Cardinality of Number Partitions

Lukas Bulwahn

June 11, 2019

Abstract

This entry provides a basic library for number partitions, defines the two-argument partition function through its recurrence relation and relates this partition function to the cardinality of number partitions. The main proof shows that the recursively-defined partition function with arguments $n$ and $k$ equals the cardinality of number partitions of $n$ with exactly $k$ parts. The combinatorial proof follows the proof sketch of Theorem 2.4.1 in Mazur’s textbook “Combinatorics: A Guided Tour” [2]. This entry can serve as starting point for various more intrinsic properties about number partitions, the partition function and related recurrence relations.

Contents

1 Additions to Isabelle’s Main Theories ................................................. 2
   1.1 Addition to Finite-Set Theory .................................................. 2
   1.2 Addition to Set-Interval Theory .............................................. 2
   1.3 Additions to Multiset Theory .................................................. 2

2 Number Partitions ........................................................................... 3
   2.1 Number Partitions as $\mathsf{nat} \Rightarrow \mathsf{nat}$ Functions .............. 3
   2.2 Bounds and Finiteness of Number Partitions .............................. 4
   2.3 Operations of Number Partitions .............................................. 6
   2.4 Number Partitions as Multisets on Natural Numbers ................. 12
      2.4.1 Relationship to Definition on Functions ............................ 12

3 Cardinality of Number Partitions .................................................... 14
   3.1 The Partition Function ............................................................ 14
   3.2 Cardinality of Number Partitions .............................................. 15
   3.3 Cardinality of Number Partitions as Multisets of Natural Numbers .................................................. 19
   3.4 Cardinality of Number Partitions with only 1-parts .................... 20
1 Additions to Isabelle’s Main Theories

theory Additions-to-Main
imports HOL-Library.Multiset
begin

1.1 Addition to Finite-Set Theory

lemma bound-domain-and-range-impl-finitely-many-functions:
finite \{f::nat\Rightarrow nat. (\forall i. f i \leq n) \land (\forall i\geq m. f i = 0)\}

proof (induct m)
case 0
have eq: \{f. (\forall i. f i \leq n) \land (\forall i. f i = 0)\} = \{(\lambda -. 0)\} by auto
from this show ?case by auto (subst eq; auto)

next
case (Suc m)
let ?S = (\lambda (y, f). f (m := y)) \cdot ({0..n} \times \{f. (\forall i. f i \leq n) \land (\forall i\geq m. f i = 0)\})

{ fix g
  assume \forall i. g i \leq n \land \forall i\geq Suc m. g i = 0
  from this have g \in ?S by (auto intro: image-eqI [where x = (g m, g(m:=0))])
}
from this have \{f. (\forall i. f i \leq n) \land (\forall i\geq Suc m. f i = 0)\} = ?S by auto
from this Suc show ?case by simp
qed

1.2 Addition to Set-Interval Theory

lemma sum-atMost-remove-nat:
assumes k \leq (n :: nat)
shows (\sum i\leq n. f i) = f k + (\sum i\in{..n}-{k}. f i)
using assms by (auto simp add: sum.remove[where x=k])

1.3 Additions to Multiset Theory

lemma set-mset-Abs-multiset:
assumes f \in multiset
shows set-mset (Abs-multiset f) = \{x. f x > 0\}
using assms unfolding set-mset-def by simp

lemma sum-mset-sum-count:
sum-mset M = (\sum i\in set-mset M. count M i \cdot i)
proof (induct M)
show sum-mset \{\#\} = (\sum i\in set-mset \{\#\}. count \{\#\} i \cdot i) by simp
next
fix M x
assume hyp: sum-mset M = (\sum i\in set-mset M. count M i \cdot i)
show \text{sum-mset}(\text{add-mset} \ x \ M) = \left(\sum_{i \in \text{set-mset}(\text{add-mset} \ x \ M). \text{count}(\text{add-mset} \ x \ M)} i \ast i\right)

\text{proof (cases } x \in \# \ M)\n\quad \text{assume } a: \neg x \in \# \ M\n\quad \text{from this have } \text{count} M x = 0 \text{ by (meson count-inI)}\n\quad \text{from } (\neg x \in \# \ M) \text{ this hyp show } ?\text{thesis}\n\quad \text{by (auto intro!: sum.cong)}
\text{next}\n\quad \text{assume } x \in \# \ M\n\quad \text{have } \text{sum-mset}(\text{add-mset} \ x \ M) = \left(\sum_{i \in \text{set-mset} M} \text{count}(\text{add-mset} \ x \ M) i \ast i\right) + x
\quad \text{using hyp by simp}\n\quad \text{also have } \ldots = \left(\sum_{i \in \text{set-mset} M} \text{count}(\text{add-mset} \ x \ M) i \ast i\right) + \text{count} M x \ast x + x
\quad \text{using } (x \in \# \ M) \text{ by (simp add: sum.remove[of - x])}\n\quad \text{also have } \ldots = \text{count}(\text{add-mset} \ x \ M) x \ast x + \left(\sum_{i \in \text{set-mset}(\text{add-mset} \ x \ M)} \text{count}(\text{add-mset} \ x \ M) i \ast i\right)
\quad \text{by simp}\n\quad \text{also have } \ldots = \left(\sum_{i \in \text{set-mset}(\text{add-mset} \ x \ M)} \text{count}(\text{add-mset} \ x \ M) i \ast i\right)
\quad \text{using } (x \in \# \ M) \text{ by (simp add: sum.remove[of - x])}\n\quad \text{finally show } ?\text{thesis} .\n\text{qed}\n\text{qed}

\text{lemma } \text{sum-mset-eq-sum-on-supersets}:\n\text{assumes } \text{finite } A \text{ set-mset } M \subseteq A\n\text{shows } \left(\sum_{i \in \text{set-mset} M} \text{count} M i \ast i\right) = \left(\sum_{i \in A} \text{count } M i \ast i\right)
\text{proof -}\n\quad \text{note } \text{finite } A \text{ (set-mset } M \subseteq A)\n\quad \text{moreover have } \forall i \in A - \text{set-mset } M. \text{count} M i \ast i = 0
\quad \text{using count-inI by fastforce}\n\quad \text{ultimately show } ?\text{thesis}\n\quad \text{by (auto intro: sum.mono-neutral-cong-left)}
\text{qed}

\text{end}

2 \text{ Number Partitions}

theory \text{Number-Partition}\n\text{imports Additions-to-Main}\n\text{begin}\n
2.1 \text{ Number Partitions as } \text{nat} \Rightarrow \text{nat} \text{ Functions}\n
\text{definition } \text{partitions} :: (\text{nat} \Rightarrow \text{nat}) \Rightarrow \text{nat} \Rightarrow \text{bool} \text{ (infix } \text{partitions 50)}\n\text{where}\n\quad p \text{ partitions } n = ((\forall i. p i \neq 0 \rightarrow 1 \leq i \land i \leq n) \land (\sum i \leq n. p i \ast i) = n)

