Fun With Tilings

Tobias Nipkow and Lawrence Paulson

February 23, 2021

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
⟨proof⟩

lemma tiling-Diff1E:
assumes t−a ∈ tiling A and a ∈ A and a ⊆ t
shows t ∈ tiling A
⟨proof⟩

lemma tiling-finite:
assumes ∆a. a ∈ A ⇒ finite a
shows t ∈ tiling A ⇒ finite t
⟨proof⟩
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.

```plaintext
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
 ⟨proof⟩
```

Sets of squares of the given colour

```plaintext
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

```plaintext
lemma Sigma-Suc1 [simp]:
  {0..< Suc n} × B = (\{n\} × B) ∪ (\{0..<n\} × B)
 ⟨proof⟩
```

```plaintext
lemma Sigma-Suc2 [simp]:
  A × {0..< Suc n} = (A × \{n\}) ∪ (A × \{0..<n\})
 ⟨proof⟩
```

```plaintext
lemma dominoes-tile-row [intro!]: \{i\} × \{0..< 2*n\} ∈ tiling domino
 ⟨proof⟩
```

```plaintext
lemma dominoes-tile-matrix: \{0..<m\} × \{0..< 2*n\} ∈ tiling domino
 ⟨proof⟩
```

```plaintext
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)
```

(proof)

lemma domino-singletons:
\[ d \in \text{domino} \implies
\begin{align*}
\exists i, j. & \text{whites} \cap d = \{(i, j)\} \land \\
\exists m, n. & \text{blacks} \cap d = \{(m, n)\}
\end{align*}
\]

(proof)

Tilings of dominoes

declare
\begin{align*}
\text{Int-Un-distrib [simp]} \\
\text{Diff-Int-distrib [simp]}
\end{align*}

lemma tiling-domino-0-1:
\[ t \in \text{tiling domino} \implies \text{card}(\text{whites} \cap t) = \text{card}(\text{blacks} \cap t) \]

(proof)

Final argument is surprisingly complex

theorem gen-mutil-not-tiling:
\[ t \in \text{tiling domino} \implies
\begin{align*}
(i + j) \mod 2 = 0 & \implies (m + n) \mod 2 = 0 \implies \\
\{(i, j), (m, n)\} \subseteq t \\
\implies (t - \{(i, j)\} - \{(m, n)\}) \notin \text{tiling domino}
\end{align*}
\]

(proof)

Apply the general theorem to the well-known case

theorem mutil-not-tiling:
\[ t = \{0..<2 \cdot \text{Suc } m\} \times \{0..<2 \cdot \text{Suc } n\} \\
\implies t - \{(0,0)\} - \{(\text{Suc}(2 \cdot m), \text{Suc}(2 \cdot n))\} \notin \text{tiling domino} \]

(proof)

3 The Mutilated Chess Board Can be Tiled by Ls

Remove an 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 \equiv \{ L0 x y | x y. True\} \cup \{ L1 x y | x y. True\} \cup \\
\{ L2 x y | x y. True\} \cup \{ L3 x y | x y. True\}

3
Lemma LinLs: L0 i j : Ls & L1 i j : Ls & L2 i j : Ls & L3 i j : Ls
(proof)

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

Definition square2 (n:nat) (i:nat) (j:nat) = \{i..< 2^n+i\} \times \{j..< 2^n+j\}

Lemma in-square2[simp]:
(a,b) : square2 n i j \iff i\leq a \land a<2^n+i \land j\leq b \land b<2^n+j
(proof)

Lemma square2-Suc: square2 (Suc n) i j =
square2 n i j \cup square2 n (2^n+i) j \cup square2 n i (2^n+j) \cup
square2 n (2^n+i) (2^n+j)
(proof)

Lemma square2-disj: square2 n i j \cap square2 n x y = {} \iff
(2^n+i \leq x \lor 2^n+x \leq i) \lor (2^n+j \leq y \lor 2^n+y \leq j) (is ?A = ?B)
(proof)

Some specific lemmas:

Lemma pos-pow2: (0::nat) < 2^n::nat
(proof)

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

Lemma Diff-insert-if: shows
B \neq {} \Rightarrow a:A \Rightarrow A - insert a B = (A-B - \{a\}) and
B \neq {} \Rightarrow a:A \Rightarrow A - insert a B = A-B
(proof)

Lemma DisjI1: A Int B = {} \Rightarrow (A-X) Int B = {}
(proof)
Lemma DisjI2: A Int B = {} \Rightarrow A Int (B-X) = {}
(proof)

The main theorem:

Theorem Ls-can-tile: i \leq a \Rightarrow a < 2^n + i \Rightarrow j \leq b \Rightarrow b < 2^n + j
\Rightarrow square2 n i j - \{(a,b)\} \in tiling Ls
(proof)

Corollary Ls-can-tile00:
a < 2^n \Rightarrow b < 2^n \Rightarrow square2 n 0 0 - \{(a,b)\} \in tiling Ls
(proof)

End
References
