Getting Started With Isabelle

Lecture IV: The Mutilated Chess Board

Lawrence C. Paulson Computer Laboratory



Lecture Outline

- The informal problem
- Inductive definitions
- The Isabelle/HOL specification
- Proof overview



After cutting off the corners, can the board be tiled with dominoes?



The point: find a suitably abstract model.



A tile is a set of points (such as squares). Given a set of tiles (such as dominoes):

- The empty set can be tiled.
- If t can be tiled, and a is a tile disjoint from t, then the set $a \cup t$ can be tiled.

For A a set of tiles, inductively define tiling(A):

```
consts tiling :: "'a set set => 'a set set"
inductive "tiling A"
    intrs
    empty "{} : tiling A"
    Un "[| a: A; t: tiling A; a <= -t |]
        ==> a Un t : tiling A"
```



Inductive Definitions in Isabelle/HOL

We get (proved from a fixed point construction)

- rules tiling.empty and tiling.Un for making tilings
- rule tiling.induct to do induction on tilings:

If property *P* holds for {} and if *P* is closed under adding a tile, then *P* holds for all tilings.



Example: The Union of Disjoint Tilings

If t, $u \in \text{tiling}(A)$ and $t \subseteq \overline{u}$ then $t \cup u \in \text{tiling}(A)$.

base case Here $t = \{\}$, so $t \cup u = u \in tiling(A)$ by assumption.

induction step Here $t = a \cup t'$, with a disjoint from t.

Assume that $a \cup t'$ is disjoint from u.

By induction $t' \cup u$ is a tiling, since t' is disjoint from u.

And $a \cup (t' \cup u)$ is a tiling, since a is disjoint from $t' \cup u$.

So $t \cup u = a \cup t' \cup u \in tiling(A)$.



The Proof Script for Our Example

tidy up remaining subgoals

qed_spec_mp "tiling_UnI";
 store th

store the theorem



The Isabelle Theory File

```
Mutil = Main +
<u>consts</u> tiling ...
consts domino :: "(nat*nat)set set"
inductive domino
                               dominoes too are inductive!
  intrs
    horiz "{(i, j), (i, Suc j)} : domino"
    vertl "{(i, j), (Suc i, j)} : domino"
constdefs
  below :: "nat => nat set" row/column numbering
   "below n == \{i. i < n\}"
  colored :: "nat => (nat*nat)set"
   "colored b == {(i,j). (i+j) \mod 2 = b}"
```

end

UNIVERSITY OF CAMBRIDGE

Two disjoint tilings form a tiling.

Simple facts about below: chess board geometry

Then some facts about tiling with dominoes:

Every row of length 2n can be tiled.

Every $m \times 2n$ board can be tiled.

Every tiling has as many black squares as white ones.

If t can be tiled, then the area obtained by removing two black squares cannot be tiled.

No $2m \times 2n$ mutilated chess board (m, n > 0) can be tiled.



The Cardinality Proof Script

```
Goal "t: tiling domino ==> \setminus
     card(colored 0 Int t) = card(colored 1 Int t)";
by (etac tiling.induct 1);
                        perform induction over tiling(A)
by (dtac domino_singletons 2);
                        a domino has a white square & a black one
by Auto tac;
by (subgoal_tac "ALL p C. C Int a = p - p \sim t' + 1);
                        lemma about the domino a and tiling t
by (Asm simp tac 1);
by (blast_tac (claset() addEs [equalityE]) 1);
                        using, and proving, this lemma
```



Follows the informal argument Admits a general proof, not just the 8×8 case Yields a short proof script:

- 15 theorems
- 2.4 tactic calls per theorem
- 4.5 seconds run time



Other Applications of Inductive Definitions

- Proof theory
- Operational semantics
- Security protocol verification
- Modelling the λ -calculus

