Fun With Tilings

Tobias Nipkow and Lawrence Paulson

March 17, 2025

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* ⇒ *'a set set*
  for *A* :: *'a set set* **where**
    empty [*simp*, *intro*]: {} ∈ *tiling A* |
    Un [*simp*, *intro*]: ⟦ *a* ∈ *A*; *t* ∈ *tiling A*; *a* ∩ *t* = {} ⟧
      ⟹ *a* ∪ *t* ∈ *tiling A*

lemma *tiling-UnI* [*intro*]:
  ⟦ *t* ∈ *tiling A*; *u* ∈ *tiling A*; *t* ∩ *u* = {} ⟧ ⟹ *t* ∪ *u* ∈ *tiling A*
  by (*induct set*: *tiling*) (*auto simp*: *Un-assoc*)

lemma *tiling-Diff1E*:
  assumes *t*−*a* ∈ *tiling A* **and** *a* ∈ *A* **and** *a* ⊆ *t*
  shows *t* ∈ *tiling A*
  by (*metis Diff-disjoint Diff-partition assms tiling.Un*)

lemma *tiling-finite*:
  assumes ⋀*a*. *a* ∈ *A* ⟹ *finite a*
  shows *t* ∈ *tiling A* ⟹ *finite t*
  by (*induct set*: *tiling*) (*use assms* **in** *auto*)

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* :: (*nat* × *nat*) *set set* **where**
horiz [*simp*]: {(*i*, *j*), (*i*, *Suc j*)} ∈ *domino* |
vertl [*simp*]: {(*i*, *j*), (*Suc i*, *j*)} ∈ *domino*

lemma *domino-finite*: *d* ∈ *domino* ⟹ *finite d*
by (*erule domino.cases*, *auto*)

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

Sets of squares of the given colour

definition
coloured :: *nat* ⇒ (*nat* × *nat*) *set* **where**
coloured b = {(*i*, *j*). (*i* + *j*) *mod* 2 = *b*}

abbreviation
whites :: (*nat* × *nat*) *set* **where**
whites ≡ *coloured* 0

abbreviation
blacks :: (*nat* × *nat*) *set* **where**
blacks ≡ *coloured* (*Suc* 0)

Chess boards

lemma *Sigma-Suc1* [*simp*]:
{0..< *Suc n*} × *B* = ({*n*} × *B*) ∪ ({0..<*n*} × *B*)
by *auto*

lemma *Sigma-Suc2* [*simp*]:
A × {0..< *Suc n*} = (*A* × {*n*}) ∪ (*A* × {0..<*n*})
by *auto*

lemma *dominoes-tile-row* [*intro*!]: {*i*} × {0..< 2∗*n*} ∈ *tiling domino*
by (*induct n*) (*auto simp flip*: *Un-insert-left Un-assoc*)

lemma *dominoes-tile-matrix*: {0..<*m*} × {0..< 2∗*n*} ∈ *tiling domino*
by (*induct m*) *auto*

coloured and Dominoes

lemma *coloured-insert* [*simp*]:
coloured b ∩ (*insert* (*i*, *j*) *t*) =
(*if* (*i* + *j*) *mod* 2 = *b then insert* (*i*, *j*) (*coloured b* ∩ *t*)
else coloured b ∩ *t*)

by (*auto simp*: *coloured-def*)

lemma *domino-singletons*:
  $d \in domino \Longrightarrow$
  $(\exists\, i\ j.\ whites \cap d = \{(i, j)\}) \wedge$
  $(\exists\, m\ n.\ blacks \cap d = \{(m, n)\})$
  by (*elim domino.cases*) (*auto simp*: *mod-Suc*)

Tilings of dominoes

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

lemma *tiling-domino-0-1*:
  $t \in tiling\ domino \Longrightarrow card(whites \cap t) = card(blacks \cap t)$
proof (*induct set*: *tiling*)
  case *empty*
  then show *?case*
    by *auto*
next
  case (*Un d t*)
  — this fact tells us that both "inserts" are non-trivial
  then have $\bigwedge p\ C.\ C \cap d = \{p\} \longrightarrow p \notin t$
    by *auto*
  moreover
  obtain *w b* **where** $whites \cap d = \{w\}$ $blacks \cap d = \{b\}$
    using *Un.hyps domino-singletons* **by** *force*
  ultimately show *?case*
    using *Un* **by** *auto*
qed

