

Fun With Tilings

Tobias Nipkow and Lawrence Paulson

December 12, 2009

Abstract

Tilings are defined inductively. It is shown that one form of mutilated chess board cannot be tiled with dominoes, while another one can be tiled with L-shaped tiles.

Sections 1 and 2 are by Paulson and described elsewhere [1]. Section 3 is by Nipkow and formalizes a well-known argument from the literature [2].

Please add further fun examples of this kind!

theory *Tilings* **imports** *Main* **begin**

1 Inductive Tiling

inductive-set

tiling :: 'a set set \Rightarrow 'a set set

for *A* :: 'a set set **where**

empty [*simp*, *intro*]: $\{\} \in \text{tiling } A$ |

Un [*simp*, *intro*]: $\llbracket a \in A; t \in \text{tiling } A; a \cap t = \{\} \rrbracket$
 $\implies a \cup t \in \text{tiling } A$

lemma *tiling-UnI* [*intro*]:

$\llbracket t \in \text{tiling } A; u \in \text{tiling } A; t \cap u = \{\} \rrbracket \implies t \cup u \in \text{tiling } A$

apply (*induct set: tiling*)

apply (*auto simp add: Un-assoc*)

done

lemma *tiling-Diff1E*:

assumes $t - a \in \text{tiling } A$ **and** $a \in A$ **and** $a \subseteq t$

shows $t \in \text{tiling } A$

proof –

from *assms(2-3)* **have** $\exists r. t = r \cup a \ \& \ r \cap a = \{\}$

by (*metis Diff-disjoint Int-commute Un-Diff-cancel Un-absorb1 Un-commute*)

thus *?thesis* **using** *assms(1,2)*

by (*auto simp: Un-Diff*)

(*metis Compl-Diff-eq Diff-Compl Diff-empty Int-commute Un-Diff-cancel*)

Un-commute double-complement tiling.Un)

qed

lemma *tiling-finite*:

assumes $\bigwedge a. a \in A \implies \text{finite } a$

shows $t \in \text{tiling } A \implies \text{finite } t$

apply (*induct set: tiling*)

using *assms apply auto*

done

2 The Mutilated Chess Board Cannot be Tiled by Dominoes

The originator of this problem is Max Black, according to J A Robinson. It was popularized as the *Mutilated Checkerboard Problem* by J McCarthy.

inductive-set *domino* :: $(\text{nat} \times \text{nat}) \text{ set set}$ **where**

horiz [*simp*]: $\{(i, j), (i, \text{Suc } j)\} \in \text{domino} \mid$

vertl [*simp*]: $\{(i, j), (\text{Suc } i, j)\} \in \text{domino}$

lemma *domino-finite*: $d \in \text{domino} \implies \text{finite } d$

by (*erule domino.cases, auto*)

declare *tiling-finite*[*OF domino-finite, simp*]

Sets of squares of the given colour

definition

coloured :: $\text{nat} \Rightarrow (\text{nat} \times \text{nat}) \text{ set}$ **where**

coloured $b = \{(i, j). (i + j) \bmod 2 = b\}$

abbreviation

whites :: $(\text{nat} \times \text{nat}) \text{ set}$ **where**

whites $\equiv \text{coloured } 0$

abbreviation

blacks :: $(\text{nat} \times \text{nat}) \text{ set}$ **where**

blacks $\equiv \text{coloured } (\text{Suc } 0)$

Chess boards

lemma *Sigma-Suc1* [*simp*]:

$\{0..< \text{Suc } n\} \times B = (\{n\} \times B) \cup (\{0..<n\} \times B)$

by *auto*

lemma *Sigma-Suc2* [*simp*]:

$A \times \{0..< \text{Suc } n\} = (A \times \{n\}) \cup (A \times \{0..<n\})$

by *auto*

lemma *dominoes-tile-row* [*intro!*]: $\{i\} \times \{0..< 2*n\} \in \text{tiling } \text{domino}$

apply (*induct n*)
apply (*simp-all del:Un-insert-left add: Un-assoc [symmetric]*)
done

lemma *dominoes-tile-matrix*: $\{0..<m\} \times \{0..< 2*n\} \in \text{tiling domino}$
by (*induct m*) *auto*

coloured and Dominoes

lemma *coloured-insert* [*simp*]:
coloured b \cap (*insert (i, j) t*) =
 (*if (i + j) mod 2 = b then insert (i, j) (coloured b* \cap *t)*
else coloured b \cap *t*)
by (*auto simp add: coloured-def*)

lemma *domino-singletons*:
d \in *domino* \implies
 ($\exists i j. \text{whites} \cap d = \{(i, j)\}$) \wedge
 ($\exists m n. \text{blacks} \cap d = \{(m, n)\}$)
apply (*erule domino.cases*)
apply (*auto simp add: mod-Suc*)
done