\text{lemma } \text{partitionsI}:\n\text{assumes } \land i. p i \neq 0 \Rightarrow 1 \leq i \land i \leq n

3
assumes \((\sum_{i \leq n} p \ i \ i) = n\)
shows \(p \text{ partitions } n\)
using assms unfolding partitions-def by auto

lemma partitionsE:
assumes \(p \text{ partitions } n\)
obtains \(\forall i. \ p \ i \neq 0 \implies 1 \leq i \land i \leq n\)
(\(\sum_{i \leq n} p \ i \ i) = n\)
using assms unfolding partitions-def by auto

lemma partitions-zero:
\(p \text{ partitions } 0 \iff p = (\lambda i. \ 0)\)
unfolding partitions-def by auto

lemma partitions-one:
\(p \text{ partitions } (\text{Suc } 0) \iff p = (\lambda i. \ 0)(1 := 1)\)
unfolding partitions-def
by (auto split: if-split-asm) (auto simp add: fun-eq-iff)

2.2 Bounds and Finiteness of Number Partitions

lemma partitions-imp-finite-elements:
assumes \(p \text{ partitions } n\)
shows finite \(\{i. \ 0 < p \ i\}\)
proof –
from assms have \(\{i. \ 0 < p \ i\} \subseteq \{..n\}\)
by (auto elim: partitionsE)
from this show ?thesis
using rev-finite-subset by blast
qed

lemma partitions-imp-multiset:
assumes \(p \text{ partitions } n\)
shows \(p \in \text{multiset}\)
using assms partitions-imp-finite-elements multiset-def by auto

lemma partitions-bounds:
assumes \(p \text{ partitions } n\)
shows \(p \ i \leq n\)
proof –
from assms have index-bounds: \((\forall i. \ p \ i \neq 0 \implies 1 \leq i \land i \leq n)\)
and sum: \((\sum_{i \leq n} p \ i \ i) = n\)
unfolding partitions-def by auto
show ?thesis
proof (cases \(1 \leq i \land i \leq n\))
case True
from True have \(\{..n\} = \text{insert } i \{i', \ i' \leq n \land i' \neq i\}\)
by blast
from sum[unfolded this] have \(p \ i \ i + (\sum_{i \in \{i', \ i' \leq n \land i' \neq i\}} p \ i \ i) = n\)
by auto
from this have \(p \ i \ i \leq n\)
by linarith
from this True show ?thesis using dual-order.trans by fastforce
next
  case False
  from this index-bounds show \( ?thesis \) by fastforce
qed

lemma partitions-parts-bounded:
  assumes \( p \) partitions \( n \)
  shows \( \sum p \{..n\} \leq n \)
proof (—)
  { fix \( i \)
    assume \( i \leq n \)
    from assms have \( p \, i \leq p \, i \times i \)
      by (auto elim!: partitionsE)
  }
from this have \( \sum p \{..n\} \leq (\sum i \leq n. \, p \, i \times i) \)
  by (auto intro: sum-mono)
also from assms have \( n \cdot (\sum i \leq n. \, p \, i \times i) = n \)
  by (auto elim!: partitionsE)
finally show \( ?thesis \).
qed

lemma finite-partitions:
  finite \{ \( p \). \( p \) partitions \( n \)\}
proof (—)
  have \( \{ \( p \). \( p \) partitions \( n \)\} \subseteq \{ f. \, (\forall i. \, f \, i \leq n) \land (\forall i. \, n + 1 \leq i \longrightarrow f \, i = 0)\}\}
    by (auto elim: partitions-bounds) (auto simp add: partitions-def)
from this bound-domain-and-range-impl-finitely-many-functions[of \( n \) \( n + 1 \)] show \( ?thesis \)
  by (simp add: finite-subset)
qed

lemma finite-partitions-k-parts:
  finite \{ \( p \). \( p \) partitions \( n \) \land \( \sum p \{..n\} = k \)\}
by (simp add: finite-partitions)

lemma partitions-remaining-Max-part:
  assumes \( p \) partitions \( n \)
  assumes \( 0 < p \, k \)
  shows \( \forall i. \, n - k < i \land i \neq k \longrightarrow p \, i = 0 \)
proof (clarify)
  fix \( i \)
  assume \( n - k < i \ i \neq k \)
  show \( p \, i = 0 \)
  proof (cases \( i \leq n \))
    assume \( i \leq n \)
    from assms have \( n \cdot (\sum i \leq n. \, p \, i \times i) = n \) and \( k \leq n \)
      by (auto elim: partitionsE)
have \( (\sum_{i \leq n} p_i \cdot i) = p_k \cdot k + (\sum_{i \in \{..n\} - \{k\}} p_i \cdot i) \)
using \( k \leq n \) sum-atMost-remove-nat by blast
also have \( \ldots = p_i \cdot i + p_k \cdot k + (\sum_{i \in \{..n\} - \{i, k\}} p_i \cdot i) \)
using \( i \leq n \) \( i \neq k \)
by (auto simp add: sum.remove[where \( x = i \)]) (metis Diff-insert)
finally have eq: \( (\sum_{i \leq n} p_i \cdot i) = p_i \cdot i + p_k \cdot k + (\sum_{i \in \{..n\} - \{i, k\}} p_i \cdot i) \).

\[ p \cdot i = 0 \]
proof (rule ccontr)
assume \( p \cdot i \neq 0 \)
have upper-bound: \( p_i \cdot i + p_k \cdot k \leq n \)
using eq n by auto
have lower-bound: \( p_i \cdot i + p_k \cdot k > n \)
using \( \langle n - k < i \rangle \langle 0 < p_k \rangle \langle k \leq n \rangle \langle p_i \neq 0 \rangle \) mult-eq-if not-less by auto
from upper-bound lower-bound show False by simp
qed
next
assume \( \neg (i \leq n) \)
from this show \( p \cdot i = 0 \)
using assms(1) by (auto elim: partitionsE)
qed

