

Rapport du DM de Coq

Adrien BLASSIAU
Corentin JUVIGNY

30 octobre 2019

Avant de commencer les deux exercices, on importe tout d'abord le module `List` comme demandé. On importe aussi `Omega` qui fournit la tactique **omega** qui se révèle être très pratique pour la gestion des expressions arithmétiques. À noter que l'on peut très bien faire sans (en utilisant des lemmes du module `Arith.Gt`, `Arith.Lt`, etc) mais cela permet de réduire de quelques lignes les preuves ...

Les théorèmes admis (donc ceux que nous n'avons pas réussi à démontrer) sont notés en **rouge**.

1 Exercice 1 : Listes et comptage

Dans la suite on travaille avec un type `A` quelconque muni d'une égalité décidable. Pour les tests, on utilise `nat` (ils sont commentés dans le fichier `.v`).

Variable `A : Type`.

Hypothesis `dec_A : ∀ (x y : A), ({x=y}+{~x=y})`.

Question 1 `occ` prend en argument un élément `x` de type `A` et une liste `l` de type `(list A)` et retourne le nombre d'occurrences de `x` dans `l`. Elle a la signature suivante :

`occ : A → list A → nat`

Question 2 On démontre le théorème suivant :

Theorem `occ_app : ∀ (x : A) l1 l2, occ x (app l1 l2) = (occ x l1) + (occ x l2)`.

Question 3 On démontre le théorème suivant :

Theorem `occ_filter : ∀ (P : A → bool) (a : A) l, occ a (filter P l) = if (P a) then occ a l else 0`.

Question 4 On démontre le théorème suivant :

Theorem `occ_map : ∀ (y : A) (z : A), y ≠ z → ∃ (f : A → A) (x : A) (l : list A), occ (f x) (map f l) ≠ occ x l`.

Question 5 On démontre le théorème de l'énoncé en deux temps, en utilisant un petit lemme intermédiaire :

Lemma `mem_diff : ∀ (l : list A) (x : A) (y : A), mem x (cons y l) → x ≠ y → mem x l`.

Theorem `mem_null_1 : ∀ (l : list A) (x : A), occ x l = 0 → ~ (mem x l)`.

Theorem `mem_null_2 : ∀ (l : list A) (x : A), ~ (mem x l) → occ x l = 0`.

Question 6 On démontre le théorème de l'énoncé en deux temps. Nous ne sommes pas parvenus à démontrer l'un des sens du théorème :

Theorem `doublon_1 : ∀ (l : list A) (x : A), nodup l → occ x l ≤ 1`.

Theorem `doublon_2 : ∀ (l : list A) (x : A), occ x l ≤ 1 → nodup l`

2 Exercice 2 : Implantation des multi-ensembles

Dans la suite on travaille avec un type T quelconque muni d'une égalité décidable. Pour les tests, on utilise nat (ils sont commentés dans le fichier `.v`).

Variable $T : \text{Type}$.

Hypothesis $T_eq_dec : \forall (x \ y : T), \{x=y\} + \{\neg x=y\}$.

2.1 Implantation des multi-ensembles à l'aide de listes d'association

Question 1 On définit ici un multi-ensemble comme une liste de couples (élément, cardinalité), de la manière suivante :

Definition $\text{multiset} := \text{list } (T \times \text{nat})$.

Voici les signatures des différentes fonctions demandées dans l'énoncé :

$\text{empty} : \text{multiset}$

$\text{singleton} : T \rightarrow \text{multiset}$

$\text{add} : T \rightarrow \text{nat} \rightarrow \text{multiset} \rightarrow \text{multiset}$

$\text{member} : T \rightarrow \text{multiset} \rightarrow \text{bool}$

$\text{union} : \text{multiset} \rightarrow \text{multiset} \rightarrow \text{multiset}$

$\text{multiplicity} : T \rightarrow \text{multiset} \rightarrow \text{nat}$

$\text{removeOne} : T \rightarrow \text{multiset} \rightarrow \text{multiset}$

$\text{removeAll} : T \rightarrow \text{multiset} \rightarrow \text{multiset}$

Question 2

Question 2. a) On définit le prédicat InMultiset de la manière suivante :

Inductive $\text{InMultiset} (x : T) (l : \text{multiset}) : \text{Prop} :=$
| $\text{inMultiset_intro} : \text{member } x \ l = \text{true} \rightarrow \text{InMultiset } x \ l$.