Tilings of dominoes

declare
Int-Un-distrib [*simp*]
Diff-Int-distrib [*simp*]

lemma *tiling-domino-0-1*:
t \in *tiling domino* $\implies \text{card}(\text{whites} \cap t) = \text{card}(\text{blacks} \cap t)$
apply (*induct set: tiling*)
apply (*drule-tac [2] domino-singletons*)
apply (*auto*)
apply (*subgoal-tac* $\forall p C. C \cap a = \{p\} \implies p \notin t$)
 — this lemma tells us that both “inserts” are non-trivial
apply (*simp (no-asm-simp)*)
apply *blast*
done

Final argument is surprisingly complex

theorem *gen-mutil-not-tiling*:
t \in *tiling domino* \implies
 (*i + j mod 2 = 0* $\implies (m + n) \text{ mod } 2 = 0 \implies$
 $\{(i, j), (m, n)\} \subseteq t$
 $\implies (t - \{(i, j)\} - \{(m, n)\}) \notin \text{tiling domino}$)
apply (*rule notI*)
apply (*subgoal-tac*
 $\text{card}(\text{whites} \cap (t - \{(i, j)\} - \{(m, n)\})) <$
 $\text{card}(\text{blacks} \cap (t - \{(i, j)\} - \{(m, n)\}))$)

apply (*force simp only: tiling-domino-0-1*)
apply (*simp add: tiling-domino-0-1 [symmetric]*)
apply (*simp add: coloured-def card-Diff2-less*)
done

Apply the general theorem to the well-known case

theorem *mutil-not-tiling*:
 $t = \{0..< 2 * \text{Suc } m\} \times \{0..< 2 * \text{Suc } n\}$
 $\implies t - \{(0,0)\} - \{(\text{Suc}(2 * m), \text{Suc}(2 * n))\} \notin \text{tiling domino}$
apply (*rule gen-mutil-not-tiling*)
apply (*blast intro!: dominoes-tile-matrix*)
apply *auto*
done

3 The Mutilated Chess Board Can be Tiled by Ls

Remove a arbitrary square from a chess board of size $2^n \times 2^n$. The result can be tiled by L-shaped tiles: \boxplus . The four possible L-shaped tiles are obtained by dropping one of the four squares from $\{(x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1)\}$:

definition *L2* ($x::\text{nat}$) ($y::\text{nat}$) = $\{(x,y), (x+1,y), (x, y+1)\}$
definition *L3* ($x::\text{nat}$) ($y::\text{nat}$) = $\{(x,y), (x+1,y), (x+1, y+1)\}$
definition *L0* ($x::\text{nat}$) ($y::\text{nat}$) = $\{(x+1,y), (x,y+1), (x+1, y+1)\}$
definition *L1* ($x::\text{nat}$) ($y::\text{nat}$) = $\{(x,y), (x,y+1), (x+1, y+1)\}$

All tiles:

definition *Ls* :: ($\text{nat} * \text{nat}$) *set set* **where**
 $Ls \equiv \{ L0 \ x \ y \mid x \ y. \ \text{True} \} \cup \{ L1 \ x \ y \mid x \ y. \ \text{True} \} \cup$
 $\{ L2 \ x \ y \mid x \ y. \ \text{True} \} \cup \{ L3 \ x \ y \mid x \ y. \ \text{True} \}$

lemma *LinLs*: $L0 \ i \ j : Ls \ \& \ L1 \ i \ j : Ls \ \& \ L2 \ i \ j : Ls \ \& \ L3 \ i \ j : Ls$
by(*fastsimp simp:Ls-def*)