2.3 Operations of Number Partitions

lemma partitions-remove1-bounds:
assumes partitions: \( p \) partitions \( n \)
assumes gr0: \( 0 < p_k \)
assumes neq: \( (p(k := p_k - 1)) \cdot i \neq 0 \)
shows \( 1 \leq i \land i \leq n - k \)
proof
from partitions neq show \( 1 \leq i \)
by (auto elim!: partitionsE split: if-split-asm)
next
from partitions gr0 have n: \( (\sum_{i \leq n} p_i \cdot i) = n \) and \( k \leq n \)
by (auto elim: partitionsE)
show \( i \leq n - k \)
proof cases
assume \( k \leq n - k \)
from \( i \leq n - k \) neq show \(?thesis
using partitions-remaining-Max-part[OF partitions gr0] not-le by force
next
assume \( \neg k \leq n - k \)
from this have \( 2 \cdot k > n \) by auto
have \( p_k = 1 \)
proof (rule ccontr)
assume \( p_k \neq 1 \)
with gr0 have \( p_k \geq 2 \) by auto
from this have \( p_k \cdot k \geq 2 \cdot k \) by simp
with \(2 \ast k > n\) have \(p k \ast k > n\) by linarith
from \((k \leq n)\) this have \((\sum i \leq n. \ p i \ast i) > n\)
  by (simp add: sum-atMost-removeNat[of k])
from this n show False by auto
qed

from neq this show ?thesis
  using partitions.remaining-Max-part[OF partitions gr0] leI
by (auto split: if-split_asm) force
qed

lemma partitions-remove1:
  assumes partitions: \(p\) partitions \(n\)
  assumes gr0: \(0 < p k\)
  shows \(p(k := p k - 1)\) partitions \((n - k)\)
proof (rule partitionsI)
  fix \(i\)
  assume \((p(k := p k - 1)) i \neq 0\)
  from this show \(1 \leq i \&\& i \leq n - k\) using partitions.remove1-bounds partitions
  gr0 by blast
next
from partitions gr0 have \(k \leq n\) by (auto elim: partitionsE)
have \((\sum i \leq n - k. \ (p(k := p k - 1)) i \ast i) = (\sum i \leq n. \ (p(k := p k - 1)) i \ast i)\)
  using partitions.remove1-bounds partitions gr0 by (auto intro!: sum_mono_neutral_left)
also have \(\ldots = (p k - 1) \ast k + (\sum i \in\{..n\} - \{k\}. \ (p(k := p k - 1)) i \ast i)\)
  using \(k \leq n\) by (simp add: sum-atMost-removeNat[where \(k=k\)])
also have \(\ldots = p k \ast k + (\sum i \in\{..n\} - \{k\}. \ p i \ast i) - k\)
  using gr0 by (simp add: diff_mult_distrib)
also have \(\ldots = (\sum i \leq n. \ p i \ast i) - k\)
  using \(k \leq n\) by (simp add: sum-atMost-removeNat[of \(k\)])
also from partitions have \(\ldots = n - k\)
  by (auto elim: partitionsE)
finally show \((\sum i \leq n - k. \ (p(k := p k - 1)) i \ast i) = n - k\).
qed

lemma partitions-insert1:
  assumes \(p\): \(p\) partitions \(n\)
  assumes \(k > 0\)
  shows \((p(k := p k + 1))\) partitions \((n + k)\)
proof (rule partitionsI)
  fix \(i\)
  assume \((p(k := p k + 1)) i \neq 0\)
  from \(p\) this \((k > 0)\) show \(1 \leq i \&\& i \leq n + k\)
  by (auto elim!: partitionsE)
next
have \((\sum i \leq n + k. \ (p(k := p k + 1)) i \ast i) = p k \ast k + (\sum i \in\{..n + k\} - \{k\}. \ p i \ast i) + k\)
  by (simp add: sum-atMost-removeNat[of \(k\)])
also have \(\ldots = p k \ast k + (\sum i \in\{..n\} - \{k\}. \ p i \ast i) + k\)
proof  
also have \(\vdash (\sum\{i \leq n. p\ i\} + k)\)  
using \(p\) by (auto intro!: sum_mono_neutral_right elim!: partitionsE)  
also have \(\vdash \vdash p\ i\ i\)  
using \(p\) by (auto elim!: partitionsE)  
finally show \(\sum\{i \leq n + k. (p(k := p\ k + 1))\ i\ i\} = n + k\).
qed

lemma count-remove1:
assumes \(p\) partitions \(n\)
assumes \(0 < p\)
shows \(\sum\{i \leq n - k. (p(k := p\ k - 1))\ i\} = (\sum\{i \leq n. p\ i\} - 1)\)
proof  
have \(k \leq n\) using \(\vdash\) assms by (auto elim!: partitionsE)

have \(\sum\{i \leq n - k. (p(k := p\ k - 1))\ i\} = (\sum\{i \leq n. (p(k := p\ k - 1))\ i\}\)
using partitions-remove1-bounds assms by (auto intro!: sum_mono_neutral_left)
also have \(\sum\{i \leq n. (p(k := p\ k - 1))\ i\} = p\ k + (\sum\{i \in\{n\} - \{k\}. p\ i\} - 1)\)
using \(\vdash 0 < p\ k\) \(k \leq n\) by (simp add: sum_atMost_remove_nat[of \(k\)])
also have \(\vdash \vdash (\sum\{i \in\{n\}. p\ i\}) - 1\)
using \(\vdash\) assms by (auto intro!: sum_mono_neutral_right elim!: partitionsE)
also have \(\vdash \vdash (\sum\{i \leq n. p\ i\}) + 1\)
using assms by (cases \(k \leq n\) (auto simp add: sum_atMost_remove_nat[of \(k\)]
elim: partitionsE)
finally show \(?\)thesis .
qed

lemma count-insert1:
assumes \(p\) partitions \(n\)
shows \(\vdash \vdash \sum\{i \leq n + k. (p(k := p\ k + 1))\ i\} = (\sum\{i \leq n. p\ i\} + 1)\)
proof  
have \(\vdash \vdash \sum\{i \leq n + k. (p(k := p\ k + 1))\ i\} = p\ k + (\sum\{i \in\{n\} + k\} - \{k\}. p\ i\) + 1\)
by (simp add: sum_atMost_remove_nat[of \(k\)])
also have \(\vdash \vdash (\sum\{i \in\{n\} + k\} - \{k\}. p\ i\) + 1\)
using assms by (auto intro!: sum_mono_neutral_right elim!: partitionsE)
also have \(\vdash \vdash (\sum\{i \leq n. p\ i\}) + 1\)
using assms by (cases \(k \leq n\) (auto simp add: sum_atMost_remove_nat[of \(k\)]
elim: partitionsE)
finally show \(?\)thesis .
qed

lemma partitions-decrease1:
assumes \(p\): \(p\) partitions \(m\)
assumes \(\vdash\) sum\( p\ \{\ .. m\\} = k\)
assumes \(\vdash 0 = p\)
shows \(\lambda i. p\ (i + 1)\) partitions \(m - k\)
proof  
from \(p\) have \(\vdash 0 = 0\) by (auto elim!: partitionsE)
\{  
fix \(i\)
assume \(\vdash\) neq\( p\ (i + 1) \neq 0\)
from $p$ this ($p = 0$) have $1 \leq i$ 

