Reflections Vol. 1 (2015) 1–100

Structural Logic

Wolfgang Scherer `Wolfgang.Scherer@gmx.de`

Abstract

A structural logic is developed based on mutual exclusive states and clauses, abstracting from propositions and truth values. It is mainly useful for computer-aided research of structural properties of propositional formulae.

KEYWORDS: *structural logic, satoku matrix, boolean satisfiability, selection problem*

Contents

1. Pre-Preface

Since Aristotle, logic[wiki-logic] has been tampered with in various ways.

references?

The "Law of Excluded Middle" for instance has been discussed at great length,and has lead to many-valued logics[wiki-mvl] and very prominently to constructive logic[wiki-cl]. A not so prominent discussion questions the "Law of Non-Contradiction", e.g. dialethism[sep-dia] or generally paraconsistent logic[sep-para].

Even the deductive rules have been generalized in proof theory[sep-proof] and formalized in sequent calculus[wiki-seq].

The propositional calculus[wiki-prop] has been successfully axiomatized only with propositional variables and operator symbols, without the notion of truth values at all.

A peculiar division of logic into sub-systems happened with mathematical logic[wiki-ml]. The nicely chainable unary and binary operators of logic are used for propositional calculus. The stubbornly unchainable function of choice (single selection) has been handed off to graph theory[wiki-graph][1.].

Most of these logical endeavors hold on to the notion of propositions as atomical objects holding truth values, with the exception of graph theory, where propositional states are ususally mapped to nodes and their conflict relationships are mapped to edges.

Graph theory comes closest to dealing with the structure of logical formulae. However, the structural elements of propositional logic, namely clauses, are thrown away as (mathematically) not necessary.

It is between propositional logic and graph theory, where structural logic fits in.

1. In the author's opinion the fact that graph theory deals with n-ary selections should be mentioned far more prominently.

2. Preface

In order to dampen possible expectations, it must be said that structural logic is not intended and does not give an answer to the P vs. NP problem. On a philosophical level it indicates that NP-completeness in regard to boolean statisfiability is basically a homemade problem based on repurposing an unsuitable model of logic. On a practical level, it indicates that the problem can be at the very least controlled, if not completely avoided in computer programs.

Discarding information that is not strictly necessary, is perfectly valid and quite common in mathematical reasoning. However, there are no objective criteria as to which information may be relevant in any given context or not. The implications *necessary* $\rightarrow$*relevant*, *not necessary* $\rightarrow$*irrelevant* are at best unsubstantiated.

Graph theory shows, that logic works with the absolute minimum requirements of unspecified objects and a binary relation (any 2 nodes are either connected or not), notably without any atomic propositional variables at all. However, the price is a significant reduction of explicit structure.

Since structure cannot be destroyed without removing the relational information between objects, the structure must still be there implicitely. Transforming a propositional problem into an independent set problem[MOUNT], encodes the clausal structure implicitely and then simply ignores the explicit structural information provided by clauses. The process is equivalent to reencoding a CNF problem as single choice problem in direct encoding, which preserves clausal integrity in the at-least-one clauses. However, the worst case of structural decomposition is the transformation of a propositional problem into an edge cover problem (keeping only the at-most-one clauses) and then trying to find a k-egde cover. This essentially requires, that the original clause structure is indirectly rediscovered[2.].

So the question arises, why relevant information is destroyed in the first place, only to miss its usefulness later on. Beyond the simple facts, that graph theory is defined without structural components and that it is well-researched, there is no objective reason, why plain graph theory should be the appropriate model of logic to examine certain problems or why there should even be an "effective" graph theoretic answer to structural questions.

Due to the inherent self-referential nature of logic, all logical problems can be mapped to subsets of logical expressions. And again, there is no objective reason to prefer any one subset to another one. So, while NAND is sufficient to express all logical problems, a human is utterly incapable of doing even the simplest reasoning with NAND alone. However, NAND-logic is extremely useful in electronics. Translating

2. Since the edge cover problem for an independent set problem appears in the right upper quadrant of a satoku matrix mapped with variable representations, not all hope is lost for reconstructing the original clauses.

AND, OR, NEG problems into NAND representations is ususally a one way street. Similar to reencoding a CNF problem in direct encoding a good amount of structural information is lost. However, at least a partial recovery into something discernible is possible, although quite tedious.

Mathematical logic was developed in an attempt to make logical problems more tractable through mathematical methods. Hence the definition with propositional variables, clauses and binary logical functions. It was also defined with the limits of a human brain in mind and tacitly with the limits of pencil and paper. The discovery, that loss of structure — originating from translating highly stuctured problems into simple elements — causes difficulties should not be surprising.

However, the expectation to recover higher-order structure with the very tools that destroyed the structure in the first place is somewhat strange. So, why translate a system of linear equations into the netherlands of a CNF formula, when there are perfectly appropriate methods to solve them more effectively? Be that as it may, when the deed is done, there is really no choice other than fixing it. The question is, what tools are needed to complete the job?

One thing is for sure. Human brains are not the best choice. Computers would be better, if they were not so unintelligent. But there is no model of logic defined exclusively for reasoning with computer programs. There is an unsubstantiated assumption, that the models that are good for humans must also be good for computers, but the example of NAND-logic already shows, that what is bad for humans is not necessarily bad for computers. Therefore any human-centric assumptions about logic simply cannot be applied uncritically to computer programs.

The other thing that can be asserted, is that a hammer is good for breaking things. It is not good for mending things.

3. Introduction

After discovering the satoku matrix and not changing a single basic operation in years, the author went through several cycles of explanatory attempts and found that it is simply not possible to express the mechanics of the satoku algorithms in any existing model of logic without loosing all meaning. Hence the decision to develop a model of logic with the sole purpose of conducting computer-aided structural reasoning. It is by all means impossible to reason in structural logic without a computer program, but the computer aid is nothing more substantial than pencil and paper.

Structural logic is complete, since all propositional problems can be readily translated into the required representation. The representation can be interpreted as a tranformation of CNF-formulas into an inverted adjacency matrix keeping the structural information provided by propositional clauses. But that is the point where the mapping ends. Some graph theoretic achievements like a polynomial time algorithm to solve 2-SAT problems come effortlessly in structural logic — though not as "effective" as in plain graph theory due to the intrinsic space requirements of $O(n^2)$. The diminuished "effectiveness" is outweighed by the usefulness of structural operations and their results (e.g. identifying proto prime implicants, which are guaranteed to deliver satisfying assignments for a satisfiabilty problem under sequential resolution). Graph theoretic algorithms can be used, if nodes are augmented to carry clause membership information that allows to restore the satoku matrix. However, this possibility is unexplored. Some transformations from propositional logic are recognizable, e.g. distributive expansion of clauses. Some operations have partial counterparts in propositional logic, but the core operations can neither be fully performed in plain graph theory nor propositional logic.

Despite a superficial resemblance of "conflict propagation" to "conflict clauses", structural logic has no relation to decision algorithms with conflict driven clause learning (DPLL+CDCL) algorithms. From the perspective of structural logic, CDCL is in fact an entirely ineffective way of discovering the constraints of a logical problem. The end point of proving *strict provability* in structural logic is a 2-state instance not a 1-state instance to prove satisfiability, which is trivially derived from a *strictly provable* satoku matrix. In fact, all 1-state and 2-state clauses (except for the last one) are effectively ornamental nuisances, kept for human brains to make sense of the structures. None of them have any purpose beyond definining the conflict relationships initially. Structural logic is therefore very resilient to encoding variations, whereas DPLL+CDCL is very susceptible to them (see section 13).

As a hint for classification in conventional contexts, structural logic as presented here implements a dynamic iterative deepening depth-first search (IDDFS) without backtracking. The results of the research recently incited the development of a computer-aided adaptive IDDFS algorithm for sudokus with a limited choice of pairing decisions, designed to be performed by a human[SCHSUD]. The algorithm solves the hardest available sudokus with a maximum search depth of 3 levels[3.].

Structural logic will certainly never be an "effective" contender in a SAT race, since its basic requirements are $O(n^2)$. But there is also nothing necessarily exponential in its operations, unless the choice to employ such methods is made.

The purpose of structural logic is to analyse the hardness of problems more accurately than the theories using variable counts as a measure instead of the actual input size given by the conflict relationships defined by the input clauses.

Structural logic is further useful to analyze and restore structural properties of hard problems, so that they can be solved with more appropriate algorithms. However, these algorithms themselves are not necessarily the domain of structural logic. E.g., it is possible to realize Gauss-Jordan eliminiation entirely within the framework of the satoku matrix, but mileage may vary substantially for other algorithms.

In short, structural logic is to propositional logic, what NAND is to DeMorgan's laws. Everything is true, but very awkward to understand. However, the results can be very useful.

3. Preliminary analysis of the relevant satoku matrix produced by hard sudoku instances indicates, that the maximum search depth of 3 levels can be easily proved in structural logic, whereas such a proof in propositional calculus seems quite hopeless.

4. States, Cells, Matrix

Structural logic does not have propositions or truth values. It deals exclusively with singular states that are either atomic states or their conflict relationships (CFR). Singular states (represented by 0 or 1) are grouped in cells, represented as state matrices. Cells are further arranged in a cell matrix.

4.1 Index Scheme

Except for defining the basic properties of states, singular states outside a matrix or cell are mostly meaningless in structural logic. Therefore, states are always referenced with full matrix indices, which are introduced here before the definition of states.

Atomic states $s_{i_{j_{i_j}}}$ and conflict relationships $s_{i_{j_{g_h}}}, i \neq g \vee j \neq h$, with $i, j, g, h = (0, 1, \dots)$ are mapped into cells c_{i_g} which are matrices of cell rows $r_{i_{j_g}}$ and cell columns (not referenced as state entity).

Cells c_{i_g} are arranged in a matrix (satoku matrix) of cell matrix rows c_i and cell matrix columns (not referenced as state entity).

As for all other indices used in this article: $e, f, x, y, z, m, n = (0, 1, \dots)$.

The index scheme for cells and states is chosen to reference increasingly detailed subsets of the global state. It is 0-based, since it mainly serves as a template for computer algorithms.

Table 1 shows the index scheme used in this article.

indexed state entity	**description**
$c_{\textit{cell-matrix-row}}$	row c_i of cells (cell-matrix-row)
$c_{\textit{cell-matrix-row}_{\textit{cell-matrix-column}}}$	single cell c_{i_g}
$r_{\textit{cell-matrix-row}_{\textit{cell-row}_{\textit{cell-matrix-column}}}}$	cell row $r_{i_{j_g}}$ containing all CFR states between an atomic state $s_{i_{j_{i_j}}}$ and an atomic cell c_{g_g}
$s_{\textit{cell-matrix-row}_{\textit{cell-row}}}$	state row s_{i_j} of all cell rows $r_{i_{j_g}}$ containing all singular states for an atomic state $s_{i_{j_{i_j}}}$
$s_{\textit{cell-matrix-row}_{\textit{cell-row}_{\textit{cell-matrix-column}_{\textit{cell-column}}}}}$	singular state $s_{i_{j_{g_h}}}$

Table 1: Index scheme

Figure 1 shows the extent covered by different levels of indexing.

s_{0_0}	1 0 0	1 1 1 1	1 1 1	1 1	c_0
s_{0_1}	0 1 0	0 1 1 1	1 1 1	1 1	
s_{0_2}	0 0 1	1 1 1 1	1 0 1	0 1	
s_{1_0}	1 0 1	1 0 0 0	0 1 1	1 1	c_{1_2}
s_{1_1}	1 1 1	0 1 0 0	1 0 1	0 1	
s_{1_2}	1 1 1	0 0 1 0	1 1 1	1 0	
s_{1_3}	1 1 1	0 0 0 1	1 1 0	1 1	
s_{2_0}	1 1 1	0 1 1 1	1 0 0	1 1	s_{2_0}
s_{2_1}	1 1 0	1 0 1 1	0 1 0	1 1	$s_{2_{1_0}}$
s_{2_2}	1 1 1	1 1 1 0	0 0 1	1 1	$s_{2_{2_{1_3}}}$
s_{3_0}	1 1 0	1 0 1 1	1 1 1	1 0	
s_{3_1}	1 1 1	1 1 0 1	1 1 1	0 1	$r_{3_{1_2}}$

.

Figure 1: Basic satoku matrix and extent of indexing

4.2 State Properties

possible
impossible

An atomic state $s_{i_{j_{i_j}}}$ is either *possible* (Pos), denoted as 1, or *impossible* (Imp), denoted as 0:

$$\forall s_{i_{j_{i_j}}} : \mathsf{Pos}(s_{i_{j_{i_j}}}) \veebar \mathsf{Imp}(s_{i_{j_{i_j}}})$$

independent
mutually exclusive

Two atomic states $s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}$ can be either *independent* (Ind) or *mutually exclusive* (Mutex, $\uparrow$):

$$\forall s_{i_{j_{i_j}}} : \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) \veebar \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}})$$

Independence and *mutual exclusion* are commutative. If state $s_{i_{j_{i_j}}}$ is *independent* of/*mutually exclusive* with state $s_{g_{h_{g_h}}}$, it follows that state $s_{g_{h_{g_h}}}$ is *independent* of/*mutually exclusive* with state $s_{i_{j_{i_j}}}$:

$$\begin{aligned} \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) &\Leftrightarrow \mathsf{Ind}(s_{g_{h_{g_h}}}, s_{i_{j_{i_j}}}) \\ \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) &\Leftrightarrow \mathsf{Mutex}(s_{g_{h_{g_h}}}, s_{i_{j_{i_j}}}) \\ s_{i_{j_{i_j}}} \uparrow s_{g_{h_{g_h}}} &\Leftrightarrow s_{g_{h_{g_h}}} \uparrow s_{i_{j_{i_j}}} \end{aligned}$$

The conflict relationship (CFR) $s_{i_{j_{g_h}}}, i \neq g \lor j \neq h$, between two atomic states $s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}$ is *possible*, if the atomic states are *independent*. If the atomic states are *mutually exclusive*, the CFR is *impossible*:

$$\begin{aligned} \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) &\Leftrightarrow \mathsf{Pos}(s_{i_{j_{g_h}}}) \\ \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) &\Leftrightarrow \mathsf{Imp}(s_{i_{j_{g_h}}}) \end{aligned}$$

Due to commutativity of Ind and Mutex, the same holds for the mirror CFR $s_{g_{h_{i_j}}}$:

$$\begin{array}{rclcl} \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) & \leftrightarrow & \mathsf{Ind}(s_{g_{h_{g_h}}}, s_{i_{j_{i_j}}}) & \Leftrightarrow & \mathsf{Pos}(s_{i_{j_{g_h}}}) \leftrightarrow \mathsf{Pos}(s_{g_{h_{i_j}}}) \\ \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) & \leftrightarrow & \mathsf{Mutex}(s_{g_{h_{g_h}}}, s_{i_{j_{i_j}}}) & \Leftrightarrow & \mathsf{Imp}(s_{i_{j_{g_h}}}) \leftrightarrow \mathsf{Imp}(s_{g_{h_{i_j}}}) \end{array}$$

This also holds, when an atomic state $s_{i_{j_{i_j}}}$ is viewed as a special conflict relationship of an atomic state with itself. Therefore, a singular state $s_{i_{j_{g_h}}}$ is defined as either an atomic state or a conflict relationship (CFR).

4.2.1 Merge Operation

When two singular states $s_{i_{j_{g_h}}}, s_{e_{f_{g_h}}}$ are merged into another singular state $s_{x_{y_{g_h}}}$:

$$s_{x_{y_{g_h}}} = \mathsf{Mrg}(s_{i_{j_{g_h}}}, s_{e_{f_{g_h}}}),$$

the resulting properties of state $s_{x_{y_{g_h}}}$ are defined by the state table 2.

$s_{i_{j_{g_h}}}$	$s_{e_{f_{g_h}}}$	$s_{x_{y_{g_h}}}$
Imp	Imp	Imp
Imp	Pos	Imp
Pos	Imp	Imp
Pos	Pos	Pos

Table 2: State table for merging two states $s_{i_{j_{g_h}}}, s_{e_{f_{g_h}}}$

It is obvious, that *impossible* is the dominant state.

The merge operation for singular states is equivalent to the function AND in propositional logic: $s_{i_{j_{g_h}}} \wedge s_{e_{f_{g_h}}} = s_{x_{y_{g_h}}}$. However, since the merge operation never just affects a singular state $s_{x_{y_{g_h}}}$, but also always the mirror state $s_{g_{h_{xy}}}$, and is generally not context free or functional, the AND function is avoided to minimize confusion.

4.2.2 Macro States

A macro state M is a group of singular states that can contain more than one singular state $s_{i_{j_{g_h}}}$. The properties *possible* and *impossible* are extended to macro states. A macro state is *possible*, if at least one of the contained singular states is *possible*. A macro state is *impossible*, if all of the contained singular states are *impossible* (*possible*

possible
impossible

bias), or if it does not contain any states.

$$
\begin{array}{lcl}
M = \{s_{i_{j_{g_h}}}\} & & \\
|M| = 0 & \Rightarrow & \mathsf{Imp}(M) \\
\exists s_{i_{j_{g_h}}} : s_{i_{j_{g_h}}} \in M \wedge \mathsf{Pos}(s_{i_{j_{g_h}}}) & \Leftrightarrow & \mathsf{Pos}(M) \\
\forall s_{i_{j_{g_h}}} : s_{i_{j_{g_h}}} \in M \wedge \mathsf{Imp}(s_{i_{j_{g_h}}}) & \Rightarrow & \mathsf{Imp}(M)
\end{array}
$$

decided
undecided

A macro state can further be either *decided* (Dec) or *undecided* (Und). A macro state is *decided*, if it has at most one *possible* singular state. A macro state is *undecided*, if it has more than one *possible* singular state. (*undecided* bias).

$$
\begin{array}{lcl}
P_m = \{s_{i_{j_{g_h}}} | s_{i_{j_{g_h}}} \in M \wedge \mathsf{Pos}(s_{i_{j_{g_h}}})\} & & \\
|P_m| \leq 1 & \Leftrightarrow & \mathsf{Dec}(M) \\
|P_m| > 1 & \Leftrightarrow & \mathsf{Und}(M)
\end{array}
$$

restricted
unrestricted

Another macro state classification is *restricted* (Rst) and *unrestricted* (Unr). A macro state is *restricted*, if it contains at least one *impossible* singular state. A macro state is *unrestricted*, if it does not contain any *impossible* singular states, i.e. it consists entirely of *possible* singular states. (*restricted* bias).

$$
\begin{array}{lcl}
I_m = \{s_{i_{j_{g_h}}} | s_{i_{j_{g_h}}} \in M \wedge \mathsf{Imp}(s_{i_{j_{g_h}}})\} & & \\
|M| = 0 & \Rightarrow & \mathsf{Rst}(M) \\
|I_m| > 0 & \Rightarrow & \mathsf{Rst}(M) \\
|I_m| = 0 & \Leftrightarrow & \mathsf{Unr}(M)
\end{array}
$$

4.2.3 Compound States

possible
impossible

A compound state C is a group of macro states M_f that can contain more than one macro state. The properties *possible* and *impossible* are extended to compound states. A compound state is *possible*, if all of the contained macro states are *possible*. A compound state is *impossible*, if one of the contained macro states is *impossible*. (*impossible* bias).

$$
\begin{array}{lcl}
C = \{M_f\} & & \\
|C| = 0 & \Rightarrow & \mathsf{Imp}(C) \\
\forall M_f : M_f \in C \wedge \mathsf{Pos}(M_f) & \Leftrightarrow & \mathsf{Pos}(C) \\
\exists M_f : M_f \in C \wedge \mathsf{Imp}(M_f) & \Rightarrow & \mathsf{Imp}(C)
\end{array}
$$

decided
undecided

A compound state can further be either *decided* or *undecided*. A compound state is *decided*, if all contained macro states, are *decided*. A compound state is *undecided*, if it has at least one *undecided* macro state. (*undecided* bias).

$$
\begin{array}{lcl}
U_c = \{M_f | M_f \in C \wedge \mathsf{Und}(M_f)\} & & \\
|U_c| = 0 & \Leftrightarrow & \mathsf{Dec}(C) \\
|U_c| > 0 & \Leftrightarrow & \mathsf{Und}(C)
\end{array}
$$

Another compound state classification is *restricted* and *unrestricted*. A compound state is *restricted*, if it contains at least one *restricted* macro state, that does not contain an atomic state. A compound state is *unrestricted*, if all contained macro states, that do not contain an atomic state, are *unrestricted*. (*restricted* bias).

restricted
unrestricted

$$\begin{array}{lll} R_c = \{M_f | M_f \in C \wedge s_{i_{j_{g_h}}} \in M_f \wedge (i \neq g \vee j \neq h) \wedge \mathsf{Rst}(M_f)\} & & \\ |C| = 0 & \Rightarrow & \mathsf{Rst}(C) \\ |R_c| > 0 & \Rightarrow & \mathsf{Rst}(C) \\ |R_c| = 0 & \Leftrightarrow & \mathsf{Unr}(C) \end{array}$$

4.3 Cell Representation

Atomic states and their conflict relationships are grouped in atomic cells c_{i_i} where the positions on the diagonal represent the atomic states $s_{i_{j_{i_j}}}$ and other positions $s_{i_{j_{i_h}}}, j \neq h$ represent the conflict relationships between the atomic states $s_{i_{j_{i_j}}}$ and $s_{i_{h_{i_h}}}$ intersecting at that position. In this article, only atomic cells with *mutually exclusive* atomic states are considered (see figure 2):

$$\forall c_{i_i} \forall s_{i_{j_{i_h}}} : c_{i_i} = \{s_{i_{j_{i_h}}}\}, j \neq h \rightarrow \mathsf{Imp}(s_{i_{j_{i_h}}})$$

$$\begin{array}{c|c|c|c} \hline s_{0_0} & 1\,0\,0 & & r_{0_{0_0}} \\ s_{0_1} & 0\,1\,0 & & r_{0_{1_0}} \\ s_{0_2} & 0\,0\,1 & & r_{0_{2_0}} \\ \hline s_{1_0} & & 1\,0\,0 & r_{1_{0_1}} \\ s_{1_1} & & 0\,1\,0 & r_{1_{1_1}} \\ s_{1_2} & & 0\,0\,1 & r_{1_{2_1}} \\ \hline \end{array} .$$

Figure 2: Atomic cells c_{i_i} with *mutually exclusive* atomic states $s_{i_{j_{i_j}}}$, $i = (0,1)$, $j = (0,1,2)$

Conflict relation cells $c_{i_g}, g \neq i$ only contain conflict relationships between atomic states $s_{i_{j_{i_j}}}$ and atomic states $s_{g_{h_{g_h}}}$ (see figure 3):

$$\forall c_{i_g} \forall s_{i_{j_{g_h}}} : c_{i_g} = \{s_{i_{j_{g_h}}} | i \neq g, \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) \rightarrow \mathsf{Pos}(s_{i_{j_{g_h}}}), \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) \rightarrow \mathsf{Imp}(s_{i_{j_{g_h}}})\}$$

$$
\begin{array}{l|c|c|l}
\hline
s_{0_0} & & 1\,1\,1 & r_{0_{0_1}} \\
s_{0_1} & & 1\,1\,1 & r_{0_{1_1}} \\
s_{0_2} & & 1\,1\,0 & r_{0_{2_1}} \\
\hline
s_{1_0} & 1\,1\,1 & & r_{1_{0_0}} \\
s_{1_1} & 1\,1\,1 & & r_{1_{1_0}} \\
s_{1_2} & 1\,1\,0 & & r_{1_{2_0}} \\
\hline
\end{array} .
$$

Figure 3: CFR cells c_{0_1}, c_{1_0} with *impossible* CFR states $s_{0_{2_{1_2}}}, s_{1_{2_{0_2}}}$ for *mutually exclusive* atomic states $s_{0_{2_{0_2}}}$ and $s_{1_{2_{1_2}}}$

Due to commutativity of *mutual exclusion*, each conflict relationship $s_{i_{j_{g_h}}}, i \neq g \vee j \neq h$, is necessarily equivalent to its diagonal mirror state $s_{g_{h_{i_j}}}$.

Since conflict relationships are intrinsic properties of atomic states, the general term "state" $s_{i_{j_g}}$ is used to reference cell row $r_{i_{j_g}}$.

If $i = g$, this includes the atomic state $s_{i_{j_{i_j}}}$ and its conflict relationships $s_{i_{j_{i_h}}}$ to the other atomic states $s_{i_{h_{i_h}}}, h \neq j$:

$$\forall s_{i_{j_i}} \forall r_{i_{j_i}} \forall s_{i_{j_{i_h}}} : s_{i_{j_i}} = r_{i_{j_i}} = \{ s_{i_{j_{i_h}}} \,|\, j \neq h \rightarrow \mathsf{Imp}(s_{i_{j_{i_h}}}) \}$$

If $i \neq g$, the cell row $r_{i_{j_g}}$ contains the conflict relationships $s_{i_{j_{g_h}}}$ between atomic state $s_{i_{j_{i_j}}}$ and the atomic states $s_{g_{h_{g_h}}}$:

$$\begin{aligned}
&\forall s_{i_{j_g}} \forall r_{i_{j_g}} \forall s_{i_{j_{g_h}}} : \\
&s_{i_{j_g}} = r_{i_{j_g}} = \{ s_{i_{j_{g_h}}} | \quad \mathsf{Ind}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) \rightarrow \mathsf{Pos}(s_{i_{j_{g_h}}}), \\
&\qquad\qquad\qquad\qquad \mathsf{Mutex}(s_{i_{j_{i_j}}}, s_{g_{h_{g_h}}}) \rightarrow \mathsf{Imp}(s_{i_{j_{g_h}}}) \}, i \neq g
\end{aligned}$$

Cell rows $r_{i_{j_g}}$ are macro states.

bound *required* A *possible decided* cell row is called *bound* (Bnd):

$$\exists s_{i_{j_{g_h}}} : \mathsf{Pos}(s_{i_{j_{g_h}}}) \wedge \mathsf{Dec}(r_{i_{j_g}}) \Leftrightarrow \mathsf{Bnd}(r_{i_{j_g}}).$$

A *possible* state $s_{i_{j_{g_h}}}$ in a *bound* cell row $r_{i_{j_g}}$ is called *required* (Req):

$$\exists s_{i_{j_{g_h}}} : \mathsf{Pos}(s_{i_{j_{g_h}}}) \wedge \mathsf{Bnd}(r_{i_{j_g}}) \Leftrightarrow \mathsf{Req}(s_{i_{j_{g_h}}}).$$

Note that *required* is not a commutative state property, since the mirror state of a cell row is a cell column.

conflict When a cell row $r_{i_{j_g}}$ is *decided* and has no *possible* states $s_{i_{j_{g_h}}}$ and is therefore *im-*

possible, it is called a *conflict* cell row (Cfl), short notation $\neg r_{i_{jg}}$:

$$\forall s_{i_{jg_h}} : s_{i_{jg_h}} \in r_{i_{jg}} \wedge \mathsf{Imp}(s_{i_{jg_h}}) \rightarrow \mathsf{Imp}(r_{i_{jg}}) \Leftrightarrow \mathsf{Cfl}(r_{i_{jg}}) \Leftrightarrow \neg r_{i_{jg}}$$

Two cell rows $r_{i_{jg}}, r_{e_{fg}}$ are *combinable* (Cmb), if their atomic states $s_{i_{j_{i_j}}}, s_{e_{f_{e_f}}}$ are *independent*: *combinable*

$$\mathsf{Ind}(s_{i_{j_{i_j}}}, s_{e_{f_{e_f}}}) \rightarrow \mathsf{Pos}(s_{i_{j_{e_f}}}) \wedge \mathsf{Pos}(s_{e_{f_{i_j}}}) \Leftrightarrow \mathsf{Cmb}(r_{i_{jg}}, r_{e_{fg}})$$

Cells c_{i_g} are both macro and compound states. They are defined as hybrid states containing cell rows $r_{i_{jg}}$. If the macro state properties of cells are referenced, the cells are denoted as macro state cells $c_{m_{i_g}}$.