Question 2. b) On définit le prédicat wf de la manière suivante :

Inductive $\text{wf} (l : \text{multiset}) : \text{Prop} :=$
| $\text{wf_intro} : (\forall x, (\text{InMultiset } x \ (\text{removeAll } x \ l) \rightarrow \text{False}) \wedge (\text{member } x \ l = \text{true} \rightarrow (\text{multiplicity } x \ l) > 0)) \rightarrow \text{wf } l$.

Question 2. c) On démontre maintenant la bonne formation de nos différentes fonctions. 2 lemmes intermédiaires ont été créés.

Nous ne sommes pas parvenus à démontrer cette propriété pour 4 fonctions :

Theorem *empty_wf* : *wf empty*.

Theorem *singleton_wf* : $\forall (x : T), \text{wf } (\text{singleton } x)$.

Lemma *plus_n_0* : $\forall n, n + 0 = n$.

Lemma *wf_plus_n* : $\forall t \ n0 \ n \ l, \text{wf } ((t, n0) :: l) \rightarrow \text{wf } ((t, n0+n) :: l)$.

Theorem *add_wf* : $\forall (x : T) (n : \text{nat}) (l : \text{multiset}), \text{wf } l \rightarrow \text{wf } (\text{add } x \ n \ l)$.

Theorem *union_wf* : $\forall (l : \text{multiset}) (l' : \text{multiset}), \text{wf } l \rightarrow \text{wf } l' \rightarrow \text{wf } (\text{union } l \ l')$.

Theorem *removeOne_wf* : $\forall (x : T) (l : \text{multiset}), \text{wf } l \rightarrow \text{wf } (\text{removeOne } x \ l)$.

Theorem *removeAll_wf* : $\forall (x : T) (l : \text{multiset}), \text{wf } l \rightarrow \text{wf } (\text{removeAll } x \ l)$.

Question 3 On démontre maintenant les différents théorèmes indiqués dans l'énoncé. 1 lemme intermédiaire a été créé.

Nous ne sommes pas parvenus à démontrer les deux derniers théorèmes :

Theorem *proof_1_1* : $\forall (x : T), \neg \text{InMultiset } x \ \text{empty}$.

Theorem *proof_2_1* : $\forall x \ y, \text{InMultiset } y \ (\text{singleton } x) \leftrightarrow x = y$.

Theorem *proof_3_1* : $\forall x, \text{multiplicity } x \ (\text{singleton } x) = 1$.

Theorem *proof_4_1* : $\forall x \ s, \text{wf } s \rightarrow (\text{member } x \ s = \text{true} \leftrightarrow \text{InMultiset } x \ s)$.

Theorem *proof_5_1* : $\forall x \ n \ s, n > 0 \rightarrow \text{InMultiset } x \ (\text{add } x \ n \ s)$.

Theorem *proof_6_1* : $\forall x \ n \ y \ s, x \neq y \rightarrow (\text{InMultiset } y \ (\text{add } x \ n \ s) \leftrightarrow \text{InMultiset } y \ s)$.

Lemma *proof_7_1_1* : $\forall x \ s, \text{multiplicity } x \ s \neq 0 \rightarrow \text{InMultiset } x \ s$.

Theorem *proof_7_1_2* : $\forall x \ s, \text{wf } s \rightarrow (\text{multiplicity } x \ s = 0 \leftrightarrow \neg \text{InMultiset } x \ s)$.

Theorem *proof_8_1* : $\forall x \ n \ s, \text{multiplicity } x \ (\text{add } x \ n \ s) = n + (\text{multiplicity } x \ s)$.

Theorem *proof_9_1* : $\forall x \ n \ y \ s, x \neq y \rightarrow \text{wf } s \rightarrow \text{multiplicity } y \ (\text{add } x \ n \ s) = \text{multiplicity } y \ s$.

Theorem *proof_10_1* : $\forall s \ t \ x, \text{wf } s \rightarrow \text{wf } t \rightarrow (\text{InMultiset } x \ (\text{union } s \ t) \leftrightarrow \text{InMultiset } x \ s \vee \text{InMultiset } x \ t)$.

Question 4 On démontre maintenant quelques théorèmes supplémentaires sur *removeOne* et *removeAll*.

Theorem *proof_11_1* : $\forall x, \text{multiplicity } x \ (\text{removeOne } x \ (\text{singleton } x)) = 0$.

Theorem *proof_12_1* : $\forall x, \text{multiplicity } x \ (\text{removeAll } x \ (\text{singleton } x)) = 0$.

Theorem *proof_13_1* : $\forall x \ l \ n, \text{multiplicity } x \ l = n \rightarrow n > 1 \rightarrow \text{multiplicity } x \ (\text{removeOne } x \ l) = n - 1$.

