S. Linton and R. Sebastiani (ed.), Proc. of the Calculemus 2001 Symposium, pp. 28-47.

On a Solution of the Mutilated Checkerboard Problem using the *Theorema* Set Theory Prover

Wolfgang Windsteiger*[1]

[1] *RISC Institute*
A-4232 Hagenberg, Austria
Wolfgang.Windsteiger@RISC.Uni-Linz.ac.at

Abstract

The Mutilated Checkerboard Problem has some tradition as a benchmark problem for automated theorem proving systems. Informally speaking, it states that an 8 by 8 checkerboard with the two opposite corners removed cannot be covered by dominoes. Various solutions using different approaches have been presented since its original statement by John McCarthy in 1964. An elegant four–line proof has been given on paper by McCarthy himself in 1995, which is based on a formulation of the original problem in set theory. Since then, the checkerboard problem stands as a benchmark problem in particular also for automated set theory provers. In this paper, we are going to present a complete proof of the checkerboard problem using the *Theorema Set Theory prover*.

1. Introduction

The Mutilated Checkerboard (british for Chessboard) Problem goes back to John McCarthy (http://www-formal.stanford.edu/jmc), who stated the problem originally in a Stanford AI memo in 1964, see (McCarthy, 1964): An 8 by 8 checkerboard with two diagonally opposite squares removed cannot be covered by dominoes each of which covers two rectilinearly adjacent squares. Different proofs of this statement have been formulated by among others Shmuel Winograd, Marvin Minsky and Dimitri Stefanyuk, none of them published according to McCarthy, and kind of a contest started to formulate the most non-creative proof

*This work has been supported by the "SFB Numerical and Symbolic Scientific Computing" (F013) at the University of Linz, the "PROVE Project" supported by the regional government of Upper Austria, and the european union "CALCULEMUS Project" (HPRN–CT–2000–00102).

of the statement. In 1993, an interactive proof using the Boyer-Moore prover NQTHM was given in (Subramanian, 1993) and (Subramanian, 1996), whereas William McCune gave a proof using MACE to find a finite model, see (McCune, 1995). In 1996, a proof in higher order logic was presented by Andrews and Bishop, see (AndrewsBishop, 1996), using the TPS system. In 1995, McCarthy himself gave a formulation of the original problem in set theory and a four-line proof, see (McCarthy, 1995). From the abstract of his article: While no present system that I know of will accept either the formal description or the proof, I claim that both should be admitted in any heavy duty set theory. At the same conference, where McCarthy presented this challenge, at the QED Workshop II in Warshaw, Grzegorz Bancerek came up with the set theoretic formulation of McCarthy accepted and checked by Mizar, which took about 400 lines, see (Bancerek, 1995).

The key to the four-line proof of McCarthy is the fact that three propositions are thrown in, which are intuitively correct when having the picture of the colored chessboard in one's mind, but, nevertheless, lack a proof in his presentation. What we shall show in the exploration of the mutilated checkerboard in *Theorema* is

- the reformulation of the definitions and the theorem into the language available in *Theorema*,
- the proof of the Theorem *and* the proofs of *all* intermediate propositions,
- a systematic build up of the "theory of dominoes and partial coverings on a chessboard" augmented also by *computations* in order to get a feeling how the new functions and predicates introduced behave on small examples.

The solution shown below is *one* possible solution in *Theorema*, neither necessarily the most elegant one nor the shortest. In particular, the computations at the beginning do not at all contribute to the final solution, but it was the intention to show, how the *Theorema* system would support the entire process of mathematical theory development. As a matter of fact, experiments with newly invented notions, i.e. computations, are most of the time the origin of the development of involved theories and fancy theorems, since some key properties of the new notions can be observed during the experiments or improvements of the computations turn out to be necessary due to tremendous computation times. This process of "every-day work of a mathematician" has been called the *creativity spiral of mathematics* in (Buchberger, 1993). It is an attempt to argue that the *Theorema* system can support the working mathematician throughout the *entire cycle* of this spiral. Moreover, the solution presented in this paper follows the style of theory exploration as explained in (Buchberger, 1999), with several exploration rounds starting from the initial definitions finally arriving at the theorem of interest. The first exploration round in Section 2 introduces the set theoretic formulation of the basic concepts needed in the formulation of the checkerboard problem and some immediate properties needed later. Exploration round 2 in Section 3 introduces the new notion of "dominoes" and leads to a proof of a lemma corresponding to one of McCarthy's unproved propositions. Round 3 described in Section 4 introduces the new concept of "partial coverings" and explores the interactions with already known concepts with a Lemma corresponding to the second unproved proposition of McCarthy's as a highlight. The final round in Section 4 collects the results in a proof of McCarthy's theorem.

All intermediate propositions that occur during the entire case study have been proven using the *Theorema* Set Theory prover developed in (Windsteiger, 2001) – except for two propositions in Section 2. However, due to space limitations for the final paper, we omit some of the proofs completely and we leave out some routine parts of the remaining proofs, which we will mark by inserting "..." for the left out proof parts. In the presentation of the proofs, we concentrate on particular features of the Set Theory prover, since this case study was originally motivated as a test for this new prover in the *Theorema* system. See (Windsteiger, 2001) for the complete proofs of *all* propositions and also for a detailed introduction into the theoretical foundations of the Set Theory prover.

2. Exploration Round 1: The Board

As a first remark, McCarthy's formulation uses a slightly different language compared to what *Theorema* offers, but the translation into *Theorema* is pretty straight-forward and we will comment on the details when giving our definitions below. Additionally, we allow ourselves to rename certain concepts according to our personal taste regarding the choice of names instead of just copying from McCarthy's model. We can define the set of pairs representing the chessboard elegantly using multiple integer ranges in the set quantifier. Set difference must be denoted by "\" in case built-in knowledge about set difference is desired. Of course, *Theorema* provides the tools for the user to define any new function or predicate using (almost) any desired notion, but if available semantics for notions provided by the *Theorema* language should be applied during computation, or if available inference rules should by applied during proving then the user must stick to the notions fixed inside the *Theorema* system.

Definition["Board",

$$\text{Board} = \{\langle i, j\rangle \underset{\substack{i=0,\ldots,7\\ j=0,\ldots,7}}{|} \}$$

$$\text{Mutilated-Board} = \text{Board} \setminus \{\langle 0,0\rangle, \langle 7,7\rangle\}$$

]

As indicated in the final proof of the theorem, there is some motivation to define the color of a pair as 0 or 1, where 0 corresponds to a black square and 1 corresponds to a white square on the chessboard. Using the Mod function, which returns the remainder when dividing by an integer, we can easily define the color as the remainder when dividing the sum of the components by 2. Just think of the components to be the "coordinates on the board".