The properties *possible* and *impossible* are extended to cells. A cell c_{i_g} is *possible*, if at least one of the contained cell rows $r_{i_{g_j}}$ is *possible*. A cell c_{i_g} is *impossible*, if all of the contained cell rows $r_{i_{g_j}}$ are *impossible*. (*possible* bias): *possible* *impossible*

$$\begin{array}{lcl} H = c_{i_g} = \{r_{i_{g_j}}\} & & \\ |H| = 0 & \Rightarrow & \mathsf{Imp}(H) \\ \exists r_{i_{g_j}} : r_{i_{g_j}} \in H \wedge \mathsf{Pos}(r_{i_{g_j}}) & \Leftrightarrow & \mathsf{Pos}(H) \\ \forall r_{i_{g_j}} : r_{i_{g_j}} \in H \wedge \mathsf{Imp}(r_{i_{g_j}}) & \Rightarrow & \mathsf{Imp}(H) \end{array}$$

A cell c_{i_g} can further be either *decided* or *undecided*. A cell c_{i_g} is *decided*, if all contained cell rows $r_{i_{g_j}}$, are *decided*. A cell c_{i_g} is *undecided*, if it has at least one *undecided* cell row $r_{i_{g_j}}$. (*undecided* bias): *decided* *undecided*

$$\begin{array}{lcl} U_h = \{r_{i_{g_j}} | r_{i_{g_j}} \in H \wedge \mathsf{Und}(r_{i_{g_j}})\} & & \\ |U_h| = 0 & \Leftrightarrow & \mathsf{Dec}(H) \\ |U_h| > 0 & \Leftrightarrow & \mathsf{Und}(H) \end{array}$$

A cell c_{i_g} is *restricted*, if it contains at least one *restricted* cell row $r_{i_{g_j}}$. A cell c_{i_g} is *unrestricted*, if all contained cell rows $r_{i_{g_j}}$ are *unrestricted*. (*restricted* bias): *restricted* *unrestricted*

$$\begin{array}{lcl} R_h = \{r_{i_{g_j}} | r_{i_{g_j}} \in H \wedge \mathsf{Rst}(r_{i_{g_j}})\} & & \\ |H| = 0 & \Rightarrow & \mathsf{Rst}(H) \\ |R_h| > 0 & \Rightarrow & \mathsf{Rst}(H) \\ |R_h| = 0 & \Leftrightarrow & \mathsf{Unr}(H) \end{array}$$

When a cell is *decided* and has no *possible* cell rows and is therefore *impossible*, it is called a *contradiction* (Ctr): *contradiction*

$$\mathsf{Imp}(c_{i_g}) \Leftrightarrow \mathsf{Ctr}(c_{i_g})$$

4.4 Satoku Matrix

A satoku matrix is a compound state containing cells c_{i_g}.

Here is a short summary of properties.

Cells c_{i_g} are generally defined as hybrid states consisting of cell rows $r_{i_{j_g}}$

The cells c_{i_i} on the diagonal of the satoku matrix contain atomic states $s_{i_{j_{i_j}}}$.

Macro state cells $c_{m_{i_g}}$ contain singular states $s_{i_{j_{g_h}}}$, which are either

- atomic states $s_{i_{j_{i_j}}}$ and their *impossible* conflict relationships $s_{i_{j_{i_h}}}, j \neq h$, or
- the conflict relationships between atomic states $s_{i_{j_{i_j}}}$ and atomic states $s_{g_{h_{g_h}}}$ if $i \neq g, j \neq h$.

A cell row $r_{i_{j_g}}, g \neq i$, of a conflict relationship cell c_{i_g} represents the conflict relationships between the atomic state $s_{i_{j_{i_j}}}$ and all atomic states $s_{g_{h_{g_h}}}$ of atomic cell c_{g_g}.

A state row s_{i_j} is a compound state, referencing the entire sequence of corresponding cell rows $r_{i_{j_g}}$ and contains all intra-cell and inter-cell conflict relationships for atomic state $s_{i_{j_{i_j}}}$:

$$s_{i_j} = \{r_{i_{j_g}}\}$$

conflict When a state row s_{i_j} has a *conflict* cell row $r_{i_{j_g}}$ and is therefore *impossible*, it is called a *conflict* state row (Cfl), short notation $\neg s_{i_j}$:

$$\mathsf{Cfl}(r_{i_{j_g}}) \rightarrow \mathsf{Imp}(s_{i_j}) \Leftrightarrow \mathsf{Cfl}(s_{i_j}) \Leftrightarrow \neg s_{i_j}$$

combinable Two state rows s_{i_j}, s_{e_f} are *combinable* (Cmb), if their atomic states $s_{i_{j_{i_j}}}, s_{e_{f_{e_f}}}$ are *independent*:

$$\mathsf{Ind}(s_{i_{j_{i_j}}}, s_{e_{f_{e_f}}}) \Leftrightarrow \mathsf{Cmb}(s_{i_j}, s_{e_f})$$

An *impossible* satoku matrix $\mathbb{S}$ is called a *contradiction*:

$$\mathsf{Imp}(c_{i_g}) \rightarrow \mathsf{Imp}(\mathbb{S}) \Leftrightarrow \mathsf{Ctr}(\mathbb{S})$$

It is advantageous for human readers, to represent *possible* singular states $s_{i_{j_{g_h}}}$ of *undecided* cell rows $r_{i_{j_g}}$ with a dash "–" contrasting the *required* "1" for a *possible* singular state $s_{i_{j_{g_h}}}$ of a *decided* cell row $r_{i_{j_g}}$. Just keep in mind, that it is no new third state indicating ternary logic. The satoku matrix from figure 1 then presents as shown in figure 4.

$$
\begin{array}{c|c|c|c|c|l}
\hline
s_{0_0} & 1 \circ \circ & - - - - & - - - & - - & c_0 \\
s_{0_1} & \circ 1 \circ & 0 - - - & - - - & - - & \\
s_{0_2} & \circ \circ 1 & - - - - & - 0 - & 0 1 & \\
\hline
s_{1_0} & - 0 - & 1 \circ \circ \circ & 0 - - & - - & c_{1_2} \\
s_{1_1} & - - - & \circ 1 \circ \circ & - 0 - & 0 1 & \\
s_{1_2} & - - - & \circ \circ 1 \circ & - - - & 1 0 & \\
s_{1_3} & - - - & \circ \circ \circ 1 & - - 0 & - - & \\
\hline
s_{2_0} & - - - & 0 - - - & 1 \circ \circ & - - & s_{2_0} \\
s_{2_1} & - - 0 & - 0 - - & \circ 1 \circ & - - & s_{2_{1_0}} \\
s_{2_2} & - - - & - - - 0 & \circ \circ 1 & - - & s_{2_{2_{1_3}}} \\
\hline
s_{3_0} & - - 0 & - 0 - - & - - - & 1 \circ & \\
s_{3_1} & - - - & - - 0 - & - - - & \circ 1 & r_{3_{1_2}} \\
\hline
\end{array}
.
$$

Figure 4: Visually enhanced satoku matrix

The satoku matrix without the cell constraints is always equivalent to an inverted adjacency matrix[wiki-am]. It can therefore be mapped to a graph or propositional formula at any time, if desired.

It is obvious, that the satoku matrix is not minimal. It would be sufficient to use only the cells on the diagonal and one half of the conflict relation cells. But that would require considering the cell columns and result in an unwieldy indexing scheme that impedes structural reasoning (for human brains). However, when implementing a computer algorithm, it is possible to implement the transformations in approximately half of the space requirements of the satoku matrix. The space requirements will still be $O(n^2)$, so algorithm complexity is not affected.

The satoku matrix is structurally complete, in that all direct consequences of conflict relationships between states are properly represented in full detail. Aside from folding rows into a column and a row part, any attempt of simplification results in a structurally incomplete logic that can no longer describe all structural properties adequately[4.].

It is obvious, that an artificially sequentialized symbolic notation (as with propositional variables and clauses in mathematical logic) cannot fully reflect all aspects of the structural properties of cells and states, most prominently the inherent self-referentiality/circularity and generally the proper conflict context. Therefore the matrix figures are the proper and definitive notation and not mere illustrations.

4. Both graph theory and propositional calculus can be interpreted as simplifications of structural logic in this sense. This is also a good example for the unsubstantiated implication *not necessary* →*irrelevant*. Just because graph theory and propositional calculus are unnecessary in this interpretation, this does not mean that they are irrelevant.

4.5 Mapping of State Properties

The mapping of singular state properties $\{impossible, possible\}$ to $\{0, 1\}$ must not be confused with other models of logic. Mapping state properties to truth values, graph nodes/edges, propositions/clauses is entirely arbitrary.

It is especially invalid to assume that an atomic state is equivalent to a propositional variable. The basic representation of an atomic state in propositional logic is a conjunction of one or more literals (negated or unnegated propositional variables). A propositional variable is therefore just a special case of a cell with two atomic states $(p, \neg p)$.

The proper representation of a cell in propositional logic is a disjunction of conjunctions of literals.

State	Truth Value	Boolean	Graph	Prop.
possible	`T`	1	edge	p
impossible	`F`	0	no edge	$\neg p$

Table 3: Standard mappings of truth values to states

The natural mapping of propositional logic into the satoku matrix is that of propositional formulae expressed as conjunction of disjunctive normal forms (CDF[SCHPDE])[5.], where the alternatives of the disjunctive normal forms (conjunctions of literals) are mapped to atomic states of a cell.

5. Deduction Rules

Structural logic has a set of deduction rules for transforming a satoku matrix $\mathbb{S}$ into a satoku matrix $\mathbb{S}'$ preserving *provability*.

5.1 Provability (Minimal Definition)

provable A satoku matrix $\mathbb{S}$ is *provable* (Prov), if it can be reduced via deduction rules to a state row s_{i_j}, which is *decided* and *possible*:

$$\exists s_{i_j}\, \forall s_{i_f} : f \neq j \wedge \mathsf{Pos}(s_{i_j}) \wedge \mathsf{Dec}(s_{i_j}) \wedge \mathsf{Imp}(s_{i_f}) \Leftrightarrow \mathsf{Prov}(\mathbb{S})$$

5. CDF must not be confused with CNF (conjunctive normal form). CNF is a restricted simplification of CDF. None of the restrictions of CNF apply to CDF. I.e., a propositional variable may appear negated and unnegated in a disjunction, propositional variables may appear more than once.

It follows that exactly one atomic state $s_{i_{j_{i_j}}}$ from each cell c_{i_i} of satoku matrix $\mathbb{S}$ must be *possible*:

$$\forall M_i : M_i = \{s_{i_{j_{i_j}}} | c_{i_i} \in \mathbb{S} \land s_{i_{j_{i_j}}} \in c_{i_i} \land \mathsf{Pos}(s_{i_{j_{i_j}}})\} \land |M_i| = 1 \Leftrightarrow \mathsf{Prov}(\mathbb{S})$$

This strict core definition of *provability* is as close to boolean satisfiability as the satoku matrix gets. Note, however, that a *possible decided* state row s_{i_j} does not imply, that all boolean variables of an underlying CDF formula are necessarily decided in such a case.

5.2 Assignments

Algorithms use two variations of assignments, value assignment (:=) and state assignment (←).

Value assignment (:=) of value *val* to variable *var* is imperative:

$$(var := val) \Rightarrow (var = val)$$

State assignment (←) of value *val*, $val \in (0, 1, \mathsf{Pos}, \mathsf{Imp})$, to singular states $s_{i_{j_{g_h}}}$ follows the merge operation (Mrg) as defined in table 2. Due to commutativity of *independence* (Ind) and *mutual exclusion* (Mutex), the mirror state $s_{g_{h_{i_j}}}$ is always adjusted accordingly.

Assigning a value *val*, to a singular state $s_{i_{j_{g_h}}}$ is therefore a shortcut for assigning the result of $\mathsf{Mrg}(s_{i_{j_{g_h}}}, val)$ to $s_{i_{j_{g_h}}}$ and the result of $\mathsf{Mrg}(s_{g_{h_{i_j}}}, val)$ to $s_{g_{h_{i_j}}}$:

$$(s_{i_{j_{g_h}}} = \mathsf{Mrg}(s_{i_{j_{g_h}}}, val)) \land (s_{g_{h_{i_j}}} = \mathsf{Mrg}(s_{g_{h_{i_j}}}, val)) \quad \Leftrightarrow \quad s_{i_{j_{g_h}}} \leftarrow val$$

A direct consequence is, that singular states $s_{i_{j_{g_h}}}$ are never "resurrected". Once a singular state $s_{i_{j_{g_h}}}$ becomes *impossible*, there is no provision that it becomes *possible* again:

$$\forall s_{i_{j_{g_h}}} \nexists \mathsf{Op} : \mathsf{Pos}(\mathsf{Op}(\mathsf{Imp}(s_{i_{j_{g_h}}})))$$

Therefore, a state assignment of property *possible* (Pos) to a singular state $s_{i_{j_{g_h}}}$ is essentially a no-op, i.e., no changes to the states $s_{i_{j_{g_h}}}, s_{g_{h_{i_j}}}$ are made:

$$s'_{i_{j_{g_h}}} = s_{i_{j_{g_h}}} \leftarrow \mathsf{Pos} \Rightarrow s'_{i_{j_{g_h}}} = s_{i_{j_{g_h}}}$$

Backtracking can still be implemented by keeping a copy of the satoku matrix.

5.3 Contradiction Check

All deduction rules terminate, when a satoku matrix $\mathbb{S}$ becomes a *contradiction*. This happens, when a cell c_{i_g} becomes a *contradiction*, which in turn happens, when all cell rows $r_{i_{j_g}}$ of cell c_{i_g} become *impossible*. In the following algorithm, the property Ctr is implemented as an attribute of the respective data structure.

Algorithm 1 (contradiction check of cell c_{i_g})**.**
if $\neg\mathsf{Ctr}(\mathbb{S})$:
    cell_status $:= \mathsf{Imp}$
    for each cell row $r_{i_{j_g}}$ ***in*** cell c_{i_g}:
        if $\mathsf{Pos}(r_{i_{j_g}})$:
            cell_status $:= \mathsf{Pos}$
            break
    if $\mathsf{Imp}(\textit{cell\_status})$:
        $\mathsf{Imp}(c_{i_g}) \Rightarrow c_{i_g} := \mathsf{Ctr} \Rightarrow \mathsf{Imp}(\mathbb{S}) \Rightarrow \mathbb{S} := \mathsf{Ctr}$
if $\mathsf{Ctr}(\mathbb{S})$:
    terminate

5.4 Conflict Propagation

When an atomic state $s_{i_{j_{i_j}}}$ becomes *impossible*, it becomes *mutually exclusive* to all other atomic states, therefore all of its CFR states $s_{i_{j_{g_h}}}, g \neq i \vee h \neq j$ become *impossible* and are updated accordingly.

Algorithm 2 (set atomic state $s_{i_{j_{i_j}}}$ *impossible* and propagate)**.**
for each singular state $s_{i_{j_{g_h}}}$ ***in*** state row s_{i_j}:
    $s_{i_{j_{g_h}}} \leftarrow \mathsf{Imp}$
perform algorithm 1 (contradiction check of cell c_{i_i})

When all CFR states $r_{i_{j_g}}, g \neq i$ between the atomic state $s_{i_{j_{i_j}}}$ and a cell c_{g_g} become *impossible*, the atomic state $s_{i_{j_{i_j}}}$ becomes *impossible*, since the condition for *provability*, that exactly one atomic state in each cell must be *possible*, can no longer be fulfilled for cell c_{g_g}, if $r_{i_{j_i}}$ becomes the only *possible* state of cell c_{i_i}.

Algorithm 3 (check for *impossible* cell row $r_{i_{j_g}}$ and propagate)**.**
if $\mathsf{Imp}(r_{i_{j_g}})$:
    perform algorithm 2 (set atomic state $s_{i_{j_{i_j}}}$ *impossible* and propagate)

Since this results in a global change, it is useful to add an informational status line reflecting global states of the satoku matrix. This status line is labelled P.

In the example of figure 5a, state row s_{0_1} was made *impossible* and the CFR states have been updated accordingly. It is obvious, that the respective row and column can be removed from the satoku matrix since the *impossible* state is no longer relevant to the *provability* of the satoku matrix.

Note that cell rows $r_{2_{1_0}}$ and $r_{3_{0_0}}$ have become *decided* and states $s_{2_{1_{0_0}}}$ and $s_{3_{0_{0_0}}}$ have become *required*.

P	− 0 −	− − − −	− − −	− −
s_{0_0}	1 ∘ ∘	− − − −	− − −	− −
s_{0_1}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_2}	∘ ∘ 1	− − − −	− 0 −	0 1
s_{1_0}	− 0 −	1 ∘ ∘ ∘	0 − −	− −
s_{1_1}	− 0 −	∘ 1 ∘ ∘	− 0 −	0 1
s_{1_2}	− 0 −	∘ ∘ 1 ∘	− − −	1 0
s_{1_3}	− 0 −	∘ ∘ ∘ 1	− − 0	− −
s_{2_0}	− 0 −	0 − − −	1 ∘ ∘	− −
s_{2_1}	1 0 0	− 0 − −	∘ 1 ∘	− −
s_{2_2}	− 0 −	− − − 0	∘ ∘ 1	− −
s_{3_0}	1 0 0	− 0 − −	− − −	1 ∘
s_{3_1}	− 0 −	− − 0 −	− − −	∘ 1

(a) Impossible state s_{0_1}

P	0 0 1	− − − −	− 0 −	− −
s_{0_0}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_1}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_2}	∘ ∘ 1	− − − −	− 0 −	0 1
s_{1_0}	0 0 1	1 ∘ ∘ ∘	0 0 1	− −
s_{1_1}	0 0 1	∘ 1 ∘ ∘	− 0 −	0 1
s_{1_2}	0 0 1	∘ ∘ 1 ∘	− 0 −	1 0
s_{1_3}	0 0 1	∘ ∘ ∘ 1	1 0 0	− −
s_{2_0}	0 0 1	0 − − −	1 ∘ ∘	− −
s_{2_1}	0 0 0	0 0 0 0	∘ ∘ ∘	0 0
s_{2_2}	0 0 1	− − − 0	∘ ∘ 1	− −
s_{3_0}	0 0 0	− 0 − −	− 0 −	1 ∘
s_{3_1}	0 0 1	− − 0 −	− 0 −	∘ 1

(b) Decided macro state cell $c_{m_{0_0}}$

Figure 5: conflict propagation

When a macro state cell $c_{m_{i_i}}$ becomes *decided* and has one *possible* state $s_{i_{j_{i_j}}}$, that state becomes globally *required*.

Figure 5b shows that macro state cell $c_{m_{0_0}}$ has become decided when state row s_{0_0} was made *impossible*. Consequently state row s_{0_2} has become *required* and its *impossible* CFR $s_{0_{2_{2_1}}}$ was propagated to all other states, causing state row s_{2_1} to become *impossible*.

The next step is shown in figure 6a where CFR $s_{0_{2_{3_0}}}$ was propagated to all other states, causing state row s_{3_0} to become *impossible*.

Since macro state cell $c_{m_{3_3}}$ has now become *decided*, and state s_{3_1} has become globally *required*, CFR $s_{3_{1_{1_2}}}$ is also propagated as shown in figure 6b.

P	0 0 1	– – – –	– 0 –	0 1
s_{0_0}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_1}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_2}	∘ ∘ 1	– – – –	– 0 –	0 1
s_{1_0}	0 0 1	1 ∘ ∘ ∘	0 0 1	0 1
s_{1_1}	0 0 1	∘ 1 ∘ ∘	– 0 –	0 1
s_{1_2}	0 0 1	∘ ∘ 1 ∘	– 0 –	0 0
s_{1_3}	0 0 1	∘ ∘ ∘ 1	1 0 0	0 1
s_{2_0}	0 0 1	0 – – –	1 ∘ ∘	0 1
s_{2_1}	0 0 0	0 0 0 0	∘ ∘ ∘	0 0
s_{2_2}	0 0 1	– – – 0	∘ ∘ 1	0 1
s_{3_0}	0 0 0	0 0 0 0	0 0 0	∘ ∘
s_{3_1}	0 0 1	– – 0 –	– 0 –	∘ 1

(a) State $s_{0_{2_{3_0}}}$ causes *impossible* state s_{3_0}

P	0 0 1	– – 0 –	– 0 –	0 1
s_{0_0}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_1}	∘ ∘ ∘	0 0 0 0	0 0 0	0 0
s_{0_2}	∘ ∘ 1	– – 0 –	– 0 –	0 1
s_{1_0}	0 0 1	1 ∘ ∘ ∘	0 0 1	0 1
s_{1_1}	0 0 1	∘ 1 ∘ ∘	– 0 –	0 1
s_{1_2}	0 0 0	∘ ∘ ∘ ∘	0 0 0	0 0
s_{1_3}	0 0 1	∘ ∘ ∘ 1	1 0 0	0 1
s_{2_0}	0 0 1	0 – 0 –	1 ∘ ∘	0 1
s_{2_1}	0 0 0	0 0 0 0	∘ ∘ ∘	0 0
s_{2_2}	0 0 1	– – 0 0	∘ ∘ 1	0 1
s_{3_0}	0 0 0	0 0 0 0	0 0 0	∘ ∘
s_{3_1}	0 0 1	– – 0 –	– 0 –	∘ 1

(b) Macro state cell $c_{m_{3_3}}$ becomes *decided*

Figure 6: conflict propagation continued

5.5 Requirement Update

When a state row s_{i_j} has a *bound* cell row $r_{i_{j_g}}, i \neq g$ with *required* CFR $s_{i_{j_{g_h}}}$, the singular states $s_{g_{h_{e_f}}}$ of state row s_{g_h} will inevitably become global should the state $s_{i_{j_{i_j}}}$ become the *required* state of macro state cell $c_{m_{i_i}}$.

The singular states of state row s_{g_h} for a *required* state $s_{i_{j_{g_h}}}$ can therefore be merged locally into the CFR states of state row s_{i_j}, even when macro state cell $c_{m_{i_i}}$ is not yet decided (see also section 8.1). Note that this is not a decision, but still conflict propagation.

Algorithm 4 (requirement update of state row s_{i_j})**.**
do
    changed $:= 0$
    for each cell row $r_{i_{j_e}}$:
        if $\mathsf{Dec}(r_{i_{j_e}}) \wedge \mathsf{Pos}(s_{i_{j_{e_f}}})$:
            for each $s_{i_{j_{g_h}}}$:
                prev_state $:= s_{i_{j_{g_h}}}$
                $s_{i_{j_{g_h}}} \leftarrow s_{e_{f_{g_h}}}$
                if $s_{i_{j_{g_h}}} \neq$ *prev_state*:
                    changed $:= 1$
until *changed* $= 0$

When algorithm 4 is applied to the orignal example (see figure 4), it presents with the additionally propagated CFR states $s_{0_{2_{1_2}}}$ and $s_{1_{2_{0_2}}}$ as shown in figure 7

P	$- - -$	$- - - -$	$- - -$	$- -$
s_{0_0}	$1 \circ \circ$	$- - - -$	$- - -$	$- -$
s_{0_1}	$\circ 1 \circ$	$0 - - -$	$- - -$	$- -$
s_{0_2}	$\circ \circ 1$	$- - 0 -$	$- 0 -$	$0 1$
s_{1_0}	$- 0 -$	$1 \circ \circ \circ$	$0 - -$	$- -$
s_{1_1}	$- - -$	$\circ 1 \circ \circ$	$- 0 -$	$0 1$
s_{1_2}	$- - 0$	$\circ \circ 1 \circ$	$- - -$	$1 0$
s_{1_3}	$- - -$	$\circ \circ \circ 1$	$- - 0$	$- -$
s_{2_0}	$- - -$	$0 - - -$	$1 \circ \circ$	$- -$
s_{2_1}	$- - 0$	$- 0 - -$	$\circ 1 \circ$	$- -$
s_{2_2}	$- - -$	$- - - 0$	$\circ \circ 1$	$- -$
s_{3_0}	$- - 0$	$- 0 - -$	$- - -$	$1 \circ$
s_{3_1}	$- - -$	$- - 0 -$	$- - -$	$\circ 1$

.

Figure 7: Original example satoku matrix *consolidated*

Finally, when a state $s_{i_{j_{g_h}}}$ changes from *possible* to *impossible*, all state rows s_{e_f} with a *required* state $s_{e_{f_{i_j}}}$ must also be updated accordingly:

$$\forall s_{e_f} : \mathsf{Req}(s_{e_{f_{i_j}}}) \wedge \mathsf{Pos}(s_{i_{j_{g_h}}}) \wedge s_{i_{j_{g_h}}} \leftarrow \mathsf{Imp} \Rightarrow s_{e_{f_{g_h}}} \leftarrow \mathsf{Imp}$$

5.6 Consolidation

Consolidation of a satoku matrix is the most important process of structural logic. Changes to the satoku matrix cannot generally be introduced in parallel. Each change must be followed by consolidation before introducing the next change[6].

A satoku matrix is *consolidated* (Con), when all deduction rules have been exhausted. *consolidated*

In conventional terms this mechanism can be interpreted as a restricted lookahead of 1 level in an iterative deepening depth-first search (IDDFS) algorithm. It can also be interpreted as partial distributive expansion of propositional clauses, restricted to the polynomial time portion of distributive expansion[SCHPDE].

6. This is the reason, why any attempt of a functional analysis in symbolic notation is utterly useless.

6. Summary of Properties

Table 4 shows a loosely structured overview of properties for structural state entitities defined so far. Conflict relationship is abbreviated as CFR.

atomic state	**CFR**	**cell row**	**cell**	**state row**	**matrix**	conditions
atomic	atomic	macro	hybrid macro	compound	compound	
independent						
mutually exclusive						
possible	*possible*	*possible*	*possible*	*possible*	*possible*	
impossible	*impossible*	*impossible*	*impossible*	*impossible*	*impossible*	
		unrestricted	*unrestricted*	*unrestricted*	*unrestricted*	$\lvert\mathsf{Imp}\rvert = 0$
		restricted	*restricted*	*restricted*	*restricted*	$\lvert\mathsf{Imp}\rvert > 0$
		undecided	*undecided*	*undecided*	*undecided*	$\lvert\mathsf{Pos}\rvert > 1$
		decided	*decided*	*decided*	*decided*	$\lvert\mathsf{Pos}\rvert \leq 1$
	required	*bound*				*possible decided*
				unconsolidated	*unconsolidated*	
				consolidated	*consolidated*	
		conflict	*contradiction*	*conflict*	*contradiction*	*impossible*
					provable	*possible* and (*decided* or *unrestricted*)

Table 4: Summary of properties

7. Mapping CDF Problems

As shown in [MOUNT], all boolean satisfiability problems in conjunctive normal form (CNF) encoding can be mapped to an independent set problem and therefore to an (inverted) adjacency matrix.