Square $2^n \times 2^n$ grid, shifted by i and j :

definition *square2* ($n::\text{nat}$) ($i::\text{nat}$) ($j::\text{nat}$) = $\{i..< 2^{\wedge}n+i\} \times \{j..< 2^{\wedge}n+j\}$

lemma *in-square2*[*simp*]:
 $(a,b) : \text{square2 } n \ i \ j \iff i \leq a \wedge a < 2^{\wedge}n+i \wedge j \leq b \wedge b < 2^{\wedge}n+j$
by(*simp add:square2-def*)

lemma *square2-Suc*: $\text{square2 } (\text{Suc } n) \ i \ j =$
 $\text{square2 } n \ i \ j \cup \text{square2 } n \ (2^{\wedge}n + i) \ j \cup \text{square2 } n \ i \ (2^{\wedge}n + j) \cup$
 $\text{square2 } n \ (2^{\wedge}n + i) \ (2^{\wedge}n + j)$
by(*auto simp:square2-def*)

lemma *square2-disj*: $\text{square2 } n \ i \ j \cap \text{square2 } n \ x \ y = \{\} \iff$
 $(2^{\wedge}n+i \leq x \vee 2^{\wedge}n+x \leq i) \vee (2^{\wedge}n+j \leq y \vee 2^{\wedge}n+y \leq j)$ (**is** $?A = ?B$)

proof –

```

{ assume ?B hence ?A by(auto simp:square2-def) }
moreover
{ assume  $\neg$  ?B
  hence (max i x, max j y) : square2 n i j  $\cap$  square2 n x y by simp
  hence  $\neg$  ?A by blast }
ultimately show ?thesis by blast
qed

```

Some specific lemmas:

```

lemma pos-pow2: (0::nat) < 2^(n::nat)
by simp

```

```

declare nat-zero-less-power-iff[simp del] zero-less-power[simp del]

```

```

lemma Diff-insert-if: shows
  B  $\neq$  {}  $\implies$  a:A  $\implies$  A - insert a B = (A-B - {a}) and
  B  $\neq$  {}  $\implies$  a ~: A  $\implies$  A - insert a B = A-B
by auto

```

```

lemma DisjI1: A Int B = {}  $\implies$  (A-X) Int B = {}
by blast

```

```

lemma DisjI2: A Int B = {}  $\implies$  A Int (B-X) = {}
by blast

```

The main theorem:

```

declare [[max-clauses = 200]]

```

```

theorem Ls-can-tilde: i  $\leq$  a  $\implies$  a < 2^n + i  $\implies$  j  $\leq$  b  $\implies$  b < 2^n + j
 $\implies$  square2 n i j - {(a,b)} : tiling Ls

```

```

proof(induct n arbitrary: a b i j)

```

```

  case 0 thus ?case by (simp add:square2-def)

```

```

next

```

```

  case (Suc n) note IH = Suc(1) and a = Suc(2-3) and b = Suc(4-5)

```

```

  hence a < 2^n+i  $\wedge$  b < 2^n+j  $\vee$ 

```

```

    2^n+i  $\leq$  a  $\wedge$  a < 2^(n+1)+i  $\wedge$  b < 2^n+j  $\vee$ 

```

```

    a < 2^n+i  $\wedge$  2^n+j  $\leq$  b  $\wedge$  b < 2^(n+1)+j  $\vee$ 

```

```

    2^n+i  $\leq$  a  $\wedge$  a < 2^(n+1)+i  $\wedge$  2^n+j  $\leq$  b  $\wedge$  b < 2^(n+1)+j (is ?A|?B|?C|?D)

```

```

  by simp arith

```

```

moreover

```

```

{ assume ?A

```

```

  hence square2 n i j - {(a,b)} : tiling Ls using IH a b by auto

```

```

moreover have square2 n (2^n+i) j - {(2^n+i, 2^n+j - 1)} : tiling Ls

```

```

  by(rule IH)(insert pos-pow2[of n], auto)

```

```

moreover have square2 n i (2^n+j) - {(2^n+i - 1, 2^n+j)} : tiling Ls

```

```

  by(rule IH)(insert pos-pow2[of n], auto)

```

```

moreover have square2 n (2^n+i) (2^n+j) - {(2^n+i, 2^n+j)} : tiling Ls

```

```

  by(rule IH)(insert pos-pow2[of n], auto)

```

```

ultimately

```

```

have square2 (n+1) i j - {(a,b)} - L0 (2^n+i - 1) (2^n+j - 1)  $\in$  tiling