Definition["Color", any[x],

$$\mathrm{color}[x] = \mathrm{Mod}[x_1 + x_2, 2]] \;]$$

`Compute`[$\{x \underset{x \in \text{Mutilated-Board}}{|} \mathrm{color}[x] = 0\}$]

$\{\langle 0,2\rangle, \langle 0,4\rangle, \langle 0,6\rangle, \langle 1,1\rangle, \langle 1,3\rangle, \langle 1,5\rangle, \langle 1,7\rangle, \langle 2,0\rangle, \langle 2,2\rangle, \langle 2,4\rangle, \langle 2,6\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 3,5\rangle, \langle 3,7\rangle, \langle 4,0\rangle, \langle 4,2\rangle, \langle 4,4\rangle, \langle 4,6\rangle, \langle 5,1\rangle, \langle 5,3\rangle, \langle 5,5\rangle, \langle 5,7\rangle, \langle 6,0\rangle, \langle 6,2\rangle, \langle 6,4\rangle, \langle 6,6\rangle, \langle 7,1\rangle, \langle 7,3\rangle, \langle 7,5\rangle\}$

`Compute`[$\{x \underset{x \in \text{Mutilated-Board}}{|} \mathrm{color}[x] = 1\}$]

$\{\langle 0,1\rangle, \langle 0,3\rangle, \langle 0,5\rangle, \langle 0,7\rangle, \langle 1,0\rangle, \langle 1,2\rangle, \langle 1,4\rangle, \langle 1,6\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 2,5\rangle, \langle 2,7\rangle, \langle 3,0\rangle, \langle 3,2\rangle, \langle 3,4\rangle, \langle 3,6\rangle, \langle 4,1\rangle, \langle 4,3\rangle, \langle 4,5\rangle, \langle 4,7\rangle, \langle 5,0\rangle, \langle 5,2\rangle, \langle 5,4\rangle, \langle 5,6\rangle, \langle 6,1\rangle, \langle 6,3\rangle, \langle 6,5\rangle, \langle 6,7\rangle, \langle 7,0\rangle, \langle 7,2\rangle, \langle 7,4\rangle, \langle 7,6\rangle\}$

`Compute`[$|\{x \underset{x \in \text{Mutilated-Board}}{|} \mathrm{color}[x] = 0\}|$]

$$30$$

`Compute`[$|\{x \underset{x \in \text{Mutilated-Board}}{|} \mathrm{color}[x] = 1\}|$]

$$32$$

The key idea for the entire proof is essentially contained in the following observation, which may be conjectured from the previous computations: the mutilated board contains more white squares than black ones. The region covered by dominoes, however, always contains equally many black and white fields. This observation is even independent of the actual size of the chessboard, see also (Subramanian, 1993). For the computation of colors it is sufficient to use the Mathematica built-in function Mod for doing the remainder calculations. However, if we want to *prove* properties of colors, we will need a definition of the Mod function. We use a definition by cases.

Definition["Mod", any[n],

$$\mathrm{Mod}[n, 2] := \begin{cases} 0 & \Leftarrow \quad \underset{j \in \mathbb{N}}{\exists} n = 2j \\ 1 & \Leftarrow \quad \textit{otherwise} \end{cases} \quad]$$

The Mod function defined in this way has some interesting interaction with the absolute value function on integers. In the spirit of theory exploration, we do not prove these two propositions now, since they would normally have been proven in earlier exploration rounds when introducing the Mod function or the absolute value function on integers using a specialized prover for this area. (Of course, we could now give a definition of the absolute value function using case distinction, but finally this would be more in the spirit of *proving going back to first principles.*) In our approach of exploring the checkerboard, we assume the Mod function and the absolute value function on integers as *completely explored*, and we assume that the following propositions have been proven in those explorations. In fact, these propositions are *the only propositions* in this case study, for which we will not provide a proof.

Proposition[“Mod property even”, any[x, y, z],

$$\begin{array}{l}(|x-z|=1) \wedge (\mathrm{Mod}[x+y,2]=0) \Rightarrow (\mathrm{Mod}[z+y,2]=1)\\(|x-z|=1) \wedge (\mathrm{Mod}[y+x,2]=0) \Rightarrow (\mathrm{Mod}[y+z,2]=1)\end{array} \quad]$$

Proposition[“Mod property odd”, any[x, y, z],

$$\begin{array}{l}(|x-z|=1) \wedge (\mathrm{Mod}[x+y,2]=1) \Rightarrow (\mathrm{Mod}[z+y,2]=0)\\(|x-z|=1) \wedge (\mathrm{Mod}[y+x,2]=1) \Rightarrow (\mathrm{Mod}[y+z,2]=0)\end{array} \quad]$$

Starting from the definitions, we can now prove color properties, like for instance the fact:

Proposition[“zero or one”, any[$x \in$ Board],

$$(\mathrm{color}[x]=0) \vee (\mathrm{color}[x]=1)] \quad]$$

`Prove`[Proposition[“zero or one”], using→ ⟨Definition[“Mod”],Definition[“Color”]⟩, built-in→Built-in[“Numbers”]]

In the proof of this proposition, we let the prover use built-in knowledge on numbers from the *Theorema* language semantics. Through this facility, we manage to smoothly integrate *computation* into the *Theorema* proving process, because application of built-in semantics knowledge during proving uses the same mechanism that is used by the top-level *Theorema* command Compute.

3. Exploration Round 2: Dominoes

Definition[“Domino”, any[x],

$$\text{domino-on-board}[x] :\Leftrightarrow \wedge \begin{cases} x \subseteq \text{Board} \\ |x| = 2 \\ \underset{\substack{x1,x2\in x\\ x1\neq x2}}{\forall} \text{adjacent}[x1, x2] \end{cases} \quad]$$

Definition[“Adjacency”, any[x, y],

$\text{adjacent}[x,y] :\Leftrightarrow ((|x_1 - y_1| = 1) \wedge (x_2 = y_2) \vee (|x_2 - y_2| = 1) \wedge (x_1 = y_1))$]

Many interesting properties will be provable when exploring these two new notions. Still, we start doing some computations first.

`Compute`[domino-on-board[$\{\langle 3,3\rangle, \langle 3,4\rangle\}$]]

True

`Compute`[domino-on-board[$\{\langle 3,3\rangle, \langle 3,4\rangle, \langle 3,5\rangle\}$]]

False

A domino is part of the board and its components are adjacent.

Proposition["dominoes adjacent", any[X],

domino-on-board[X] $\Rightarrow X \subseteq \text{Board} \wedge \underset{\substack{x,y \in X \\ x \neq y}}{\exists} \text{adjacent}[x,y]$]

`Prove`[Proposition["dominoes adjacent"],using→ Definition["Domino"]]

Prove:

(Proposition (dominoes adjacent)) ...,

under the assumption:

(Definition (Domino)) ....