In fact, the example satoku matrix (see figures 4, 7) was constructed by mapping the following propositional formula:

$$\begin{array}{ll} (\ \ a \vee \ \ b \vee \ \ c) & \wedge \\ (\neg b \vee \ \ c \vee \neg d \vee \neg e) & \wedge \\ (\ \ b \vee \neg c \vee \ \ e) & \wedge \\ (\neg c \vee \ \ d) & \end{array}$$

CNF is a a special case of the more general CDF[SCHPDE], where the alternatives of a CNF clause are conjunctions. The single variable of a CNF clause alternative is just considered the special case of a conjunction with one variable.

Algorithm 5 (Mapping of CDF problem to satoku matrix)**.**
With the CDF problem P consisting of clauses C_i
  with alternatives $A_{i_j}, i = 0..(|P| - 1), j = 0..(|C_i| - 1)$
Create a satoku matrix $\mathbb{S}$
for each clause C_i:
  Add a cell-matrix row c_i with $j + 1$ state rows to $\mathbb{S}$
  Set all atomic states and CFR states to *possible*
  Set all CFR states $s_{i_{j_{i_h}}}, h \neq j$ to *impossible*
for each state row s_{i_j}:
  for each cell row $r_{i_{j_g}}, g > i$:
    for each CFR $s_{i_{j_{g_h}}}$:
      if $A_{i_j} \wedge A_{g_h} = F$:
        $s_{i_{j_{g_h}}} \leftarrow \mathsf{Imp}$

7.1 Maximizing Conflicts

Traditionally, propositional formulae are reduced as much as possible. This makes perfect sense, if reasoning is conducted with pencil and paper[7]. Decision algorithms also follow that convention by eliminating as many clauses as possible.

7. I could not find any rationale for CNF, other than the desire of human minds to reduce the amount of redundancy. Although syllogistic reasoning[BROWN] based on Blake's theory of syllogistic formulas does require CNF to acquire the set of prime implicants, it is entirely irrelevant to structural logic.

However, the positive effect of an increased number of *impossible* conflict relationships in structural logic has an interesting effect, which seems counter-intuitive at first.

Using the tautology:

$$(p \vee q \vee r) = ((p) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q \wedge r)) \ .$$

the original CNF formula can be transformed to the CDF formula:

$$\begin{array}{llll}
(& (\ a) & \vee & \\
 & (\neg a \wedge \ b) & \vee & \\
 & (\neg a \wedge \neg b \wedge \ c) & &) \wedge \\
(& (\neg b) & \vee & \\
 & (\ b \wedge \ c) & \vee & \\
 & (\ b \wedge \neg c \wedge \neg d) & \vee & \\
 & (\ b \wedge \neg c \wedge \ d \wedge \neg e) & &) \wedge \\
(& (\ b) & \vee & \\
 & (\neg b \wedge \neg c) & \vee & \\
 & (\neg b \wedge \ c \wedge \ e) & &) \wedge \\
(& (\neg c) & \vee & \\
 & (\ c \wedge \ d) & &)
\end{array}$$

which results in the *consolidated* satoku matrix on the right side of figure 8, in contrast to the *consolidated* satoku matrix of the original "optimized" problem on the left side.

P	$- - -$	$- - - -$	$- - -$	$- -$
s_{0_0}	$1 \circ \circ$	$- - - -$	$- - -$	$- -$
s_{0_1}	$\circ 1 \circ$	$0 - - -$	$- - -$	$- -$
s_{0_2}	$\circ \circ 1$	$- - 0 -$	$- 0 -$	$0\ 1$
s_{1_0}	$- 0 -$	$1 \circ \circ \circ$	$0 - -$	$- -$
s_{1_1}	$- - -$	$\circ 1 \circ \circ$	$- 0 -$	$0\ 1$
s_{1_2}	$- - 0$	$\circ \circ 1 \circ$	$- - -$	$1\ 0$
s_{1_3}	$- - -$	$\circ \circ \circ 1$	$- - 0$	$- -$
s_{2_0}	$- - -$	$0 - - -$	$1 \circ \circ$	$- -$
s_{2_1}	$- - 0$	$- 0 - -$	$\circ 1 \circ$	$- -$
s_{2_2}	$- - -$	$- - - 0$	$\circ \circ 1$	$- -$
s_{3_0}	$- - 0$	$- 0 - -$	$- - -$	$1 \circ$
s_{3_1}	$- - -$	$- - 0 -$	$- - -$	$\circ 1$

.

P	$- - -$	$- - - -$	$- - -$	$- -$
s_{0_0}	$1 \circ \circ$	$- - - -$	$- - -$	$- -$
s_{0_1}	$\circ 1 \circ$	$0 - - -$	$1\ 0\ 0$	$- -$
s_{0_2}	$\circ \circ 1$	$1\ 0\ 0\ 0$	$0\ 0\ 1$	$0\ 1$
s_{1_0}	$- 0 -$	$1 \circ \circ \circ$	$0 - -$	$- -$
s_{1_1}	$- - 0$	$\circ 1 \circ \circ$	$1\ 0\ 0$	$0\ 1$
s_{1_2}	$- - 0$	$\circ \circ 1 \circ$	$1\ 0\ 0$	$1\ 0$
s_{1_3}	$- - 0$	$\circ \circ \circ 1$	$1\ 0\ 0$	$1\ 0$
s_{2_0}	$- - 0$	$0 - - -$	$1 \circ \circ$	$- -$
s_{2_1}	$1\ 0\ 0$	$1\ 0\ 0\ 0$	$\circ 1 \circ$	$1\ 0$
s_{2_2}	$- 0 -$	$1\ 0\ 0\ 0$	$\circ \circ 1$	$0\ 1$
s_{3_0}	$- - 0$	$- 0 - -$	$- - 0$	$1 \circ$
s_{3_1}	$- - -$	$- - 0\ 0$	$- 0 -$	$\circ 1$

.

Figure 8: satoku matrix for plain CNF problem with maximized conflicts

Note: This technique to enrich the CNF formula with redundant information is not essential and generally does not work with propositional problems in direct encoding (although it does reduce the amount of binary at-most-one clauses by identifying redundancies). See section 8.1 for the general principle in the satoku matrix.

Note: This "trick" is not proprietary to structural logic, it is just as well available for the usual mapping of CNF problems to graphs.

7.2 Mapping Propositonal Variables

Although structural logic has no concept of propositional variables, it is still possible to map them to a satoku matrix in a natural manner. This is achieved by adding clauses of the form:

$$(p \lor \neg p)$$

for each propositional variable p to a CDF formula.

7.3 Mapping Example

Here is a 3-variable "AND" with added variable clauses on the left side and the extended formula with maximized conflicts (see section 7.1) on the right side:

$$\begin{array}{ll}
(\neg a \lor \neg b \lor c) & \land \\
(\neg a \lor b \lor \neg c) & \land \\
(\neg a \lor b \lor c) & \land \\
(a \lor \neg b \lor \neg c) & \land \\
(a \lor \neg b \lor c) & \land \\
(a \lor b \lor \neg c) & \land \\
(a \lor b \lor c) & \land \\
(a \lor \neg a) & \land \\
(b \lor \neg b) & \land \\
(c \lor \neg c) &
\end{array}$$

$$\begin{array}{llll}
(& (\neg a) & \lor & \\
 & (a \land \neg b) & \lor & \\
 & (a \land b \land c) & &) \land \\
(& (\neg a) & \lor & \\
 & (a \land b) & \lor & \\
 & (a \land \neg b \land \neg c) & &) \land \\
(& (\neg a) & \lor & \\
 & (a \land b) & \lor & \\
 & (a \land \neg b \land c) & &) \land \\
(& (a) & \lor & \\
 & (\neg a \land \neg b) & \lor & \\
 & (\neg a \land b \land \neg c) & &) \land \\
(& (a) & \lor & \\
 & (\neg a \land \neg b) & \lor & \\
 & (\neg a \land b \land c) & &) \land \\
(& (a) & \lor & \\
 & (\neg a \land b) & \lor & \\
 & (\neg a \land \neg b \land \neg c) & &) \land \\
(& (a) & \lor & \\
 & (\neg a \land b) & \lor & \\
 & (\neg a \land \neg b \land c) & &) \land \\
(& (a) & \lor & \\
 & (\neg a) & &) \land \\
(& (b) & \lor & \\
 & (\neg b) & &) \land \\
(& (c) & \lor & \\
 & (\neg c) & &)
\end{array}$$

The satoku matrix for the unmaximized formula is shown in figure 9.

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −	0 1	− −	− −	$\neg a \vee$
s_{0_1}	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− −	0 1	− −	$\neg b \vee$
s_{0_2}	∘ ∘ 1	− − 0	− − −	− − 0	− − −	− − 0	− − −	− −	− −	1 0	c
s_{1_0}	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −	0 1	− −	− −	$\neg a \vee$
s_{1_1}	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− −	1 0	− −	$b \vee$
s_{1_2}	− − 0	∘ ∘ 1	− − 0	− − −	− − 0	− − −	− − 0	− −	− −	0 1	$\neg c$
s_{2_0}	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	0 1	− −	− −	$\neg a \vee$
s_{2_1}	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− −	1 0	− −	$b \vee$
s_{2_2}	− − −	− − 0	∘ ∘ 1	− − 0	− − −	− − 0	− − −	− −	− −	1 0	c
s_{3_0}	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −	1 0	− −	− −	$a \vee$
s_{3_1}	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− −	0 1	− −	$\neg b \vee$
s_{3_2}	− − 0	− − −	− − 0	∘ ∘ 1	− − 0	− − −	− − 0	− −	− −	0 1	$\neg c$
s_{4_0}	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −	1 0	− −	− −	$a \vee$
s_{4_1}	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− −	0 1	− −	$\neg b \vee$
s_{4_2}	− − −	− − 0	− − −	− − 0	∘ ∘ 1	− − 0	− − −	− −	− −	1 0	c
s_{5_0}	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −	1 0	− −	− −	$a \vee$
s_{5_1}	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− −	1 0	− −	$b \vee$
s_{5_2}	− − 0	− − −	− − 0	− − −	− − 0	∘ ∘ 1	− − 0	− −	− −	0 1	$\neg c$
s_{6_0}	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘	1 0	− −	− −	$a \vee$
s_{6_1}	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− −	1 0	− −	$b \vee$
s_{6_2}	− − −	− − 0	− − −	− − 0	− − −	− − 0	∘ ∘ 1	− −	− −	1 0	c
s_{7_0}	0 − −	0 − −	0 − −	− − −	− − −	− − −	− − −	1 ∘	− −	− −	a
s_{7_1}	− − −	− − −	− − −	0 − −	0 − −	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{8_0}	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	− − −	− −	1 ∘	− −	b
s_{8_1}	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{9_0}	− − −	− − 0	− − −	− − 0	− − −	− − 0	− − −	− −	− −	1 ∘	c
s_{9_1}	− − 0	− − −	− − 0	− − −	− − 0	− − −	− − 0	− −	− −	∘ 1	$\neg c$

.

Figure 9: satoku matrix for plain 3-variable “AND”

The added propositional variable clauses are separated from the relevant core satoku matrix, since they are redundant and not necessary to decide the matrix. This follows directly from the properties of an adjacency matrix and the corresponding independent set problem.

The satoku matrix in figure 10 resulting from the propositional formula, transformed to maximize conflicts, shows some interesting properties.

P	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 0	1 0	
s_{0_0}	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a$
s_{0_1}	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$a \wedge \neg b$
s_{0_2}	○ ○ 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 0	1 0	$a \wedge b \wedge c$
s_{1_0}	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a$
s_{1_1}	0 0 1	○ 1 ○	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 0	1 0	$a \wedge b$
s_{1_2}	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$a \wedge \neg b \wedge \neg c$
s_{2_0}	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a$
s_{2_1}	0 0 1	0 1 0	○ 1 ○	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 0	1 0	$a \wedge b$
s_{2_2}	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$a \wedge \neg b \wedge c$
s_{3_0}	0 0 1	0 1 0	0 1 0	1 ○ ○	1 0 0	1 0 0	1 0 0	1 0	1 0	1 0	a
s_{3_1}	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a \wedge \neg b$
s_{3_2}	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a \wedge b \wedge \neg c$
s_{4_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 ○ ○	1 0 0	1 0 0	1 0	1 0	1 0	a
s_{4_1}	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a \wedge \neg b$
s_{4_2}	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0	0 0	0 0	$\neg a \wedge b \wedge c$
s_{5_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 ○ ○	1 0 0	1 0	1 0	1 0	a
s_{5_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0	0 0	0 0	$\neg a \wedge b$
s_{5_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0	0 0	0 0	$\neg a \wedge \neg b \wedge \neg c$
s_{6_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 ○ ○	1 0	1 0	1 0	a
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0	0 0	0 0	$\neg a \wedge b$
s_{6_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0	0 0	0 0	$\neg a \wedge \neg b \wedge c$
s_{7_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 ○	1 0	1 0	a
s_{7_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○	0 0	0 0	$\neg a$
s_{8_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 ○	1 0	b
s_{8_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	○ ○	0 0	$\neg b$
s_{9_0}	0 0 1	0 1 0	0 1 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0	1 0	1 ○	c
s_{9_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	○ ○	$\neg c$

Figure 10: satoku matrix for 3-variable "AND" with maximized conflicts

Foremost, all macro state cells of the *consolidated* matrix are *decided*. It follows directly from the deduction rules, namely requirement update (section 5.5), that all *possible* states are necessarily equivalent.

The solution for the mapped propositional formula can be directly derived by examining the *decided* 2-state macro state cells $c_{m_{7_7}}, c_{m_{8_8}}, c_{m_{9_9}}$, representing the propositional variables. Only the atomic states representing a, b and c are *possible*, the states for $\neg a$, $\neg b$ and $\neg c$ are *impossible*. Therefore the solution to the original problem is $a \wedge b \wedge c$.

7.4 Mapping a Satoku Matrix to CNF

Here is a minimal algorithm for mapping a satoku matrix to a propositional formula (CNF).

- Each atomic state of a state row s_{i_j} is mapped to a propositional variable.
- Each cell c_i is mapped to a propositional clause, with unnegated propositional variables of the corresponding state rows s_{i_j} as alternatives.
- Each *impossible* inter-cell conflict relationship $s_{i_{j_{g_h}}}, \mathsf{Imp}(s_{i_{j_{g_h}}})$, for a state row s_{i_j} is mapped as implication $s_{i_j} \rightarrow \neg s_{g_h}$, transformed to a disjunction $\neg s_{i_j} \vee \neg s_{g_h}$.

Note: It is obvious that once an inter-cell CFR $s_{i_{j_{g_h}}}$ is mapped to $\neg s_{i_j} \vee \neg s_{g_h}$, mapping the mirror CFR $s_{g_{h_{i_j}}}$ to $s_{g_h} \rightarrow \neg s_{i_j}$, transformed to $\neg s_{g_h} \vee \neg s_{i_j}$ is redundant.

Note: If it is not obvious, that mapping the intra-cell conflict relationships can be omitted, consider the difference between one or more propositional variables becoming true and selecting exactly one alternative from a clause.

Algorithm 6 (Minimal mapping of satoku matrix to CNF)**.**
start formula
for each cell c_{i_i}:
    start disjunction
    for each cell row $r_{i_{j_i}}$:
        add variable p_{i_j} to disjunction
    add disjunction to formula

for each status row s_{i_j}:
    for each cell row $r_{i_{j_g}}$ in status row $s_{i_j}, g > i$:
        for each singular state $s_{i_{j_{g_h}}}$ in cell row $r_{i_{j_g}}$:
            if $\mathsf{Imp}(s_{i_{j_{g_h}}})$:
                add disjunction $(\neg p_{i_j} \vee \neg p_{g_h})$ to formula

The result of this algorithm is strictly CNF, albeit in many cases with an entirely different set of variables than the original CNF problem.

Applying algorithm 6 to the satoku matrix in figure 9, results in the following propositional formula:

$$
\begin{array}{llllll}
(s_{0_0} \vee s_{0_1} \vee s_{0_2}) & \wedge & (s_{1_0} \vee s_{1_1} \vee s_{1_2}) & \wedge & (s_{2_0} \vee s_{2_1} \vee s_{2_2}) & \wedge \\
(s_{3_0} \vee s_{3_1} \vee s_{3_2}) & \wedge & (s_{4_0} \vee s_{4_1} \vee s_{4_2}) & \wedge & (s_{5_0} \vee s_{5_1} \vee s_{5_2}) & \wedge \\
(s_{6_0} \vee s_{6_1} \vee s_{6_2}) & \wedge & (s_{7_0} \vee s_{7_1}) & \wedge & (s_{8_0} \vee s_{8_1}) & \wedge \\
(s_{9_0} \vee s_{9_1}) & \wedge & & & & \\
(\neg s_{0_0} \vee \neg s_{3_0}) & \wedge & (\neg s_{0_0} \vee \neg s_{4_0}) & \wedge & (\neg s_{0_0} \vee \neg s_{5_0}) & \wedge \\
(\neg s_{0_0} \vee \neg s_{6_0}) & \wedge & (\neg s_{0_0} \vee \neg s_{7_0}) & \wedge & (\neg s_{0_1} \vee \neg s_{1_1}) & \wedge \\
(\neg s_{0_1} \vee \neg s_{2_1}) & \wedge & (\neg s_{0_1} \vee \neg s_{5_1}) & \wedge & (\neg s_{0_1} \vee \neg s_{6_1}) & \wedge \\
(\neg s_{0_1} \vee \neg s_{8_0}) & \wedge & (\neg s_{0_2} \vee \neg s_{1_2}) & \wedge & (\neg s_{0_2} \vee \neg s_{3_2}) & \wedge \\
(\neg s_{0_2} \vee \neg s_{5_2}) & \wedge & (\neg s_{0_2} \vee \neg s_{9_1}) & \wedge & (\neg s_{1_0} \vee \neg s_{3_0}) & \wedge \\
(\neg s_{1_0} \vee \neg s_{4_0}) & \wedge & (\neg s_{1_0} \vee \neg s_{5_0}) & \wedge & (\neg s_{1_0} \vee \neg s_{6_0}) & \wedge \\
(\neg s_{1_0} \vee \neg s_{7_0}) & \wedge & (\neg s_{1_1} \vee \neg s_{3_1}) & \wedge & (\neg s_{1_1} \vee \neg s_{4_1}) & \wedge \\
(\neg s_{1_1} \vee \neg s_{8_1}) & \wedge & (\neg s_{1_2} \vee \neg s_{2_2}) & \wedge & (\neg s_{1_2} \vee \neg s_{4_2}) & \wedge \\
(\neg s_{1_2} \vee \neg s_{6_2}) & \wedge & (\neg s_{1_2} \vee \neg s_{9_0}) & \wedge & (\neg s_{2_0} \vee \neg s_{3_0}) & \wedge \\
(\neg s_{2_0} \vee \neg s_{4_0}) & \wedge & (\neg s_{2_0} \vee \neg s_{5_0}) & \wedge & (\neg s_{2_0} \vee \neg s_{6_0}) & \wedge \\
(\neg s_{2_0} \vee \neg s_{7_0}) & \wedge & (\neg s_{2_1} \vee \neg s_{3_1}) & \wedge & (\neg s_{2_1} \vee \neg s_{4_1}) & \wedge \\
(\neg s_{2_1} \vee \neg s_{8_1}) & \wedge & (\neg s_{2_2} \vee \neg s_{3_2}) & \wedge & (\neg s_{2_2} \vee \neg s_{5_2}) & \wedge \\
(\neg s_{2_2} \vee \neg s_{9_1}) & \wedge & (\neg s_{3_0} \vee \neg s_{7_1}) & \wedge & (\neg s_{3_1} \vee \neg s_{5_1}) & \wedge \\
(\neg s_{3_1} \vee \neg s_{6_1}) & \wedge & (\neg s_{3_1} \vee \neg s_{8_0}) & \wedge & (\neg s_{3_2} \vee \neg s_{4_2}) & \wedge \\
(\neg s_{3_2} \vee \neg s_{6_2}) & \wedge & (\neg s_{3_2} \vee \neg s_{9_0}) & \wedge & (\neg s_{4_0} \vee \neg s_{7_1}) & \wedge \\
(\neg s_{4_1} \vee \neg s_{5_1}) & \wedge & (\neg s_{4_1} \vee \neg s_{6_1}) & \wedge & (\neg s_{4_1} \vee \neg s_{8_0}) & \wedge \\
(\neg s_{4_2} \vee \neg s_{5_2}) & \wedge & (\neg s_{4_2} \vee \neg s_{9_1}) & \wedge & (\neg s_{5_0} \vee \neg s_{7_1}) & \wedge \\
(\neg s_{5_1} \vee \neg s_{8_1}) & \wedge & (\neg s_{5_2} \vee \neg s_{6_2}) & \wedge & (\neg s_{5_2} \vee \neg s_{9_0}) & \wedge \\
(\neg s_{6_0} \vee \neg s_{7_1}) & \wedge & (\neg s_{6_1} \vee \neg s_{8_1}) & \wedge & (\neg s_{6_2} \vee \neg s_{9_1}) &
\end{array}
$$

Figure 11 shows an excerpt of the *unconsolidated* satoku matrix derived from that formula. This is a very destructurized version of a 3-variable "AND".

P	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{0_0}	1 ∘ ∘	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	0 −	0 −	0 −	0 −	0 −	−−
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	0 −
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{1_0}	−−−	1 ∘ ∘	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	− 0
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{2_0}	−−−	−−−	1 ∘ ∘	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{2_1}	−−−	−−−	∘ 1 ∘	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{2_2}	−−−	−−−	∘ ∘ 1	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{3_0}	−−−	−−−	−−−	1 ∘ ∘	−−−	−−−	−−−	−−	−−	−−	− 0	−−	−−	−−	−−	−−
s_{3_1}	−−−	−−−	−−−	∘ 1 ∘	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{3_2}	−−−	−−−	−−−	∘ ∘ 1	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{4_0}	−−−	−−−	−−−	−−−	1 ∘ ∘	−−−	−−−	−−	−−	−−	−−	− 0	−−	−−	−−	−−
s_{4_1}	−−−	−−−	−−−	−−−	∘ 1 ∘	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{4_2}	−−−	−−−	−−−	−−−	∘ ∘ 1	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{5_0}	−−−	−−−	−−−	−−−	−−−	1 ∘ ∘	−−−	−−	−−	−−	−−	−−	− 0	−−	−−	−−
s_{5_1}	−−−	−−−	−−−	−−−	−−−	∘ 1 ∘	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{5_2}	−−−	−−−	−−−	−−−	−−−	∘ ∘ 1	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{6_0}	−−−	−−−	−−−	−−−	−−−	−−−	1 ∘ ∘	−−	−−	−−	−−	−−	−−	− 0	−−	−−
s_{6_1}	−−−	−−−	−−−	−−−	−−−	−−−	∘ 1 ∘	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{6_2}	−−−	−−−	−−−	−−−	−−−	−−−	∘ ∘ 1	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{7_0}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	1 ∘	−−	−−	−−	−−	−−	−−	− 0	−−
s_{7_1}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	∘ 1	−−	−−	−−	−−	−−	−−	−−	−−
s_{8_0}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	1 ∘	−−	−−	−−	−−	−−	−−	−−
s_{8_1}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	∘ 1	−−	−−	−−	−−	−−	−−	−−
s_{9_0}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	1 ∘	−−	−−	−−	−−	−−	−−
s_{9_1}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	∘ 1	−−	−−	−−	−−	−−	−−
s_{10_0}	0 −−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	1 ∘	−−	−−	−−	−−	−−
s_{10_1}	−−−	−−−	−−−	0 −−	−−−	−−−	−−−	−−	−−	−−	∘ 1	−−	−−	−−	−−	−−
s_{11_0}	0 −−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	1 ∘	−−	−−	−−	−−
s_{11_1}	−−−	−−−	−−−	−−−	0 −−	−−−	−−−	−−	−−	−−	−−	∘ 1	−−	−−	−−	−−
s_{12_0}	0 −−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	1 ∘	−−	−−	−−
s_{12_1}	−−−	−−−	−−−	−−−	−−−	0 −−	−−−	−−	−−	−−	−−	−−	∘ 1	−−	−−	−−
s_{13_0}	0 −−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	1 ∘	−−	−−
s_{13_1}	−−−	−−−	−−−	−−−	−−−	−−−	0 −−	−−	−−	−−	−−	−−	−−	∘ 1	−−	−−
s_{14_0}	0 −−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	1 ∘	−−
s_{14_1}	−−−	−−−	−−−	−−−	−−−	−−−	−−−	0 −	−−	−−	−−	−−	−−	−−	∘ 1	−−
s_{15_0}	− 0 −	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	1 ∘
s_{15_1}	−−−	− 0 −	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	∘ 1

.

Figure 11: satoku matrix for 3-variable "AND" from direct encoding (*unconsolidated*)

Figure 12 shows an excerpt of the *consolidated* satoku matrix derived from the direct encoding formula. The consolidation algorithm has restored the core problem including the representation of the original variables, exactly as shown in figure 9.

