

Résumé

Le but de ce miniprojet est d'utiliser la programmation logique et le langage Prolog pour résoudre un problème.

1 Problème

Après une rude deuxième année à l'ENSEEIH SN, vous rentrez enfin chez vous. Bien évidemment, vous avez choisi l'avion, Toulouse oblige¹...Le passager devant vous à l'enregistrement est un touriste qui voudrait ramener des souvenirs dans son sac-à-dos. Malheureusement, il ne peut pas tout prendre, car la compagnie aérienne ne l'autorisera pas à prendre plus de 20kg de bagages avec lui dans l'avion. Évidemment, le passager veut prendre un maximum d'objets. Ne sachant pas comment résoudre son problème, celui-ci se tourne vers vous et vous demande de l'aider à choisir des objets.

Cela tombe bien, puisque vous avez sur vous votre ordinateur portable équipé de la dernière version de GNU Prolog. Un petit programme logique devrait permettre de résoudre facilement ce problème...

On suppose que les objets que le passager veut prendre sont les suivants² :

objet	poids (kg)
morceau du pont Saint Pierre	10
cachous	1
stock de ballons de rugueubi	7
goodies Airbus et Cité de l'Espace	6
ordinateur salle TP N7	20

2 Questions

- on va représenter les poids des objets à l'aide d'un prédicat `poids/2` tel que `poids(0, P)` est vrai si `P` est la masse en kg de l'objet `0`. Définir le prédicat `poids` et les faits associés.
- définir un prédicat `poids_sac/2` tel que `poids_sac(S, P)` est vrai si `P` est le poids total de la liste d'objets `S`.
- définir un prédicat `sous_liste/2`, tel que `sous_liste(L, M)` est vrai si `M` est une sous-liste de `L`, c'est-à-dire si tous les éléments de `M` sont dans `L`, et s'ils sont rangés dans le même ordre que dans `L`. Par exemple, les sous-listes de `[a, b]` sont `[]`, `[a]`, `[b]`, et `[a, b]`.
- définir un prédicat `acceptable/2`, tel que `acceptable(L, S)` est vrai si `S` est une liste d'objets choisis dans la liste `L`, et si le poids total des objets de `S` ne dépasse pas 20 kg.
- définir, en utilisant la négation par échec, un prédicat `meilleur_poids/2`, tel que `meilleur_poids(L, S)` est vrai si `S` est une liste acceptable d'objets, choisis dans la liste `L`, de poids maximal : il n'y a pas moyen de choisir dans `L` une autre liste acceptable d'objets de poids total strictement supérieur au poids total des objets de `S`.
- on suppose maintenant que l'on a en plus un prédicat `utilise/2` qui représente l'utilité d'un objet pour le voyageur. Définir une relation `meilleure_utilite/2` tel que `meilleure_utilite(L, S)` est vrai si `S` est une liste d'objets de poids maximal et d'utilité maximale.
- la définition précédente de `meilleur_poids` n'est pas efficace. Nous allons chercher dans cette question une meilleure définition.

(a) soient les prédicats `q` et `p` définis comme suit :

```
q([],_,_) :- fail.  
q([X|_],[],[X]).  
q([X|RR],[X|RC],S) :- q(RR,RC,S).  
q([X|_],[Y|RC],S) :- X \= Y, S = [X,Y|RC].  
  
p(R,C,[C]) :- \+ (q(R,C,_)).  
p(R,C,[C|V]) :- q(R,C,S), p(R,S,V).
```

1. Le train c'est bien aussi...et moins polluant!

2. On pourra bien évidemment utiliser d'autres objets dans les questions pour tester les relations développées.

Le prédicat **fail** est toujours évalué à faux.

À quoi sert le prédicat **p**? On pourra évaluer par exemple $p([a,b],[],X)$, $p([a,b],[a],X)$, $p([a,b],[b],X)$, $p([a,b],[a,b],X)$.

- (b) donner une nouvelle définition de **meilleur_poids** n'utilisant pas la négation par échec et utilisant **p**. Cette nouvelle définition devra être plus efficace que la précédente. On pourra écrire une disjonction si besoin est avec Prolog grâce à l'opérateur **;**. Par exemple, la clause programme suivante :

```
A :- ((B1, B2, B3);  
      (C1,C2))
```

nous dit que **A** est vrai si

- **B1**, **B2** et **B3** sont vrais *ou*
- **C1** et **C2** sont vrais

3 Documents à rendre

Vous devez déposer pour le **23 avril 2021 à 23h00** sur le dépôt Moodle du cours le source d'un programme Prolog répondant aux questions ci-dessus. **Le fichier source devra impérativement s'appeler knapsack-login1-login2.pl où login1 et login2 sont les logins de votre binôme.** Les réponses aux questions pourront y être insérées sous forme de commentaires.

License CC BY-NC-SA 3.0



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license (CC BY-NC-SA 3.0)

You are free to Share (copy, distribute and transmit) and to Remix (adapt) this work under the following conditions:



Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial – You may not use this work for commercial purposes.



Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

See <http://creativecommons.org/licenses/by-nc-sa/3.0/> for more details.