We assume

(1) domino-on-board[X_0],

and show

(2) $X_0 \subseteq \text{Board} \wedge \underset{x,y}{\exists} ((x \in X_0 \wedge y \in X_0 \wedge (x \neq y)) \wedge \text{adjacent}[x,y])$.

We prove the individual conjunctive parts of (2):
Formula (1), by (Definition (Domino)), implies:

(3) $(|X_0| = 2) \wedge \underset{x1,x2}{\forall} (x1 \in X_0 \wedge x2 \in X_0 \wedge (x1 \neq x2) \Rightarrow \text{adjacent}[x1,x2]) \wedge X_0 \subseteq$ Board.

Formula (2.1) is true because it is identical to (3.3).
Formula (1), by (Definition (Domino)), implies:

(4) $(|X_0| = 2) \wedge \underset{x1,x2}{\forall} (x1 \in X_0 \wedge x2 \in X_0 \wedge (x1 \neq x2) \Rightarrow \text{adjacent}[x1,x2]) \wedge X_0 \subseteq$ Board.

From what we already know follows:
From (4.1) we can infer

(5) $X1_0 \in X_0$,

(6) $X1_1 \in X_0$,

(7) $X1_0 \neq X1_1$.

From (4.3) we can infer

(8) $\underset{X2}{\forall} (X2 \in X_0 \Rightarrow X2 \in \text{Board})$.

Now, let $y := X1_0$. Thus, for proving (2.2) it is sufficient to prove:

(12) $\underset{x}{\exists} ((x \in X_0 \wedge X1_0 \in X_0 \wedge (x \neq X1_0)) \wedge \text{adjacent}[x, X1_0])$.

Now, let $x := X1_1$. Thus, for proving (12) it is sufficient to prove:

(68) $(X1_1 \in X_0 \wedge X1_0 \in X_0 \wedge (X1_1 \neq X1_0)) \wedge \text{adjacent}[X1_1, X1_0]$.

Formula (68.1.1) is true because it is identical to (6).
Formula (68.1.2) is true because it is identical to (5).
Proof of (68.1.3) $X1_1 \neq X1_0$:
We prove (68.1.3) by contradiction.
We assume

(69) $X1_1 = X1_0$,

and show *a contradiction*.
Formula (7), by (69), implies:

(72) $X1_0 \neq X1_0$.

Using available computation rules we can simplify the knowledge base:
Formula (72) simplifies to

(73) *False*,

Formula (a contradiction) is true because the assumption (73) is false.
Formula (68.2), using (4.2), is implied by:

(74) $X1_0 \in X_0 \wedge X1_1 \in X_0 \wedge (X1_1 \neq X1_0)$.

Formula (74.1) is true because it is identical to (5).
Formula (74.2) is true because it is identical to (6).
We prove (74.3) by contradiction.
We assume

(75) $X1_1 = X1_0$,

and show *a contradiction*.
Formula (7), by (75), implies:

(78) $X1_0 \neq X1_0$.

Using available computation rules we can simplify the knowledge base:
Formula (78) simplifies to

(79) *False*,

Formula (a contradiction) is true because the assumption (79) is false. □

We want to emphasize the activation of a special inference rule that allows to choose certain elements from a set known to be finite in the previous proof. This rule is applied when deducing formulae (5), (6), and (7) from formula (4.1).

Adjacent fields have opposite colors.

Proposition[“adjacent of black are white”, any[x, y],
adjacent[x, y] $\wedge$ color[x] $= 0 \Rightarrow$ color[y] $= 1$]]

`Prove`[Proposiiton[“adjacent of black are white”],
using → ⟨Proposition[“Mod property even”],
Definition[“Adjacency”], Definition[“Color”]⟩]

See (Windsteiger, 2001) for this proof and also for the proof of the analog proposition for white fields.

Proposition[“adjacent of white are black”, any[x, y],
adjacent[x, y] $\wedge$ color[x] $= 1 \Rightarrow$ color[y] $= 0$]]

Using these propositions, we can now prove one of the statements from McCarthy’s proof. (We need knowledge about built-in numbers in order to be able to do simplification on arithmetic expressions.)

Lemma[“different color”, any[X],
domino-on-board[X] $\Rightarrow \underset{u,v \in X}{\exists} ((\text{color}[u] = 0) \wedge (\text{color}[v] = 1))$]

`Prove`[Lemma[“different color”],
using → ⟨ Proposition[“dominoes adjacent”],
Proposition[“zero or one”], Proposition[“adjacent of black are white”],
Proposition[“adjacent of white are black”]⟩,
built-in → Built-in[“Numbers”]]

Prove:

(Lemma (different color)) . . .,

under the assumptions:

(Proposition (dominoes adjacent)) . . .,

(Proposition (zero or one)) $\underset{x}{\forall} (x \in \text{Board} \Rightarrow (\text{color}[x] = 0) \vee (\text{color}[x] = 1))$, (Proposition (adjacent of black are white)) $\underset{x,y}{\forall}$ (adjacent$[x, y] \wedge (\text{color}[x] = 0) \Rightarrow$ $(\text{color}[y] = 1)$),

(Proposition (adjacent of white are black)) $\underset{x,y}{\forall}$ (adjacent$[x, y] \wedge (\text{color}[x] = 1) \Rightarrow$ $(\text{color}[y] = 0)$).

Formula (Proposition (dominoes adjacent)) simplifies to

(1) $\underset{X}{\forall}$ domino-on-board$[X] \Rightarrow X \subseteq \text{Board} \wedge \underset{x,y}{\exists} (x \in X \wedge y \in X \wedge (x \neq y) \wedge$ adjacent$[x, y]$),

Using available computation rules we evaluate (Lemma (different color)):

(2) $\underset{X}{\forall}$ (domino-on-board$[X] \Rightarrow \underset{u,v}{\exists} (u \in X \wedge v \in X \wedge (\text{color}[u] = 0) \wedge (\text{color}[v] = 1)))$.

Formula (1) is simplified to:

(3) $\underset{X}{\forall}$ (domino-on-board$[X] \Rightarrow X \subseteq \text{Board}) \wedge \underset{X}{\forall} \Big($domino-on-board$[X] \Rightarrow \underset{x,y}{\exists} (x \in X \wedge y \in X \wedge (x \neq y) \wedge \text{adjacent}[x, y])\Big)$.

By (3.2), we can take an appropriate Skolem function such that

(4) $\underset{X}{\forall}$ (domino-on-board$[X] \Rightarrow x_0[X] \in X \wedge y_0[X] \in X \wedge (x_0[X] \neq y_0[X]) \wedge$ adjacent$[x_0[X], y_0[X]]$),

We assume

(5) domino-on-board$[X_0]$,

and show

(6) $\underset{u,v}{\exists} (u \in X_0 \wedge v \in X_0 \wedge (\text{color}[u] = 0) \wedge (\text{color}[v] = 1))$.