by (fastforce elim!: partitionsE simp add: leSuc-eq)

moreover have $i \leq m - k$

proof (rule contr)

assume i-greater: $\neg i \leq m - k$

from $p$ have $s$: $(\sum i \leq m. \ p \ i \ i) = m$

by (auto elim!: partitionsE)

from $p$ sum have $k \leq m$

using partitions-parts-bounded by fastforce

from neq $p$ have $i + 1 \leq m$ by (auto elim!: partitionsE)

from i-greater have $i > m - k$ by simp

have ineq1: $i + 1 > (m - k) + 1$

using i-greater by simp

have ineq21: $(\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j * j) \geq (\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j$

using ($p = 0 = 0$): not-less by (fastforce intro!: sum-mono)

have ineq22a: $(\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j = (\sum j \leq m. \ p \ j) - 1$

using ($i + 1 \leq m$): neq by (simp add: sum.remove{where $x=i+1$})

have ineq22: $(\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j \geq k - 1$

using sum neq ineq22a by auto

have ineq2: $(\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j * j) \geq k - 1$

using ineq21 ineq22 by auto

have $(\sum i \leq m. \ p \ i \ i) = (p + (i + 1) - 1) * (i + 1) + (\sum i \in\{..m\} \ - \ {i + 1}. \ p \ i \ i)$

using ($i + 1 \leq m$): neq

by (subst sum.remove{where $x=i+1$}) auto

also have ... = $(i + 1) + (\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j * j)

using ($i + 1 \leq m$): neq

by (subst sum.remove{where $x=i+1$ and $g$==j. \ $p(i + 1 := p \ (i + 1) - 1) \ j * j) * j\})$

(auto simp add: mult-eq-if)

finally have $(\sum i \leq m. \ p \ i \ i) = i + 1 + (\sum j \leq m. \ (p(i + 1 := p \ (i + 1) - 1))) \ j * j) * j\)\)

moreover have ... > $m$ using ineq1 ineq2 \ ($k \leq m$) \ ($p \ i + 1$) \ neq 0\) by linarith

ultimately have $(\sum i \leq m. \ p \ i \ i) > m$ by simp

from $s$ this show $False$ by simp

qed

ultimately have $1 \leq i \wedge i \leq m - k$ ..

} note bounds = this

show $(\exists \ i. \ p \ (i + 1))$ partitions $m - k$

proof (rule partitionsI)

fix $i$

assume $p \ (i + 1) \ neq 0$

from bounds this show $1 \leq i \wedge i \leq m - k$ .

next

have geq: $\forall \ i. \ p \ i \ i \ geq p \ i$

using ($p \ 0 = 0$): not-less by fastforce

have $(\sum i \leq m - k. \ p \ (i + 1) * i) = (\sum i \leq m. \ p \ (i + 1) * i)$
lemma count-decrease1:
  assumes p: p partitions m
  assumes sum: sum p {..m} = k
  assumes p 1 = 0
  shows sum (λi. p (i + 1)) {..m - k} = k

proof
  have ... = (∑ i∈Suc ' {..m}. p i * (i - 1))
    by (auto simp add: sum.reindex)
  also have ... = (∑ i≤Suc m. p i * (i - 1))
    using p 0 = 0
    by (simp add: atMost-Suc-eq-insert-0 zero-notin-Suc-image)
  also have ... = (∑ i≤m. p i * (i - 1))
    using p by (auto elimination: partitionsE)
  also have ... = (∑ i≤m. p i * i) - (∑ i≤m. p i)
    using p by (simp only: sum_subtractf_nat)
  also have ... = m - k
    using sum p by (auto elimination: partitionsE)
  finally show (∑ i≤m - k. p (i + 1) * i) = m - k
    qed

lemma partitions-increase1:
  assumes partitions: p partitions m - k
  assumes k: sum p {..m - k} = k
  shows (λi. p (i - 1)) partitions m
proof (rule partitionsI)
  fix i
  assume p (i - 1) ≠ 0
  from partitions this k show 1 ≤ i ∧ i ≤ m
    by (cases k) (auto elimination: partitionsE)
next
  from k partitions have k ≤ m
    using linear partitions-zero by force
  have eq-0: ∀ i>m - k. p i = 0
    using partitions by (auto elimination: partitionsE)
  from partitions have s: (∑ i≤m - k. p i * i) = m - k
    by (auto elimination: partitionsE)
  have (∑ i≤m. p (i - 1) * i) = (∑ i≤Suc m. p (i - 1) * i)
    using partitions k
    by (cases k) (auto elimination: partitionsE)
  also have (∑ i≤Suc m. p (i - 1) * i) = (∑ i≤m. p i * (i + 1))
    by subst sum.atMost-Suc-shift
  also have ... = (∑ i≤m - k. p i * i + (∑ i≤m - k. p i))
    using eq-0
    by (auto intro: sum_mono_neutral_right)
  also have ... = m - k + k
    using s k by simp
  also have ... = m
    using k ≤ m by simp
  finally show (∑ i≤m. p (i - 1) * i) = m
    qed
proof 
-  
from p have p 0 = 0 by (auto elim!: partitionsE)
have sum (λi. p (i + 1)) {...m - k} = sum (λi. p (i + 1)) {...m}
  using partitions-decrease1[OF assms]
  by (auto intro: sum mono-neutral-left elim!: partitionsE)
also have ... = sum (λi. p (i + 1)) {...m} by (simp add: atLeastAtMost)
also have ... = sum (λi. p i) {Suc 0.. Suc m}
  by (simp only: One-nat-def add-Suc-right add-0-right sum.reindex-cong[of Suc])
  case 0
from this show thesis using (p 0 = 0) by simp
next
  case (Suc m')
  { fix x assume Suc 0 ≤ x x ≤ m
    from this Suc have x ∈ Suc ‘ {...m'}
      by (auto intro!: image-eqI[where x=x - 1])
  }
from this Suc show thesis
  by (intro sum.reindex-cong[of Suc]) auto
qed
also have (∑ i≤m-1. p i) = (∑ i≤m.. p i)
proof 
  { fix i
    assume 0 < p i i ≤ m
    from assms this have i ≤ m - 1
      using (p 0 = 0; partitions-increase1) by (cases k) (auto elim!: partitionsE)
  }
from this show thesis
  by (auto intro: sum mono-neutral-cong-left)
also have ... = (\(\sum_{i \leq m - k} p \cdot i\)) using partitions by (auto intro: sum.mononic-neutral-right elim!: partitionsE)
also have ... = k using k by auto
finally show theorem.