P	−−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
s_{0_0}	1 ∘ ∘	−−−	−−−	0−−	0−−	0−−	0−−	0 1	−−	−−	0 1	0 1	0 1	0 1	0 1	−−
s_{0_1}	∘ 1 ∘	−0−	−0−	−−−	−−−	−0−	−0−	−−	0 1	−−	−−	−−	−−	−−	−−	0 1
s_{0_2}	∘ ∘ 1	−−0	−−−	−−0	−−−	−−0	−−−	−−	−−	1 0	−−	−−	−−	−−	−−	−−
s_{1_0}	−−−	1 ∘ ∘	−−−	0−−	0−−	0−−	0−−	0 1	−−	−−	−−	−−	−−	−−	−−	−−
s_{1_1}	−0−	∘ 1 ∘	−−−	−0−	−0−	−−−	−−−	−−	1 0	−−	−−	−−	−−	−−	−−	1 0
s_{1_2}	−−0	∘ ∘ 1	−−0	−−−	−−0	−−−	−−0	−−	−−	0 1	−−	−−	−−	−−	−−	−−
s_{2_0}	−−−	−−−	1 ∘ ∘	0−−	0−−	0−−	0−−	0 1	−−	−−	−−	−−	−−	−−	−−	−−
s_{2_1}	−0−	−−−	∘ 1 ∘	−0−	−0−	−−−	−−−	−−	1 0	−−	−−	−−	−−	−−	−−	−−
s_{2_2}	−−−	−−0	∘ ∘ 1	−−0	−−−	−−0	−−−	−−	−−	1 0	−−	−−	−−	−−	−−	−−
s_{3_0}	0−−	0−−	0−−	1 ∘ ∘	−−−	−−−	−−−	1 0	−−	−−	1 0	−−	−−	−−	1 0	−−
s_{3_1}	−−−	−0−	−0−	∘ 1 ∘	−−−	−0−	−0−	−−	0 1	−−	−−	−−	−−	−−	−−	−−
s_{3_2}	−−0	−−−	−−0	∘ ∘ 1	−−0	−−−	−−0	−−	−−	0 1	−−	−−	−−	−−	−−	−−
s_{4_0}	0−−	0−−	0−−	−−−	1 ∘ ∘	−−−	−−−	1 0	−−	−−	−−	1 0	−−	−−	1 0	−−
s_{4_1}	−−−	−0−	−0−	−−−	∘ 1 ∘	−0−	−0−	−−	0 1	−−	−−	−−	−−	−−	−−	−−
s_{4_2}	−−−	−−0	−−−	−−0	∘ ∘ 1	−−0	−−−	−−	−−	1 0	−−	−−	−−	−−	−−	−−
s_{5_0}	0−−	0−−	0−−	−−−	−−−	1 ∘ ∘	−−−	1 0	−−	−−	−−	−−	1 0	−−	1 0	−−
s_{5_1}	−0−	−−−	−−−	−0−	−0−	∘ 1 ∘	−−−	−−	1 0	−−	−−	−−	−−	−−	−−	−−
s_{5_2}	−−0	−−−	−−0	−−−	−−0	∘ ∘ 1	−−0	−−	−−	0 1	−−	−−	−−	−−	−−	−−
s_{6_0}	0−−	0−−	0−−	−−−	−−−	−−−	1 ∘ ∘	1 0	−−	−−	−−	−−	−−	1 0	1 0	−−
s_{6_1}	−0−	−−−	−−−	−0−	−0−	−−−	∘ 1 ∘	−−	1 0	−−	−−	−−	−−	−−	−−	−−
s_{6_2}	−−−	−−0	−−−	−−0	−−−	−−0	∘ ∘ 1	−−	−−	1 0	−−	−−	−−	−−	−−	−−
s_{7_0}	0−−	0−−	0−−	−−−	−−−	−−−	−−−	1 ∘	−−	−−	−−	−−	−−	−−	1 0	−−
s_{7_1}	−−−	−−−	−−−	0−−	0−−	0−−	0−−	∘ 1	−−	−−	−−	−−	−−	−−	−−	−−
s_{8_0}	−0−	−−−	−−−	−0−	−0−	−−−	−−−	−−	1 ∘	−−	−−	−−	−−	−−	−−	−−
s_{8_1}	−−−	−0−	−0−	−−−	−−−	−0−	−0−	−−	∘ 1	−−	−−	−−	−−	−−	−−	−−
s_{9_0}	−−−	−−0	−−−	−−0	−−−	−−0	−−−	−−	−−	1 ∘	−−	−−	−−	−−	−−	−−
s_{9_1}	−−0	−−−	−−0	−−−	−−0	−−−	−−0	−−	−−	∘ 1	−−	−−	−−	−−	−−	−−
s_{10_0}	0−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	1 ∘	−−	−−	−−	−−	−−
s_{10_1}	−−−	−−−	−−−	0−−	−−−	−−−	−−−	−−	−−	−−	∘ 1	−−	−−	−−	−−	−−
s_{11_0}	0−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	1 ∘	−−	−−	−−	−−
s_{11_1}	−−−	−−−	−−−	−−−	0−−	−−−	−−−	−−	−−	−−	−−	∘ 1	−−	−−	−−	−−
s_{12_0}	0−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	1 ∘	−−	−−	−−
s_{12_1}	−−−	−−−	−−−	−−−	−−−	0−−	−−−	−−	−−	−−	−−	−−	∘ 1	−−	−−	−−
s_{13_0}	0−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	1 ∘	−−	−−
s_{13_1}	−−−	−−−	−−−	−−−	−−−	−−−	0−−	−−	−−	−−	−−	−−	−−	∘ 1	−−	−−
s_{14_0}	0−−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	1 ∘	−−
s_{14_1}	−−−	−−−	−−−	0−−	0−−	0−−	0−−	0 1	−−	−−	−−	−−	−−	−−	∘ 1	−−
s_{15_0}	−0−	−−−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	1 ∘
s_{15_1}	−−−	−0−	−−−	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−	−−	−−	∘ 1

Figure 12: satoku matrix for 3-variable "AND" from direct encoding (*consolidated*)

The effects of direct encoding on CDCL Solvers are illustrated in [SCHCDCL].

8. Satoku Matrix Transformations

The satoku matrix offers various ways to transform one state representation into another state representation, preserving *provability*. A 3-variable propositional XOR as shown in figure 13 is chosen as an example to demonstrate structural analysis of propositional problems with the satoku matrix. Due to the XOR structure, the DPLL resolution algorithm delivers no useful results. There are also no clauses to be learned.

$$\begin{array}{ll} (\neg a \vee \neg b \vee c) & \wedge \\ (\neg a \vee b \vee \neg c) & \wedge \\ (a \vee \neg b \vee \neg c) & \wedge \\ (a \vee b \vee c) & \end{array}$$

Figure 13: 3-variable XOR

8.1 Advance Decisions

superset

A state row s_{i_j} is said to be a superset of state row s_{e_f}, when state row s_{i_j} and state row s_{e_f} are *combinable* in a *consolidated* satoku matrix $\mathbb{S}$ and all *impossible* CFR states $s_{e_{f_{g_h}}}$ of *undecided* cell rows $r_{e_{f_g}}$ also appear as *impossible* CFR states $s_{i_{j_{g_h}}}$ in state row s_{i_j}:

$$\begin{array}{rl} & \mathsf{Con}(\mathbb{S}) \wedge \mathsf{Cmb}(s_{i_j}, s_{e_f}) \wedge \\ & \forall r_{e_{f_g}} \forall s_{e_{f_{g_h}}} : \mathsf{Und}(r_{e_{f_g}}) \wedge \mathsf{Imp}(s_{e_{f_{g_h}}}) \rightarrow \mathsf{Imp}(s_{i_{j_{g_h}}}) \\ \Leftrightarrow & s_{i_j} \supseteq s_{e_f} \end{array}$$

When s_{i_j} is a superset of state row $s_{g_h}, i \neq g$, state row s_{i_j} can be transformed to require state row s_{g_h} without affecting *provability* of a satoku matrix $\mathbb{S}$.

The proof uses the obvious fact, when state row s_{i_j} is a superset of state row s_{e_f}, that for all *impossible* CFR states $s_{e_{f_{g_h}}}$ of the *undecided* cell rows $r_{e_{f_g}}$ of a state row s_{e_f} the state row s_{g_h} has an *impossible* CFR $s_{g_{h_{e_f}}}$ (due to mirror property). Since state row s_{i_j} is a superset of state row s_{e_f}, CFR $s_{i_{j_{g_h}}}$ must also be *impossible*. Therefore, CFR $s_{g_{h_{i_j}}}$ must also be *impossible* (mirror property):

$$\forall s_{g_h} : s_{i_j} \supseteq s_{e_f} \wedge \mathsf{Mutex}(s_{g_h}, s_{e_f}) \rightarrow \mathsf{Mutex}(s_{g_h}, s_{i_j})$$

The *provability* of a satoku matrix $\mathbb{S}$ would only change, if there was a state row s_{g_h}, where CFR $s_{g_{h_{e_f}}}$ was *impossible* and CFR $s_{g_{h_{i_j}}}$ was *required* and therefore *possible*. However, we have just shown, that such a condition cannot exist in a *consolidated* satoku matrix $\mathbb{S}$, when $s_{i_j} \supseteq s_{e_f}$.

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	− − −	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	− 0 −	− − −	− 0 −	− −	0 1	− −	
s_{0_2}	∘ ∘ 1	− − 0	− − 0	− − −	− −	− −	1 0	
s_{1_0}	− − −	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	− 0 −	∘ 1 ∘	− 0 −	− − −	− −	1 0	− −	
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − 0	− −	− −	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	− − −	1 0	− −	− −	
s_{2_1}	− − −	− 0 −	∘ 1 ∘	− 0 −	− −	0 1	− −	
s_{2_2}	− − 0	− − −	∘ ∘ 1	− − 0	− −	− −	0 1	
s_{3_0}	0 − −	0 − −	− − −	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	− 0 −	− − −	− 0 −	∘ 1 ∘	− −	1 0	− −	
s_{3_2}	− − −	− − 0	− − 0	∘ ∘ 1	− −	− −	1 0	
s_{4_0}	0 − −	0 − −	− − −	− − −	1 ∘	− −	− −	a
s_{4_1}	− − −	− − −	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − −	− 0 −	− − −	− −	1 ∘	− −	b
s_{5_1}	− − −	− 0 −	− − −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− − −	− − 0	− − 0	− − −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− − −	− − −	− − 0	− −	− −	∘ 1	$\neg c$

.

(a) ex-xor-3.v-000

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	− − −	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 −	− − −	− 0 −	− −	0 1	− −	
s_{0_2}	∘ ∘ 1	0 − 0	− − 0	− − −	− −	− −	1 0	
s_{1_0}	− 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	− 0 −	∘ 1 ∘	− 0 −	− − −	− −	1 0	− −	
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − 0	− −	− −	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	− − −	1 0	− −	− −	
s_{2_1}	− − −	− 0 −	∘ 1 ∘	− 0 −	− −	0 1	− −	
s_{2_2}	− − 0	− − −	∘ ∘ 1	− − 0	− −	− −	0 1	
s_{3_0}	0 − −	0 − −	− − −	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	− 0 −	− − −	− 0 −	∘ 1 ∘	− −	1 0	− −	
s_{3_2}	− − −	− − 0	− − 0	∘ ∘ 1	− −	− −	1 0	
s_{4_0}	0 − −	0 − −	− − −	− − −	1 ∘	− −	− −	a
s_{4_1}	− − −	− − −	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − −	− 0 −	− − −	− −	1 ∘	− −	b
s_{5_1}	− − −	− 0 −	− − −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− − −	− − 0	− − 0	− − −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− − −	− − −	− − 0	− −	− −	∘ 1	$\neg c$

.

(b) ex-xor-3.v-001

Figure 14: Advance decision stage 1

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	− 0 0	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 1	− − −	1 0 0	1 0	0 1	0 1	
s_{0_2}	∘ ∘ 1	0 1 0	1 0 0	− − −	1 0	1 0	1 0	
s_{1_0}	1 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	0 0 −	∘ 1 ∘	− 0 −	− − −	− −	1 0	− −	
s_{1_2}	0 − 0	∘ ∘ 1	− − −	− − 0	− −	− −	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	− − −	1 0	− −	− −	
s_{2_1}	− − 0	− 0 −	∘ 1 ∘	− 0 −	− −	0 1	− −	
s_{2_2}	− − 0	− − −	∘ ∘ 1	− − 0	− −	− −	0 1	
s_{3_0}	0 − −	0 − −	− − −	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	− 0 −	− − −	− 0 −	∘ 1 ∘	− −	1 0	− −	
s_{3_2}	− 0 −	− − 0	− − 0	∘ ∘ 1	− −	− −	1 0	
s_{4_0}	0 − −	0 − −	− − −	− − −	1 ∘	− −	− −	a
s_{4_1}	1 0 0	− − −	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − −	− 0 −	− − −	− −	1 ∘	− −	b
s_{5_1}	− − 0	− 0 −	− − −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− 0 −	− − 0	− − 0	− − −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− − −	− − −	− − 0	− −	− −	∘ 1	$\neg c$

.

(a) ex-xor-3.v-002

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	1 0 0	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 1	− − −	1 0 0	1 0	0 1	0 1	
s_{0_2}	∘ ∘ 1	0 1 0	1 0 0	− − −	1 0	1 0	1 0	
s_{1_0}	1 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	0 0 1	∘ 1 ∘	1 0 0	− − −	1 0	1 0	1 0	
s_{1_2}	0 1 0	∘ ∘ 1	− − −	1 0 0	1 0	0 1	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	− − −	1 0	− −	− −	
s_{2_1}	− − 0	− 0 −	∘ 1 ∘	− 0 −	− −	0 1	− −	
s_{2_2}	− − 0	− 0 −	∘ ∘ 1	− − 0	− −	− −	0 1	
s_{3_0}	0 − −	0 − −	− − −	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	− 0 −	− − 0	− 0 −	∘ 1 ∘	− −	1 0	− −	
s_{3_2}	− 0 −	− − 0	− − 0	∘ ∘ 1	− −	− −	1 0	
s_{4_0}	0 − −	0 − −	− − −	− − −	1 ∘	− −	− −	a
s_{4_1}	1 0 0	1 0 0	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − 0	− 0 −	− − −	− −	1 ∘	− −	b
s_{5_1}	− − 0	− 0 −	− − −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− 0 −	− − 0	− − 0	− − −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− 0 −	− − −	− − 0	− −	− −	∘ 1	$\neg c$

.

(b) ex-xor-3.v-003

Figure 15: Advance decision stage 2

The satoku matrix, generated from the propositional formula in figure 13, is shown in figure 14a. In figure 14b, state row s_{1_0} has the same *impossible* CFR states as state row s_{0_0} and cell row $r_{1_{0_0}}$ has therefore been transformed to require state row s_{0_0}.

After consolidation, cell row $r_{0_{0_1}}$ has been transformed to require state row s_{1_0} in figure 15a. Figure 15b shows the satoku matrix after consolidation.

Since requirement update algorithm 4 merges all *impossible* singular states from a state row s_{g_h} into state row s_{e_f}, if cell row $r_{e_{f_g}}$ is *bound* and CFR $s_{e_{f_{g_h}}}$ is *required*. Therefore, state row s_{e_f} becomes a superset of state row s_{g_h}.

It follows that if state row s_{i_j} is a superset of state row s_{e_f}, it is then also a superset of state row s_{g_h} by transitivity. Therefore *bound* cell rows $r_{e_{f_g}}$ with required CFR $s_{e_{f_{g_h}}}$ can be omitted, when determining whether state row s_{i_j} is a superset of state row s_{e_f}.

Proof. Let state row s_{e_f} be a superset of state row s_{g_h}. Let state row s_{i_j} be a superset of state row s_{e_f}, but *mutually exclusive* with state row s_{g_h}. It follows that CFR $s_{i_{j_{g_h}}}$ is *impossible*. Therefore, mirror state $s_{g_{h_{i_j}}}$ is also *impossible*. Since state row s_{e_f} is a superset of s_{g_h}, CFR $s_{e_{f_{i_j}}}$ must also be *impossible* and CFR $s_{i_{j_{e_f}}}$ must also be *impossible*. Therefore, s_{i_j} is *mutually exclusive* with s_{e_f}, which means that $s_{i_j} \not\supseteq s_{e_f}$, since s_{i_j} and s_{e_f} must be *combinable* to satisfy the superset conditions. □

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	1 0 0	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 1	− − −	1 0 0	1 0	0 1	0 1	
s_{0_2}	∘ ∘ 1	0 1 0	1 0 0	− − −	1 0	1 0	1 0	
s_{1_0}	1 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	0 0 1	∘ 1 ∘	1 0 0	− − −	1 0	1 0	1 0	
s_{1_2}	0 1 0	∘ ∘ 1	− − −	1 0 0	1 0	0 1	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	− 0 0	1 0	− −	− −	
s_{2_1}	− − 0	− 0 −	∘ 1 ∘	0 0 −	− −	0 1	− −	
s_{2_2}	− − 0	− 0 −	∘ ∘ 1	0 − 0	− −	− −	0 1	
s_{3_0}	0 − −	0 − −	− 0 0	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	− 0 −	− − 0	0 0 −	∘ 1 ∘	− −	1 0	− −	
s_{3_2}	− 0 −	− − 0	0 − 0	∘ ∘ 1	− −	− −	1 0	
s_{4_0}	0 − −	0 − −	− − −	− − −	1 ∘	− −	− −	a
s_{4_1}	1 0 0	1 0 0	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − 0	− 0 −	− − −	− −	1 ∘	− −	b
s_{5_1}	− − 0	− 0 −	− − −	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− 0 −	− − 0	− − 0	− − −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− 0 −	− − −	− − 0	− −	− −	∘ 1	$\neg c$

(a) ex-xor-3.v-005

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	1 0 0	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 1	1 0 0	1 0 0	1 0	0 1	0 1	
s_{0_2}	∘ ∘ 1	0 1 0	1 0 0	1 0 0	1 0	1 0	1 0	
s_{1_0}	1 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	0 0 1	∘ 1 ∘	1 0 0	1 0 0	1 0	1 0	1 0	
s_{1_2}	0 1 0	∘ ∘ 1	1 0 0	1 0 0	1 0	0 1	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	1 0 0	1 0	− −	− −	
s_{2_1}	1 0 0	1 0 0	∘ 1 ∘	0 0 1	0 1	0 1	1 0	
s_{2_2}	1 0 0	1 0 0	∘ ∘ 1	0 1 0	0 1	1 0	0 1	
s_{3_0}	0 − −	0 − −	1 0 0	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	1 0 0	1 0 0	0 0 1	∘ 1 ∘	0 1	1 0	0 1	
s_{3_2}	1 0 0	1 0 0	0 1 0	∘ ∘ 1	0 1	0 1	1 0	
s_{4_0}	0 − −	0 − −	1 0 0	1 0 0	1 ∘	− −	− −	a
s_{4_1}	1 0 0	1 0 0	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − 0	− 0 −	− − 0	− −	1 ∘	− −	b
s_{5_1}	− − 0	− 0 −	− − 0	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− 0 −	− − 0	− − 0	− 0 −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− 0 −	− 0 −	− − 0	− −	− −	∘ 1	$\neg c$

(b) ex-xor-3.v-006

Figure 16: Advance Decision stage 3

In figure 16a, state row s_{2_0} has the same *impossible* CFR states as state row s_{3_0} and cell row $r_{3_{0_2}}$ has therefore been transformed to require state row s_{2_0}, and cell row $r_{2_{0_3}}$ has been transformed to require state row s_{3_0}. Figure 16b shows the satoku matrix after consolidation.

The advance decision deduction rule allows to transform a satoku matrix based on a CNF problem into a form which is similar to maximizing conflicts (see section 7.1) (compare figure 16b and figure 17) without the need to resort to propositional logic. In contrast to maximizing conflicts this method is resilient to different encodings.

Applying the advance decision deduction rule, is essentially an arbitrary decision that has been made in advance, without an excplicit necessity.

As a rule of thumb, without further analysis, advance decisions for state rows that are equal, i.e. one requires the other, are more desirable. Advance decisions where the most equal state rows can be made to require each other are most desirable.

8.2 Redundancy Removal

The satoku matrix generated from the propositional formula in figure 13 with maximized conflicts is shown in figure 17. The satoku matrix is identical to the one obtained by advance decisions in figure 16b.

P	− − −	− − −	− − −	− − −	− −	− −	− −	
s_{0_0}	1 ∘ ∘	1 0 0	0 − −	0 − −	0 1	− −	− −	
s_{0_1}	∘ 1 ∘	0 0 1	1 0 0	1 0 0	1 0	0 1	0 1	
s_{0_2}	∘ ∘ 1	0 1 0	1 0 0	1 0 0	1 0	1 0	1 0	
s_{1_0}	1 0 0	1 ∘ ∘	0 − −	0 − −	0 1	− −	− −	
s_{1_1}	0 0 1	∘ 1 ∘	1 0 0	1 0 0	1 0	1 0	1 0	
s_{1_2}	0 1 0	∘ ∘ 1	1 0 0	1 0 0	1 0	0 1	0 1	
s_{2_0}	0 − −	0 − −	1 ∘ ∘	1 0 0	1 0	− −	− −	
s_{2_1}	1 0 0	1 0 0	∘ 1 ∘	0 0 1	0 1	0 1	1 0	
s_{2_2}	1 0 0	1 0 0	∘ ∘ 1	0 1 0	0 1	1 0	0 1	
s_{3_0}	0 − −	0 − −	1 0 0	1 ∘ ∘	1 0	− −	− −	
s_{3_1}	1 0 0	1 0 0	0 0 1	∘ 1 ∘	0 1	1 0	0 1	
s_{3_2}	1 0 0	1 0 0	0 1 0	∘ ∘ 1	0 1	0 1	1 0	
s_{4_0}	0 − −	0 − −	1 0 0	1 0 0	1 ∘	− −	− −	a
s_{4_1}	1 0 0	1 0 0	0 − −	0 − −	∘ 1	− −	− −	$\neg a$
s_{5_0}	− 0 −	− − 0	− 0 −	− − 0	− −	1 ∘	− −	b
s_{5_1}	− − 0	− 0 −	− − 0	− 0 −	− −	∘ 1	− −	$\neg b$
s_{6_0}	− 0 −	− − 0	− − 0	− 0 −	− −	− −	1 ∘	c
s_{6_1}	− − 0	− 0 −	− 0 −	− − 0	− −	− −	∘ 1	$\neg c$

Figure 17: satoku matrix for 3-variable “XOR” with maximized conflicts

The states in c_{0_0} and c_{1_1} in figure 17 are structurally equivalent aside from permutation of their states. I.e., for each state s_{0_j} there is a corresponding *required* state $s_{0_{j_{1_h}}}$.

The reverse is also true. It is therefore obvious, that in a *consolidated* satoku matrix the states in c_{1_1} cannot have a different effect on *provability* than the states in c_{0_0}. c_{1_1} is therefore redundant and can be removed. The same argument holds for cells c_{2_2} and c_{3_3}.

The reduced satoku matrix is shown in figure 18.

P	$---$	$---$	$--$	$--$	$--$	
s_{0_0}	$1\circ\circ$	$0--$	$0\,1$	$--$	$--$	
s_{0_1}	$\circ 1 \circ$	$1\,0\,0$	$1\,0$	$0\,1$	$0\,1$	
s_{0_2}	$\circ\circ 1$	$1\,0\,0$	$1\,0$	$1\,0$	$1\,0$	
s_{1_0}	$0--$	$1\circ\circ$	$1\,0$	$--$	$--$	
s_{1_1}	$1\,0\,0$	$\circ 1 \circ$	$0\,1$	$0\,1$	$1\,0$	
s_{1_2}	$1\,0\,0$	$\circ\circ 1$	$0\,1$	$1\,0$	$0\,1$	
s_{2_0}	$0--$	$1\,0\,0$	$1\,\circ$	$--$	$--$	a
s_{2_1}	$1\,0\,0$	$0--$	$\circ\,1$	$--$	$--$	$\neg a$
s_{3_0}	$-0-$	$-0-$	$--$	$1\,\circ$	$--$	b
s_{3_1}	$--0$	$--0$	$--$	$\circ\,1$	$--$	$\neg b$
s_{4_0}	$-0-$	$--0$	$--$	$--$	$1\,\circ$	c
s_{4_1}	$--0$	$-0-$	$--$	$--$	$\circ\,1$	$\neg c$

Figure 18: Redundancies removed from satoku matrix for 3-variable "XOR"

redundant covers

In a *consolidated* satoku matrix $\mathbb{S}$, if all cell rows $r_{i_{j_g}}$ of a cell $c_{i_g}, i \neq g$, are *bound*, the cell c_{g_g} is *redundant* (Red), since all its atomic states with their direct conflict relationships are fully represented by at least one of the atomic states of cell c_{i_i}. It is said that Cell c_{i_i} *covers* (Cov) cell c_{g_g}:

$$\forall r_{i_{j_g}} : \mathsf{Con}(\mathbb{S}) \wedge r_{i_{j_g}} \in c_{i_g} \wedge \mathsf{Bnd}(r_{i_{j_g}}) \wedge i \neq g \Leftrightarrow \mathsf{Cov}(c_{i_i}, c_{g_g}) \Leftrightarrow \mathsf{Red}(c_{g_g}).$$

Note that the cells do not have to have the same number of atomic states. E.g., in figure 18, c_{1_2} consists of CFR states for 3 atomic states and *covers* c_{2_2}, which has 2 atomic states.

8.3 Merging Cells

Two or more cells can be merged into a single cell, by adding requirements for all state combinations of the merged cells to a single cell.

The first step of the procedure for cells c_{0_0}, c_{1_1} is shown in figure 19. A new cell c_{5_5} with 9 states has been added in a new section and *impossible* conflict relationships have been added to conflict relationship cell c_{5_0}, such that each state from cell c_{0_0} will become *required* 3 times, when the satoku matrix is *consolidated*.

P	−−−	−−−	−−	−−	−−	−−−−−−−−−	
s_{0_0}	1 ∘ ∘	0 − −	0 1	− −	− −	− − − 0 0 0 0 0 0	
s_{0_1}	∘ 1 ∘	1 0 0	1 0	0 1	0 1	0 0 0 − − − 0 0 0	
s_{0_2}	∘ ∘ 1	1 0 0	1 0	1 0	1 0	0 0 0 0 0 0 − − −	
s_{1_0}	0 − −	1 ∘ ∘	1 0	− −	− −	− − − − − − − − −	
s_{1_1}	1 0 0	∘ 1 ∘	0 1	0 1	1 0	− − − − − − − − −	
s_{1_2}	1 0 0	∘ ∘ 1	0 1	1 0	0 1	− − − − − − − − −	
s_{2_0}	0 − −	1 0 0	1 ∘	− −	− −	− − − − − − − − −	a
s_{2_1}	1 0 0	0 − −	∘ 1	− −	− −	− − − − − − − − −	$\neg a$
s_{3_0}	− 0 −	− 0 −	− −	1 ∘	− −	− − − − − − − − −	b
s_{3_1}	− − 0	− − 0	− −	∘ 1	− −	− − − − − − − − −	$\neg b$
s_{4_0}	− 0 −	− − 0	− −	− −	1 ∘	− − − − − − − − −	c
s_{4_1}	− − 0	− 0 −	− −	− −	∘ 1	− − − − − − − − −	$\neg c$
s_{5_0}	− 0 0	− − −	− −	− −	− −	1 ∘ ∘ ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_1}	− 0 0	− − −	− −	− −	− −	∘ 1 ∘ ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_2}	− 0 0	− − −	− −	− −	− −	∘ ∘ 1 ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_3}	0 − 0	− − −	− −	− −	− −	∘ ∘ ∘ 1 ∘ ∘ ∘ ∘ ∘	
s_{5_4}	0 − 0	− − −	− −	− −	− −	∘ ∘ ∘ ∘ 1 ∘ ∘ ∘ ∘	
s_{5_5}	0 − 0	− − −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ 1 ∘ ∘ ∘	
s_{5_6}	0 0 −	− − −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ 1 ∘ ∘	
s_{5_7}	0 0 −	− − −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ ∘ 1 ∘	
s_{5_8}	0 0 −	− − −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ ∘ ∘ 1	.

Figure 19: Merge cells c_{0_0}, c_{1_1} for 3-variable "XOR", require states from c_{0_0}

The second step is shown in figure 19, where *impossible* conflict relationships have been added to cell c_{5_1} such that each state from c_{1_1} becomes *required* 3 times when the satoku matrix is *consolidated*. The pattern in cells c_{5_0}, c_{5_1} shows, that the 9 states cover all possible combinations of the states in cells c_{0_0}, c_{1_1}.

P	−−−	−−−	−−	−−	−−	−−−−−−−−−	
s_{0_0}	1 ∘ ∘	0 − −	0 1	− −	− −	− − − 0 0 0 0 0 0	
s_{0_1}	∘ 1 ∘	1 0 0	1 0	0 1	0 1	0 0 0 − − − 0 0 0	
s_{0_2}	∘ ∘ 1	1 0 0	1 0	1 0	1 0	0 0 0 0 0 0 − − −	
s_{1_0}	0 − −	1 ∘ ∘	1 0	− −	− −	− 0 0 − 0 0 − 0 0	
s_{1_1}	1 0 0	∘ 1 ∘	0 1	0 1	1 0	0 − 0 0 − 0 0 − 0	
s_{1_2}	1 0 0	∘ ∘ 1	0 1	1 0	0 1	0 0 − 0 0 − 0 0 −	
s_{2_0}	0 − −	1 0 0	1 ∘	− −	− −	− − − − − − − − −	a
s_{2_1}	1 0 0	0 − −	∘ 1	− −	− −	− − − − − − − − −	$\neg a$
s_{3_0}	− 0 −	− 0 −	− −	1 ∘	− −	− − − − − − − − −	b
s_{3_1}	− − 0	− − 0	− −	∘ 1	− −	− − − − − − − − −	$\neg b$
s_{4_0}	− 0 −	− − 0	− −	− −	1 ∘	− − − − − − − − −	c
s_{4_1}	− − 0	− 0 −	− −	− −	∘ 1	− − − − − − − − −	$\neg c$
s_{5_0}	− 0 0	− 0 0	− −	− −	− −	1 ∘ ∘ ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_1}	− 0 0	0 − 0	− −	− −	− −	∘ 1 ∘ ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_2}	− 0 0	0 0 −	− −	− −	− −	∘ ∘ 1 ∘ ∘ ∘ ∘ ∘ ∘	
s_{5_3}	0 − 0	− 0 0	− −	− −	− −	∘ ∘ ∘ 1 ∘ ∘ ∘ ∘ ∘	
s_{5_4}	0 − 0	0 − 0	− −	− −	− −	∘ ∘ ∘ ∘ 1 ∘ ∘ ∘ ∘	
s_{5_5}	0 − 0	0 0 −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ 1 ∘ ∘ ∘	
s_{5_6}	0 0 −	− 0 0	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ 1 ∘ ∘	
s_{5_7}	0 0 −	0 − 0	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ ∘ 1 ∘	
s_{5_8}	0 0 −	0 0 −	− −	− −	− −	∘ ∘ ∘ ∘ ∘ ∘ ∘ ∘ 1	.