Final argument is surprisingly complex

theorem *gen-mutil-not-tiling*:
  assumes $t \in tiling\ domino$ $(i + j)\ mod\ 2 = 0$
    $(m + n)\ mod\ 2 = 0$ $\{(i, j), (m, n)\} \subseteq t$
  shows $t - \{(i, j)\} - \{(m, n)\} \notin tiling\ domino$
proof −
  have $card\ (whites \cap (t - \{(i,j)\} - \{(m,n)\}))$
      $< card\ (blacks \cap (t - \{(i,j)\} - \{(m,n)\}))$
    using *assms* **unfolding** *tiling-domino-0-1* [*symmetric*]
    by (*simp flip*: *tiling-domino-0-1*) (*simp add*: *coloured-def card-Diff2-less*)
  then show *?thesis*
    by (*metis nat-neq-iff tiling-domino-0-1*)
qed

Apply the general theorem to the well-known case

theorem *mutil-not-tiling*:
  assumes $t = \{0..< 2 * Suc\ m\} \times \{0..< 2 * Suc\ n\}$

shows $t - \{(0,0)\} - \{(Suc(2 * m), Suc(2 * n))\} \notin tiling\ domino$
proof −
have $t \in tiling\ domino$
using *assms dominoes-tile-matrix* **by** *blast*
with *assms* **show** *?thesis*
by (*intro gen-mutil-not-tiling*) *auto*
qed

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: ⊞. 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*::*nat*) (*y*::*nat*) = {(*x*,*y*), (*x*+*1*,*y*), (*x*, *y*+*1*)}
definition *L3* (*x*::*nat*) (*y*::*nat*) = {(*x*,*y*), (*x*+*1*,*y*), (*x*+*1*, *y*+*1*)}
definition *L0* (*x*::*nat*) (*y*::*nat*) = {(*x*+*1*,*y*), (*x*,*y*+*1*), (*x*+*1*, *y*+*1*)}
definition *L1* (*x*::*nat*) (*y*::*nat*) = {(*x*,*y*), (*x*,*y*+*1*), (*x*+*1*, *y*+*1*)}

All tiles:

definition *Ls* :: (*nat* ∗ *nat*) *set set* **where**
Ls ≡ { *L0 x y* | *x y*. *True*} ∪ { *L1 x y* | *x y*. *True*} ∪
{ *L2 x y* | *x y*. *True*} ∪ { *L3 x y* | *x y*. *True*}

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

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

definition *square2* (*n*::*nat*) (*i*::*nat*) (*j*::*nat*) = {*i*..< *2*^*n*+*i*} × {*j*..< *2*^*n*+*j*}

lemma *in-square2*[*simp*]:
(*a*,*b*) : *square2 n i j* ⟷ *i*≤*a* ∧ *a*<*2*^*n*+*i* ∧ *j*≤*b* ∧ *b*<*2*^*n*+*j*
by(*simp add*:*square2-def*)

lemma *square2-Suc*: *square2* (*Suc n*) *i j* =
square2 n i j ∪ *square2 n* (*2*^*n* + *i*) *j* ∪ *square2 n i* (*2*^*n* + *j*) ∪
square2 n (*2*^*n* + *i*) (*2*^*n* + *j*)
by *auto*

lemma *square2-disj*: *square2 n i j* ∩ *square2 n x y* = {} ⟷
(*2*^*n*+*i* ≤ *x* ∨ *2*^*n*+*x* ≤ *i*) ∨ (*2*^*n*+*j* ≤ *y* ∨ *2*^*n*+*y* ≤ *j*) (**is** *?A* = *?B*)
proof−
{ assume *?B* **hence** *?A* **by**(*auto simp*:*square2-def*) **}**
moreover
{ assume ¬ *?B*
hence (*max i x*, *max j y*) : *square2 n i j* ∩ *square2 n x y* **by** *simp*
hence ¬ *?A* **by** *blast* **}**
ultimately show *?thesis* **by** *blast*

qed

Some specific lemmas:

lemma *pos-pow2*: $(0::nat) < 2\hat{}n$
by *simp*

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

lemma *Diff-insert-if*: **shows**
$B \neq \{\} \Longrightarrow a \in A \Longrightarrow A - insert\ a\ B = (A - B - \{a\})$ **and**
$B \neq \{\} \Longrightarrow a \notin A \Longrightarrow A - insert\ a\ B = A - B$
by *auto*

lemma *DisjI1*: $A \cap B = \{\} \Longrightarrow (A - X) \cap B = \{\}$
by *blast*

lemma *DisjI2*: $A \cap B = \{\} \Longrightarrow A \cap (B - X) = \{\}$
by *blast*