2.4 Number Partitions as Multisets on Natural Numbers

definition number-partition :: nat \Rightarrow nat multiset \Rightarrow bool
where
number-partition n N = (\(\sum\) mset N = n \land 0 \notin \# N)

2.4.1 Relationship to Definition on Functions

lemma count-partitions-iff:
\(\text{count N partitions n} \iff \text{number-partition n N}\)
proof
assume count N partitions n
from this have (\(\forall i. \text{count N i \neq 0} \rightarrow 1 \leq i \land i \leq n\)) (\(\sum_{i \leq n. \text{count N i} \cdot i}\)) = n
unfolding \(\text{Number-Partition.partitions-def}\) by auto
moreover from this have set-mset N \subseteq {..n} by auto
moreover have finite {..n} by auto
ultimately have sum-mset N = n
using sum-mset-sum-count sum-mset-eq-sum-on-supersets by presburger
moreover have 0 \notin \# N
using (\(\forall i. \text{count N i \neq 0} \rightarrow 1 \leq i \land i \leq n\)) by auto
ultimately show number-partition n N
unfolding number-partition-def by auto

next
assume number-partition n N
from this have sum-mset N = n \land 0 \notin \# N
unfolding number-partition-def by auto
{}
fix i
assume count N i \neq 0
have 1 \leq i \land i \leq n
proof
from (0 \notin \# N) (count N i \neq 0) show 1 \leq i
using Suc.le-eq by auto
from (sum-mset N = n) (count N i \neq 0) show i \leq n
using multi-member-split by fastforce
qed

moreover from (sum-mset N = n) have (\(\sum_{i \leq n. \text{count N i} \cdot i}\)) = n
by (metis atMost-iff calculation finite-atMost not-in-iff subsetI sum-mset-eq-sum-on-supersets sum-mset-sum-count)
ultimately show count N partitions n
by (rule partitionsI) auto
lemma partitions-iff-Abs-multiset:
p partitions n ←→ finite \{ x. 0 < p x \} ∧ number-partition n (Abs-multiset p)
proof
  assume p partitions n
  from this have bounds: (\forall i. p i \neq 0 \rightarrow 1 \leq i \land i \leq n)
  and sum: (\sum i \leq n. p i \ast i) = n
  unfolding partitions-def by auto
  from \langle p partitions n \rangle have p \in multiset by (rule partitions-imp-multiset)
  from \langle p partitions n \rangle have finite \{ x. 0 < p x \}
    by (rule partitions-imp-finite-elements)
  moreover from \langle p \in multiset \rangle bounds have \neg 0 \in# Abs-multiset p
  using count-eq-zero-iff by force
  moreover from \langle p \in multiset \rangle this sum have sum-mset (Abs-multiset p) = n
  proof
    have (\sum i \in \{ x. 0 < p x \}. p i \ast i) = (\sum i \leq n. p i \ast i)
      using bounds by (auto intro: sum_mono_neutral_cong_left)
    from \langle p \in multiset \rangle this sum show sum-mset (Abs-multiset p) = n
      by (simp add: sum-mset-sum_count set-mset-Abs-multiset)
  qed
  ultimately show finite \{ x. 0 < p x \} ∧ number-partition n (Abs-multiset p)
  unfolding number-partition-def by auto
next
  assume finite \{ x. 0 < p x \} ∧ number-partition n (Abs-multiset p)
  from this have finite \{ x. 0 < p x \} 0 \notin# Abs-multiset p
  sum-mset (Abs-multiset p) = n
  proof
    fix i
    assume p i \neq 0
    from \langle \neg 0 \in# Abs-multiset p \rangle \langle p \in multiset \rangle have p 0 = 0
      using count-inI by force
    from this \langle p i \neq 0 \rangle show 1 \leq i
      by (metis One_nat_def leI less-Suc0)
    show i \leq n
      proof (rule ccontr)
        assume \neg i \leq n
        from this have i > n
          using le_less_linear by blast
        from this \langle p i \neq 0 \rangle have p i \ast i > n
          by (auto simp add: less_le_trans)
        from \langle p i \neq 0 \rangle have (\sum i \in \{ x. 0 < p x \}. p i \ast i) = p i \ast i + (\sum i \in \{ x. 0 < p x \} - \{ i \}. p i \ast i)
using \( \{x. 0 < p \times x \} \)
by (simp add: subst sum.insert-remove[symmetric])
also from \( \{p \times i > n\} \) have \( \ldots > n \) by auto
finally show False using \( \{\sum i \in \{x. 0 < p \times x\}. p \times i = n\} \) by blast
qed

moreover have \( \{\sum i \leq n. p \times i\} \) = \( n \)
proof -
  have \( \{\sum i \leq n. p \times i\} = \{\sum i \in \{x. 0 < p \times x\}. p \times i\} \)
  using bounds by (auto intro: sum.mono-neutral-cong-right)
  from this show \(?thesis \)
  using \( \{\sum i \in \{x. 0 < p \times x\}. p \times i = n\} \) by simp
qed

ultimately show \( p \) partitions \( n \) by (auto intro: partitionsI)
qed

lemma size-nat-multiset-eq:
  fixes \( N :: \text{nat multiset} \)
  assumes number-partition \( n N \)
  shows \( \text{size} N = \text{sum} \ (\text{count} N) \{..n\} \)
proof -
  have set-mset \( N \subseteq \{..\text{sum-mset} N\} \)
    by (auto dest: multi-member-split)
  have \( \text{size} N = \text{sum} \ (\text{count} N) \ (\text{set-mset} N) \)
    by (rule size-multiset-overloaded-eq)
  also have \( \ldots = \text{sum} \ (\text{count} N) \ \{..\text{sum-mset} N\} \)
    using \( \{\text{set-mset} N \subseteq \{..\text{sum-mset} N\}\} \)
    by (auto intro: sum.mono-neutral-cong-left count-inI)
  finally show \(?thesis \)
    using \( \{\text{number-partition} n N\} \)
    unfolding number-partition-def by auto
qed

end

3  Cardinality of Number Partitions

theory Card-Number-Partitions
imports Number-Partition
begin

3.1  The Partition Function

fun \( \text{Partition} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \)
where
  \( \text{Partition} \ 0 \ 0 = 1 \)
| \( \text{Partition} \ 0 \ (\text{Suc} \ k) = 0 \)
| \( \text{Partition} \ (\text{Suc} \ m) \ 0 = 0 \)
\[\text{Partition} (\text{Suc } m) (\text{Suc } k) = \text{Partition} m k + \text{Partition} (m - k) (\text{Suc } k)\]