Figure 20: Merge cells c_{0_0}, c_{1_1} for 3-variable "XOR", require states from c_{1_1}

After consolidation and removal of *impossible* states, cell c_{5_5} contains 4 *possible* states as shown in figure 21. Since, by construction, all states from cells c_{0_0}, c_{1_1} have been transformed and are represented in cell c_{5_5}, cells c_{0_0}, c_{1_1} are now *redundant* and can be removed. Since the variable representations in cells $c_{2_2}, c_{3_3}, c_{4_4}$ are also *redundant*, as previously shown, the satoku matrix can be reduced to cell c_{5_5} only.

Cell c_{5_5} has only states that are *decided*. The 4 possible solutions for the original propositional formula in figure 13 can be easily derived by constructing the 4 possible variants which cause cell c_{5_5} to become *decided*.

P	− − −	− − −	− −	− −	− −	− − − −	
s_{0_0}	1 ∘ ∘	0 − −	0 1	− −	− −	− − 0 0	
s_{0_1}	∘ 1 ∘	1 0 0	1 0	0 1	0 1	0 0 1 0	
s_{0_2}	∘ ∘ 1	1 0 0	1 0	1 0	1 0	0 0 0 1	
s_{1_0}	0 − −	1 ∘ ∘	1 0	− −	− −	0 0 − −	
s_{1_1}	1 0 0	∘ 1 ∘	0 1	0 1	1 0	1 0 0 0	
s_{1_2}	1 0 0	∘ ∘ 1	0 1	1 0	0 1	0 1 0 0	
s_{2_0}	0 − −	1 0 0	1 ∘	− −	− −	0 0 − −	a
s_{2_1}	1 0 0	0 − −	∘ 1	− −	− −	− − 0 0	$\neg a$
s_{3_0}	− 0 −	− 0 −	− −	1 ∘	− −	0 − 0 −	b
s_{3_1}	− − 0	− − 0	− −	∘ 1	− −	− 0 − 0	$\neg b$
s_{4_0}	− 0 −	− − 0	− −	− −	1 ∘	− 0 0 −	c
s_{4_1}	− − 0	− 0 −	− −	− −	∘ 1	0 − − 0	$\neg c$
s_{5_0}	1 0 0	0 1 0	0 1	0 1	1 0	1 ∘ ∘ ∘	
s_{5_1}	1 0 0	0 0 1	0 1	1 0	0 1	∘ 1 ∘ ∘	
s_{5_2}	0 1 0	1 0 0	1 0	0 1	0 1	∘ ∘ 1 ∘	
s_{5_3}	0 0 1	1 0 0	1 0	1 0	1 0	∘ ∘ ∘ 1	

.

Figure 21: Merge cells c_{0_0}, c_{1_1} for 3-variable "XOR", add states from c_{1_1}

Note, that the satisfiability of the mapped propositional problem can be deduced without actually deciding the macro state cell $c_{m_{5_5}}$.

The possible solutions for the propositional problem can also be directly mapped from the decided conflict relations $r_{5_{i_2}}, r_{5_{i_3}}, r_{5_{i_4}}$:

$$
\begin{array}{ll}
(\neg a \wedge \neg b \wedge c) & \vee \\
(\neg a \wedge b \wedge \neg c) & \vee \\
(a \wedge \neg b \wedge \neg c) & \vee \\
(a \wedge b \wedge c) &
\end{array}
$$

Observe, that merging was not really necessary at all, since the possible solutions to the propositional problem already appear in conflict relations $r_{0_{1_a}}, r_{0_{2_a}}, r_{1_{1_a}}, r_{1_{2_a}}, a = (2, 3, 4)$.

Also, *provability* of the satoku matrix can already be deduced from each of the *decided* relevant conflict relations $r_{0_{1_1}}, r_{0_{2_1}}, r_{1_{1_0}}, r_{1_{2_0}}$ of states $s_{0_1}, s_{0_2}, s_{1_1}, s_{1_2}$ independently.

When mapping the conflict relations in $r_{0_{0_a}}, r_{1_{0_a}}, a = (2, 3, 4)$ to their propositional variable equivalents, they represent partial assignments for the boolean satisfiability problem.

Obviously, merging cells is equivalent to performing a distributive expansion of propositional clauses. However, the number of required operations to perform the distributive expansion over the 4 original propositional clauses has required less potentially exponential steps than doing it in the usual manner. The 3×3 merge can even be reduced to a $2 + 1 + 1$ merge, by leaving out any combinations that would become *impossible*.

So, trivially a representation of all possible solutions to a mapped propositional problem can be generated by merging all cells of a satoku matrix into a single cell. If such a cell is *possible*, trivially the mapped propositional formula is satisfiable.

A good algorithm for merging is to merge only source cells c_{i_i}, c_{e_e}, if the projected resulting cell s_{x_x} has less *possible* atomic states than the sum of *possible* atomic states of the source cells c_{i_i}, c_{e_e}:

Algorithm 7 (merge for satoku matrix reduction)**.**
for each cell row c_i:
  $\mathit{min\_pos\_cell} := (-1, -1)$ ***for each*** conflict relationship cell $c_{i_j}, j > i$:
    $\mathit{cell\_pos\_count} := |c_{i_i}| + |c_{j_j}|$
    if $\mathit{cell\_pos\_count} < |\mathsf{Mrg}(c_{i_i}, c_{j_j})|$:
      if $\mathit{min\_pos\_cell}[0] < 0 \lor \mathit{min\_pos\_cell}[1] > \mathit{cell\_pos\_count}$:
        $\mathit{min\_pos\_cell} := (j, \mathit{cell\_pos\_count})$
  if $\mathit{min\_pos\_cell}[0] >= 0$:
    $j := \mathit{min\_pos\_cell}[1]$
    $s_{x_x} := \mathsf{Mrg}(c_{i_i}, c_{j_j})$
    remove source cells c_{i_i}, c_{j_j}
    decrement i

This algorithm is guaranteed to make the satoku matrix smaller.

Merging potentially increases the number of *impossible* singular states by merging partially disjoint cell rows. At least it may tighten the conflict context by merging subset cell rows with less *impossible* singular states into superset cell rows with more *impossible* singular states.

Therefore a point could be made for merging cells when the sum of *possible* atomic states of the source cells is equal to the projected size of the merge result. However, there are not necessarily any changes in the conflict context and there is no change in the number of 2-state partitions for the source cells and the merged cell, if the number of atomic states in each source cell is even.

9. Indirect Conflicts

By construction, all direct consequences of *mutual exclusion* between atomic states, as well as all direct consequences of *mutual exclusion* between atomic states and cells are represented in a *consolidated* satoku matrix.

For the source of contradictions that leaves only indirect conflicts $r_{x_{y_g}}$, which are consequences of merging 2 cell rows $r_{i_{j_g}}, r_{e_{f_g}}$, when merging state rows s_{i_j}, s_{e_f}. So the first condition for an indirect conflict are 2 *combinable* state rows s_{i_j}, s_{e_f} in a *consolidated* satoku matrix $\mathbb{S}$:

$$\mathsf{Con}(\mathbb{S}) \land s_{i_j}, s_{e_f} \in \mathbb{S} \land \mathsf{Cmb}(s_{i_j}, s_{e_f})$$

9.1 Immediate Indirect Conflicts

For immediate indirect conflicts, it is necessary that merging two *combinable* cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ results in a *conflict* cell row $r_{x_{y_g}}$. Therefore, one cell row must complement the other, i.e. for each *possible* CFR $s_{i_{j_{g_h}}}$, there must be an *impossible* CFR $s_{e_{f_{g_h}}}$, and for each *possible* CFR $s_{e_{f_{g_h}}}$ there must be an *impossible* CFR $s_{i_{j_{g_h}}}$:

$$\begin{array}{rl} & \exists r_{i_{j_g}} \exists r_{e_{f_g}} \forall s_{i_{j_{g_h}}} \forall s_{e_{f_{g_h}}} : \\ & \qquad s_{i_{j_{g_h}}} \in r_{i_{j_g}} \land s_{e_{f_{g_h}}} \in r_{e_{f_g}} \\ & \quad \land \quad (\mathsf{Pos}(s_{i_{j_{g_h}}}) \rightarrow \mathsf{Imp}(s_{e_{f_{g_h}}})) \\ & \quad \land \quad (\mathsf{Pos}(s_{e_{f_{g_h}}}) \rightarrow \mathsf{Imp}(s_{i_{j_{g_h}}})) \\ \Leftrightarrow & \mathsf{Mrg}(r_{i_{j_g}}, r_{e_{f_g}}) \rightarrow \mathsf{Cfl} \end{array}$$

Is is obvious that the immediate result of merging an *unrestricted* cell row $r_{i_{j_g}}$ with another cell row $r_{e_{f_g}}$ in a *consolidated* satoku matrix $\mathbb{S}$ cannot produce a *conflict* cell row $r_{x_{y_g}}$, unless $r_{e_{f_g}}$ is already an *impossible* cell row. However, the consolidation algorithm makes state $s_{e_{f_{e_f}}}$ *impossible*, so it is no longer *combinable* with any other state:

$$\begin{array}{rl} & \quad \langle \; 1 \;\; 1 \;\; 1 \;\; 1 \; \rangle \; r_{i_{j_g}} \\ & \land \langle \; ? \;\; ? \;\; ? \;\; ? \; \rangle \; r_{e_{f_g}} \\ \hline & \quad \langle \; 0 \;\; 0 \;\; 0 \;\; 0 \; \rangle \; r_{x_{y_g}} \\ \Rightarrow & r_{e_{f_g}} = \langle \; 0 \;\; 0 \;\; 0 \;\; 0 \; \rangle \\ \Rightarrow & \mathsf{Imp}(r_{e_{f_g}}) \Rightarrow \mathsf{Imp}(s_{e_f}) \Rightarrow \mathsf{Imp}(s_{e_{f_{e_f}}}) \\ \Rightarrow & \forall s_{i_{j_{i_j}}} : \neg \mathsf{Cmb}(s_{e_{f_{e_f}}}, s_{i_{j_{i_j}}}) \end{array}$$

Therefore both cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ must be *restricted* in order for merging to result in a *conflict* cell row $\mathsf{Mrg}(r_{i_{j_g}}, r_{e_{f_g}}) \to \mathsf{Cfl}(r_{x_{y_g}})$.

Both cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ must also be *undecided*.

Proof. If cell row $r_{i_{j_g}}$ is *bound* with *required* state $s_{i_{j_{g_h}}}$, then state $s_{e_{f_{g_h}}}$ of cell row $r_{e_{f_g}}$ must be *impossible* for a *conflict* merge result. Consequently, if CFR $s_{e_{f_{g_h}}}$ is *impossible*, then CFR $s_{g_{h_{e_f}}}$ must also be *impossible* due to commutativity of *mutual exclusion*. Since *required* CFR $s_{i_{j_{g_h}}}$ causes all singular states of state row s_{g_h} to be merged into state row s_{i_j} during consolidation, CFR $s_{i_{j_{e_f}}}$ must also be *impossible*. But this violates the condition that state rows s_{i_j}, s_{e_f} must be *combinable*. $\square$

The following example illustrates this. Starting out with *bound* cell row $r_{0_{1_2}}$ in figure 22a, CFR $s_{1_{0_{2_1}}}$ is set *impossible* to satisfy the conditions for an immediate indirect conflict in figure 22b. This also causes CFR $s_{2_{1_{1_0}}}$ to become *impossible*. Matrix consolidation causes CFR $s_{0_{1_{1_0}}}$ and therefore CFR $s_{1_{0_{0_1}}}$ to become impossible in figure 22c. Therefore state row s_{0_1} and state row s_{1_0} are no longer *combinable*.

P	$- - - -$	$- - - -$	$- - - -$
s_{0_0}	$1 \circ \circ \circ$	$- - - -$	$- - - -$
s_{0_1}	$\circ 1 \circ \circ$	$- - - -$	$0\,1\,0\,0$
s_{0_2}	$\circ \circ 1 \circ$	$- - - -$	$- - - -$
s_{0_3}	$\circ \circ \circ 1$	$- - - -$	$- - - -$
s_{1_0}	$- - - -$	$1 \circ \circ \circ$	$- - - -$
s_{1_1}	$- - - -$	$\circ 1 \circ \circ$	$- - - -$
s_{1_2}	$- - - -$	$\circ \circ 1 \circ$	$- - - -$
s_{1_3}	$- - - -$	$\circ \circ \circ 1$	$- - - -$
s_{2_0}	$- 0 - -$	$- - - -$	$1 \circ \circ \circ$
s_{2_1}	$- - - -$	$- - - -$	$\circ 1 \circ \circ$
s_{2_2}	$- 0 - -$	$- - - -$	$\circ \circ 1 \circ$
s_{2_3}	$- 0 - -$	$- - - -$	$\circ \circ \circ 1$

(a) *bound* cell row $r_{0_{1_2}}$

P	$- - - -$	$- - - -$	$- - - -$
s_{0_0}	$1 \circ \circ \circ$	$- - - -$	$- - - -$
s_{0_1}	$\circ 1 \circ \circ$	$- - - -$	$0\,1\,0\,0$
s_{0_2}	$\circ \circ 1 \circ$	$- - - -$	$- - - -$
s_{0_3}	$\circ \circ \circ 1$	$- - - -$	$- - - -$
s_{1_0}	$- - - -$	$1 \circ \circ \circ$	$- 0 - -$
s_{1_1}	$- - - -$	$\circ 1 \circ \circ$	$- - - -$
s_{1_2}	$- - - -$	$\circ \circ 1 \circ$	$- - - -$
s_{1_3}	$- - - -$	$\circ \circ \circ 1$	$- - - -$
s_{2_0}	$- 0 - -$	$- - - -$	$1 \circ \circ \circ$
s_{2_1}	$- - - -$	$0 - - -$	$\circ 1 \circ \circ$
s_{2_2}	$- 0 - -$	$- - - -$	$\circ \circ 1 \circ$
s_{2_3}	$- 0 - -$	$- - - -$	$\circ \circ \circ 1$

(b) complementary CFR $s_{1_{0_{2_1}}}$

P	$- - - -$	$- - - -$	$- - - -$
s_{0_0}	$1 \circ \circ \circ$	$- - - -$	$- - - -$
s_{0_1}	$\circ 1 \circ \circ$	$0 - - -$	$0\,1\,0\,0$
s_{0_2}	$\circ \circ 1 \circ$	$- - - -$	$- - - -$
s_{0_3}	$\circ \circ \circ 1$	$- - - -$	$- - - -$
s_{1_0}	$- 0 - -$	$1 \circ \circ \circ$	$- 0 - -$
s_{1_1}	$- - - -$	$\circ 1 \circ \circ$	$- - - -$
s_{1_2}	$- - - -$	$\circ \circ 1 \circ$	$- - - -$
s_{1_3}	$- - - -$	$\circ \circ \circ 1$	$- - - -$
s_{2_0}	$- 0 - -$	$- - - -$	$1 \circ \circ \circ$
s_{2_1}	$- - - -$	$0 - - -$	$\circ 1 \circ \circ$
s_{2_2}	$- 0 - -$	$- - - -$	$\circ \circ 1 \circ$
s_{2_3}	$- 0 - -$	$- - - -$	$\circ \circ \circ 1$

(c) CFR $s_{0_{1_{1_0}}} \to \neg\mathsf{Cmb}(s_{0_1}, s_{1_0})$

Figure 22: Construct complementary cell row $r_{1_{0_2}}$ for *bound* cell row $r_{0_{1_2}}$

In a *restricted undecided* cell row there must be at least 1 *impossible* state and 2 *possible* states. Therefore, there is a minimum of 3 states in a *restricted undecided* cell row. Any 2 *restricted undecided* 3-state cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ share at least one common *possible* state $s_{i_{j_{g_h}}}, s_{e_{f_{g_h}}}$. Making either one state *impossible*, causes the respective cell row $r_{i_{j_g}}, r_{e_{f_g}}$ to become *decided*, since it now has 1 *possible* state and 2 *impossible* states. Therefore, there cannot be any immediate indirect conflicts in 3-state cell rows.

At least 4 singular states are required for an immediate indirect conflict in 2 cell rows $r_{i_{j_g}}, r_{e_{f_g}}$, 2 of them are *possible*, 2 of them are *impossible*. For each *impossible* state

$s_{i_{j_{g_h}}}$ in cell row $r_{i_{j_g}}$ the corresponding state $s_{e_{f_{g_h}}}$ in cell row $r_{e_{f_g}}$ is *possible*. For each *impossible* state $s_{e_{f_{g_h}}}$ in cell row $r_{e_{f_g}}$ the corresponding state $s_{i_{j_{g_h}}}$ in cell row $r_{i_{j_g}}$ is *possible*.

The example in figure 23 shows that an immediate indirect conflict $(r_{0_{0_2}}, r_{1_{0_2}})$ is not backpropagated $(s_{0_{0_{1_0}}} \leftarrow \mathsf{Imp})$, when detected.

P	− − − −	− − − −	− − − −	− − − −
s_{0_0}	1 ∘ ∘ ∘	− − − −	0 − 0 −	− − − −
s_{0_1}	∘ 1 ∘ ∘	− − − −	− − − −	0 − − −
s_{0_2}	∘ ∘ 1 ∘	− − − −	− − − −	0 − − −
s_{0_3}	∘ ∘ ∘ 1	− − − −	− − − −	0 − − −
s_{1_0}	− − − −	1 ∘ ∘ ∘	− 0 − 0	− − − −
s_{1_1}	− − − −	∘ 1 ∘ ∘	− − − −	0 − − −
s_{1_2}	− − − −	∘ ∘ 1 ∘	− − − −	0 − − −
s_{1_3}	− − − −	∘ ∘ ∘ 1	− − − −	0 − − −
s_{2_0}	0 − − −	− − − −	1 ∘ ∘ ∘	0 − − −
s_{2_1}	− − − −	0 − − −	∘ 1 ∘ ∘	− − − −
s_{2_2}	0 − − −	− − − −	∘ ∘ 1 ∘	0 − − −
s_{2_3}	− − − −	0 − − −	∘ ∘ ∘ 1	− − − −
s_{3_0}	1 0 0 0	− 0 0 0	0 − 0 −	1 ∘ ∘ ∘
s_{3_1}	− − − −	− − − −	− − − −	∘ 1 ∘ ∘
s_{3_2}	− − − −	− − − −	− − − −	∘ ∘ 1 ∘
s_{3_3}	− − − −	− − − −	− − − −	∘ ∘ ∘ 1

(a) Partially consolidated

P	− − − −	− − − −	− − − −	0 − − −
s_{0_0}	1 ∘ ∘ ∘	− − − −	0 − 0 −	0 − − −
s_{0_1}	∘ 1 ∘ ∘	− − − −	− − − −	0 − − −
s_{0_2}	∘ ∘ 1 ∘	− − − −	− − − −	0 − − −
s_{0_3}	∘ ∘ ∘ 1	− − − −	− − − −	0 − − −
s_{1_0}	− − − −	1 ∘ ∘ ∘	− 0 − 0	0 − − −
s_{1_1}	− − − −	∘ 1 ∘ ∘	− − − −	0 − − −
s_{1_2}	− − − −	∘ ∘ 1 ∘	− − − −	0 − − −
s_{1_3}	− − − −	∘ ∘ ∘ 1	− − − −	0 − − −
s_{2_0}	0 − − −	− − − −	1 ∘ ∘ ∘	0 − − −
s_{2_1}	− − − −	0 − − −	∘ 1 ∘ ∘	0 − − −
s_{2_2}	0 − − −	− − − −	∘ ∘ 1 ∘	0 − − −
s_{2_3}	− − − −	0 − − −	∘ ∘ ∘ 1	0 − − −
s_{3_0}	0 0 0 0	0 0 0 0	0 0 0 0	∘ ∘ ∘ ∘
s_{3_1}	− − − −	− − − −	− − − −	∘ 1 ∘ ∘
s_{3_2}	− − − −	− − − −	− − − −	∘ ∘ 1 ∘
s_{3_3}	− − − −	− − − −	− − − −	∘ ∘ ∘ 1

(b) Consolidated satoku matrix $\mathbb{S}$

Figure 23: Immediate indirect conflict

Matrix consolidation is therefore extended with an algorithm to detect immediate indirect conflicts.

Algorithm 8 (detect immediate indirect conflicts)**.**
for each status row s_{i_j}:
  for each cell row $r_{i_{j_g}}$ ***in*** status row s_{i_j}:
    if $|r_{i_{j_g}}| >= 4 \wedge \mathsf{Rst}(r_{i_{j_g}}) \wedge \mathsf{Und}(r_{i_{j_g}})$:
      for each status row $s_{e_f}, e > i$:
        if $\mathsf{Rst}(r_{e_{f_g}}) \wedge \mathsf{Und}(r_{e_{f_g}}) \wedge (\mathsf{Mrg}(r_{i_{j_g}}, r_{e_{f_g}}) \rightarrow \mathsf{Cfl})$:
          $s_{i_{j_{e_f}}} \leftarrow \mathsf{Imp}$

9.2 Hidden Indirect Conflicts

For a hidden indirect conflict in a *consolidated* satoku matrix $\mathbb{S}$, it is necessary, that merging 2 *combinable* state rows $s_{i_j}, s_{e_f}, i \neq e$ with *undecided restricted* cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ results in a series of 1 or more *bound* cell rows $r_{p_{q_r}}$ triggering additional merges and finally revealing a *conflict* cell row $r_{x_{y_g}}$ during consolidation of the satoku matrix $\mathbb{S}$.

This is demonstrated in figures 24 and 25:

P	−−−	−−−	−−−	−−−	−−−	
s_{0_0}	1 ∘ ∘	−−−	−−0	−−0	−0−	$s_{0_{0_g}}$
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	0−−	
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	0−−	
s_{1_0}	−−−	1 ∘ ∘	−0−	−0−	−−0	$s_{1_{0_g}}$
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	0−−	
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	0−−	
s_{2_0}	−−−	−−−	1 ∘ ∘	0−−	−−−	
s_{2_1}	−−−	0−−	∘ 1 ∘	−−−	−−−	
s_{2_2}	0−−	−−−	∘ ∘ 1	−−−	−−−	
s_{3_0}	−−−	−−−	0−−	1 ∘ ∘	−−−	
s_{3_1}	−−−	0−−	−−−	∘ 1 ∘	−−−	
s_{3_2}	0−−	−−−	−−−	∘ ∘ 1	−−−	
s_{4_0}	−0 0	−0 0	−−−	−−−	1 ∘ ∘	$s_{0_0} \wedge s_{1_0}$
s_{4_1}	0−−	−−−	−−−	−−−	∘ 1 ∘	$\neg s_{0_0}$
s_{4_2}	−−−	0−−	−−−	−−−	∘ ∘ 1	$\neg s_{1_0}$

(a) Request merge of s_{0_0}, s_{1_0}

P	−−−	−−−	−−−	−−−	−−−	
s_{0_0}	1 ∘ ∘	−−−	−−0	−−0	−0−	$s_{0_{0_g}}$
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	0−−	
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	0−−	
s_{1_0}	−−−	1 ∘ ∘	−0−	−0−	−−0	$s_{1_{0_g}}$
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	0−−	
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	0−−	
s_{2_0}	−−−	−−−	1 ∘ ∘	0−−	−−−	
s_{2_1}	−−−	0−−	∘ 1 ∘	−−−	0−−	
s_{2_2}	0−−	−−−	∘ ∘ 1	−−−	0−−	
s_{3_0}	−−−	−−−	0−−	1 ∘ ∘	−−−	
s_{3_1}	−−−	0−−	−−−	∘ 1 ∘	0−−	
s_{3_2}	0−−	−−−	−−−	∘ ∘ 1	0−−	
s_{4_0}	1 0 0	1 0 0	−0 0	−0 0	1 ∘ ∘	$s_{0_0} \wedge s_{1_0}$
s_{4_1}	0−−	−−−	−−−	−−−	∘ 1 ∘	$\neg s_{0_0}$
s_{4_2}	−−−	0−−	−−−	−−−	∘ ∘ 1	$\neg s_{1_0}$

(b) Satisfy *required* $s_{4_{0_{0_0}}}, s_{4_{0_{1_0}}}$

Figure 24: Hidden indirect conflict stage 1

P	−−−	−−−	−−−	−−−	−−−	
s_{0_0}	1 ∘ ∘	−−−	−−0	−−0	−0−	$s_{0_{0_g}}$
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	0−−	
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	0−−	
s_{1_0}	−−−	1 ∘ ∘	−0−	−0−	−−0	$s_{1_{0_g}}$
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	0−−	
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	0−−	
s_{2_0}	−−−	−−−	1 ∘ ∘	0−−	−−−	
s_{2_1}	−−−	0−−	∘ 1 ∘	−−−	0−−	
s_{2_2}	0−−	−−−	∘ ∘ 1	−−−	0−−	
s_{3_0}	−−−	−−−	0−−	1 ∘ ∘	0−−	
s_{3_1}	−−−	0−−	−−−	∘ 1 ∘	0−−	
s_{3_2}	0−−	−−−	−−−	∘ ∘ 1	0−−	
s_{4_0}	1 0 0	1 0 0	1 0 0	0 0 0	1 ∘ ∘	$s_{0_0} \wedge s_{1_0}$
s_{4_1}	0−−	−−−	−−−	−−−	∘ 1 ∘	$\neg s_{0_0}$
s_{4_2}	−−−	0−−	−−−	−−−	∘ ∘ 1	$\neg s_{1_0}$

(a) Satisfy $\mathsf{Req}(s_{4_{0_{2_0}}}) \rightarrow \mathsf{Imp}(s_{4_{0_3}})$

P	−−−	−−−	−−−	−−−	0−−	
s_{0_0}	1 ∘ ∘	0−−	−−0	−−0	0 0 1	$s_{0_{0_g}}$
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	0−−	
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	0−−	
s_{1_0}	0−−	1 ∘ ∘	−0−	−0−	0 1 0	$s_{1_{0_g}}$
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	0−−	
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	0−−	
s_{2_0}	−−−	−−−	1 ∘ ∘	0−−	0−−	
s_{2_1}	−−−	0−−	∘ 1 ∘	−−−	0−−	
s_{2_2}	0−−	−−−	∘ ∘ 1	−−−	0−−	
s_{3_0}	−−−	−−−	0−−	1 ∘ ∘	0−−	
s_{3_1}	−−−	0−−	−−−	∘ 1 ∘	0−−	
s_{3_2}	0−−	−−−	−−−	∘ ∘ 1	0−−	
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	$s_{0_0} \wedge s_{1_0}$
s_{4_1}	0−−	−−−	−−−	−−−	∘ 1 ∘	$\neg s_{0_0}$
s_{4_2}	−−−	0−−	−−−	−−−	∘ ∘ 1	$\neg s_{1_0}$

(b) *consolidated* satoku matrix

Figure 25: Hidden indirect conflict stage 2

It is obvious, that the result of merging any cell row $r_{i_{jg}}$ with an *unrestricted* cell row $r_{e_{fg}}$ produces the same CFR states as given in $r_{i_{jg}}$. So, no new merges are triggered in this case.