Formula (5), by (3.1), implies:

(7) $X_0 \subseteq \text{Board}$.

From (7) we can infer

(8) $\underset{X1}{\forall} (X1 \in X_0 \Rightarrow X1 \in \text{Board})$.

Formula (5), by (4), implies:

(9) adjacent$[x_0[X_0], y_0[X_0]] \wedge x_0[X_0] \in X_0 \wedge y_0[X_0] \in X_0 \wedge (x_0[X_0] \neq y_0[X_0])$.

From (9.2) we can infer

(10) $X_0 \neq \{\}$.

Formula (9.1), by (Proposition (adjacent of black are white)), implies:

(12) $(\text{color}[x_0[X_0]] = 0) \Rightarrow (\text{color}[y_0[X_0]] = 1)$.

Formula (9.1), by (Proposition (adjacent of white are black)), implies:

(13) $(\text{color}[x_0[X_0]] = 1) \Rightarrow (\text{color}[y_0[X_0]] = 0)$.

Formula (9.2), by (8), implies:

$x_0[X_0] \in \text{Board}$,

which, by (Proposition (zero or one)), implies:

(14) $(\text{color}[x_0[X_0]] = 0) \vee (\text{color}[x_0[X_0]] = 1)$.

Formula (9.3), by (8), implies:

$y_0[X_0] \in \text{Board}$, which, by (Proposition (zero or one)), implies:

(15) $(\text{color}[y_0[X_0]] = 0) \vee (\text{color}[y_0[X_0]] = 1)$.

We prove (6) by case distinction using (15).
Case (15.1) $\text{color}[y_0[X_0]] = 0$:
We prove (6) by case distinction using (14).
Case (14.1) $\text{color}[x_0[X_0]] = 0$:
Formula (14.1), by (12), implies:
$\text{color}[y_0[X_0]] = 1$,
which, by (15.1), implies:

(17) $0 = 1$.

Using available computation rules we can simplify the knowledge base:
Formula (17) simplifies to

(18) *False*,

Formula (6) is true because the assumption (18) is false.
Case (14.2) $\text{color}[x_0[X_0]] = 1$:
Now, let $u := y_0[X_0]$. Thus, for proving (6) it is sufficient to prove:

(20) $\underset{v}{\exists} (y_0[X_0] \in X_0 \wedge v \in X_0 \wedge (\text{color}[y_0[X_0]] = 0) \wedge (\text{color}[v] = 1))$.

Using available computation rules we evaluate (20) using (15.1), (14.2), (10), and (9.4) as additional assumption(s) for simplification:

(24) $\underset{v}{\exists} (y_0[X_0] \in X_0 \wedge v \in X_0 \wedge (\text{color}[v] = 1))$.

Now, let $v := x_0[X_0]$. Thus, for proving (24) it is sufficient to prove:

(25) $y_0[X_0] \in X_0 \wedge x_0[X_0] \in X_0 \wedge (\text{color}[x_0[X_0]] = 1)$.

Using available computation rules we evaluate (25) using (15.1), (14.2), (10), and (9.4) as additional assumption(s) for simplification:

(27) $y_0[X_0] \in X_0 \wedge x_0[X_0] \in X_0$.

We prove the individual conjunctive parts of (27):
Proof of (27.1) $y_0[X_0] \in X_0$:

Formula (27.1) is true because it is identical to (9.3).
Proof of (27.2) $x_0[X_0] \in X_0$:
Formula (27.2) is true because it is identical to (9.2).
Case (15.2) color$[y_0[X_0]] = 1$:
We prove (6) by case distinction using (14).
Case (14.1) color$[x_0[X_0]] = 0$:
... (similar to above)
Case (14.2) color$[x_0[X_0]] = 1$:
... (similar to above) □

We can now study also some interactions between dominoes, adjacency, and colors. From here on, we list the propositions and leave out some of the proofs due to lack of space. For complete proofs of the propositions see (Windsteiger, 2001)].

Two fields in a domino must be either identical or adjacent.

Proposition[“same or adjacent in domino”, any$[X, x, y]$,
domino-on-board$[X] \wedge x \in X \wedge y \in X \Rightarrow x = y \vee$ adjacent$[x, y]$]]

One domino can neither contain two distinct black fields nor two distinct white fields.

Proposition[“black in domino unique”, any$[X, x, y]$,
domino-on-board$[X] \wedge x \in X \wedge y \in X \wedge$ color$[x] = 0 \wedge$ color$[y] = 0 \Rightarrow x = y$]

Proposition[“white in domino unique”, any$[X, x, y]$,
domino-on-board$[X] \wedge x \in X \wedge y \in X \wedge$ color$[x] = 1 \wedge$ color$[y] = 1 \Rightarrow x = y$]

4. Exploration Round 3: Partial Coverings

Definition[“Partial Covering”, any$[z]$,
$$\text{partial-covering}[z] :\Leftrightarrow \underset{x \in z}{\forall}\ \text{domino-on-board}[x] \wedge \underset{\substack{x,y \in z \\ x \neq y}}{\forall} (x \cap y = \emptyset)$$]

The picture to have in mind of a partial covering is that of a set of non-overlapping dominoes. Since we have a clear understanding of dominoes from the previous exploration we skip computations with partial coverings. In this exploration round, we are heading towards the second statement in McCarthy’s proof, which, sloppily formulated, says that in a partial covering there are as many black fields as there are white fields. Intuitively, one thinks to have a very clear understanding of the notion “cardinality” that captures the “number of elements in a set”, in particular if the sets under consideration are “finite sets”.

However, the definition of cardinality of sets tells us that in order to prove that two sets have equal cardinality we have to find a bijective mapping from one set into the other. Alternatively, we could implement a special prover for cardinality that bears in it the clear understanding on finite cardinalities in form of special inference rules. In the spirit of the transition of knowledge from the knowledge base to the level of inference rules, see (Windsteiger, 2001), this would happen after having completed the exploration of the cardinality notion in set theory. We have not implemented such a prover yet, thus, we have to construct the bijection. Before we start proving, we again check by computation whether the statement of interest at least holds in some example:

Definition["pc",
$pc := \{\{\langle 1,2\rangle, \langle 1,3\rangle\}, \{\langle 2,2\rangle, \langle 2,3\rangle\}, \{\langle 3,3\rangle, \langle 3,4\rangle\}, \{\langle 2,1\rangle, \langle 3,1\rangle\}, \{\langle 3,2\rangle, \langle 4,2\rangle\}\}$]

`Compute`[partial-covering[pc],using→ Definition["pc"]]

$$True$$

`Compute`[$\{u \underset{u\in\bigcup pc}{|} \text{color}[u] = 0\}$, using→ Definition["pc"]]

$$\{\langle 1,3\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 4,2\rangle\}$$