**Lemma Partition-less:**
- Assumes \(m < k\)
- Shows \(\text{Partition } m k = 0\)
- Using assms by (induct \(m \ k\) rule: Partition.induct) auto

**Lemma Partition-sum-Partition-diff:**
- Assumes \(k \leq m\)
- Shows \(\text{Partition } m k = (\sum_{i \leq k} \text{Partition} (m - k) i)\)
- Using assms by (induct \(m \ k\) rule: Partition.induct) auto

**Lemma Partition-parts1:**
- \(\text{Partition} (\text{Suc } m) (\text{Suc } 0) = 1\)
- By (induct \(m\)) auto

**Lemma Partition-diag:**
- \(\text{Partition} (\text{Suc } m) (\text{Suc } m) = 1\)
- By (induct \(m\)) auto

**Lemma Partition-diag1:**
- \(\text{Partition} (\text{Suc } (\text{Suc } m)) (\text{Suc } m) = 1\)
- By (induct \(m\)) auto

**Lemma Partition-parts2:**
- Shows \(\text{Partition} m 2 = m \div 2\)
- Proof (induct \(m\) rule: nat-less-induct)
  - Fix \(m\)
  - Assume hypothesis: \(\forall n<m. \text{Partition } n 2 = n \div 2\)
  - Have \((m = 0 \lor m = 1) \lor m \geq 2\) by auto
  - From this show \(\text{Partition } m 2 = m \div 2\)
  - Proof
    - Assume \(m = 0 \lor m = 1\)
    - From this show ?thesis by (auto simp add: numerals(2))
  - Next
    - Assume \(2 \leq m\)
    - From this obtain \(m'\) where \(m' = \text{Suc} (\text{Suc } m)\) by (metis add-2-eq-Suc le-Suc-ex)
    - From hypothesis this have \(\text{Partition } m' 2 = m' \div 2\) by simp
    - From this \(m'\) show ?thesis using \(\text{Partition-parts1} \text{Partition.simps(4)}[\text{of Suc } m' \text{ Suc } 0]\) \(\text{div2-Suc-Suc}\)
    - By (simp add: numerals(2) del: \text{Partition.simps})
  - Qed
- Qed

### 3.2 Cardinality of Number Partitions

**Lemma set-rewrite1:**
- \(\{p. \ p \text{ partitions } \text{Suc } m \land \text{sum } p \{..\text{Suc } m\} = \text{Suc } k \land p \ 1 \neq 0\}\)
\[
(p \cdot p(1 := p 1 + 1)) \cdot \{p \cdot p \text{ partitions } m \land \text{sum } p \{..m\} = k\} \quad (\text{is } ?S = \ ?T)
\]

proof

\{ 
  \begin{align*}
  \text{fix } p \\
  \text{assume } \text{assms: } p \text{ partitions } \text{Suc } m \text{ sum } p \{..\text{Suc } m\} = \text{Suc } k \land p 1 < p 1 \\
  \text{have } p(1 := p 1 - 1) \text{ partitions } m \\
  \text{using } \text{assms by } (\text{metis partitions-remove1 diff-Suc-1}) \\
  \text{moreover have } \sum_{i \leq m.} (p(1 := p 1 - 1)) i = k \\
  \text{using } \text{assms by } (\text{metis count-remove1 diff-Suc-1}) \\
  \text{ultimately have } p(1 := p 1 - 1) \in \{p \cdot p \text{ partitions } m \land \text{sum } p \{..m\} = k\} \\
  \text{by simp} \\
  \text{moreover have } p = p(1 := p 1 - 1, 1 := (p(1 := p 1 - 1)) 1 + 1) \\
  \text{using } (0 < p 1) \text{ by auto} \\
  \text{ultimately have } p \in (\lambda p. p(1 := p 1 + 1)) \cdot \{p \cdot p \text{ partitions } m \land \text{sum } p \{..m\} = k\} \text{ by blast} \\
\end{align*}
\}

from this show \( ?S \subseteq ?T \) by blast

next

\{ 
  \begin{align*}
  \text{fix } p \\
  \text{assume } \text{assms: } p \text{ partitions } m \text{ sum } p \{..m\} = k \\
  \text{have } p(1 := p 1 + 1) \text{ partitions } \text{Suc } m \quad (\text{is } ?g1) \\
  \text{using } \text{assms by } (\text{metis partitions-insert1 Suc-eq-plus1 zero-less-one}) \\
  \text{moreover have } \text{sum } p(1 := p 1 + 1) \{..\text{Suc } m\} = \text{Suc } k \quad (\text{is } ?g2) \\
  \text{using } \text{assms by } (\text{metis count-insert1 Suc-eq-plus1}) \\
  \text{moreover have } p(1 := p 1 + 1) 1 \neq 0 \quad (\text{is } ?g3) \text{ by auto} \\
  \text{ultimately have } ?g1 \land ?g2 \land ?g3 \text{ by simp} \\
\end{align*}
\}

from this show \( ?T \subseteq ?S \) by auto

qed

lemma \text{set-rewrite2:}

\{p \cdot p \text{ partitions } m \land \text{sum } p \{..m\} = k \land p 1 = 0\}

= (\lambda p. (\lambda i. p (i - 1))) \cdot \{p \cdot p \text{ partitions } (m - k) \land \text{sum } p \{..m - k\} = k\}

(is \( ?S = ?T \))

proof

\{ 
  \begin{align*}
  \text{fix } p \\
  \text{assume } \text{assms: } p \text{ partitions } m \text{ sum } p \{..m\} = k \land p 1 = 0 \\
  \text{have } (\lambda i. p (i + 1)) \text{ partitions } m - k \\
  \text{using } \text{assms partitions-decrease1 by blast} \\
  \text{moreover from } \text{assms have } \text{sum } (\lambda i. p (i + 1)) \{..m - k\} = k \\
  \text{using } \text{assms count-decrease1 by blast} \\
  \text{ultimately have } (\lambda i. p (i + 1)) \in \{p \cdot p \text{ partitions } m - k \land \text{sum } p \{..m - k\} = k\} \text{ by simp} \\
  \text{moreover have } p = (\lambda i. p ((i - 1) + 1)) \\
  \text{proof } (\text{rule ext}) \\
  \text{fix } i \text{ show } p = p (i - 1 + 1) \\
\end{align*}
\}
using assms by (cases i) (auto elim: partitionsE)

qed
ultimately have \( p \in (\lambda p. (\lambda i. p\ (i - 1))) \cdot \{ p. p \text{ partitions } m - k \land \text{sum } p\ \{..m - k\} = k\} \) by auto