```

```

Ls

```

```

    using a b (?A)
    by (clarsimp simp: square2-Suc L0-def Un-Diff Diff-insert-if)
      (fastsimp intro!: tiling-UnI DisjI1 DisjI2 square2-disj[THEN iffD2]
        simp: Int-Un-distrib2)
  } moreover
  { assume ?B
    hence square2 n (2^n+i) j - {(a,b)} : tiling Ls using IH a b by auto
    moreover have square2 n i j - {(2^n+i - 1, 2^n+j - 1)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n i (2^n+j) - {(2^n+i - 1, 2^n+j)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n (2^n+i) (2^n+j) - {(2^n+i, 2^n+j)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    ultimately
    have square2 (n+1) i j - {(a,b)} - L1 (2^n+i - 1) (2^n+j - 1) ∈ tiling
  }
Ls
    using a b (?B)
    by (simp add: square2-Suc L1-def Un-Diff Diff-insert-if le-diff-conv2)
      (fastsimp intro!: tiling-UnI DisjI1 DisjI2 square2-disj[THEN iffD2]
        simp: Int-Un-distrib2)
  } moreover
  { assume ?C
    hence square2 n i (2^n+j) - {(a,b)} : tiling Ls using IH a b by auto
    moreover have square2 n i j - {(2^n+i - 1, 2^n+j - 1)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n (2^n+i) j - {(2^n+i, 2^n+j - 1)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n (2^n+i) (2^n+j) - {(2^n+i, 2^n+j)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    ultimately
    have square2 (n+1) i j - {(a,b)} - L3 (2^n+i - 1) (2^n+j - 1) ∈ tiling
  }
Ls
    using a b (?C)
    by (simp add: square2-Suc L3-def Un-Diff Diff-insert-if le-diff-conv2)
      (fastsimp intro!: tiling-UnI DisjI1 DisjI2 square2-disj[THEN iffD2]
        simp: Int-Un-distrib2)
  } moreover
  { assume ?D
    hence square2 n (2^n+i) (2^n+j) - {(a,b)} : tiling Ls using IH a b by auto
    moreover have square2 n i j - {(2^n+i - 1, 2^n+j - 1)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n (2^n+i) j - {(2^n+i, 2^n+j - 1)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    moreover have square2 n i (2^n+j) - {(2^n+i - 1, 2^n+j)} : tiling Ls
      by(rule IH)(insert pos-pow2[of n], auto)
    ultimately
    have square2 (n+1) i j - {(a,b)} - L2 (2^n+i - 1) (2^n+j - 1) ∈ tiling
  }
Ls
    using a b (?D)

```

```

    by (simp add: square2-Suc L2-def Un-Diff Diff-insert-if le-diff-conv2)
      (fastsimp intro!: tiling-UnI DisjI1 DisjI2 square2-disj [THEN iffD2]
        simp: Int-Un-distrib2)
  } moreover
  have ?A  $\implies$  L0  $(2^{\hat{n}} + i - 1) (2^{\hat{n}} + j - 1) \subseteq \text{square2 } (n+1) i j - \{(a, b)\}$ 
    using a b by (simp add: L0-def) arith moreover
  have ?B  $\implies$  L1  $(2^{\hat{n}} + i - 1) (2^{\hat{n}} + j - 1) \subseteq \text{square2 } (n+1) i j - \{(a, b)\}$ 
    using a b by (simp add: L1-def) arith moreover
  have ?C  $\implies$  L3  $(2^{\hat{n}} + i - 1) (2^{\hat{n}} + j - 1) \subseteq \text{square2 } (n+1) i j - \{(a, b)\}$ 
    using a b by (simp add: L3-def) arith moreover
  have ?D  $\implies$  L2  $(2^{\hat{n}} + i - 1) (2^{\hat{n}} + j - 1) \subseteq \text{square2 } (n+1) i j - \{(a, b)\}$ 
    using a b by (simp add: L2-def) arith
  ultimately show ?case by simp (metis LinLs tiling-Diff1E)
qed

corollary Ls-can-tile00:
  a <  $2^{\hat{n}}$   $\implies$  b <  $2^{\hat{n}}$   $\implies$  square2 n 0 0 - {(a, b)}  $\in$  tiling Ls
by (rule Ls-can-tile) auto

end

```

References

- [1] Lawrence C. Paulson. A simple formalization and proof for the mutilated chess board. *Logic J. of the IGPL*, 9(3), 2001.
- [2] Velleman. *How to Prove it*. Cambridge University Press, 1994.