`Compute`[$\{u \underset{u\in\bigcup pc}{|} \text{color}[u] = 1\}$, using→ Definition["pc"]]

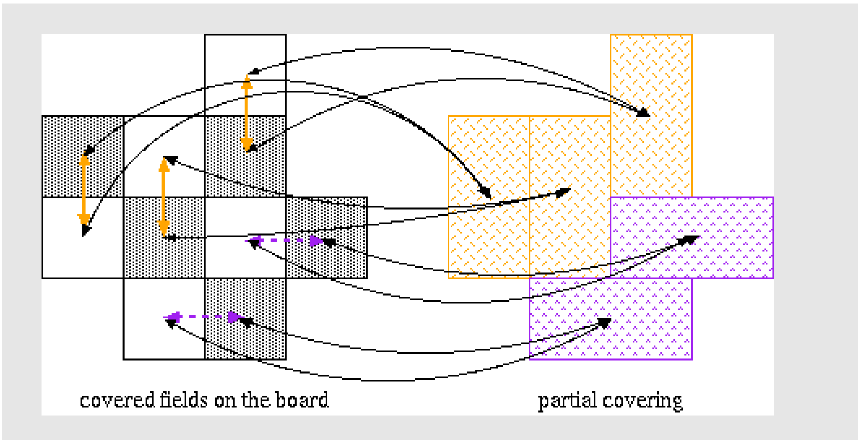
$$\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle, \langle 3,4\rangle\}$$

`Compute`[$|\{u \underset{u\in\bigcup pc}{|} \text{color}[u] = 0\}| = |\{u \underset{u\in\bigcup pc}{|} \text{color}[u] = 1\}|$,
using → Definition["pc"]]

$$True$$

At least in this example, it is really true that there are equally many black fields as white fields in the covering. Note, that this property does certainly not hold for any set of fields on a board, as one can easily verify by a counter-example.

Definition["no pc",
$no - pc := \{\{\langle 1,2\rangle, \langle 1,3\rangle\}, \{\langle 2,2\rangle, \langle 2,3\rangle\}, \{\langle 3,3\rangle, \langle 3,4\rangle\}, \{\langle 2,1\rangle, \langle 3,1\rangle\}, \{\langle 3,1\rangle, \langle 4,1\rangle\}\}$]



`Compute`[partial-covering[no-pc],using→ Definition[“no pc”]]

$$\mathit{False}$$

`Compute`[$\{u \underset{u \in \cup no-pc}{|} \text{color}[u] = 0\}$, using→ Definition[“no pc”]]

$$\{\langle 1,3\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,3\rangle\}$$

`Compute`[$\{u \underset{u \in \cup no-pc}{|} \text{color}[u] = 1\}$, using→ Definition[“no pc”]]

$$\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 4,1\rangle\}$$

An observation, which can be made in the computations is, that the number of black fields corresponds to the number of dominoes in the partial covering! The same is true for the number of white fields, which enforces the number of black fields to coincide with the number of white fields. This idea is also depicted in Figure 1.

We therefore investigate the interplay between covered fields on the board and dominoes contained in the partial covering. A first observation is the uniqueness of the domino in the covering, in which a covered field is contained.

Proposition[“covering maps to unique domino”, any[partial-covering[z]],

$$\underset{u \in \cup z}{\forall} \ \underset{\substack{x,y \in z \\ u \in x \wedge u \in y}}{\forall} x = y$$]

We can even show for each covered field on the board the unique existence of a adomino containing this field. Note that we must formulate the uniqueness property explicitly by means of predicate logic, because the predicate logic prover does not yet contain special inference rules for the ∃!-quantifier (there exists a unique).

Proposition[“covering defines unique domino”, any[partial-covering[z]],
$\underset{u\in\cup z}{\forall}\ \underset{x\in z}{\exists}\ u \in x \wedge \text{domino-on-board}[x] \wedge \underset{y\in z}{\forall}\ u \in y \Rightarrow x = y$]

`Prove`[Proposition[“covering defines unique domino”],
using → Definition[“Partial Covering”]]

Prove:

(Proposition (covering defines unique domino)) . . .,

under the assumption:

(Definition (Partial Covering)) . . ..

. . .

We assume

(4) $u_0 \in \cup z_0$,

and show

(5) $\underset{x}{\exists}\ x \in z_0 \wedge u_0 \in x \wedge \text{domino-on-board}[x] \wedge \underset{y}{\forall}\ y \in z_0 \Rightarrow u_0 \in y \Rightarrow x = y$.

From (4) we know by definition of the big ∪-operator that we can choose an appropriate value such that

(6) $z1_0 \in z_0$,

(7) $u_0 \in z1_0$.

Now, let $x := z1_0$. Thus, for proving (5) it is sufficient to prove:

(10) $z1_0 \in z_0 \wedge u_0 \in z1_0 \wedge \text{domino-on-board}[z1_0] \wedge \underset{y}{\forall}\ (y \in z_0 \Rightarrow (u_0 \in y \Rightarrow (z1_0 = y)))$.

We prove the individual conjunctive parts of (10):
Formula (10.1) is true because it is identical to (6).
Formula (10.2) is true because it is identical to (7).
Formula (2), by (Definition (Partial Covering)), implies:

(12) $\underset{x}{\forall}\ (x \in z_0 \Rightarrow \text{domino-on-board}[x]) \wedge \underset{x,y}{\forall}\ (x \in z_0 \wedge y \in z_0 \wedge (x \neq y) \Rightarrow (x \cap y = \{\}))$.

Formula (6), by (12.1), implies:

(13) domino-on-board[$z1_0$].

Formula (10.3) is true because it is identical to (13).
Proof of (10.4) $\underset{y}{\forall} (y \in z_0 \Rightarrow (u_0 \in y \Rightarrow (z1_0 = y)))$:

We assume

(14) $y_0 \in z_0$,

We assume

(17) $u_0 \in y_0$

and show

(18) $z1_0 = y_0$.

From (17) together with (7) we know

(20) $u_0 \in y_0 \cap z1_0$.

From (20) we can infer

(21) $y_0 \cap z1_0 \neq \{\}$.

Formula (2), by (Definition (Partial Covering)), implies:

(22) $\underset{x}{\forall} (x \in z_0 \Rightarrow \text{domino-on-board}[x]) \wedge \underset{x,y}{\forall} (x \in z_0 \wedge y \in z_0 \wedge (x \neq y) \Rightarrow (x \cap y = \{\}))$.

Formula (21), by (22.2), implies:

(23) $\neg(y_0 \in z_0 \wedge z1_0 \in z_0 \wedge (y_0 \neq z1_0))$.

Formula (23) is simplified to

(25) $(y_0 \notin z_0) \vee (z1_0 \notin z_0) \vee (y_0 = z1_0)$,

From (14) and (25) we obtain by resolution

(26) $(z1_0 \notin z_0) \vee (y_0 = z1_0)$.

