Cardinality of Multisets

Lukas Bulwahn

August 16, 2018

Abstract

This entry provides three lemmas to count the number of multisets of a given size and finite carrier set. The first lemma provides a cardinality formula assuming that the multiset's elements are chosen from the given carrier set. The latter two lemmas provide formulas assuming that the multiset's elements also cover the given carrier set, i.e., each element of the carrier set occurs in the multiset at least once.

The proof of the first lemma uses the argument of the recurrence relation for counting multisets [1]. The proof of the second lemma is straightforward, and the proof of the third lemma is easily obtained using the first cardinality lemma. A challenge for the formalization is the derivation of the required induction rule, which is a special combination of the induction rules for finite sets and natural numbers. The induction rule is derived by defining a suitable inductive predicate and transforming the predicate's induction rule.

Contents

1 Cardinality of Multisets

theory *Card-Multisets*
imports
HOL−Library.Multiset
begin

1.1 Additions to Multiset Theory

lemma *mset-set-set-mset-subseteq*:

  mset-set (*set-mset* M) $\subseteq\#$ M
proof (*induct* M)
  case *empty*
  show *?case* **by** *simp*
next
  case (*add* x M)
  from *this* **show** *?case*
  proof (*cases* $x \in\# M$)
    assume $x \in\# M$
    from *this* **have** *mset-set* (*set-mset* (M + {#x#})) = *mset-set* (*set-mset* M)
      by (*simp add*: *insert-absorb*)
    from *this add.hyps* **show** *?thesis*
      using *subset-mset.order.trans* **by** *fastforce*
  next
    assume $\neg$ $x \in\# M$
    from *this add.hyps* **have** {#x#} + *mset-set* (*set-mset* M) $\subseteq\#$ M + {#x#}
      by (*simp add*: *insert-subset-eq-iff*)
    from *this* ‹$\neg$ $x \in\# M$› **show** *?thesis* **by** *simp*
  qed
qed

lemma *size-mset-set-eq-card*:
  assumes *finite* A
  shows *size* (*mset-set* A) = *card* A
using *assms* **by** (*induct* A) *auto*

lemma *card-set-mset-leq*:
  card (*set-mset* M) $\leq$ *size* M
by (*induct* M) (*auto simp add*: *card-insert-le-m1*)

1.2 Lemma to Enumerate Sets of Multisets

lemma *set-of-multisets-eq*:
  assumes $x \notin A$
  shows {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = *Suc* k} =
    {M. *set-mset* M $\subseteq$ A $\wedge$ *size* M = *Suc* k} $\cup$
    (λM. M + {#x#}) ‘ {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = k}
proof −
  from ‹$x \notin A$› **have** {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = *Suc* k} =
    {M. *set-mset* M $\subseteq$ A $\wedge$ *size* M = *Suc* k} $\cup$
    {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = *Suc* k $\wedge$ $x \in\# M$}
    by *auto*
  moreover have {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = *Suc* k $\wedge$ $x \in\# M$} =
    (λM. M + {#x#}) ‘ {M. *set-mset* M $\subseteq$ *insert* x A $\wedge$ *size* M = k} (**is** *?S* =
?T)
  proof
    show *?S* $\subseteq$ *?T*
    proof
      fix M

```
      assume M ∈ ?S
      from this have M = M − {#x#} + {#x#} by auto
      moreover have M − {#x#} ∈ {M. set-mset M ⊆ insert x A ∧ size M = k}
      proof −
        have set-mset (M − {#x#} + {#x#}) ⊆ insert x A
          using ‹M ∈ ?S› by force
        moreover have size (M − {#x#} + {#x#}) = Suc k ∧ x ∈# M − {#x#} + {#x#}
          using ‹M ∈ ?S› by force
        ultimately show ?thesis by force
      qed
      ultimately show M ∈ ?T by auto
    qed
  next
    show ?T ⊆ ?S by force
  qed
  ultimately show ?thesis by auto
qed
```

