

Exercice A (*Datalog*)

Le tramway fait des siennes. Une relation $Acces(x,y,n,E)$, mise à jour régulièrement, indique que la station y suit la station x sur la ligne n , et que le tramway est dans l'état E de x à y : si E vaut 1 alors la ligne fonctionne de x à y , et sinon la ligne ne fonctionne pas de x à y .

[On répondra en **datalog**. On vérifiera la correction des clauses en particulier celle des clauses récursives, et on expliquera la signification des relations auxiliaires utilisés.]

- (A.i) Quels sont les triplets (x,y,n) tels que la station y suit la station x sur la ligne n .
- (A.ii) Quels sont les triplets (x,y,n) tels que la station y suit la station x sur la ligne n , et que le tramway fonctionne de x à y .
- (A.iii) Quels sont les couples de stations consécutives qui sont reliées dans un sens et dans l'autre par un tramway en fonction?
- (A.iv) Quels sont les couples (x,y) de stations tels que l'on peut aller de x à y avec un tramway en fonction en suivant la même ligne?
- (A.v) Quels sont les couples (x,y) de stations tels que l'on peut aller de x à y avec un tramway en fonction, en changeant de ligne si besoin est?
- (A.vi) Quels sont les couples de stations (x,y) tels que l'on puisse aller avec un tramway en fonction de x à y mais pas revenir avec un tramway en fonction de y à x ?
- (A.vii) Un voyageur situé à la station u veut bien faire en tout au plus une station de tramway à pieds. A quelles stations peut-il accéder?

Exercice B (*Listes en prolog*)

(B.i) Définir le prédicat **appartient(X,L)** qui est vrai lorsque l'élément **X** appartient à la liste **L**.

(B.ii) Définir le prédicat **non_appartient(X,[])** qui est vrai lorsque l'élément **X** n'appartient pas à la liste **L**.

(B.iii) Définir le prédicat **sans_repetition(L)** qui est vrai lorsque la liste **L** ne contient pas deux fois le même élément.

(B.iv) Définir le prédicat **ajout_tete(X,L1,L2)** qui est vrai lorsque **L2** est la liste obtenue à partir de **L1** par ajout en tête de l'élément **X**.

(B.v) Définir le prédicat **ajout_queue(X,L1,L2)** qui est vrai lorsque **L2** est la liste obtenue à partir de **L1** par ajout en queue de l'élément **X**.

(B.vi) Définir le prédicat **supprimer(X,L1,L2)** qui est vrai lorsque **L2** est la liste obtenue à partir de **L1** en supprimant la première occurrence de **X** s'il y en a une, et, lorsqu'il n'y en a pas lorsque **L1=L2**.

(B.vii) Définir le prédicat **supprimer_fin(L1,L2)** qui est vrai lorsque **L2** est la liste obtenue à partir de **L1** en supprimant son dernier élément, ou lorsque **L1=L2=[]**.

(B.viii) Définir le prédicat **fusion(L1,L2,L3)** qui est vrai lorsque **L3** est obtenue à partir de **L1** et **L2** en prenant alternativement un élément dans **L1** et un dans **L2** et en adjoignant en queue les éléments non encore utilisés de la liste la plus longue parmi **L1** et **L2**.

(B.ix) Définir le prédicat **concatener(L1,L2,L3)** qui est vrai lorsque **L3** est la liste dont les éléments sont d'abord ceux de **L1** puis ceux de **L2**.

(B.x) Définir le prédicat **inverser(L1,L2)** qui est vrai lorsque **L2** est constituée des même éléments que **L1** mais en sens inverse.

(B.xi) Définir le prédicat **commun(L1,L2,L3)** qui est vrai lorsque **L3** est la liste sans répétition des éléments communs à **L1** et à **L2**. telle que l'ordre d'apparition des éléments dans **L3** est l'ordre de leur première apparition dans **L1** suivie de **L2**.

(B.xii) Définir le prédicat **ens(L1,L2)** qui est vrai lorsque **L2** est obtenue à partir de **L1** par suppression de toutes les occurrences d'un élément *sauf la dernière*. La liste **L2** a les même éléments que **L1**, mais sans répétition.

(B.xiii) Définir le prédicat **reunion(L1,L2,L3)** qui, en supposant que **L1** et **L2** sont sans répétition, est vrai lorsque **L3** est sans répétition et contient tous les éléments de **L1** et de **L2** dans l'ordre où ils apparaissent dans **L1** suivie de **L2**.

(B.xiv) Définir le prédicat **reunionbis(L1,L2,L3)** qui est vrai lorsque **L3** contient tous les éléments de **L1** et de **L2**, au plus une fois, et dans l'ordre dans leur dernière apparition dans **L1** suivie de **L2**. Ce prédicat ne requiert pas que **L1** et **L2** soient sans répétition.