From (6) and (26) we obtain by resolution

(27) $y_0 = z1_0$.

Using available computation rules we evaluate (18) using (27), (21), (19), (8), and (9) as additional assumption(s) for simplification:

$z1_0 = y_0 \triangleright$

$z1_0 = y_0 \triangleright$ (by Mathematica's FullSimplify)

True.

(28) *True*.

Formula (28) is true because it is the constant True. □

(Note the final step in the last proof: we could verify $z1_0 = y_0$ *by computation*, because the standard setting of the option "*SimplifyFormula* → *True*" applies term simplification using additionally certain assumptions from the knowledge base. Essentially, we use the Mathematica built-in FullSimplify for term simplification, which allows also assumptions to be used during simplification, where we pass all number arithmetic related formulae from the knowledge base. In the proof above, this yields the simplification problem $z1_0 = y_0$ under the assumption $y_0 = z1_0$ (from formula (27)), which yields True. In the above example, it would not be necessary to use FullSimplify in order to validate $z1_0 = y_0$ but in other examples it proves useful.) Now, since for all $u \in \cup z$, where z is a partial covering, there exists a *unique* $x \in z$ with $u \in x \wedge$ domino-on-board$[x]$, we may define a function using the "such that" quantifier $\ni$. We will address this domino as "*the* covering domino of u".

Definition["Covering Domino", any[u, partial-covering[z]], with[$u \in \cup z$]
covering-domino$_z[u] := \underset{x \in z}{\ni}\ u \in x \wedge$ domino-on-board$[x]$]

The domino that contains a covered field is *the* covering domino of just this field.

Proposition["own covering domino", any[x, partial-covering[z], domino-on-board[d]],
$x \in d \wedge d \in z \wedge x \in \cup z \Rightarrow$ covering-domino$_z[x] = d$)]

`Prove`[Proposition["own covering domino"],
using→ ⟨ Proposition["covering maps to unique domino"],
Definition["Covering Domino"] ⟩]

Prove (we only show the parts dealing with the $\ni$-quantifier):
...
Formula (Definition (Covering Domino)) simplifies to

(2) $\underset{u,z}{\forall}$ partial-covering$[z] \wedge u \in \cup z \Rightarrow$ covering-domino$_z[u] := \underset{x}{\ni}\ u \in x \wedge$ domino-on-board$[x] \wedge x \in z$,

From (2) we can infer by expansion of the "such that"-quantifier

(3) $\underset{u,z}{\forall}$ (partial-covering$[z] \wedge u \in \cup z \Rightarrow u \in$ covering-domino$_z[u] \wedge$ domino-on-board[covering-domino$_z[u]] \wedge$ covering-domino$_z[u] \in z$).

...
Proof of (12.6) covering-domino$_{z_0}[x_0] \in z_0$:
Formula (6.3), by (3), implies:

(15) partial-covering$[z_0] \Rightarrow$ domino-on-board[covering-domino$_{z_0}[x_0]] \wedge$
$x_0 \in$ covering-domino$_{z_0}[x_0] \wedge$ covering-domino$_{z_0}[x_0] \in z_0$.

From (4.1) and (15) we obtain by modus ponens

(16) domino-on-board[covering-domino$_{z_0}[x_0]] \wedge x_0 \in$ covering-domino$_{z_0}[x_0] \wedge$
covering-domino$_{z_0}[x_0] \in z_0$.

Formula (12.6) is true because it is identical to (16.3). □

If two covered black (white respectively) fields have the same covering domino in a partial covering, they must be identical.

Proposition[“black and same covering domino:identical”, any$[u, v,$ partial-covering$[z]]$,
$$\text{covering-domino}_z[u] = \text{covering-domino}_z[v] \wedge u \in \cup z \wedge v \in \cup z \wedge \text{color}[u] = 0 \wedge \text{color}[v] = 0 \Rightarrow u = v \quad]$$

Proposition[“white and same covering domino:identical”, any$[u, v,$ partial-covering$[z]]$,
$$\text{covering-domino}_z[u] = \text{covering-domino}_z[v] \wedge u \in \cup z \wedge v \in \cup z \wedge \text{color}[u] = 1 \wedge \text{color}[v] = 1 \Rightarrow u = v \quad]$$

The covering-domino function is a bijective mapping from the set of covered black fields to the set of dominoes in a partial covering.

Lemma[“covering domino bijective from black fields”, any[partial-covering$[z]]$,
$$\text{covering-domino}_z :: \{u_{u \in \cup z} \mid \text{color}[u] = 0\} \overset{bij}{\to} \{d_{d \in z} \mid \text{domino-on-board}[d]\} \quad]$$

`Prove`[Lemma[“covering domino bijective from black fields”],
using→ ⟨Proposition[“own covering domino”], Lemma[“different color”],
Proposition[“black and same covering domino: identical”] ⟩]

Prove:

(Lemma (covering domino bijective from black fields)) . . .,

under the assumptions:

(Proposition (own covering domino)) . . .,

(Lemma (different color)) . . .,

(Proposition (black and same covering domino: identical)) . . ..

Using available computation rules we can simplify the knowledge base: Formula (Lemma (different color)) simplifies to

(1) $\underset{X}{\forall}$ (domino-on-board$[X] \Rightarrow \underset{u,v}{\exists} (u \in X \wedge v \in X \wedge (\text{color}[u] = 0) \wedge (\text{color}[v] = 1)))$,

By (1), we can take an appropriate Skolem function such that

(2) $\underset{X}{\forall}$ (domino-on-board$[X] \Rightarrow u_0[X] \in X \wedge v_0[X] \in X \wedge (\text{color}[u_0[X]] = 0) \wedge (\text{color}[v_0[X]] = 1))$,

We assume

(3) partial-covering$[z_0]$,

and show

(4) $\text{covering-domino}_{z_0} :: \{\underset{u}{u|}\ u \in \cup z_0 \wedge (\text{color}[u] = 0)\} \overset{bij}{\rightarrow} \cdot \{\underset{d}{d|}\ \text{domino-on-board}[d] \wedge d \in z_0\}$.

In order to show that $\text{covering-domino}_{z_0}$ in (4) is bijective, we have to prove:
Injectivity of $\text{covering-domino}_{z_0}$:

(5) $\text{covering-domino}_{z_0} :: \{\underset{u}{u|}\ u \in \cup z_0 \wedge (\text{color}[u] = 0)\} \overset{inj}{\rightarrow} \cdot \{\underset{d}{d|}\ \text{domino-on-board}[d] \wedge d \in z_0\}$.

