

TP numéro 4

Intelligence artificielle, ENSIIE

D'après Michele Pagani

Semestre 4, 2019–20

L'objectif de ce TP est de développer un programme en ProLog capable d'aider un usager d'un réseau de transport (par exemple la RATP) à identifier le trajet lui permettant de se rendre d'une station à une autre, en respectant certaines conditions (horaire de départ, horaire d'arrivée, etc.)

Exercice 1 : Représentation de notre réseaux de transport

L'ensemble des lignes sera définie par le prédicat suivant :

```
ligne(Nom, Type, LArret, LTrajetAller, LTrajetRetour)
```

où :

- **Nom** décrit le nom de la ligne, il peut être un numéro 1, 256, ou une lettre majuscule A, B, etc.
- **Type** décrit le type de moyen de transport, soit **metro**, **rer** ou **bus**.
- **LArret** est une liste des paires [[A1, T1], [A2, T2], ..., [An, Tn]] où Ai décrit le nom d'un arrêt desservi par la ligne et Ti le temps nécessaire à parcourir la distance entre A(i-1) et Ai. On suppose un temps constant dans les deux directions (de A(i-1) vers Ai et de Ai vers A(i-1));
- **LTrajetAller** est le triplet ayant comme premier élément l'horaire du premier départ de la ligne de A1 vers An, comme deuxième élément l'intervalle en minutes entre un départ et l'autre, et comme dernier élément l'horaire du dernier départ de la ligne de A1. Les horaires sont représentés sous forme d'une paire des nombres, le premier élément représentant les heures (de 0 jusqu'à 23), le deuxième élément les minutes (de 0 jusqu'à 59). Par exemple le triplet [[5,15], 5, [1,30]] signifie que il y aura un départ de A1 chaque 5 minutes à partir de 05h15 et jusqu'à 1h30;
- **LTrajetRetour** est le triplet ayant comme premier élément l'horaire du premier départ de la ligne de An vers A1, comme deuxième élément l'intervalle en minutes entre un départ et l'autre, et comme dernier élément l'horaire du dernier départ de la ligne de An.

Par exemple on peut imaginer de définir la ligne 11 du métro parisienne comme suit :

```
ligne(11, metro,
    [
        [mairie_lilas, 0],
        [porte_lilas, 3],
        [telegraphe, 1],
        [place_fetes, 1],
        [jourdain, 1],
        [pyrenees, 1],
        [belleville, 2],
        [goncourt, 2],
        [republique, 3],
        [arts_metiers, 2],
        [rambuteau, 1],
        [hotel_de_ville, 1],
        [chatelet, 1]
    ], [[5,15], 5, [1,30]], [[5,0], 5, [2,0]]
)
```

On trouvera dans `metro.pl` une petite base pour tester.

Exercice 2 : Recherche des itinéraires

1. Le but de cet exercice est de développer les outils nécessaires à manipuler les horaires. On vous suggère de représenter les horaires sous forme d'une paire des nombres [Heures, Minutes], le premier élément représentant les heures (de 0 jusqu'à 23), le deuxième élément les minutes (de 0 jusqu'à 59). Définir le prédicat `addh(X,M,R)`, qui est vrai quand R est l'horaire obtenu en sommant les minutes M à l'horaire X, par exemple :
`addh([13, 34], 30, [14, 4])` est vrai
`addh([10, 14], 25, [14, 4])` est faux
2. Étant donné un ensemble de lignes, on étudie le problème de savoir si un ligne passe par deux arrêts, éventuellement en respectant un choix de l'horaire de départs ou d'arrivée. Définir les prédicats suivants :
`lig(Arret1,Arret2,Ligne)`, qui est vrai quand Ligne passe de l'Arret1 à l'Arret2 ;
`ligtot(Arret1,Arret2,Ligne,Horaire)`, qui est vrai quand Ligne passe de l'Arret1 à l'Arret2 en partant après Horaire ;
`ligtard(Arret1,Arret2,Ligne,Horaire)`, qui est vrai quand Ligne passe de l'Arret1 à l'Arret2 en arrivant avant Horaire.
3. Maintenant on considère le problème de savoir s'il y a un itinéraire entre deux arrêts (éventuellement avec des échanges de moyen de transport). Les itinéraires seront représentés par des listes qui doivent contenir comme informations l'arrêt et l'horaire de départ et d'arrivée, et tous les arrêts et horaires des échanges. Étant donné un ensemble de lignes, écrire les prédicats suivants :
`itinTot(Arret1,Arret2,horaire,Parcours)` qui est vrai quand Parcours décrit un itinéraire de Arret1 à Arret2 qui part après Horaire ;
`itinTard(Arret1,Arret2,horaire,Parcours)` qui est vrai quand Parcours décrit un itinéraire de Arret1 à Arret2 qui arrive avant Horaire ;