Theorem *proof_14_1* : $\forall x \ l, \text{wf } l \rightarrow \neg (\text{InMultiset } x \ (\text{removeAll } x \ l))$.

Theorem *proof_15_1* : $\forall x \ l, \text{multiplicity } x \ l > 1 \rightarrow \text{InMultiset } x \ (\text{removeOne } x \ l)$.

2.2 Implantation Fonctionnelle des multi-ensembles

Question 1 On définit ici un multi-ensemble comme une fonction qui encode les multiplicités, de la manière suivante :

Definition *multiset_2* := $T \rightarrow \text{nat}$.

Voici les signatures des différentes fonctions demandées dans l'énoncé :

empty_2 : *multiset_2*

singleton_2 : $T \rightarrow \text{multiset}_2$

add_2 : $T \rightarrow \text{nat} \rightarrow \text{multiset}_2 \rightarrow \text{multiset}_2$

member_2 : $T \rightarrow \text{multiset}_2 \rightarrow \text{bool}$

union_2 : $\text{multiset}_2 \rightarrow \text{multiset}_2 \rightarrow \text{multiset}_2$

multiplicity_2 : $T \rightarrow \text{multiset}_2 \rightarrow \text{nat}$

removeOne_2 : $T \rightarrow \text{multiset}_2 \rightarrow \text{multiset}_2$

removeAll_2 : $T \rightarrow \text{multiset}_2 \rightarrow \text{multiset}_2$

Question 2 On définit le prédicat *InMultiset_2* de la manière suivante :

Inductive *InMultiset_2* ($x : T$) ($l : \text{multiset}_2$) : Prop :=
| *inMultiset_2_intro* : *member_2* x l = true \rightarrow *InMultiset_2* x l .

Question 3 On démontre maintenant les différents théorèmes indiqués dans l'énoncé ainsi que ceux sur les fonctions *removeOne* et *removeAll*. 1 lemme intermédiaire a été créé.

Theorem *proof_1_2* : $\forall (x : T), \neg \text{InMultiset}_2 x \text{ empty}_2$.

Theorem *proof_2_2* : $\forall x y, \text{InMultiset}_2 y (\text{singleton}_2 x) \leftrightarrow x = y$.

Theorem *proof_3_2* : $\forall x, \text{multiplicity}_2 x (\text{singleton}_2 x) = 1$.

Theorem *proof_4_2* : $\forall x s, \text{member}_2 x s = \text{true} \leftrightarrow \text{InMultiset}_2 x s$.

Theorem *proof_5_2* : $\forall x n s, n > 0 \rightarrow \text{InMultiset}_2 x (\text{add}_2 x n s)$.

Theorem *proof_6_2* : $\forall x n y s, x \neq y \rightarrow (\text{InMultiset}_2 y (\text{add}_2 x n s) \leftrightarrow \text{InMultiset}_2 y s)$.

Lemma *proof_7_2_1* : $\forall x s, s x \neq 0 \rightarrow \text{InMultiset}_2 x s$.

Theorem *proof_7_2_2* : $\forall x s, \text{multiplicity}_2 x s = 0 \leftrightarrow \neg \text{InMultiset}_2 x s$.

Theorem *proof_8_2* : $\forall x n s, \text{multiplicity}_2 x (\text{add}_2 x n s) = n + (\text{multiplicity}_2 x s)$.

Theorem *proof_9_2* : $\forall x n y s, x \neq y \rightarrow \text{multiplicity}_2 y (\text{add}_2 x n s) = \text{multiplicity}_2 y s$.

Theorem *proof_10_2* : $\forall s t x, (\text{InMultiset}_2 x (\text{union}_2 s t) \leftrightarrow \text{InMultiset}_2 x s \vee \text{InMultiset}_2 x t)$.

Theorem *proof_11_2* : $\forall x, \text{multiplicity}_2 x (\text{removeOne}_2 x (\text{singleton}_2 x)) = 0$.

Theorem *proof_12_2* : $\forall x, \text{multiplicity}_2 x (\text{removeAll}_2 x (\text{singleton}_2 x)) = 0$.

Theorem *proof_13_2* : $\forall x l n, \text{multiplicity}_2 x l = n \rightarrow n > 1 \rightarrow \text{multiplicity}_2 x (\text{removeOne}_2 x l) = n - 1$.

Theorem *proof_14_2* : $\forall x\ l, \sim(\text{InMultiset_2 } x\ (\text{removeAll_2 } x\ l))$.

Theorem *proof_15_2* : $\forall x\ l, \text{multiplicity_2 } x\ l > 1 \rightarrow \text{InMultiset_2 } x\ (\text{removeOne_2 } x\ l)$.