It is further obvious, that merging any cell row $r_{i_{jg}}$ with a *bound* cell row $r_{e_{fg}}$ and *required* CFR $s_{e_{fg_h}}$ can only produce a *bound* result cell row $r_{x_{yg}}$ with *required* CFR $s_{x_{yg_h}}$. However, the *required* state row s_{g_h} has already been merged into state row s_{e_f} during a previous consolidation of the satoku matrix $\mathbb{S}$ and a new merge of state row s_{g_h} cannot reveal anything new.

Therefore cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ must both be *restricted* and *undecided*.

As shown previously, this implies that cell rows $r_{i_{j_g}}, r_{e_{f_g}}$ must have a minimum number of 3 singular states, in order for them to produce *bound* cell rows as merge results which in turn trigger a new merge.

9.3 2-State Cells

Sections 9.1 and 9.2 already show that there cannot be any indirect conflicts in a satoku matrix with a maximum cell size of 2. However, to make it perfectly clear, it is summarized here.

The 4 possible cell row states for 2-state cells are:

$\langle\ 0\ \ 0\ \rangle$	*impossible*	*decided*	*restricted*
$\langle\ 0\ \ 1\ \rangle$	*possible*	*decided*	*restricted*
$\langle\ 1\ \ 0\ \rangle$	*possible*	*decided*	*restricted*
$\langle\ 1\ \ 1\ \rangle$	*possible*	*undecided*	*unrestricted*

While states $\langle 01 \rangle$ and $\langle 10 \rangle$ are *restricted*, they are not *undecided* as required for an indirect conflict. Since there are no other *restricted* states available it is not possible to construct an indirect conflict in a *consolidated* satoku matrix consisting of 2-state cells. Without indirect conflicts it is also not possible to construct an indirect *contradiction*. Therefore, if a *consolidated* satoku matrix consisting of 2-state cells is *possible*, nothing else needs to be shown for *provability*.

While an indirect conflict can be constructed in an *unconsolidated* 2-state satoku matrix (see CFR states $r_{0_{0_2}}$ and $r_{1_{0_2}}$ in figure 26a), consolidation always resolves these conditions for 2-state cells. E.g., in figure 26b, satisfying *required* state $s_{0_{0_{2_1}}}$ already leaves states $s_{0_{0_{0_0}}}$ and $s_{1_{1_{1_1}}}$ *mutually exclusive* in cell row $r_{0_{0_1}}$.

P	−−	−−	−−	
s_{0_0}	1 ∘	−−	0 −	$r_{0_{0_2}}$
s_{0_1}	∘ 1	−−	−−	
s_{1_0}	−−	1 ∘	− 0	$r_{1_{0_2}}$
s_{1_1}	−−	∘ 1	−−	
s_{2_0}	0 −	−−	1 ∘	
s_{2_1}	−−	0 −	∘ 1	

(a) Indirect conflict in $r_{0_{0_2}}, r_{1_{0_2}}$

P	−−	−−	−−	
s_{0_0}	1 ∘	0 1	0 1	$r_{0_{0_1}}$
s_{0_1}	∘ 1	−−	−−	
s_{1_0}	0 1	1 ∘	1 0	
s_{1_1}	−−	∘ 1	−−	
s_{2_0}	0 1	−−	1 ∘	
s_{2_1}	−−	0 1	∘ 1	

(b) $\mathsf{Req}(s_{0_{0_{2_1}}}) \to \mathsf{Mutex}(s_{0_{0_{0_0}}}, s_{1_{1_{1_1}}})$

Figure 26: Indirect conflict in *unconsolidated* 2-state satoku matrix

9.4 Refined Provability

As shown, the definition of *provability* can be extended in the following manner.

A satoku matrix is *provable*, if there exists a sequence of successive decisions according to the transformation rules that decides the satoku matrix and does not result in a *contradiction*. This definition of *provability* is the closest analog to boolean satisfiability. *provable*

If all cell rows $r_{i_{jg}}, i \neq g$ of a state row s_{i_j} are *bound* in a *consolidated* satoku matrix $\mathbb{S}$, the corresponding macro state cell $c_{m_{i_i}}$ can be decided by forcing state $s_{i_{j_{i_j}}}$ to become the *required* state for $c_{m_{i_i}}$.

State row s_{i_j} is a superset for each *bound* cell row $r_{i_{jg}}$. All *impossible* states of state rows s_{g_h} required by cell rows $r_{i_{jg}}$ are therefore already present in state row s_{i_j}, so no combination of state rows s_{g_h} can produce a *conflict*, as previously shown. Thus consolidation reduces the satoku matrix $\mathbb{S}$ to the *possible decided* state.

It is therefore not necessary to actually decide a satoku matrix in order to deduce *provability*. Showing that state row s_{i_j} exists in a *consolidated* satoku matrix $\mathbb{S}$, is sufficient.

A *consolidated* satoku matrix $\mathbb{S}$ is *strictly provable*, if successive arbitrary decisions of *undecided* cells according to the transformation rules cannot result in a *contradiction*. There is no equivalent for this definition in propositional logic, since the special cases where it is obvious are primarily trivial. Whereas in structural logic *strict provability* is the standard case. *strictly provable*

It follows trivially, that a *consolidated* satoku matrix $\mathbb{S}$ is *strictly provable*, if it is *possible* and all cell rows $r_{i_{jg}}$ are either *unrestricted* or *decided*. In this case, no indirect conflicts are possible, since they require at least 2 *combinable undecided restricted* cell rows $r_{i_{jg}}, r_{e_{fg}}$.

Specifically, any *consolidated* satoku matrix $\mathbb{S}$, which consists exclusively of 1-state and 2-state cells is *strictly provable*. It is therefore sufficient to reduce a satoku matrix $\mathbb{S}$ to cells with a maximum of 2 states to determine *strict provability*.

If a *consolidated* satoku matrix $\mathbb{S}$ has a state row s_{i_j} whose cell rows $r_{i_{jg}}$ only have a maximum of 2 *possible* states $s_{i_{jgx}}, s_{i_{jgy}}$, the corresponding state $s_{i_{j_{i_j}}}$ can be forced global (as the *required* state of macro cell state $c_{m_{i_i}}$). After consolidation, satoku matrix $\mathbb{S}$ is either a contradiction Ctr or it is *strictly provable*.

According to this definiton, the satoku matrix reduced to cell c_{5_5} in figure 21 is *strictly provable* since all states are *decided* and no further arbitrary decision of *undecided* cells can be made.

10. Advanced Satoku Matrix Transformations

There are some satoku matrix transformations, which are easier to prove with *strict provability* and especially the fact that any satoku matrix consisting of 2-state cells is *strictly provable* (see section 9.3).

10.1 2-State Splitting

core sub-matrix
2-state sub-matrix

In a *consolidated* satoku matrix $\mathbb{S}$, any *possible* sub-matrix of cells consisting of a maximum of 2 *possible* singular states is *strictly provable*. It can therefore be separated from satoku matrix $\mathbb{S}$ as 2-state sub-matrix $\mathbb{S}_2$ without affecting *provability*, leaving the core sub-matrix $\mathbb{C}$, that still needs to be proved.

Proof. It is obvious, that all 1-state cells $c_{m_{i_i}}$ are *decided*. If a 1-state cell is *impossible* $c_{m_{i_i}}$, the satoku matrix $\mathbb{S}$ becomes a contradiction (Ctr). If a 1-state cell $c_{m_{i_i}}$ is *possible* (see figure 28a, cell $c_{m_{4_4}}$), state $s_{i_{0_{i_0}}}$ is required by all other states $s_{e_{f_{e_f}}}, e \neq i$. Consolidaton therefore propagates all *impossible* singular states $s_{i_{0_{g_h}}}, g \neq i$ to all other states $s_{e_{f_{e_f}}}, e \neq i$. In a *consolidated* satoku matrix $\mathbb{S}$, all *possible* 1-state cells $c_{m_{i_i}}$ can therefore be removed without affecting *provability*.

P	−−−	−−−	−−−	−−−	−−	−−	−−	−−	−−
s_{0_0}	1 ∘ ∘	−−−	−−−	−−−	0 −	− 0	−−	0 −	−−
s_{0_1}	∘ 1 ∘	−−−	−−−	−−−	− 0	− 0	−−	−−	−−
s_{0_2}	∘ ∘ 1	−−−	−−−	−−−	− 0	0 −	− 0	−−	−−
s_{1_0}	−−−	1 ∘ ∘	−−−	−−−	− 0	−−	−−	−−	−−
s_{1_1}	−−−	∘ 1 ∘	−−−	−−−	−−	0 −	−−	−−	−−
s_{1_2}	−−−	∘ ∘ 1	−−−	−−−	−−	−−	−−	− 0	−−
s_{2_0}	−−−	−−−	1 ∘ ∘	−−−	−−	− 0	−−	−−	−−
s_{2_1}	−−−	−−−	∘ 1 ∘	−−−	−−	−−	0 −	−−	−−
s_{2_2}	−−−	−−−	∘ ∘ 1	−−−	−−	−−	−−	0 −	−−
s_{3_0}	−−−	−−−	−−−	1 ∘ ∘	−−	−−	−−	−−	−−
s_{3_1}	−−−	−−−	−−−	∘ 1 ∘	−−	−−	−−	−−	−−
s_{3_2}	−−−	−−−	−−−	∘ ∘ 1	−−	−−	−−	−−	−−
s_{4_0}	0 −−	−−−	−−−	−−−	1 ∘	−−	−−	−−	−−
s_{4_1}	− 0 0	0 −−	−−−	−−−	∘ 1	−−	−−	−−	−−
s_{5_0}	−− 0	− 0 −	−−−	−−−	−−	1 ∘	−−	−−	−−
s_{5_1}	0 0 −	−−−	0 −−	−−−	−−	∘ 1	−−	−−	−−
s_{6_0}	−−−	−−−	− 0 −	−−−	−−	−−	1 ∘	−−	−−
s_{6_1}	−− 0	−−−	−−−	−−−	−−	−−	∘ 1	−−	−−
s_{7_0}	0 −−	−−−	−− 0	−−−	−−	−−	−−	1 ∘	−−
s_{7_1}	−−−	−− 0	−−−	−−−	−−	−−	−−	∘ 1	−−
s_{8_0}	−−−	−−−	−−−	−−−	−−	−−	−−	−−	1 ∘
s_{8_1}	−−−	−−−	−−−	−−−	−−	−−	−−	−−	∘ 1

(a) *unconsolidated* satoku matrix $\mathbb{S}$

P	0 −−	−−−	−−−	−−−	1 0	−−	−−	−−	−−
s_{0_0}	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0 0
s_{0_1}	∘ 1 ∘	− 0 −	−−−	−−−	1 0	1 0	−−	−−	−−
s_{0_2}	∘ ∘ 1	−− 0	0 0 1	−−−	1 0	0 1	1 0	0 1	−−
s_{1_0}	0 −−	1 ∘ ∘	−−−	−−−	1 0	−−	−−	−−	−−
s_{1_1}	0 0 1	∘ 1 ∘	0 0 1	−−−	1 0	0 1	1 0	0 1	−−
s_{1_2}	0 1 0	∘ ∘ 1	−− 0	−−−	1 0	1 0	−−	1 0	−−
s_{2_0}	0 1 0	− 0 −	1 ∘ ∘	−−−	1 0	1 0	−−	−−	−−
s_{2_1}	0 1 0	− 0 −	∘ 1 ∘	−−−	1 0	1 0	0 1	−−	−−
s_{2_2}	0 −−	−− 0	∘ ∘ 1	−−−	1 0	−−	−−	0 1	−−
s_{3_0}	0 −−	−−−	−−−	1 ∘ ∘	1 0	−−	−−	−−	−−
s_{3_1}	0 −−	−−−	−−−	∘ 1 ∘	1 0	−−	−−	−−	−−
s_{3_2}	0 −−	−−−	−−−	∘ ∘ 1	1 0	−−	−−	−−	−−
s_{4_0}	0 −−	−−−	−−−	−−−	1 ∘	−−	−−	−−	−−
s_{4_1}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘	0 0	0 0	0 0	0 0
s_{5_0}	0 1 0	− 0 −	−−−	−−−	1 0	1 ∘	−−	−−	−−
s_{5_1}	0 0 1	−− 0	0 0 1	−−−	1 0	∘ 1	1 0	0 1	−−
s_{6_0}	0 −−	−−−	− 0 −	−−−	1 0	−−	1 ∘	−−	−−
s_{6_1}	0 1 0	− 0 −	−−−	−−−	1 0	1 0	∘ 1	−−	−−
s_{7_0}	0 1 0	− 0 −	−− 0	−−−	1 0	1 0	−−	1 ∘	−−
s_{7_1}	0 −−	−− 0	−−−	−−−	1 0	−−	−−	∘ 1	−−
s_{8_0}	0 −−	−−−	−−−	−−−	1 0	−−	−−	−−	1 ∘
s_{8_1}	0 −−	−−−	−−−	−−−	1 0	−−	−−	−−	∘ 1

(b) *consolidated* satoku matrix $\mathbb{S}$

Figure 27: 2-State splitting stage 1

A CFR cell row $r_{i_{j_g}}$ for a state row $s_{i_j}, c_{i_i} \in \mathbb{C}$, and a cell $c_{g_g}, c_{g_g} \in \mathbb{S}_2$, consists of 2 CFR states $s_{i_{j_{g_h}}}$. So $r_{i_{j_g}}$ is either

- *impossible*, which eliminates state row s_{i_j} entirely from satoku matrix $\mathbb{S}$ reducing cell $c_{m_{i_i}}$ to a 1-state cell (see figure 27b, state row s_{4_1}, cell row $r_{4_{1_4}}$, cell $c_{m_{4_4}}$)(see figure 28a, state row s_{4_1}, cell $c_{m_{4_4}}$), or
- *unrestricted*, which allows any of the CFR states $s_{i_{j_{g_h}}}$ without restrictions, or
- *restricted* and *possible*.

If cell row $r_{i_{j_g}}$ is *restricted* and *possible* it is also necessarily *bound* with *required* state $s_{i_{j_{g_h}}}$. Therefore consolidation merges all *impossible* singular states from state row s_{g_h} into state row s_{i_j}. In a *consolidated* satoku matrix $\mathbb{S}$, state row s_{g_h} is then no longer necessary to decide core sub-matrix $\mathbb{C}$.

If all cell rows $r_{i_{j_g}}, c_{i_i} \in \mathbb{C} \wedge c_{g_g} \in \mathbb{S}_2$, are *unrestricted*, all CFR states $s_{i_{j_{g_h}}}$ are *possible* and therefore the corresponding mirror states $s_{g_{h_{i_j}}}$ must also be *possible* (see figure 29a, cell rows $r_{i_{j_6}}, i = 0 \ldots 2$ and cell rows $r_{6_{j_g}}, j = 0 \ldots 1, j = 0 \ldots 2$). This means, that cell c_{g_g} is *independent* of any cell c_{i_i} in core sub-matrix $\mathbb{C}$ and therfore, cell c_{g_g} can be removed without affecting *provability* of core sub-matrix $\mathbb{C}$. □

P	−−	−−−	−−−	−−−	1	−−	−−	−−	−−	−−
s_{0_0}	1 ∘	−0−	−−−	−−−	1	1 0	−−	−−	−−	1 0
s_{0_1}	∘ 1	−−0	0 0 1	−−−	1	0 1	1 0	0 1	−−	0 1
s_{1_0}	−−	1 ∘ ∘	−−−	−−−	1	−−	−−	−−	−−	−−
s_{1_1}	0 1	∘ 1 ∘	0 0 1	−−−	1	0 1	1 0	0 1	−−	0 1
s_{1_2}	1 0	∘ ∘ 1	−−0	−−−	1	1 0	−−	1 0	−−	1 0
s_{2_0}	1 0	−0−	1 ∘ ∘	−−−	1	1 0	−−	−−	−−	1 0
s_{2_1}	1 0	−0−	∘ 1 ∘	−−−	1	1 0	0 1	−−	−−	1 0
s_{2_2}	−−	−−0	∘ ∘ 1	−−−	1	−−	−−	0 1	−−	−−
s_{3_0}	−−	−−−	−−−	1 ∘ ∘	1	−−	−−	−−	−−	−−
s_{3_1}	−−	−−−	−−−	∘ 1 ∘	1	−−	−−	−−	−−	−−
s_{3_2}	−−	−−−	−−−	∘ ∘ 1	1	−−	−−	−−	−−	−−
s_{4_0}	−−	−−−	−−−	−−−	1	−−	−−	−−	−−	−−
s_{5_0}	1 0	−0−	−−−	−−−	1	1 ∘	−−	−−	−−	1 0
s_{5_1}	0 1	−−0	0 0 1	−−−	1	∘ 1	1 0	0 1	−−	0 1
s_{6_0}	−−	−−−	−0−	−−−	1	−−	1 ∘	−−	−−	−−
s_{6_1}	1 0	−0−	−−−	−−−	1	1 0	∘ 1	−−	−−	1 0
s_{7_0}	1 0	−0−	−−0	−−−	1	1 0	−−	1 ∘	−−	1 0
s_{7_1}	−−	−−0	−−−	−−−	1	−−	−−	∘ 1	−−	−−
s_{8_0}	−−	−−−	−−−	−−−	1	−−	−−	−−	1 ∘	−−
s_{8_1}	−−	−−−	−−−	−−−	1	−−	−−	−−	∘ 1	−−
s_{9_0}	1 0	−0−	−−−	−−−	1	1 0	−−	−−	−−	1 ∘
s_{9_1}	0 1	−−0	0 0 1	−−−	1	0 1	1 0	0 1	−−	∘ 1

(a) *impossible* state rows removed

P	−−−	−−−	−−−	−−	−−	−−	−−	−−
s_{0_0}	1 ∘ ∘	−−−	−−−	−−	−−	−−	−−	−−
s_{0_1}	∘ 1 ∘	0 0 1	−−−	0 1	1 0	0 1	−−	0 1
s_{0_2}	∘ ∘ 1	−−0	−−−	1 0	−−	1 0	−−	1 0
s_{1_0}	−0−	1 ∘ ∘	−−−	1 0	−−	−−	−−	1 0
s_{1_1}	−0−	∘ 1 ∘	−−−	1 0	0 1	−−	−−	1 0
s_{1_2}	−−0	∘ ∘ 1	−−−	−−	−−	0 1	−−	−−
s_{2_0}	−−−	−−−	1 ∘ ∘	−−	−−	−−	−−	−−
s_{2_1}	−−−	−−−	∘ 1 ∘	−−	−−	−−	−−	−−
s_{2_2}	−−−	−−−	∘ ∘ 1	−−	−−	−−	−−	−−
s_{3_0}	−0−	−−−	−−−	1 ∘	−−	−−	−−	1 0
s_{3_1}	−−0	0 0 1	−−−	∘ 1	1 0	0 1	−−	0 1
s_{4_0}	−−−	−0−	−−−	−−	1 ∘	−−	−−	−−
s_{4_1}	−0−	−−−	−−−	1 0	∘ 1	−−	−−	1 0
s_{5_0}	−0−	−−0	−−−	1 0	−−	1 ∘	−−	1 0
s_{5_1}	−−0	−−−	−−−	−−	−−	∘ 1	−−	−−
s_{6_0}	−−−	−−−	−−−	−−	−−	−−	1 ∘	−−
s_{6_1}	−−−	−−−	−−−	−−	−−	−−	∘ 1	−−
s_{7_0}	−0−	−−−	−−−	1 0	−−	−−	−−	1 ∘
s_{7_1}	−−0	0 0 1	−−−	0 1	1 0	0 1	−−	∘ 1

(b) 1-state cell removed, re-ordered

Figure 28: 2-State splitting stage 2

If the core sub-matrix $\mathbb{C}$ has been proved, decided state rows can be substituted into the unsplit satoku matrix $\mathbb{S}$ to determine the effect on the states in 2-state sub-matrix $\mathbb{S}_2$.

It is also immaterial, whether the 2-state sub-matrix $\mathbb{S}_2$ is actually removed or not. The 2 intersections between core sub-matrix $\mathbb{C}$ and 2-state cell sub-matrix are necessarily irrelevant to any argument developed in the core sub-matrix $\mathbb{C}$.

10.2 Distractor Reduction

State rows $s_{i_j}, s_{i_f}, j \neq f$, within the same matrix cell row c_i are *mutually exclusive* by definition. Lifting the restriction that the state rows s_{i_j}, s_{i_f} must be *combinable*, allows to define the superset relation between intra-cell state rows s_{i_j}, s_{i_f} as follows.

intra-cell superset A state row s_{i_j} is said to be a superset of state row $s_{i_f}, j \neq f$ in a *consolidated* satoku matrix $\mathbb{S}$ when all *impossible* CFR states $s_{i_{f_{g_h}}}$ of *undecided* cell rows $r_{i_{f_g}}$ also appear as *impossible* CFR states $s_{i_{j_{g_h}}}$ in state row s_{i_j}:

$$\begin{aligned} & \mathsf{Con}(\mathbb{S}) \wedge s_{i_{j_{i_j}}} \in c_{i_i} \wedge s_{i_{f_{i_f}}} \in c_{i_i} \wedge j \neq f \wedge \\ & \forall r_{i_{f_g}} \forall s_{i_{f_{g_h}}} : \mathsf{Und}(r_{i_{f_g}}) \wedge \mathsf{Imp}(s_{i_{f_{g_h}}}) \rightarrow \mathsf{Imp}(s_{i_{j_{g_h}}}) \\ \Leftrightarrow \; & s_{i_j} \supseteq s_{i_f} \end{aligned}$$

P	− − −	− − −	− − −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘	− − −	− − −	− −	− −	− −	− −	− −
s_{0_1}	∘ 1 ∘	0 0 1	− − −	0 1	1 0	0 1	− −	0 1
s_{0_2}	∘ ∘ 1	− − 0	− − −	1 0	− −	1 0	− −	1 0
s_{1_0}	− 0 −	1 ∘ ∘	− − −	1 0	− −	− −	− −	1 0
s_{1_1}	− 0 −	∘ 1 ∘	− − −	1 0	0 1	− −	− −	1 0
s_{1_2}	− − 0	∘ ∘ 1	− − −	− −	− −	0 1	− −	− −
s_{2_0}	− − −	− − −	1 ∘ ∘	− −	− −	− −	− −	− −
s_{2_1}	− − −	− − −	∘ 1 ∘	− −	− −	− −	− −	− −
s_{2_2}	− − −	− − −	∘ ∘ 1	− −	− −	− −	− −	− −
s_{3_0}	− 0 −	− − −	− − −	1 ∘	− −	− −	− −	1 0
s_{3_1}	− − 0	0 0 1	− − −	∘ 1	1 0	0 1	− −	0 1
s_{4_0}	− − −	− 0 −	− − −	− −	1 ∘	− −	− −	− −
s_{4_1}	− 0 −	− − −	− − −	1 0	∘ 1	− −	− −	1 0
s_{5_0}	− 0 −	− − 0	− − −	1 0	− −	1 ∘	− −	1 0
s_{5_1}	− − 0	− − −	− − −	− −	− −	∘ 1	− −	− −
s_{6_0}	− − −	− − −	− − −	− −	− −	− −	1 ∘	− −
s_{6_1}	− − −	− − −	− − −	− −	− −	− −	∘ 1	− −
s_{7_0}	− 0 −	− − −	− − −	1 0	− −	− −	− −	1 ∘
s_{7_1}	− − 0	0 0 1	− − −	0 1	1 0	0 1	− −	∘ 1

(a) *distractor* state rows in core sub-matrix $\mathbb{C}$

P	− − 0	− − −	− − −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘	− − −	− − −	− −	− −	− −	− −	− −
s_{0_1}	∘ 1 ∘	0 0 1	− − −	0 1	1 0	0 1	− −	0 1
s_{0_2}	∘ ∘ ∘	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0 0
s_{1_0}	− 0 0	1 ∘ ∘	− − −	1 0	− −	− −	− −	1 0
s_{1_1}	− 0 0	∘ 1 ∘	− − −	1 0	0 1	− −	− −	1 0
s_{1_2}	− − 0	∘ ∘ 1	− − −	− −	− −	0 1	− −	− −
s_{2_0}	− − 0	− − −	1 ∘ ∘	− −	− −	− −	− −	− −
s_{2_1}	− − 0	− − −	∘ 1 ∘	− −	− −	− −	− −	− −
s_{2_2}	− − 0	− − −	∘ ∘ 1	− −	− −	− −	− −	− −
s_{3_0}	− 0 0	− − −	− − −	1 ∘	− −	− −	− −	1 0
s_{3_1}	− − 0	0 0 1	− − −	∘ 1	1 0	0 1	− −	0 1
s_{4_0}	− − 0	− 0 −	− − −	− −	1 ∘	− −	− −	− −
s_{4_1}	− 0 0	− − −	− − −	1 0	∘ 1	− −	− −	1 0
s_{5_0}	− 0 0	− − 0	− − −	1 0	− −	1 ∘	− −	1 0
s_{5_1}	− − 0	− − −	− − −	− −	− −	∘ 1	− −	− −
s_{6_0}	− − 0	− − −	− − −	− −	− −	− −	1 ∘	− −
s_{6_1}	− − 0	− − −	− − −	− −	− −	− −	∘ 1	− −
s_{7_0}	− 0 0	− − −	− − −	1 0	− −	− −	− −	1 ∘
s_{7_1}	− − 0	0 0 1	− − −	0 1	1 0	0 1	− −	∘ 1

(b) $\mathsf{Dst}(s_{0_2}, s_{0_0})$ made *impossible*

Figure 29: Distractor reduction stage 1

An intra-cell superset row s_{i_j} of state row s_{i_f} is called a *distractor* state row s_{i_j} (Dst) for state row s_{i_f}: *distractor*

$$s_{i_j} \supseteq s_{i_f} \Leftrightarrow \mathsf{Dst}(s_{i_j}, s_{i_f})$$

Figure 29a shows several *distractor* state rows:

$\mathsf{Dst}(s_{0_0}, s_{0_1}), \mathsf{Dst}(s_{0_1}, s_{0_0}), \mathsf{Dst}(s_{0_2}, s_{0_0}), \mathsf{Dst}(s_{0_2}, s_{0_1}),$
$\mathsf{Dst}(s_{1_0}, s_{1_1}), \mathsf{Dst}(s_{1_1}, s_{1_0}),$
$\mathsf{Dst}(s_{2_0}, s_{2_1}), \mathsf{Dst}(s_{2_0}, s_{2_2}), \mathsf{Dst}(s_{2_1}, s_{2_0}), \mathsf{Dst}(s_{2_1}, s_{2_2}), \mathsf{Dst}(s_{2_2}, s_{2_0}), \mathsf{Dst}(s_{2_2}, s_{2_1})$

A *distractor* state row s_{i_j} can be removed from a *consolidated* satoku matrix $\mathbb{S}$ (see $\mathsf{Dst}(s_{0_2}, s_{0_0})$ in figure 29b).