In order to show injectivity of $\text{covering-domino}_{z_0}$ in (5) we assume

(7) $u3_0 \in \{\underset{u}{u|}\ u \in \cup z_0 \wedge (\text{color}[u] = 0)\}$,

(8) $u4_0 \in \{\underset{u}{u|}\ u \in \cup z_0 \wedge (\text{color}[u] = 0)\}$,

(9) $\text{covering-domino}_{z_0}[u3_0] = \text{covering-domino}_{z_0}[u4_0]$.

and show

(10) $u3_0 = u4_0$.

From what we already know follows:
From (7) we can infer

(11) $u3_0 \in \cup z_0 \wedge (\text{color}[u3_0] = 0)$.

From (8) we can infer

(12) $u4_0 \in \cup z_0 \wedge (\text{color}[u4_0] = 0)$.

Formula (10), using (Proposition (black and same covering domino: identical)), is implied by:

(17) $\underset{z}{\exists}\ (\text{partial-covering}[z] \wedge u3_0 \in \cup z \wedge u4_0 \in \cup z \wedge (\text{color}[u3_0] = 0) \wedge (\text{color}[u4_0] = 0) \wedge (\text{covering-domino}_z[u3_0] = \text{covering-domino}_z[u4_0]))$.

Using available computation rules we evaluate (17) using (11.2), (12.2), and (9) as additional assumption(s) for simplification:

(18) $\underset{z}{\exists}\ (\text{partial-covering}[z] \wedge u3_0 \in \cup z \wedge u4_0 \in \cup z \wedge (\text{covering-domino}_z[u3_0] = \text{covering-domino}_z[u4_0]))$.

Now, let $z := z_0$. Thus, for proving (18) it is sufficient to prove:

(19) $\text{partial-covering}[z_0] \wedge u3_0 \in \cup z_0 \wedge u4_0 \in \cup z_0 \wedge (\text{covering-domino}_{z_0}[u3_0] = \text{covering-domino}_{z_0}[u4_0])$.

Using available computation rules we evaluate (19) using (11.2), (12.2), and (9) as additional assumption(s) for simplification:

(20) $\text{partial-covering}[z_0] \wedge u3_0 \in \cup z_0 \wedge u4_0 \in \cup z_0$.

We prove the individual conjunctive parts of (20):
Formula (20.1) is true because it is identical to (3).
Formula (20.2) is true because it is identical to (11.1).
Formula (20.3) is true because it is identical to (12.1).
Surjectivity of covering-domino$_{z_0}$:

(6) covering-domino$_{z_0}$ :: $\{u \underset{u}{|} u \in \cup z_0 \wedge (\text{color}[u] = 0)\} \overset{surj}{\rightarrow} \{d \underset{d}{|} \text{domino-on-board}[d] \wedge d \in z_0\}$.

In order to show surjectivity of covering-domino$_{z_0}$ in (6) we assume

(21) $d1_0 \in \{d \underset{d}{|} \text{domino-on-board}[d] \wedge d \in z_0\}$,

and show

(22) $\underset{u5}{\exists}\ u5 \in \{u \underset{u}{|} u \in \cup z_0 \wedge (\text{color}[u] = 0)\} \wedge (\text{covering-domino}_{z_0}[u5] = d1_0)$.

From what we already know follows:
From (21) we can infer

(23) domino-on-board$[d1_0] \wedge d1_0 \in z_0$.

In order to prove (22) we have to show:

(24) $\underset{u5}{\exists}\ ((u5 \in \cup z_0 \wedge (\text{color}[u5] = 0)) \wedge (\text{covering-domino}_{z_0}[u5] = d1_0))$.

Using available computation rules we evaluate (24):

(25) $\underset{u5}{\exists}\ (u5 \in \cup z_0 \wedge (\text{color}[u5] = 0) \wedge (\text{covering-domino}_{z_0}[u5] = d1_0))$.

Formula (23.1), by (2), implies:

(26) $u_0[d1_0] \in d1_0 \wedge v_0[d1_0] \in d1_0 \wedge (\text{color}[u_0[d1_0]] = 0) \wedge (\text{color}[v_0[d1_0]] = 1)$.

Now, let $u5 := u_0[d1_0]$. Thus, for proving (25) it is sufficient to prove:

(27) $u_0[d1_0] \in \cup z_0 \wedge (\text{color}[u_0[d1_0]] = 0) \wedge (\text{covering-domino}_{z_0}[u_0[d1_0]] = d1_0)$.

Using available computation rules we evaluate (27) using (26.3)and (26.4) as additional assumption(s) for simplification:

(28) $u_0[d1_0] \in \cup z_0 \wedge (\text{covering-domino}_{z_0}[u_0[d1_0]] = d1_0)$.

We prove the individual conjunctive parts of (28):
In order to show (28.1) we have to show

(29) $\underset{z3}{\exists}\ (u_0[d1_0] \in z3 \wedge z3 \in z_0)$.

Now, let $z3 := d1_0$. Thus, for proving (29) it is sufficient to prove:

(30) $u_0[d1_0] \in d1_0 \wedge d1_0 \in z_0$.

We prove the individual conjunctive parts of (30):
Formula (30.1) is true because it is identical to (26.1).

Formula (30.2) is true because it is identical to (23.2).
Formula (28.2), using (Proposition (own covering domino)), is implied by:

(31) domino-on-board$[d1_0] \wedge$ partial-covering$[z_0] \wedge d1_0 \in z_0 \wedge u_0[d1_0] \in d1_0 \wedge u_0[d1_0] \in \cup z_0$.

We prove the individual conjunctive parts of (31):
Formula (31.1) is true because it is identical to (23.1).
Formula (31.2) is true because it is identical to (3).
Formula (31.3) is true because it is identical to (23.2).
Formula (31.4) is true because it is identical to (26.1). In order to show (31.5) we have to show

(32) $\underset{z4}{\exists} (u_0[d1_0] \in z4 \wedge z4 \in z_0)$.

Now, let $z4 := d1_0$. Thus, for proving (32) it is sufficient to prove:

(33) $u_0[d1_0] \in d1_0 \wedge d1_0 \in z_0$.

We prove the individual conjunctive parts of (33):
Formula (33.1) is true because it is identical to (26.1).
Formula (33.2) is true because it is identical to (23.2). □

This, of course, was the key step for finally proving equal cardinality of the set of white fields and the set of black fields in a partial covering, because an analog proof yields a bijection from the white covered fields to the set of dominoes, which proves equal cardinality of black and white covered fields.

Corollary["covering domino bijective from white fields", any[partial-covering[z]],

$\quad$ covering-domino$_z$ :: $\{u \underset{u \in \cup z}{|} \text{color}[u] = 1\} \overset{bij}{\rightarrow} \{d \underset{d \in z}{|} \text{domino-on-board}[d]\}$]

There are as many black covered fields on the board as there are dominoes in the partial covering.