from this show \( ?S \subseteq ?T \) by auto

next

{ 
  fix \( p \)
  assume assms: \( p \text{ partitions } m - k \land \text{sum } p\ \{..m - k\} = k\)
  from assms have \( (\lambda i. p\ (i - 1)) \text{ partitions } m \) (is \( ?g1 \))
    using partitions-increase1 by blast
  moreover from assms have \( \sum i \leq m. p\ (i - 1) = k \) (is \( ?g2 \))
    using count-increase1 by blast
  moreover from assms have \( p 0 = 0 \) (is \( ?g3 \))
    by (auto elim!: partitionsE)
  ultimately have \( ?g1 \land ?g2 \land ?g3 \) by simp
}

from this show \( ?T \subseteq ?S \) by auto

qed

theorem card-partitions-k-parts:
  \( \text{card } \{ p. p \text{ partitions } n \land (\sum i \leq n. p i) = k\} = \text{Partition } n k \)
proof (induct n k rule: Partition.induct)
  case 1
  have eq: \( \{ p. p = (\lambda x. 0) \land p 0 = 0\} = \{ (\lambda x. 0)\} \) by auto
  show \( \text{card } \{ p. p \text{ partitions } 0 \land \text{sum } p\ \{..0\} = 0\} = \text{Partition } 0 0 \)
    by (simp add: partitions-zero eq)

next
  case (2 k)
  have eq: \( \{ p. p = (\lambda x. 0) \land p 0 = \text{Suc } k\} = \{\} \) by auto
  show \( \text{card } \{ p. p \text{ partitions } 0 \land \text{sum } p\ \{..\text{Suc } k\} = \text{Suc } k\} = \text{Partition } 0 \ (\text{Suc } k) \)
    by (simp add: partitions-zero eq)

next
  case (3 m)
  have eq: \( \{ p. p \text{ partitions } \text{Suc } m \land \text{sum } p\ \{..\text{Suc } m\} = 0\} = \{\} \)
    by (fastforce elim!: partitionsE simp add: le-Suc-eq)
  from this show \( \text{card } \{ p. p \text{ partitions } \text{Suc } m \land \text{sum } p\ \{..\text{Suc } m\} = 0\} = \text{Partition } (\text{Suc } m) 0 \)
    by (simp only: Partition.simps card-empty)

next
  case (4 m k)
  let \( ?set1 = \{ p. p \text{ partitions } \text{Suc } m \land \text{sum } p\ \{..\text{Suc } m\} = \text{Suc } k \land p\ 1 \neq 0\} \)
  let \( ?set2 = \{ p. p \text{ partitions } \text{Suc } m \land \text{sum } p\ \{..\text{Suc } m\} = \text{Suc } k \land p\ 1 = 0\} \)
  have \( \text{finite } \{ p. p \text{ partitions } \text{Suc } m\} \)
    by (simp add: finite-partitions)
  from this have \( \text{finite-sets: finite } ?set1 \text{ finite } ?set2 \) by simp+
  have \( \text{set-eq: } \{ p. p \text{ partitions } \text{Suc } m \land \text{sum } p\ \{..\text{Suc } m\} = \text{Suc } k\} = ?set1 \cup ?set2 \by simp \)
    by auto
have disjoint: \(?set1 \cap \?set2 = \{\}\) by auto

have inj1: inj-on (\(\lambda p. p(1 := p + 1)\)) \(\{p.\) p partitions m \& sum p \{..m\} = k\}
  by (auto intro!: inj-onI) (metis diff-Suc-1 fun-upd-idem-iff fun-upd-upd)

have inj2: inj-on (\(\lambda p. p (i - 1)\)) \(\{p.\) p partitions m - k \& sum p \{..m - k\} = Suc k\}
  by (auto intro!: inj-onI simp add: fun-eq-iff) (metis add-cancel-right)

have card1: card \(?set1 = Partition m \ k\)
  using inj1 4(1) by (simp only: set-rev rewrite1 card-image)

have card2: card \(?set2 = Partition (m - \ k\) (Suc k\)
  using inj2 4(2) by (simp only: set-rev rewrite2 card-image diff-Suc-Suc)

have card \(\{p.\) p partitions Suc m \& sum p \{..Suc m\} = Suc k\} = Partition m \ k + Partition (m - \ k\) (Suc k\)
  using finite-sets disjoint by (simp only: set-eq Un disjoint card1 card2)

from this show card \(\{p.\) p partitions Suc m \& sum p \{..Suc m\} = Suc k\} = Partition (Suc m\) (Suc k\)
  by auto

qed

theorem card-partitions:
  card \(\{p.\) p partitions n\} = (\(\sum k \leq n.\) Partition n \ k\)

proof –
  have seteq: \(\{p.\) p partitions n\} = \(\bigcup((\lambda k. \{p.\) p partitions n \& (\(\sum i \leq n.\) p i) = k\}) \cdot \{..n\})\)
    by (auto intro: partitions-parts-bounded)
  have finite: \(?k. finite \{p.\) p partitions n \& sum p \{..n\} = k\}
    by (simp add: finite-partitions)
  have card \(\{p.\) p partitions n\} = card \((\bigcup((\lambda k. \{p.\) p partitions n \& (\(\sum i \leq n.\) p i) = k\}) \cdot \{..n\})\)
    using finite by (simp add: seteq)
  also have \(\sum x \leq n.\) card \(\{p.\) p partitions n \& sum p \{..n\} = x\}
    using finite by (subst card-UN-disjoint) auto
  also have \(\sum k \leq n.\) Partition n \ k\)
    by (simp add: card-partitions-k-parts)
  finally show \(?thesis\).

qed

lemma card-partitions-atmost-k-parts:
  card \(\{p.\) p partitions n \& sum p \{..n\} \leq k\} = Partition (n + \ k\)

proof –
  have card \(\{p.\) p partitions n \& sum p \{..n\} \leq k\) =
    card \((\bigcup((\lambda k'. \{p.\) p partitions n \& sum p \{..n\} = k'\}) \cdot \{..k\}))\)
    proof –
      have \(\{p.\) p partitions n \& sum p \{..n\} \leq k\) =
        \((\bigcup k' \leq k. \{p.\) p partitions n \& sum p \{..n\} = k'\}) by auto
      from this show \(?thesis\) by simp
    qed
  also have card \((\bigcup((\lambda k'. \{p.\) p partitions n \& sum p \{..n\} = k'\}) \cdot \{..k\})) =
    sum (\(\lambda k'.\) card \(\{p.\) p partitions n \& sum p \{..n\} = k'\}) \{..k\}\)
    using finite-partitions-k-parts by (subst card-UN-disjoint) auto
also have \( \sum (\lambda k'. \text{Partition } n \ k') \{..k\} \)
using \text{card-partitions-k-parts} by \text{simp}
also have \( \text{Partition } (n + k) \ k \)
using \text{Partition-sum-Partition-diff} by \text{simp}
finally show \( \text{thesis} \).
qed