Proof. The argument is the same as for advance decisions. If a state row s_{g_h} is *combinable* with state row s_{i_j} it is also *combinable* with state row s_{i_f}, unless both cell row $r_{i_{j_g}}$ and cell row $r_{i_{f_g}}$ are *bound*. Therefore, state row s_{i_j} can be removed, as it does not offer different choices for merging than state row s_{i_f}. □

After removing the *distractor* state row $\mathsf{Dst}(s_{0_2}, s_{0_0})$ from figure 29b, the re-ordered satoku matrix $\mathbb{S}$ in figure 30a is already *strictly provable*, since it has only *unrestricted* and *bound* cell rows in core sub-matrix $\mathbb{C}$. With two more *distractor* state row removals the satoku matrix $\mathbb{S}$ is reduced to a 2-state cell matrix in figure 30b.

P	− − −	− − −	− −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘	− − −	1 0	− −	− −	− −	1 0	1 0
s_{0_1}	∘ 1 ∘	− − −	1 0	0 1	− −	− −	1 0	1 0
s_{0_2}	∘ ∘ 1	− − −	− −	− −	0 1	− −	− −	− −
s_{1_0}	− − −	1 ∘ ∘	− −	− −	− −	− −	− −	− −
s_{1_1}	− − −	∘ 1 ∘	− −	− −	− −	− −	− −	− −
s_{1_2}	− − −	∘ ∘ 1	− −	− −	− −	− −	− −	− −
s_{2_0}	− − −	− − −	1 ∘	− −	− −	− −	1 0	1 0
s_{2_1}	0 0 1	− − −	∘ 1	1 0	0 1	− −	0 1	− −
s_{3_0}	− 0 −	− − −	− −	1 ∘	− −	− −	− −	− −
s_{3_1}	− − −	− − −	1 0	∘ 1	− −	− −	1 0	1 0
s_{4_0}	− − 0	− − −	1 0	− −	1 ∘	− −	1 0	1 0
s_{4_1}	− − −	− − −	− −	− −	∘ 1	− −	− −	− −
s_{5_0}	− − −	− − −	− −	− −	− −	1 ∘	− −	− −
s_{5_1}	− − −	− − −	− −	− −	− −	∘ 1	− −	− −
s_{6_0}	− − −	− − −	1 0	− −	− −	− −	1 ∘	1 0
s_{6_1}	0 0 1	− − −	0 1	1 0	0 1	− −	∘ 1	− −
s_{7_0}	− − −	− − −	− −	− −	− −	− −	− −	1 ∘
s_{7_1}	0 0 1	− − −	0 1	1 0	0 1	− −	0 1	∘ 1

(a) *impossible* state row removed, re-ordered

P	− −	− −	− −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘	− −	1 0	0 1	− −	− −	1 0	1 0
s_{0_1}	∘ 1	− −	− −	− −	0 1	− −	− −	− −
s_{1_0}	− −	1 ∘	− −	− −	− −	− −	− −	− −
s_{1_1}	− −	∘ 1	− −	− −	− −	− −	− −	− −
s_{2_0}	− −	− −	1 ∘	− −	− −	− −	1 0	1 0
s_{2_1}	0 1	− −	∘ 1	1 0	0 1	− −	0 1	− −
s_{3_0}	0 −	− −	− −	1 ∘	− −	− −	− −	− −
s_{3_1}	− −	− −	1 0	∘ 1	− −	− −	1 0	1 0
s_{4_0}	− 0	− −	1 0	− −	1 ∘	− −	1 0	1 0
s_{4_1}	− −	− −	− −	− −	∘ 1	− −	− −	− −
s_{5_0}	− −	− −	− −	− −	− −	1 ∘	− −	− −
s_{5_1}	− −	− −	− −	− −	− −	∘ 1	− −	− −
s_{6_0}	− −	− −	1 0	− −	− −	− −	1 ∘	1 0
s_{6_1}	0 1	− −	0 1	1 0	0 1	− −	∘ 1	− −
s_{7_0}	− −	− −	− −	− −	− −	− −	− −	1 ∘
s_{7_1}	0 1	− −	0 1	1 0	0 1	− −	0 1	∘ 1

(b) reduced to 2-state cells, by removing $\mathsf{Dst}(s_{0_0}, s_{0_1})$ and $\mathsf{Dst}(s_{1_0}, s_{1_1})$

Figure 30: Distractor reduction stage 2

Distractors appear quite often in propostional formulas which have been transformed to conform to k-SAT by adding additional variables[8.]:

$$
\begin{array}{llllll}
(\neg a \vee \neg b & \vee \neg x00_0) & \wedge & (\neg c \vee & x00_0 \vee \neg x00_1) & \wedge & (\neg d \vee \neg e & \vee & x00_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x01_0) & \wedge & (\neg c \vee & x01_0 \vee \neg x01_1) & \wedge & (\neg d \vee e & \vee & x01_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x02_0) & \wedge & (\neg c \vee & x02_0 \vee \neg x02_1) & \wedge & (d \vee \neg e & \vee & x02_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x03_0) & \wedge & (\neg c \vee & x03_0 \vee \neg x03_1) & \wedge & (d \vee e & \vee & x03_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x04_0) & \wedge & (c \vee & x04_0 \vee \neg x04_1) & \wedge & (\neg d \vee \neg e & \vee & x04_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x05_0) & \wedge & (c \vee & x05_0 \vee \neg x05_1) & \wedge & (\neg d \vee e & \vee & x05_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x06_0) & \wedge & (c \vee & x06_0 \vee \neg x06_1) & \wedge & (d \vee \neg e & \vee & x06_1) & \wedge \\
(\neg a \vee \neg b & \vee \neg x07_0) & \wedge & (c \vee & x07_0 \vee \neg x07_1) & \wedge & (d \vee e & \vee & x07_1) &
\end{array}
$$

The corresponding satoku matrix $\mathbb{S}$ in figure 31 does not present trivially simple.

P	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –
s_{0_0}	1 ∘ ∘	1 0 0	1 0 0	1 0 0	0 – –	0 – –	– – –	0 – –	– – –	0 – –	– – –
s_{0_1}	∘ 1 ∘	0 – –	0 – –	0 – –	1 0 0	1 0 0	– – –	1 0 0	– – 0	1 0 0	– – –
s_{0_2}	∘ ∘ 1	0 – –	0 – –	0 – –	1 0 0	1 0 0	– 0 –	1 0 0	– – –	1 0 0	– – –
s_{1_0}	1 0 0	1 ∘ ∘	1 0 0	1 0 0	0 – –	0 – –	– – –	0 – –	– – –	0 – –	– – –
s_{1_1}	0 – –	∘ 1 ∘	0 – –	0 – –	1 0 0	1 0 0	– – –	1 0 0	– – 0	1 0 0	– – –
s_{1_2}	0 – –	∘ ∘ 1	0 – –	0 – –	1 0 0	1 0 0	– – 0	1 0 0	– – –	1 0 0	– – –
s_{2_0}	1 0 0	1 0 0	1 ∘ ∘	1 0 0	0 – –	0 – –	– – –	0 – –	– – –	0 – –	– – –
s_{2_1}	0 – –	0 – –	∘ 1 ∘	0 – –	1 0 0	1 0 0	– – –	1 0 0	– – 0	1 0 0	– – –
s_{2_2}	0 – –	0 – –	∘ ∘ 1	0 – –	1 0 0	1 0 0	– – –	1 0 0	– – –	1 0 0	– 0 –
s_{3_0}	1 0 0	1 0 0	1 0 0	1 ∘ ∘	0 – –	0 – –	– – –	0 – –	– – –	0 – –	– – –
s_{3_1}	0 – –	0 – –	0 – –	∘ 1 ∘	1 0 0	1 0 0	– – –	1 0 0	– – 0	1 0 0	– – –
s_{3_2}	0 – –	0 – –	0 – –	∘ ∘ 1	1 0 0	1 0 0	– – –	1 0 0	– – –	1 0 0	– – 0
s_{4_0}	0 – –	0 – –	0 – –	0 – –	1 ∘ ∘	1 0 0	– – –	1 0 0	– – –	1 0 0	– – –
s_{4_1}	1 0 0	1 0 0	1 0 0	1 0 0	∘ 1 ∘	0 – –	– – –	0 – –	– – 0	0 – –	– – –
s_{4_2}	1 0 0	1 0 0	1 0 0	1 0 0	∘ ∘ 1	0 – –	– 0 –	0 – –	– – –	0 – –	– – –
s_{5_0}	0 – –	0 – –	0 – –	0 – –	1 0 0	1 ∘ ∘	– – –	1 0 0	– – –	1 0 0	– – –
s_{5_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	∘ 1 ∘	– – –	0 – –	– – 0	0 – –	– – –
s_{5_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	∘ ∘ 1	– – 0	0 – –	– – –	0 – –	– – –
s_{6_0}	– – –	– – –	– – –	– – –	– – –	– – –	1 ∘ ∘	– – –	– – –	– – –	0 – –
s_{6_1}	– – 0	– – –	– – –	– – –	– – 0	– – –	∘ 1 ∘	– – –	– – –	– – –	1 0 0
s_{6_2}	– – –	– – 0	– – –	– – –	– – –	– – 0	∘ ∘ 1	– – –	– – –	– – –	1 0 0
s_{7_0}	0 – –	0 – –	0 – –	0 – –	1 0 0	1 0 0	– – –	1 ∘ ∘	– – –	1 0 0	– – –
s_{7_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	0 – –	– – –	∘ 1 ∘	– – 0	0 – –	– – –
s_{7_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	0 – –	– – –	∘ ∘ 1	– – –	0 – –	– 0 –
s_{8_0}	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	1 ∘ ∘	– – –	– – –
s_{8_1}	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	∘ 1 ∘	– – –	– – –
s_{8_2}	– 0 –	– 0 –	– 0 –	– 0 –	– 0 –	– 0 –	– – –	– 0 –	∘ ∘ 1	– 0 –	– – –
s_{9_0}	0 – –	0 – –	0 – –	0 – –	1 0 0	1 0 0	– – –	1 0 0	– – –	1 ∘ ∘	– – –
s_{9_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	0 – –	– – –	0 – –	– – 0	∘ 1 ∘	– – –
s_{9_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 – –	0 – –	– – –	0 – –	– – –	∘ ∘ 1	– – 0
s_{10_0}	– – –	– – –	– – –	– – –	– – –	– – –	0 – –	– – –	– – –	– – –	1 ∘ ∘
s_{10_1}	– – –	– – –	– – 0	– – –	– – –	– – –	1 0 0	– – 0	– – –	– – –	∘ 1 ∘
s_{10_2}	– – –	– – –	– – –	– – 0	– – –	– – –	1 0 0	– – –	– – –	– – 0	∘ ∘ 1

Figure 31: *distractor* s_{8_2} for s_{8_0} or s_{8_1}

8. More often than not, turning perfectly polynomial-time problems into exponential ones.

However, after removal of *distractor* s_{8_2} for state row s_{8_0}, and re-ordering satoku matrix $\mathbb{S}$ to separate core sub-matrix $\mathbb{C}$ from 2-state sub-matrix $\mathbb{S}_2$, 8 more *distractors* are revealed in figure 32.

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− −
s_{0_0}	1 ∘ ∘	1 0 0	1 0 0	1 0 0	0 − −	0 − −	− − −	0 − −	0 − −	− − −	− −
s_{0_1}	∘ 1 ∘	0 − −	0 − −	0 − −	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− − −	− −
s_{0_2}	∘ ∘ 1	0 − −	0 − −	0 − −	1 0 0	1 0 0	− 0 −	1 0 0	1 0 0	− − −	− −
s_{1_0}	1 0 0	1 ∘ ∘	1 0 0	1 0 0	0 − −	0 − −	− − −	0 − −	0 − −	− − −	− −
s_{1_1}	0 − −	∘ 1 ∘	0 − −	0 − −	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− − −	− −
s_{1_2}	0 − −	∘ ∘ 1	0 − −	0 − −	1 0 0	1 0 0	− − 0	1 0 0	1 0 0	− − −	− −
s_{2_0}	1 0 0	1 0 0	1 ∘ ∘	1 0 0	0 − −	0 − −	− − −	0 − −	0 − −	− − −	− −
s_{2_1}	0 − −	0 − −	∘ 1 ∘	0 − −	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− − −	− −
s_{2_2}	0 − −	0 − −	∘ ∘ 1	0 − −	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− 0 −	− −
s_{3_0}	1 0 0	1 0 0	1 0 0	1 ∘ ∘	0 − −	0 − −	− − −	0 − −	0 − −	− − −	− −
s_{3_1}	0 − −	0 − −	0 − −	∘ 1 ∘	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− − −	− −
s_{3_2}	0 − −	0 − −	0 − −	∘ ∘ 1	1 0 0	1 0 0	− − −	1 0 0	1 0 0	− − 0	− −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	1 0 0	− − −	1 0 0	1 0 0	− − −	− −
s_{4_1}	1 0 0	1 0 0	1 0 0	1 0 0	∘ 1 ∘	0 − −	− − −	0 − −	0 − −	− − −	− −
s_{4_2}	1 0 0	1 0 0	1 0 0	1 0 0	∘ ∘ 1	0 − −	− 0 −	0 − −	0 − −	− − −	− −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	1 0 0	1 ∘ ∘	− − −	1 0 0	1 0 0	− − −	− −
s_{5_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	∘ 1 ∘	− − −	0 − −	0 − −	− − −	− −
s_{5_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	∘ ∘ 1	− − 0	0 − −	0 − −	− − −	− −
s_{6_0}	− − −	− − −	− − −	− − −	− − −	− − −	1 ∘ ∘	− − −	− − −	0 − −	− −
s_{6_1}	− − 0	− − −	− − −	− − −	− − 0	− − −	∘ 1 ∘	− − −	− − −	1 0 0	− −
s_{6_2}	− − −	− − 0	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	1 0 0	− −
s_{7_0}	0 − −	0 − −	0 − −	0 − −	1 0 0	1 0 0	− − −	1 ∘ ∘	1 0 0	− − −	− −
s_{7_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	0 − −	− − −	∘ 1 ∘	0 − −	− − −	− −
s_{7_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	0 − −	− − −	∘ ∘ 1	0 − −	− 0 −	− −
s_{8_0}	0 − −	0 − −	0 − −	0 − −	1 0 0	1 0 0	− − −	1 0 0	1 ∘ ∘	− − −	− −
s_{8_1}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	0 − −	− − −	0 − −	∘ 1 ∘	− − −	− −
s_{8_2}	1 0 0	1 0 0	1 0 0	1 0 0	0 − −	0 − −	− − −	0 − −	∘ ∘ 1	− − 0	− −
s_{9_0}	− − −	− − −	− − −	− − −	− − −	− − −	0 − −	− − −	− − −	1 ∘ ∘	− −
s_{9_1}	− − −	− − −	− − 0	− − −	− − −	− − −	1 0 0	− − 0	− − −	∘ 1 ∘	− −
s_{9_2}	− − −	− − −	− − −	− − 0	− − −	− − −	1 0 0	− − −	− − 0	∘ ∘ 1	− −
s_{10_0}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	1 ∘
s_{10_1}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1

.

Figure 32: removal of $\mathsf{Dst}(s_{8_2}, s_{8_0})$ reveals more *distractors*

Removing *distractors* $\mathsf{Dst}(s_{0_2}, s_{0_0})$, $\mathsf{Dst}(s_{1_2}, s_{1_0})$, $\mathsf{Dst}(s_{2_2}, s_{2_0})$, $\mathsf{Dst}(s_{3_2}, s_{3_0})$, $\mathsf{Dst}(s_{4_2}, s_{4_0})$, $\mathsf{Dst}(s_{5_2}, s_{5_0})$, $\mathsf{Dst}(s_{7_2}, s_{7_0})$, $\mathsf{Dst}(s_{8_2}, s_{8_0})$ in figure 32 reveals still 2 more *distractors* in figure 33.

P	−−	−−	−−	−−	−−	−−	−−−	−−	−−	−−−	−−
s_{0_0}	1 $\circ$	1 0	1 0	1 0	0 1	0 1	−−−	0 1	0 1	−−−	−−
s_{0_1}	$\circ$ 1	0 1	0 1	0 1	1 0	1 0	−−−	1 0	1 0	−−−	−−
s_{1_0}	1 0	1 $\circ$	1 0	1 0	0 1	0 1	−−−	0 1	0 1	−−−	−−
s_{1_1}	0 1	$\circ$ 1	0 1	0 1	1 0	1 0	−−−	1 0	1 0	−−−	−−
s_{2_0}	1 0	1 0	1 $\circ$	1 0	0 1	0 1	−−−	0 1	0 1	−−−	−−
s_{2_1}	0 1	0 1	$\circ$ 1	0 1	1 0	1 0	−−−	1 0	1 0	−−−	−−
s_{3_0}	1 0	1 0	1 0	1 $\circ$	0 1	0 1	−−−	0 1	0 1	−−−	−−
s_{3_1}	0 1	0 1	0 1	$\circ$ 1	1 0	1 0	−−−	1 0	1 0	−−−	−−
s_{4_0}	0 1	0 1	0 1	0 1	1 $\circ$	1 0	−−−	1 0	1 0	−−−	−−
s_{4_1}	1 0	1 0	1 0	1 0	$\circ$ 1	0 1	−−−	0 1	0 1	−−−	−−
s_{5_0}	0 1	0 1	0 1	0 1	1 0	1 $\circ$	−−−	1 0	1 0	−−−	−−
s_{5_1}	1 0	1 0	1 0	1 0	0 1	$\circ$ 1	−−−	0 1	0 1	−−−	−−
s_{6_0}	−−	−−	−−	−−	−−	−−	1 $\circ$ $\circ$	−−	−−	0 −−	−−
s_{6_1}	−−	−−	−−	−−	−−	−−	$\circ$ 1 $\circ$	−−	−−	1 0 0	−−
s_{6_2}	−−	−−	−−	−−	−−	−−	$\circ$ $\circ$ 1	−−	−−	1 0 0	−−
s_{7_0}	0 1	0 1	0 1	0 1	1 0	1 0	−−−	1 $\circ$	1 0	−−−	−−
s_{7_1}	1 0	1 0	1 0	1 0	0 1	0 1	−−−	$\circ$ 1	0 1	−−−	−−
s_{8_0}	0 1	0 1	0 1	0 1	1 0	1 0	−−−	1 0	1 $\circ$	−−−	−−
s_{8_1}	1 0	1 0	1 0	1 0	0 1	0 1	−−−	0 1	$\circ$ 1	−−−	−−
s_{9_0}	−−	−−	−−	−−	−−	−−	0 −−	−−	−−	1 $\circ$ $\circ$	−−
s_{9_1}	−−	−−	−−	−−	−−	−−	1 0 0	−−	−−	$\circ$ 1 $\circ$	−−
s_{9_2}	−−	−−	−−	−−	−−	−−	1 0 0	−−	−−	$\circ$ $\circ$ 1	−−
s_{10_0}	−−	−−	−−	−−	−−	−−	−−−	−−	−−	−−−	1 $\circ$
s_{10_1}	−−	−−	−−	−−	−−	−−	−−−	−−	−−	−−−	$\circ$ 1

.

Figure 33: still more *distractors* after *distractor* removal

Dropping redundancies and re-ordering the satoku matrix in figure 33 results in the satoku matrix shown in figure 34. State rows s_{0_1}, s_{0_2}, s_{1_1}, s_{1_2}, containing only *decided* cell rows in core sub-matrix $\mathbb{C}$, show that satoku matrix $\mathbb{S}$ is *strictly provable*.

P	−−−	−−−	−−	−−
s_{0_0}	1 $\circ$ $\circ$	0 −−	−−	−−
s_{0_1}	$\circ$ 1 $\circ$	1 0 0	−−	−−
s_{0_2}	$\circ$ $\circ$ 1	1 0 0	−−	−−
s_{1_0}	0 −−	1 $\circ$ $\circ$	−−	−−
s_{1_1}	1 0 0	$\circ$ 1 $\circ$	−−	−−
s_{1_2}	1 0 0	$\circ$ $\circ$ 1	−−	−−
s_{2_0}	−−−	−−−	1 $\circ$	−−
s_{2_1}	−−−	−−−	$\circ$ 1	−−
s_{3_0}	−−−	−−−	−−	1 $\circ$
s_{3_1}	−−−	−−−	−−	$\circ$ 1

.

Figure 34: satoku matrix $\mathbb{S}$ *strictly provable*

Although not necessary, removing *distractors* $\mathsf{Dst}(s_{0_2}, s_{0_1})$, $\mathsf{Dst}(s_{1_2}, s_{0_1})$, in figure 34 reduces satoku matrix $\mathbb{S}$ to a 2-state cell matrix in figure 35, also showing that satoku matrix $\mathbb{S}$ is *strictly provable*.

P	−−	−−	−−	−−
s_{0_0}	1 $\circ$	0 1	−−	−−
s_{0_1}	$\circ$ 1	1 0	−−	−−
s_{1_0}	0 1	1 $\circ$	−−	−−
s_{1_1}	1 0	$\circ$ 1	−−	−−
s_{2_0}	−−	−−	1 $\circ$	−−
s_{2_1}	−−	−−	$\circ$ 1	−−
s_{3_0}	−−	−−	−−	1 $\circ$
s_{3_1}	−−	−−	−−	$\circ$ 1

.

Figure 35: reduced to 2-state cells

10.2.1 Special Properties of 2-State Distractors

When a cell c_{i_i} has 2 atomic states $s_{i_{j_{i_j}}}, s_{i_{f_{i_f}}}, j \neq f, |c_{i_i}| = 2$, and state row s_{i_j} has an *impossible* CFR $s_{i_{j_{g_h}}}, g \neq i$, in an *undecided* cell row $r_{i_{j_g}}$ then state row s_{i_j} can only be a distractor, if cell row $r_{i_{f_g}}$ is *unrestricted* or *bound*.

Proof. If both state rows s_{i_j}, s_{i_f} are *mutually exclusive* with the same state $s_{g_{h_{g_h}}}$, then CFR $s_{i_{j_{g_h}}}$ is *impossible*, which implies that CFR $s_{g_{h_{i_j}}}$ is also *impossible*. Further CFR $s_{i_{f_{g_h}}}$ is *impossible*, which implies that CFR $s_{g_{h_{i_f}}}$ is also *impossible*. Since cell row $r_{g_{h_i}}$ has only 2 CFR states $s_{g_{h_{i_j}}}, s_{g_{h_{i_f}}}$, which are both *impossible*, cell row $r_{g_{h_i}}$ is a *conflict*, which means that the entire state row s_{g_h} is *impossible* and is therefore removed. However, this also removes the *mutually exclusive* CFR states $s_{i_{j_{g_h}}}, s_{i_{f_{g_h}}}$. □

If state row s_{i_f} for a 2-state cell c_{i_i} does not have any *impossible* CFR states $s_{i_{f_{g_h}}}, g \neq i$, at all, making state row $s_{i_j}, j \neq f$, *impossible*, is the equivalent of unit propagation in DPLL.

10.3 Status Row Variables

To express that status row s_{i_j} must either be selected (become the *required* state row of cell-matrix row c_i) or not, create a 2-state cell c_{e_e}, make CFR $s_{e_{0_{i_j}}}$ *required* (by setting CFR states $s_{e_{0_{i_g}}}, g \neq j$, *impossible*) and make CFR $s_{e_{1_{i_j}}}$ *impossible*.

P	− − −	− − −	− − −	− −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	− 0
s_{0_1}	∘ 1 ∘	− − −	− − −	0 −
s_{0_2}	∘ ∘ 1	− − −	− − −	0 −
s_{1_0}	− − −	1 ∘ ∘	− − −	− −
s_{1_1}	0 − −	∘ 1 ∘	− − −	− −
s_{1_2}	− − −	∘ ∘ 1	− − −	− −
s_{2_0}	− − −	− − −	1 ∘ ∘	− −
s_{2_1}	0 − −	− − −	∘ 1 ∘	− −
s_{2_2}	− − −	− − −	∘ ∘ 1	− −
s_{3_0}	− 0 0	− − −	− − −	1 ∘
s_{3_1}	0 − −	− − −	− − −	∘ 1

(a) ex-status-row-variables/ex-status-row-variables-000

P	− − −	− − −	− − −	− −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	1 0
s_{0_1}	∘ 1 ∘	− − −	− − −	0 1
s_{0_2}	∘ ∘ 1	− − −	− − −	0 1
s_{1_0}	− − −	1 ∘ ∘	− − −	− −
s_{1_1}	0 − −	∘ 1 ∘	− − −	0 1
s_{1_2}	− − −	∘ ∘ 1	− − −	− −
s_{2_0}	− − −	− − −	1 ∘ ∘	− −
s_{2_1}	0 − −	− − −	∘ 1 ∘	0 1
s_{2_2}	− − −	− − −	∘ ∘ 1	− −
s_{3_0}	1 0 0	− 0 −	− 0 −	1 ∘
s_{3_1}	0 − −	− − −	− − −	∘ 1

(b) ex-status-row-variables/ex-status-row-variables-001

10.4 OR-NONE Cell Construction

OR-NONE cell
OR-NONE status row

To expresses that one or more of several status rows s_{x_y} or none of them must be selected, create status row variables $c_{e_e} \in \mathbb{V}$ for all status rows s_{x_y}.

Create a cell c_{i_i} with $|\mathbb{V}| + 1$ states.

P	− − −	− − −	− − −	− −	− −	− −	− − − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	1 0	0 1	0 1	− − − −
s_{0_1}	∘ 1 ∘	− − −	− − −	0 1	− −	− −	− − − −
s_{0_2}	∘ ∘ 1	− − −	− − −	0 1	− −	− −	− − − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− −	0 1	− −	− − − −
s_{1_1}	0 − −	∘ 1 ∘	− − −	0 1	1 0	− −	− − − −
s_{1_2}	− − −	∘ ∘ 1	− − −	− −	0 1	− −	− − − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− −	− −	0 1	− − − −
s_{2_1}	0 − −	− − −	∘ 1 ∘	0 1	− −	1 0	− − − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− −	− −	0 1	− − − −
s_{3_0}	1 0 0	− 0 −	− 0 −	1 ∘	0 1	0 1	− 0 0 0
s_{3_1}	0 − −	− − −	− − −	∘ 1	− −	− −	0 − − −
s_{4_0}	0 − −	0 1 0	− − −	0 1	1 ∘	− −	− − 0 0
s_{4_1}	− − −	− 0 −	− − −	− −	∘ 1	− −	− 0 − −
s_{5_0}	0 − −	− − −	0 1 0	0 1	− −	1 ∘	− − − 0
s_{5_1}	− − −	− − −	− 0 −	− −	− −	∘ 1	− − 0 −
s_{6_0}	− − −	− − −	− − −	− 0	− −	− −	1 ∘ ∘ ∘
s_{6_1}	− − −	− − −	− − −	0 −	− 0	− −	∘ 1 ∘ ∘
s_{6_2}	− − −	− − −	− − −	0 −	0 −	− 0	∘ ∘ 1 ∘
s_{6_3}	− − −	− − −	− − −	0 −	0 −	0 −	∘ ∘ ∘ 1

(a) ex-status-row-variables/ex-status-row-variables-002

P	− − −	− − −	− − −	− −	− −	− −	− − − −	
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	1 0	0 1	0 1	1 0 0 0	
s_{0_1}	∘ 1 ∘	− − −	− − −	0 1	− −	− −	0 − − −	
s_{0_2}	∘ ∘ 1	− − −	− − −	0 1	− −	− −	0 − − −	
s_{1_0}	− − −	1 ∘ ∘	− − −	− −	0 1	− −	− 0 − −	
s_{1_1}	0 − −	∘ 1 ∘	− − −	0 1	1 0	− −	0 1 0 0	
s_{1_2}	− − −	∘ ∘ 1	− − −	− −	0 1	− −	− 0 − −	
s_{2_0}	− − −	− − −	1 ∘ ∘	− −	− −	0 1	− − 0 −	
s_{2_1}	0 − −	− − −	∘ 1 ∘	0 1	− −	1 0	0 − − 0	
s_{2_2}	− − −	− − −	∘ ∘ 1	− −	− −	0 1	− − 0 −	
s_{3_0}	1 0 0	− 0 −	− 0 −	1 ∘	0 1	0 1	1 0 0 0	
s_{3_1}	0 − −	− − −	− − −	∘ 1	− −	− −	0 − − −	
s_{4_0}	0 − −	0 1 0	− − −	0 1	1 ∘	− −	0 1 0 0	
s_{4_1}	− − −	− 0 −	− − −	− −	∘ 1	− −	− 0 − −	
s_{5_0}	0 − −	− − −	0 1 0	0 1	− −	1 ∘	0 − − 0	
s_{5_1}	− − −	− − −	− 0 −	− −	− −	∘ 1	− − 0 −	
s_{6_0}	1 0 0	− 0 −	− 0 −	1 0	0 1	0 1	1 ∘ ∘ ∘	
s_{6_1}	0 − −	0 1 0	− − −	0 1	1 0	− −	∘ 1 ∘ ∘	
s_{6_2}	0 − −	− 0 −	0 1 0	0 1	0 1	1 0	∘ ∘ 1 ∘	$\mathsf{Dst}(s_{6_2}, s_{6_1})$
s_{6_3}	0 − −	− 0 −	− 0 −	0 1	0 1	0 1	∘ ∘ ∘ 1	$\mathsf{Dst}(s_{6_3}, s_{6_0})$

(b) ex-status-row-variables/ex-status-row-variables-003

For each status row variable c_{e_e}, allocate a status row s_{i_j}.