Corollary["number of dominoes", any[partial-covering[z]],

$$|\{u \underset{u \in \cup z}{|} \text{color}[u] = 0\}| = |\{d \underset{d \in z}{|} \text{domino-on-board}[d]\}|$$
$$|\{u \underset{u \in \cup z}{|} \text{color}[u] = 1\}| = |\{d \underset{d \in z}{|} \text{domino-on-board}[d]\}|$$]

Lemma["equally many black and white covered fields", any[any[z]],

$\quad$ partial-covering[z] $\Rightarrow |\{u \underset{u \in \cup z}{|} \text{color}[u] = 0\}| = |\{u \underset{u \in \cup z}{|} \text{color}[u] = 1\}|$]

This lemma corresponds to the second unproved statement of McCarthy's four-line proof. See (Windsteiger, 2001) for the complete proofs.

Exploration Round 4: The Final Theorem

Now we can go for the final theorem, which, after having explored dominoes, colors, and partial coverings extensively, follows rather easily from the Lemma on the number of black and white covered fields proven at the end of the previous exploration round.

The mutilated checkerboard cannot be covered by dominoes.

Theorem[“mutilated checkerboard”,
$\neg\underset{z}{\exists}$ (partial-covering$[z] \wedge \bigcup z =$ Mutilated-Board)]

`Prove`[Theorem[“mutilated checkerboard”],
using → ⟨Lemma[“equally many black and white covered fields”]⟩,
built-in → ⟨ Built-in[“Tuples”][Subscript, AngleBracket], Built-in[“Numbers”],
Definition[“Board”], Definition[“color”]⟩]

Prove:

(Theorem (mutilated checkerboard)) $\neg\underset{z}{\exists}$ (partial-covering$[z] \wedge \bigcup z =$ Mutilated-Board),

under the assumption:

(Lemma (equally many black and white covered fields)) . . ..

We prove (Theorem (mutilated checkerboard)) by contradiction. We assume

(1) $\underset{z}{\exists}$ (partial-covering$[z] \wedge \cup z =$ Mutilated-Board),

and show *a contradiction*. Formula (1) simplifies to

(2) $\underset{z}{\exists}$ (partial-covering$[z] \wedge (\cup z = \{\langle 0,1\rangle, \langle 0,2\rangle, << 58 >>, \langle 7,5\rangle, \langle 7,6\rangle\})$),

By (2) we can take appropriate values such that:

(3) partial-covering$[z_0] \wedge (\cup z_0 = \{\langle 0,1\rangle, \langle 0,2\rangle, << 58 >>, \langle 7,5\rangle, \langle 7,6\rangle\})$.

Formula (3.1), by (Lemma (equally many black and white covered fields)), implies:

$|\{u \underset{u \in \cup z_0}{|} \text{color}[u] = 0\}| = |\{u \underset{u \in \cup z_0}{|} \text{color}[u] = 1\}|$, which, by (3.2), implies:

$$(16)\quad \begin{array}{l} |\{u \underset{u \in \cup\{\langle 0,1\rangle,\langle 0,2\rangle,<<58>>,\langle 7,5\rangle,\langle 7,6\rangle\}}{|} \text{color}[u] = 0\}| = \\ \qquad |\{u \underset{u \in \cup\{\langle 0,1\rangle,\langle 0,2\rangle,<<58>>,\langle 7,5\rangle,\langle 7,6\rangle\}}{|} \text{color}[u] = 1\}| \end{array}.$$

Formula (16) simplifies to

(17) *False*,

Formula (a contradiction) is true because the assumption (17) is false. □

Conclusion

We presented a complete exploration of the checkerboard, dominoes, and partial coverings on a board, finally leading to a complete solution of the Mutilated Checkerboard Problem of McCarthy. This exploration is at the same time a nice demonstration of cooperation of proving and computing, which shows a particular strength of the *Theorema* system, namely the uniform language of expressions that allows proving and computing in the same logical frame without having need of translation between different representations suited for proving or computing. It thereby proves both

- the suitability of *Theorema* as a system supporting the entire cycle of mathematical research work and
- the power of the *Theorema* Set Theory prover as a specialized tool for tackling problems formulated in set theory.

In particular, we want to point out the two-fold intergration between proving and computing in the *Theorema* system. On the one hand, proving and computing are integrated by offering the user commands Prove and Compute embedded in a coherent language frame. On the other hand, the Set Theory prover incorporates *computation* as an *inference step* for finite language constructs, like for instance the finite set representing the mutilated checkerboard in the proof of the main theorem above. The computation in the proof of the theorem can of course only be performed for a given size of the checkerboard, in the above case 8 by 8. For arbitrary size, the final proof cannot be done by computation but it would require some reasoning on *cardinalities* of (finite) sets. This place is, however, the only place in the entire exploration, where the concrete size of the checkerboard is used in a proof, thus, generalization to the n by n case would require only providing additional knowledge on finite cardinalities in order to show that the number of white fields can't be equal to the number of black fields on the mutilated board.

References

P. Andrews and M. Bishop. On Sets, Types, Fixed Points, and Checkerboards. In Pierangelo Miglioli and Ugo Moscato and Daniele Mundici and Mario Ornaghi, editor, *Theorem Proving with Analytic Tableaux and Related Methods. 5th International Workshop, TABLEAUX '96, Palermo*, number 1071 in LNAI, pages 1-15, Springer, May 1996.

G. Bancerek. The Mutilated Chessboard Problem - checked by MIZAR. In Roman Matuszewski, editor, *The QED Workshop II*, pages 25-26, Warshaw University. 1995. Technical Report L/1/95.

B. Buchberger. Mathematica: A System For Doing Mathematics by Computer?.

In A. Miola and M. Temperini, editor, *Advances in the Design of Symbolic Computation Systems*, pages 2-29, Springer Verlag, Wien - New York, 1997. Invited Talk at DISCO 93, Gmunden, Austria.

B. Buchberger. Theory Exploration Versus Theorem Proving. In A. Armando and T. Jebelean, editor, *Electronic Notes in Theoretical Computer Science*, volume 23-3 of pages 67-69, Elsevier, 1999. CALCULEMUS Workshop, University of Trento, Trento, Italy.

J. McCarthy. A Tough Nut for Proof Procedures. Stanford AI project memo, 1964.

J. McCarthy. The Mutilated Checkerboard in Set Theory. In Roman Matuszewski, editor, *The QED Workshop II*, pages 25-26, Warshaw University. 1995. Technical Report L/1/95.

W. McCune. Another Crack in a Tough Nut. *Association for Automated Reasoning Newsletter*, (31):1-3, 1995.

S. Subramanian. *A Mechanized Framework for Specifying Problem Domains and Verifying Plans*. University of Austin, Texas, 1993.

S. Subramanian. An interactive solution to the n x n mutilated checkerboard problem. *Journal of Logic and Computation*, 6(4):573-598, August 1996.

Wolfgang Windsteiger. *A Set Theory Prover in Theorema: Implementation and Practical Applications*. RISC Institute, 2001.