3.3 Cardinality of Number Partitions as Multisets of Natural Numbers

\text{lemma bij-betw-multiset-number-partition-with-size}:
\text{bij-betw count } \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N = k\} \ \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} = k\}
\text{proof (rule bij-betw-byWitness[where } f' = \text{Abs-multiset}]\}
\text{show } \forall N \in \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N = k\}. \ \text{Abs-multiset } (\text{count } N) = N
\quad \text{using count-inverse by blast}
\text{show } \forall p \in \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} = k\}. \ \text{count } (\text{Abs-multiset } p) = p
\quad \text{by (auto simp add: multiset-def partitions-imp-finite-elements)}
\text{show count' } \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N = k\} \subseteq \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} = k\}
\quad \text{by (auto simp add: count-partitions-iff size-nat-multiset-eq)}
\text{show Abs-multiset' } \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} = k\} \subseteq \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N = k\}
\quad \text{using partitions-iff-Abs-multiset size-nat-multiset-eq partitions-imp-multiset by fastforce}
\text{qed}

\text{lemma bij-betw-multiset-number-partition-with-atmost-size}:
\text{bij-betw count } \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N \leq k\} \ \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} \leq k\}
\text{proof (rule bij-betw-byWitness[where } f' = \text{Abs-multiset}]\}
\text{show } \forall N \in \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N \leq k\}. \ \text{Abs-multiset } (\text{count } N) = N
\quad \text{using count-inverse by blast}
\text{show } \forall p \in \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} \leq k\}. \ \text{count } (\text{Abs-multiset } p) = p
\quad \text{by (auto simp add: multiset-def partitions-imp-finite-elements)}
\text{show count' } \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N \leq k\} \subseteq \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} \leq k\}
\quad \text{by (auto simp add: count-partitions-iff size-nat-multiset-eq)}
\text{show Abs-multiset' } \{p. \ p \ \text{partitions } n \ \wedge \ \text{sum } p \ \{..n\} \leq k\} \subseteq \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N \leq k\}
\quad \text{using partitions-iff-Abs-multiset size-nat-multiset-eq partitions-imp-multiset by fastforce}
\text{qed}

\text{theorem card-number-partitions-with-atmost-k-parts}:
\text{shows card } \{N. \text{number-partition } n \ N \ \wedge \ \text{size } N \leq x\} = \text{Partition } (n + x) \ x
\text{proof – –}
have bij-betw count \{N. number-partition n N \land size N \leq x\} \{p. p partitions n \land sum p \{..n\} \leq x\}
  by (rule bij-betw-multiset-number-partition-with-atmost-size)
from this have card \{N. number-partition n N \land size N \leq x\} = card \{p. p partitions n \land sum p \{..n\} \leq x\}
  by (rule bij-betw-same-card)
also have card \{p. p partitions n \land sum p \{..n\} \leq x\} = \text{Partition} (n + x) x
  by (rule card-partitions-atmost-k-parts)
finally show thesis .
qed

theorem card-partitions-with-k-parts:
  card \{N. number-partition n N \land size N = k\} = \text{Partition} n k
proof
  have bij-betw count \{N. number-partition n N \land size N = k\} \{p. p partitions n \land sum p \{..n\} = k\}
    by (rule bij-betw-multiset-number-partition-with-size)
  from this have card \{N. number-partition n N \land size N = k\} = card \{p. p partitions n \land sum p \{..n\} = k\}
    by (rule bij-betw-same-card)
  also have \ldots = \text{Partition} n k by (rule card-partitions-k-parts)
finally show thesis .
qed

3.4 Cardinality of Number Partitions with only 1-parts

lemma number-partition1-eq-replicate-mset:
  \{N. (\forall n. n \in \# N \rightarrow n = 1) \land number-partition n N\} = \{\text{replicate-mset n 1}\}
proof
  show \{N. (\forall n. n \in \# N \rightarrow n = 1) \land number-partition n N\} \subseteq \{\text{replicate-mset n 1}\}
proof
    fix N
    assume N: N \in \{N. (\forall n. n \in \# N \rightarrow n = 1) \land number-partition n N\}
    have N = \text{replicate-mset n 1}
      proof (rule multiset-eqI)
        fix i
        have count N i = sum-mset N
          proof cases
            assume N = \#
              from this show thesis by auto
          next
            assume N \neq \#
              from this N have 1 \in \# N by blast
              from this N show thesis
                by (auto simp add: sum-mset-sum-count sum.remove[where x=1] simp del: One-nat-def)
          qed
        from N this show count N i = count (replicate-mset n 1) i
unfolding number-partition-def by (auto intro: count-inI)

qed from this show \( N \in \{\text{replicate-mset } n 1\} \) by simp

qed

next

show \( \{\text{replicate-mset } n 1\} \subseteq \{N. (\forall n. n \in \# N \rightarrow n = 1) \land \text{number-partition } n N\} \)

unfolding number-partition-def by auto

qed

lemma card-number-partitions-with-only-parts-1-eq-1:
  assumes \( n \leq x \)
  shows \( \text{card } \{N. (\forall n. n \in \# N \rightarrow n = 1) \land \text{number-partition } n N \land \text{size } N \leq x\} = 1 \) (is \( \text{card } ?N = -\))

proof –
  have \( \forall N \in \{N. (\forall n. n \in \# N \rightarrow n = 1) \land \text{number-partition } n N\}. \text{size } N = n \)

unfolding number-partition1-eq-replicate-mset by simp

from this number-partition1-eq-replicate-mset \( \langle n \leq x \rangle \) have \( ?N = \{\text{replicate-mset } n 1\} \) by auto

from this show \( ?\text{thesis} \) by simp

qed

lemma card-number-partitions-with-only-parts-1-eq-0:
  assumes \( x < n \)
  shows \( \text{card } \{N. (\forall n. n \in \# N \rightarrow n = 1) \land \text{number-partition } n N \land \text{size } N \leq x\} = 0 \) (is \( \text{card } ?N = -\))

proof –
  have \( \forall N \in \{N. (\forall n. n \in \# N \rightarrow n = 1) \land \text{number-partition } n N\}. \text{size } N = n \)

unfolding number-partition1-eq-replicate-mset by simp

from this number-partition1-eq-replicate-mset \( \langle x < n \rangle \) have \( ?N = \{\} \) by auto

from this show \( ?\text{thesis} \) by (simp only: card-empty)

qed

end

References