The main theorem:

theorem *Ls-can-tile*: $i \leq a \Longrightarrow a < 2\hat{}n + i \Longrightarrow j \leq b \Longrightarrow b < 2\hat{}n + j$
$\Longrightarrow$ *square2 n i j* $- \{(a,b)\} \in$ *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 $a<2\hat{}n+i \wedge b<2\hat{}n+j \vee$
$2\hat{}n+i\leq a \wedge a<2\hat{}(n+1)+i \wedge b<2\hat{}n+j \vee$
$a<2\hat{}n+i \wedge 2\hat{}n+j\leq b \wedge b<2\hat{}(n+1)+j \vee$
$2\hat{}n+i\leq a \wedge a<2\hat{}(n+1)+i \wedge 2\hat{}n+j\leq b \wedge b<2\hat{}(n+1)+j$ (**is** *?A*|*?B*|*?C*|*?D*)
by *simp arith*
moreover
{ assume *?A*
hence *square2 n i j* $- \{(a,b)\} \in$ *tiling Ls* **using** *IH a b* **by** *auto*
moreover
have *square2 n* $(2\hat{}n+i)$ *j* $- \{(2\hat{}n+i,2\hat{}n+j - 1)\} \in$ *tiling Ls*
square2 n i $(2\hat{}n+j) - \{(2\hat{}n+i - 1,\ 2\hat{}n+j)\} \in$ *tiling Ls*
square2 n $(2\hat{}n+i)$ $(2\hat{}n+j) - \{(2\hat{}n+i,\ 2\hat{}n+j)\} \in$ *tiling Ls*
by (*simp-all add*: *IH le-add-diff pos-pow2*)
ultimately
have *square2* $(n+1)$ *i j* $- \{(a,b)\} -$ *L0* $(2\hat{}n+i - 1)$ $(2\hat{}n+j - 1) \in$ *tiling Ls*
using *a b* ‹*?A*›
by (*clarsimp simp*: *square2-Suc L0-def Un-Diff Diff-insert-if*)
(*fastforce intro*!: *tiling-UnI DisjI1 DisjI2 square2-disj*[*THEN iffD2*]
simp:*Int-Un-distrib2*)
} moreover
{ assume *?B*
hence *square2 n* $(2\hat{}n+i)$ *j* $- \{(a,b)\} \in$ *tiling Ls* **using** *IH a b* **by** *auto*
moreover

```
  have square2 n i j − {(2^n+i − 1,2^n+j − 1)} ∈ tiling Ls
      square2 n i (2^n+j) − {(2^n+i − 1, 2^n+j)} ∈ tiling Ls
      square2 n (2^n+i) (2^n+j) − {(2^n+i, 2^n+j)} ∈ tiling Ls
    by (simp-all add: IH le-add-diff pos-pow2)
  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)
      (fastforce 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
      square2 n (2^n+i) j − {(2^n+i, 2^n+j − 1)} ∈ tiling Ls
      square2 n (2^n+i) (2^n+j) − {(2^n+i, 2^n+j)} ∈ tiling Ls
    by (simp-all add: IH le-add-diff pos-pow2)
  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)
      (fastforce 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
      square2 n (2^n+i) j − {(2^n+i, 2^n+j − 1)} ∈ tiling Ls
      square2 n i (2^n+j) − {(2^n+i − 1, 2^n+j)} ∈ tiling Ls
    by (simp-all add: IH le-add-diff pos-pow2)
  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)
      (fastforce intro!: tiling-UnI DisjI1 DisjI2 square2-disj[THEN iffD2]
          simp:Int-Un-distrib2)
} moreover
have ?A ⟹ L0 (2^n + i − 1) (2^n + j − 1) ⊆ square2 (n+1) i j − {(a, b)}
  using a b by(simp add:L0-def) arith moreover
have ?B ⟹ L1 (2^n + i − 1) (2^n + j − 1) ⊆ square2 (n+1) i j − {(a, b)}
  using a b by(simp add:L1-def) arith moreover
have ?C ⟹ L3 (2^n + i − 1) (2^n + j − 1) ⊆ square2 (n+1) i j − {(a, b)}
  using a b by(simp add:L3-def) arith moreover
have ?D ⟹ L2 (2^n + i − 1) (2^n + j − 1) ⊆ 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*:
$a < 2\hat{}n \Longrightarrow b < 2\hat{}n \Longrightarrow$ *square2 n 0 0* $- \{(a, b)\} \in$ *tiling Ls*
by(*intro 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.