1.3 Derivation of Suitable Induction Rule

```
context
begin

private inductive R :: 'a set ⇒ nat ⇒ bool
where
  finite A ⟹ R A 0
| R {} k
| finite A ⟹ x ∉ A ⟹ R A (Suc k) ⟹ R (insert x A) k ⟹ R (insert x A) (Suc k)

private lemma R-eq-finite:
  R A k ⟷ finite A
proof
  assume R A k
  from this show finite A by cases auto
next
  assume finite A
  from this show R A k
  proof (induct A)
    case empty
    from this show ?case by (rule R.intros(2))
  next
    case insert
    from this show ?case
    proof (induct k)
      case 0
      from this show ?case
```

by (*intro R.intros*(*1*) *finite.insertI*)
next
case *Suc*
from *this* **show** *?case*
by (*metis R.simps Zero-neq-Suc diff-Suc-1*)
qed
qed
qed

lemma *finite-set-and-nat-induct*[*consumes 1*, *case-names zero empty step*]:
assumes *finite A*
assumes $\bigwedge A$. *finite A* $\Longrightarrow$ *P A 0*
assumes $\bigwedge k$. *P* {} *k*
assumes $\bigwedge A\ k\ x$. *finite A* $\Longrightarrow$ $x \notin A$ $\Longrightarrow$ *P A* (*Suc k*) $\Longrightarrow$ *P* (*insert x A*) *k* $\Longrightarrow$ *P* (*insert x A*) (*Suc k*)
shows *P A k*
proof −
from ‹*finite A*› **have** *R A k* **by** (*subst R-eq-finite*)
from *this assms*(*2*−*4*) **show** *?thesis* **by** (*induct A k*) *auto*
qed

end

1.4 Finiteness of Sets of Multisets

lemma *finite-multisets*:
assumes *finite A*
shows *finite* {*M*. *set-mset M* $\subseteq$ *A* $\wedge$ *size M* = *k*}
using *assms*
proof (*induct A k rule*: *finite-set-and-nat-induct*)
case *zero*
from *this* **show** *?case* **by** *auto*
next
case *empty*
from *this* **show** *?case* **by** *auto*
next
case (*step A k x*)
from *this* **show** *?case*
using *set-of-multisets-eq*[*OF* ‹$x \notin A$›] **by** *simp*
qed

1.5 Cardinality of Multisets

lemma *card-multisets*:
assumes *finite A*
shows *card* {*M*. *set-mset M* $\subseteq$ *A* $\wedge$ *size M* = *k*} = (*card A* + *k* − *1*) *choose k*
using *assms*
proof (*induct A k rule*: *finite-set-and-nat-induct*)
case (*zero A*)
assume *finite* (*A* :: *'a set*)

