(Ligne,_,LArret,_,_), recherche_arret(Arret1, LArret, LArret1), recherche_arret(Arret2,LArret1,_).
ligtot	Il vérifie si Arret1 est sur la première ligne, puis vérifie si Arret2 est sur la première ligne et après Arret1. Il calcule l'heure de départ la plus proche pour chaque ligne Ensuite vous mettez les résultats obtenus dans une liste Ensuite vous calculez la plus petite valeur parmi toutes les valeurs pour connaître l'heure la plus proche de la date de départ indiquée	rcovList1(A1,A2,[H1,H2],L,B5):- lig(A1,A2,L), ligne(L,_,_,[[X1,X2],Y,_],_), B1 is H1*60+H2, B2 is X1*60+X2, (B1<B2 -> B5 is B2 ; B3 is abs(B1-B2), B4 is (B3//Y)+1, B5 is B4*Y+B2). listR1(A1,A2,H,L,LM):- findall([L,B5],(rcovList1(A1,A2,H,L,B5)),LM). min([], L,MAX,L,MAX). min([[L,M]|T],Lact,MAX,L1,MAX1):- (M<MAX -> min(T,L,M,L1,MAX1) ; min(T,Lact,MAX,L1,MAX1)).

		ligtot(A1,A2,L,H):-
		  listR1(A1,A2,H,L,LM),
		  min(LM,L,10000000,L,_).

ligtard	Vérifie si une ligne (Ligne) arrive le plus tard possible avant un certain horaire (Horaire) entre les arrêts Arret1 et Arret2.	calcTout([], 0).
		calcTout([[_|X]|T], S):-
		  calcTout(T, S1),
		  S is S1 + X.
		
		rcovList2(A1, A2, [H1, H2], L, B5):-
		  lig(A1, A2, L),
		  ligne(L,_,A,[_,Y,[X1,X2]],_),
		  B1 is H1*60 + H2,
		  B2 is X1*60 + X2,
		  calcTout(A, S),
		  B10 is S + B2,
		  (B1 > B10
		  -> B5 is B10
		  ;
		  B3 is abs(B10 - B1),
		  B4 is (B3 // Y) + 1,
		  B5 is B10 - B4*Y
		  ).
		
		listR2(A1,A2,H,L,LM):-
		  findall([L,B5],(rcovList2(A1,A2,H,L,B5)),LM).
		
		max([], L,MAX,L,MAX).
		max([[L,M]|T],Lact,MAX,L1,MAX1):-
		  (M>MAX
		  -> max(T,L,M,L1,MAX1)
		  ; max(T,Lact,MAX,L1,MAX1)
		  ).
		
		ligtard(A1,A2,L,H):-
		  listR2(A1,A2,H,L,LM),
		  max(LM,L,1,L,_).

itinTot	Vérifie si une liste (Parcours) décrit un itinéraire de l'arrêt Arret1 à l'arrêt Arret2 qui part le plus tôt possible après un certain horaire (Horaire).	
itinTard	Vérifie si une liste (Parcours) décrit un itinéraire de l'arrêt Arret1 à l'arrêt	

	Arret2 qui arrive le plus tard possible avant un certain horaire (Horaire).	

V. Limitations et Améliorations Possibles

Limitations Potentielles :

1. Complexité Temporelle : Risque de coût élevé avec des réseaux étendus.
2. Gestion des Retards : Ne prend pas en compte explicitement les retards.
3. Variété des Moyens de Transport : Modèle actuel limité à une seule ligne.
4. Optimisation des Itinéraires : L'optimisation n'est pas explicitement traitée.

Améliorations Possibles :

1. Intégration de Données en Temps Réel : Mise à jour dynamique des horaires et des retards.
2. Interface Utilisateur : Développement d'une interface conviviale.
3. Extension des Critères d'Optimisation : Permettre aux utilisateurs de spécifier leurs préférences.
4. Validation des Données : Ajout de mécanismes pour garantir l'intégrité des données.
5. Modélisation Plus Complexes des Transports : Prise en compte de situations plus complexes.

VI. Conclusion

En résumé, notre application de Prolog pour la recherche d'itinéraires dans un réseau de transport a réussi à simuler efficacement les arrêts, les liaisons et les horaires. Les itinéraires qui respectent les contraintes temporelles peuvent être trouvés grâce aux prédicats créés. Cependant, des limites telles que la gestion des retards et la complexité temporelle ont été identifiées.

L'intégration de données en temps réel, le développement d'une interface utilisateur conviviale, l'extension des critères d'optimisation, la validation des données et une modélisation plus complexe des moyens de transport sont des améliorations potentielles.

Malgré ces considérations, la mise en œuvre offre une base solide, et les suggestions d'amélioration pourraient renforcer sa pertinence dans des contextes de planification de trajets plus complexes.