```
theory mutilated-ideal
imports Main

begin

definition chessboard :: (nat × nat) set where
  chessboard = {..7} × {..7}

definition mchessboard :: (nat × nat) set where
  mchessboard = chessboard − {(0, 0), (7, 7)}

datatype domino =
  horizontal nat × nat
| vertical   nat × nat

fun covers :: domino ⇒ (nat × nat) set where
  covers (horizontal (i, j)) = {(i, j), (i+1, j)}
| covers (vertical (i, j)) = {(i, j), (i, j+1)}

definition white :: (nat × nat) set where
  white = {p. even (fst p + snd p)}

definition flip :: (nat × nat) ⇒ (nat × nat) where
  flip p = (7 − fst p, snd p)

theorem mutilated-chessboard:
  fixes s :: domino set
  assumes h1: ∀ x ∈ s. ∀ y ∈ s. x ≠ y ⟶ covers x ∩ covers y = {} and
          h1a: finite s
  shows (⋃ x ∈ s. covers x) ≠ mchessboard
proof
  have bij-betw flip (chessboard ∩ white) (chessboard ∩ −white)
    sorry
  hence h2: card (chessboard ∩ white) = card (chessboard ∩ −white)
    sorry
  have (0, 0) ∈ chessboard ∩ white and (7, 7) ∈ chessboard ∩ white
    sorry
  with h2 have h3: card (mchessboard ∩ white) < card (mchessboard ∩ −white)
    sorry
  have ∀ x. card (covers x ∩ white) = card (covers x ∩ −white)
    sorry
  with h1 have h4: card ((⋃ x ∈ s. covers x) ∩ white) = card ((⋃ x ∈ s. covers
x) ∩ −white)
    sorry
  assume (⋃ x ∈ s. covers x) = mchessboard
  with h3 h4 show False
    sorry
qed
```

end