have {*M*. *set-mset M* ⊆ *A* ∧ *size M* = *0*} = {{#}} **by** *auto*
from *this* **show** *card* {*M*. *set-mset M* ⊆ *A* ∧ *size M* = *0*} = *card A* + *0* − *1 choose 0*
by *simp*
next
case (*empty k*)
show *card* {*M*. *set-mset M* ⊆ {} ∧ *size M* = *k*} = *card* {} + *k* − *1 choose k*
by (*cases k*) (*auto simp add*: *binomial-eq-0*)
next
case (*step A k x*)
let $?S_1$ = {*M*. *set-mset M* ⊆ *A* ∧ *size M* = *Suc k*}
and $?S_2$ = {*M*. *set-mset M* ⊆ *insert x A* ∧ *size M* = *k*}
assume *hyps1*: *card* $?S_1$ = *card A* + *Suc k* − *1 choose Suc k*
assume *hyps2*: *card* $?S_2$ = *card* (*insert x A*) + *k* − *1 choose k*
have *finite-sets*: *finite* $?S_1$ *finite* ((λ*M*. *M* + {#*x*#}) ‘ $?S_2$)
using ‹*finite A*› **by** (*auto simp add*: *finite-multisets*)
have *inj*: *inj-on* (λ*M*. *M* + {#*x*#}) $?S_2$ **by** (*rule inj-onI*) *auto*
have *card* {*M*. *set-mset M* ⊆ *insert x A* ∧ *size M* = *Suc k*} =
card ($?S_1$ ∪ (λ*M*. *M* + {#*x*#}) ‘ $?S_2$)
using *set-of-multisets-eq* ‹*x* ∉ *A*› **by** *fastforce*
also have ... = *card* $?S_1$ + *card* ((λ*M*. *M* + {#*x*#}) ‘ $?S_2$)
using *finite-sets* ‹*x* ∉ *A*› **by** (*subst card-Un-disjoint*) *auto*
also have ... = *card* $?S_1$ + *card* $?S_2$
using *inj* **by** (*auto intro*: *card-image*)
also have ... = *card A* + *Suc k* − *1 choose Suc k* + (*card* (*insert x A*) + *k* − *1 choose k*)
using *hyps1 hyps2* **by** *simp*
also have ... = *card* (*insert x A*) + *Suc k* − *1 choose Suc k*
using ‹*x* ∉ *A*› ‹*finite A*› **by** *simp*
finally show *?case* **.**
qed

lemma *card-too-small-multisets-covering-set*:
assumes *finite A*
assumes *k* < *card A*
shows *card* {*M*. *set-mset M* = *A* ∧ *size M* = *k*} = *0*
proof −
from ‹*k* < *card A*› **have** *eq*: {*M*. *set-mset M* = *A* ∧ *size M* = *k*} = {}
using *card-set-mset-leq Collect-empty-eq leD* **by** *auto*
from *this* **show** *?thesis* **by** (*metis card-empty*)
qed

lemma *card-multisets-covering-set*:
assumes *finite A*
assumes *card A* ≤ *k*
shows *card* {*M*. *set-mset M* = *A* ∧ *size M* = *k*} = (*k* − *1*) *choose* (*k* − *card A*)
proof −
have {*M*. *set-mset M* = *A* ∧ *size M* = *k*} = (λ*M*. *M* + *mset-set A*) ‘

$\{M.\ set\text{-}mset\ M \subseteq A \wedge size\ M = k - card\ A\}$ (**is** $?S = ?f$ ‘ $?T$)
proof
show $?S \subseteq ?f$ ‘ $?T$
proof
fix M
assume $M \in ?S$
from *this* **have** $M = M - mset\text{-}set\ A + mset\text{-}set\ A$
by (*auto simp add*: *mset-set-set-mset-subseteq subset-mset.diff-add*)
moreover from ‹$M \in ?S$› **have** $M - mset\text{-}set\ A \in ?T$
by (*auto simp add*: *mset-set-set-mset-subseteq size-Diff-submset size-mset-set-eq-card in-diffD*)
ultimately show $M \in ?f$ ‘ $?T$ **by** *auto*
qed
next
from ‹*finite A*› ‹$card\ A \leq k$› **show** $?f$ ‘ $?T \subseteq ?S$
by (*auto simp add*: *size-mset-set-eq-card*)+
qed
moreover have *inj-on ?f ?T* **by** (*rule inj-onI*) *auto*
ultimately have $card\ ?S = card\ ?T$ **by** (*simp add*: *card-image*)
also have $\ldots = card\ A + (k - card\ A) - 1\ choose\ (k - card\ A)$
using ‹*finite A*› **by** (*simp only*: *card-multisets*)
also have $\ldots = (k - 1)\ choose\ (k - card\ A)$
using ‹$card\ A \leq k$› **by** *auto*
finally show *?thesis* **.**
qed

end

References

[1] Wikipedia. Multiset — wikipedia, the free encyclopedia, 2016. [Online; accessed 23-June-2016].