In cell row $r_{i_{j_e}}$ make first alternative of status row variable $s_{i_{j_{e_0}}}$ *required* (by setting $s_{i_{j_{e_1}}}$ *impossible*).

In all following cell rows $r_{i_{g_e}}, g > j$, make first alternative of status row variable $s_{i_{g_{e_0}}}$ *impossible*.

Cell c_{i_i} is called OR-NONE cell, and the last, unallocated status row s_{i_h} is called OR-NONE status row.

When a status row s_{x_y} is chosen from the core sub-matrix $\mathbb{C}$ and all *mutually exclusive* status rows $s_{p_q}, p \neq x$ in core sub-matrix $\mathbb{C}$ are determined, the OR-NONE status row s_{i_h} of their OR-NONE cell c_{i_i} will be a *distractor* for the chosen status row s_{x_y}. OR cell

After *distractor* removal, cell c_{i_i} becomes an OR cell, expressing the fact that at least one of the status rows s_{x_y}, s_{p_q}, must be selected.

P	− − −	− − −	− − −	− −	− −	− −	− − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	1 0	0 1	0 1	1 0 0
s_{0_1}	∘ 1 ∘	− − −	− − −	0 1	− −	− −	0 − −
s_{0_2}	∘ ∘ 1	− − −	− − −	0 1	− −	− −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− −	0 1	− −	− 0 −
s_{1_1}	0 − −	∘ 1 ∘	− − −	0 1	1 0	− −	0 1 0
s_{1_2}	− − −	∘ ∘ 1	− − −	− −	0 1	− −	− 0 −
s_{2_0}	− − −	− − −	1 ∘ ∘	− −	− −	0 1	− − 0
s_{2_1}	0 − −	− − −	∘ 1 ∘	0 1	− −	1 0	0 − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− −	− −	0 1	− − 0
s_{3_0}	1 0 0	− 0 −	− 0 −	1 ∘	0 1	0 1	1 0 0
s_{3_1}	0 − −	− − −	− − −	∘ 1	− −	− −	0 − −
s_{4_0}	0 − −	0 1 0	− − −	0 1	1 ∘	− −	0 1 0
s_{4_1}	− − −	− 0 −	− − −	− −	∘ 1	− −	− 0 −
s_{5_0}	0 − −	− − −	0 1 0	0 1	− −	1 ∘	0 − −
s_{5_1}	− − −	− − −	− 0 −	− −	− −	∘ 1	− − 0
s_{6_0}	1 0 0	− 0 −	− 0 −	1 0	0 1	0 1	1 ∘ ∘
s_{6_1}	0 − −	0 1 0	− − −	0 1	1 0	− −	∘ 1 ∘
s_{6_2}	0 − −	− 0 −	0 1 0	0 1	0 1	1 0	∘ ∘ 1

(a) ex-status-row-variables/ex-status-row-variables-004

P	− − −	− − −	− − −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	1 0	0 1	0 1	1 0
s_{0_1}	∘ 1 ∘	0 1 0	− − −	0 1	1 0	− −	0 1
s_{0_2}	∘ ∘ 1	0 1 0	− − −	0 1	1 0	− −	0 1
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	1 0	0 1	0 1	1 0
s_{1_1}	0 − −	∘ 1 ∘	− − −	0 1	1 0	− −	0 1
s_{1_2}	1 0 0	∘ ∘ 1	− 0 −	1 0	0 1	0 1	1 0
s_{2_0}	− − −	− − −	1 ∘ ∘	− −	− −	0 1	− −
s_{2_1}	0 − −	0 1 0	∘ 1 ∘	0 1	1 0	1 0	0 1
s_{2_2}	− − −	− − −	∘ ∘ 1	− −	− −	0 1	− −
s_{3_0}	1 0 0	− 0 −	− 0 −	1 ∘	0 1	0 1	1 0
s_{3_1}	0 − −	0 1 0	− − −	∘ 1	1 0	− −	0 1
s_{4_0}	0 − −	0 1 0	− − −	0 1	1 ∘	− −	0 1
s_{4_1}	1 0 0	− 0 −	− 0 −	1 0	∘ 1	0 1	1 0
s_{5_0}	0 − −	0 1 0	0 1 0	0 1	1 0	1 ∘	0 1
s_{5_1}	− − −	− − −	− 0 −	− −	− −	∘ 1	− −
s_{6_0}	1 0 0	− 0 −	− 0 −	1 0	0 1	0 1	1 ∘
s_{6_1}	0 − −	0 1 0	− − −	0 1	1 0	− −	∘ 1

(b) ex-status-row-variables/ex-status-row-variables-005-color

If there are more than two status rows left in OR cell c_{i_i} after distractor removal, we have found a distributed multivalue variable (pidgeon/hole problem).

If the OR cell c_{i_i} is already part of core sub-matrix $\mathbb{C}$, it is not essential and could be removed from the satoku matrix $\mathbb{S}$. However, non-essential cells may still be very useful, even to the point of making *provability* polynomial instead of exponential [HERTEL]. It is, however, useful, to separate such cells from the core sub-matrix $\mathbb{C}$.

P	−−−	−−−	−−−	−−	−−	−−	−−	−−	−−−−−
s_{0_0}	1 ∘ ∘	−0−	−0−	1 0	0 1	−−	−−	1 0	0−−0 0
s_{0_1}	∘ 1 ∘	0 1 0	−−−	0 1	−−	0 1	0 1	0 1	−0 0−−
s_{0_2}	∘ ∘ 1	0 1 0	−−−	0 1	−−	0 1	0 1	0 1	−0 0−−
s_{1_0}	1 0 0	1 ∘ ∘	−0−	1 0	0 1	0 1	1 0	1 0	0 0 1 0 0
s_{1_1}	0−−	∘ 1 ∘	−−−	0 1	−−	0 1	0 1	0 1	−0 0−−
s_{1_2}	1 0 0	∘ ∘ 1	−0−	1 0	0 1	1 0	0 1	1 0	0 1 0 0 0
s_{2_0}	−−−	−−−	1 ∘ ∘	−−	0 1	−−	−−	−−	0−−−−
s_{2_1}	0−−	0 1 0	∘ 1 ∘	0 1	1 0	0 1	0 1	0 1	1 0 0 0 0
s_{2_2}	−−−	−−−	∘ ∘ 1	−−	0 1	−−	−−	−−	0−−−−
s_{3_0}	1 0 0	−0−	−0−	1 ∘	0 1	−−	−−	1 0	0−−0 0
s_{3_1}	0−−	0 1 0	−−−	∘ 1	−−	0 1	0 1	0 1	−0 0−−
s_{4_0}	0−−	0 1 0	0 1 0	0 1	1 ∘	0 1	0 1	0 1	1 0 0 0 0
s_{4_1}	−−−	−−−	−0−	−−	∘ 1	−−	−−	−−	0−−−−
s_{5_0}	1 0 0	0 0 1	−0−	1 0	0 1	1 ∘	0 1	1 0	0 1 0 0 0
s_{5_1}	−−−	−−0	−−−	−−	−−	∘ 1	−−	−−	−0−−−
s_{6_0}	1 0 0	1 0 0	−0−	1 0	0 1	0 1	1 ∘	1 0	0 0 1 0 0
s_{6_1}	−−−	0−−	−−−	−−	−−	−−	∘ 1	−−	−−0−−
s_{7_0}	1 0 0	−0−	−0−	1 0	0 1	−−	−−	1 ∘	0−−0 0
s_{7_1}	0−−	0 1 0	−−−	0 1	−−	0 1	0 1	∘ 1	−0 0−−
s_{8_0}	0−−	0 1 0	0 1 0	0 1	1 0	0 1	0 1	0 1	1 ∘ ∘ ∘ ∘
s_{8_1}	1 0 0	0 0 1	−0−	1 0	0 1	1 0	0 1	1 0	∘ 1 ∘ ∘ ∘
s_{8_2}	1 0 0	1 0 0	−0−	1 0	0 1	0 1	1 0	1 0	∘ ∘ 1 ∘ ∘
s_{8_3}	0−−	0 1 0	−0−	0 1	0 1	0 1	0 1	0 1	∘ ∘ ∘ 1 ∘
s_{8_4}	0−−	0 1 0	−0−	0 1	0 1	0 1	0 1	0 1	∘ ∘ ∘ ∘ 1

(a) ex-status-row-variables/ex-status-row-variables-007

P	−−−	0−−	−−−	−−	−−	−−	0 1	−−	−−
s_{0_0}	1 ∘ ∘	0 0 1	−0−	1 0	0 1	1 0	0 1	1 0	0 1
s_{0_1}	∘ 1 ∘	0 1 0	0 1 0	0 1	1 0	0 1	0 1	0 1	1 0
s_{0_2}	∘ ∘ 1	0 1 0	0 1 0	0 1	1 0	0 1	0 1	0 1	1 0
s_{1_0}	0 0 0	∘ ∘ ∘	0 0 0	0 0	0 0	0 0	0 0	0 0	0 0
s_{1_1}	0−−	∘ 1 ∘	0 1 0	0 1	1 0	0 1	0 1	0 1	1 0
s_{1_2}	1 0 0	∘ ∘ 1	−0−	1 0	0 1	1 0	0 1	1 0	0 1
s_{2_0}	1 0 0	0 0 1	1 ∘ ∘	1 0	0 1	1 0	0 1	1 0	0 1
s_{2_1}	0−−	0 1 0	∘ 1 ∘	0 1	1 0	0 1	0 1	0 1	1 0
s_{2_2}	1 0 0	0 0 1	∘ ∘ 1	1 0	0 1	1 0	0 1	1 0	0 1
s_{3_0}	1 0 0	0 0 1	−0−	1 ∘	0 1	1 0	0 1	1 0	0 1
s_{3_1}	0−−	0 1 0	0 1 0	∘ 1	1 0	0 1	0 1	0 1	1 0
s_{4_0}	0−−	0 1 0	0 1 0	0 1	1 ∘	0 1	0 1	0 1	1 0
s_{4_1}	1 0 0	0 0 1	−0−	1 0	∘ 1	1 0	0 1	1 0	0 1
s_{5_0}	1 0 0	0 0 1	−0−	1 0	0 1	1 ∘	0 1	1 0	0 1
s_{5_1}	0−−	0 1 0	0 1 0	0 1	1 0	∘ 1	0 1	0 1	1 0
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	∘ ∘	0 0	0 0
s_{6_1}	−−−	0−−	−−−	−−	−−	−−	∘ 1	−−	−−
s_{7_0}	1 0 0	0 0 1	−0−	1 0	0 1	1 0	0 1	1 ∘	0 1
s_{7_1}	0−−	0 1 0	0 1 0	0 1	1 0	0 1	0 1	∘ 1	1 0
s_{8_0}	0−−	0 1 0	0 1 0	0 1	1 0	0 1	0 1	0 1	1 ∘
s_{8_1}	1 0 0	0 0 1	−0−	1 0	0 1	1 0	0 1	1 0	∘ 1

(b) ex-status-row-variables/ex-status-row-variables-008

11. Gaussian Elimination with 3-Variable XORs

$$
\begin{array}{llll}
(& (\neg a \wedge \neg b \wedge c) & \vee & \\
 & (\neg a \wedge b \wedge \neg c) & \vee & \\
 & (a \wedge \neg b \wedge \neg c) & \vee & \\
 & (a \wedge b \wedge c) & &) \wedge \\
(& (\neg a \wedge \neg d \wedge f) & \vee & \\
 & (\neg a \wedge d \wedge \neg f) & \vee & \\
 & (a \wedge \neg d \wedge \neg f) & \vee & \\
 & (a \wedge d \wedge f) & &) \wedge \\
(& (\neg b \wedge \neg e \wedge \neg f) & \vee & \\
 & (\neg b \wedge e \wedge f) & \vee & \\
 & (b \wedge e \wedge \neg f) & \vee & \\
 & (b \wedge \neg e \wedge f) & &)
\end{array}
$$

P	− − − −	− − − −	− − − −	− −	− −	− −	− −	− −	− −	
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	− − 0 0	0 1	0 1	1 0	− −	− −	− −	$a \veebar b \veebar c = 1$
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	0 0 − −	0 1	1 0	0 1	− −	− −	− −	
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	− − 0 0	1 0	0 1	0 1	− −	− −	− −	
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	0 0 − −	1 0	1 0	1 0	− −	− −	− −	
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	0 − 0 −	0 1	− −	− −	0 1	− −	1 0	$a \veebar d \veebar f = 1$
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	− 0 − 0	0 1	− −	− −	1 0	− −	0 1	
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	− 0 − 0	1 0	− −	− −	0 1	− −	0 1	
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	0 − 0 −	1 0	− −	− −	1 0	− −	1 0	
s_{2_0}	− 0 − 0	0 − − 0	1 ∘ ∘ ∘	− −	0 1	− −	− −	0 1	0 1	$a \veebar e \veebar f = 0$
s_{2_1}	− 0 − 0	− 0 0 −	∘ 1 ∘ ∘	− −	0 1	− −	− −	1 0	1 0	
s_{2_2}	0 − 0 −	0 − − 0	∘ ∘ 1 ∘	− −	1 0	− −	− −	1 0	0 1	
s_{2_3}	0 − 0 −	− 0 0 −	∘ ∘ ∘ 1	− −	1 0	− −	− −	0 1	1 0	
s_{3_0}	0 0 − −	0 0 − −	− − − −	1 ∘	− −	− −	− −	− −	− −	a
s_{3_1}	− − 0 0	− − 0 0	− − − −	∘ 1	− −	− −	− −	− −	− −	$\neg a$
s_{4_0}	0 − 0 −	− − − −	0 0 − −	− −	1 ∘	− −	− −	− −	− −	b
s_{4_1}	− 0 − 0	− − − −	− − 0 0	− −	∘ 1	− −	− −	− −	− −	$\neg b$
s_{5_0}	− 0 0 −	− − − −	− − − −	− −	− −	1 ∘	− −	− −	− −	c
s_{5_1}	0 − − 0	− − − −	− − − −	− −	− −	∘ 1	− −	− −	− −	$\neg c$
s_{6_0}	− − − −	0 − 0 −	− − − −	− −	− −	− −	1 ∘	− −	− −	d
s_{6_1}	− − − −	− 0 − 0	− − − −	− −	− −	− −	∘ 1	− −	− −	$\neg d$
s_{7_0}	− − − −	− − − −	0 − − 0	− −	− −	− −	− −	1 ∘	− −	e
s_{7_1}	− − − −	− − − −	− 0 0 −	− −	− −	− −	− −	∘ 1	− −	$\neg e$
s_{8_0}	− − − −	− 0 0 −	0 − 0 −	− −	− −	− −	− −	− −	1 ∘	f
s_{8_1}	− − − −	0 − − 0	− 0 − 0	− −	− −	− −	− −	− −	∘ 1	$\neg f$

.

Figure 40: 3 XOR Gauss example - mapped from CDF

P	− − − −	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	− − 0 0	0 1	0 1	1 0	− −	− −	− −	0 1	− −	$a \veebar b \veebar \neg c = 0$
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	0 0 − −	0 1	1 0	0 1	− −	− −	− −	1 0	− −	
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	− − 0 0	1 0	0 1	0 1	− −	− −	− −	1 0	− −	
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	0 0 − −	1 0	1 0	1 0	− −	− −	− −	0 1	− −	
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	0 − 0 −	0 1	− −	− −	0 1	− −	1 0	− −	1 0	$a \veebar \neg d \veebar f = 0$
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	− 0 − 0	0 1	− −	− −	1 0	− −	0 1	− −	0 1	
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	− 0 − 0	1 0	− −	− −	0 1	− −	0 1	− −	1 0	
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	0 − 0 −	1 0	− −	− −	1 0	− −	1 0	− −	0 1	
s_{2_0}	− 0 − 0	0 − − 0	1 ∘ ∘ ∘	− −	0 1	− −	− −	0 1	0 1	− −	− −	$a \veebar e \veebar f = 0$
s_{2_1}	− 0 − 0	− 0 0 −	∘ 1 ∘ ∘	− −	0 1	− −	− −	1 0	1 0	− −	− −	
s_{2_2}	0 − 0 −	0 − − 0	∘ ∘ 1 ∘	− −	1 0	− −	− −	1 0	0 1	− −	− −	
s_{2_3}	0 − 0 −	− 0 0 −	∘ ∘ ∘ 1	− −	1 0	− −	− −	0 1	1 0	− −	− −	
s_{3_0}	0 0 − −	0 0 − −	− − − −	1 ∘	− −	− −	− −	− −	− −	− −	− −	a
s_{3_1}	− − 0 0	− − 0 0	− − − −	∘ 1	− −	− −	− −	− −	− −	− −	− −	$\neg a$
s_{4_0}	0 − 0 −	− − − −	0 0 − −	− −	1 ∘	− −	− −	− −	− −	− −	− −	b
s_{4_1}	− 0 − 0	− − − −	− − 0 0	− −	∘ 1	− −	− −	− −	− −	− −	− −	$\neg b$
s_{5_0}	− 0 0 −	− − − −	− − − −	− −	− −	1 ∘	− −	− −	− −	0 1	− −	c
s_{5_1}	0 − − 0	− − − −	− − − −	− −	− −	∘ 1	− −	− −	− −	1 0	− −	$\neg c$
s_{6_0}	− − − −	0 − 0 −	− − − −	− −	− −	− −	1 ∘	− −	− −	− −	0 1	d
s_{6_1}	− − − −	− 0 − 0	− − − −	− −	− −	− −	∘ 1	− −	− −	− −	1 0	$\neg d$
s_{7_0}	− − − −	− − − −	0 − − 0	− −	− −	− −	− −	1 ∘	− −	− −	− −	e
s_{7_1}	− − − −	− − − −	− 0 0 −	− −	− −	− −	− −	∘ 1	− −	− −	− −	$\neg e$
s_{8_0}	− − − −	− 0 0 −	0 − 0 −	− −	− −	− −	− −	− −	1 ∘	− −	− −	f
s_{8_1}	− − − −	0 − − 0	− 0 − 0	− −	− −	− −	− −	− −	∘ 1	− −	− −	$\neg f$
s_{9_0}	0 − − 0	− − − −	− − − −	− −	− −	0 1	− −	− −	− −	1 ∘	− −	$\neg c$
s_{9_1}	− 0 0 −	− − − −	− − − −	− −	− −	1 0	− −	− −	− −	∘ 1	− −	c
s_{10_0}	− − − −	− 0 − 0	− − − −	− −	− −	− −	0 1	− −	− −	− −	1 ∘	$\neg d$
s_{10_1}	− − − −	0 − 0 −	− − − −	− −	− −	− −	1 0	− −	− −	− −	∘ 1	d

.

Figure 41: 3 XOR Gauss example - results = 0

P	− − − −	− − − −	− − − −	− −	− −	− −	− −	− −	− −	
s_{0_0}	1 ○ ○ ○	− − 0 0	− − 0 0	0 1	0 1	− −	− −	0 1	− −	$a \veebar b \veebar \neg c = 0$
s_{0_1}	○ 1 ○ ○	− − 0 0	0 0 − −	0 1	1 0	− −	− −	1 0	− −	
s_{0_2}	○ ○ 1 ○	0 0 − −	− − 0 0	1 0	0 1	− −	− −	1 0	− −	
s_{0_3}	○ ○ ○ 1	0 0 − −	0 0 − −	1 0	1 0	− −	− −	0 1	− −	
s_{1_0}	− − 0 0	1 ○ ○ ○	0 − 0 −	0 1	− −	− −	1 0	− −	1 0	$a \veebar \neg d \veebar f = 0$
s_{1_1}	− − 0 0	○ 1 ○ ○	− 0 − 0	0 1	− −	− −	0 1	− −	0 1	
s_{1_2}	0 0 − −	○ ○ 1 ○	− 0 − 0	1 0	− −	− −	0 1	− −	1 0	
s_{1_3}	0 0 − −	○ ○ ○ 1	0 − 0 −	1 0	− −	− −	1 0	− −	0 1	
s_{2_0}	− 0 − 0	0 − − 0	1 ○ ○ ○	− −	0 1	0 1	0 1	− −	− −	$a \veebar e \veebar f = 0$
s_{2_1}	− 0 − 0	− 0 0 −	○ 1 ○ ○	− −	0 1	1 0	1 0	− −	− −	
s_{2_2}	0 − 0 −	0 − − 0	○ ○ 1 ○	− −	1 0	1 0	0 1	− −	− −	
s_{2_3}	0 − 0 −	− 0 0 −	○ ○ ○ 1	− −	1 0	0 1	1 0	− −	− −	
s_{3_0}	0 0 − −	0 0 − −	− − − −	1 ○	− −	− −	− −	− −	− −	a
s_{3_1}	− − 0 0	− − 0 0	− − − −	○ 1	− −	− −	− −	− −	− −	$\neg a$
s_{4_0}	0 − 0 −	− − − −	0 0 − −	− −	1 ○	− −	− −	− −	− −	b
s_{4_1}	− 0 − 0	− − − −	− − 0 0	− −	○ 1	− −	− −	− −	− −	$\neg b$
s_{5_0}	− − − −	− − − −	0 − − 0	− −	− −	1 ○	− −	− −	− −	e
s_{5_1}	− − − −	− − − −	− 0 0 −	− −	− −	○ 1	− −	− −	− −	$\neg e$
s_{6_0}	− − − −	− 0 0 −	0 − 0 −	− −	− −	− −	1 ○	− −	− −	f
s_{6_1}	− − − −	0 − − 0	− 0 − 0	− −	− −	− −	○ 1	− −	− −	$\neg f$
s_{7_0}	0 − − 0	− − − −	− − − −	− −	− −	− −	− −	1 ○	− −	$\neg c$
s_{7_1}	− 0 0 −	− − − −	− − − −	− −	− −	− −	− −	○ 1	− −	c
s_{8_0}	− − − −	− 0 − 0	− − − −	− −	− −	− −	− −	− −	1 ○	$\neg d$
s_{8_1}	− − − −	0 − 0 −	− − − −	− −	− −	− −	− −	− −	○ 1	d

Figure 42: 3 XOR Gauss example - condensed

Gauss-Jordan elimination.

$$
\begin{array}{cccccccl}
1 & 1 & 0 & 0 & 1 & 0 & = 0 & \\
1 & 0 & 0 & 1 & 0 & 1 & = 0 & \mid + R_1, \mathrm{mod} 2 \\
0 & 1 & 1 & 1 & 0 & 0 & = 0 &
\end{array}
$$

$$
\begin{array}{cccccccl}
1 & 1 & 0 & 0 & 1 & 0 & = 0 & \\
0 & 1 & 0 & 1 & 1 & 1 & = 0 & \\
0 & 1 & 1 & 1 & 0 & 0 & = 0 & \mid + R_2, \mathrm{mod} 2
\end{array}
$$

$$
\begin{array}{cccccccl}
1 & 1 & 0 & 0 & 1 & 0 & = 0 & \mid + R_2, \mathrm{mod} 2 \\
0 & 1 & 0 & 1 & 1 & 1 & = 0 & \\
0 & 0 & 1 & 0 & 1 & 1 & = 0 &
\end{array}
$$

$$
\begin{array}{ccccccc}
1 & 0 & 0 & 1 & 0 & 1 & = 0 \\
0 & 1 & 0 & 1 & 1 & 1 & = 0 \\
0 & 0 & 1 & 0 & 1 & 1 & = 0
\end{array}
$$

Transformed CDF formula.

$$
\begin{array}{lll}
(& (\neg a \wedge \neg f \wedge \neg d) & \vee \\
 & (\neg a \wedge \quad f \wedge \quad d) & \vee \\
 & (\quad a \wedge \neg f \wedge \quad d) & \vee \\
 & (\quad a \wedge \quad f \wedge \neg d) &) \wedge \\
(& (\neg b \wedge \neg f \wedge \neg c \wedge \neg d) & \vee \\
 & (\neg b \wedge \neg f \wedge \quad c \wedge \quad d) & \vee \\
 & (\neg b \wedge \quad f \wedge \neg c \wedge \quad d) & \vee \\
 & (\neg b \wedge \quad f \wedge \quad c \wedge \neg d) & \vee \\
 & (\quad b \wedge \neg f \wedge \neg c \wedge \quad d) & \vee \\
 & (\quad b \wedge \neg f \wedge \quad c \wedge \neg d) & \vee \\
 & (\quad b \wedge \quad f \wedge \neg c \wedge \neg d) & \vee \\
 & (\quad b \wedge \quad f \wedge \quad c \wedge \quad d) &) \wedge \\
(& (\neg e \wedge \neg c \wedge \neg d) & \vee \\
 & (\neg e \wedge \quad c \wedge \quad d) & \vee \\
 & (\quad e \wedge \neg c \wedge \quad d) & \vee \\
 & (\quad e \wedge \quad c \wedge \neg d) &)
\end{array}
$$

P	− − − −	− − − − − − − −	− − − −	− −	− −	− −	− −	− −	− −	
s_{0_0}	1 ∘ ∘ ∘	− 0 0 0 0 − 0 0	− 0 0 −	0 1	− −	− −	0 1	− −	0 1	$a \veebar f \veebar \neg d = 0$
s_{0_1}	∘ 1 ∘ ∘	0 0 − 0 0 0 0 −	0 − − 0	0 1	− −	− −	1 0	− −	1 0	
s_{0_2}	∘ ∘ 1 ∘	0 − 0 0 − 0 0 0	0 − − 0	1 0	− −	− −	0 1	− −	1 0	
s_{0_3}	∘ ∘ ∘ 1	0 0 0 − 0 0 − 0	− 0 0 −	1 0	− −	− −	1 0	− −	0 1	
s_{1_0}	1 0 0 0	1 ∘ ∘ ∘ ∘ ∘ ∘ ∘	1 0 0 0	0 1	0 1	0 1	0 1	0 1	0 1	$b \veebar f \veebar \neg c \veebar \neg d = 0$
s_{1_1}	0 0 1 0	∘ 1 ∘ ∘ ∘ ∘ ∘ ∘	0 1 0 0	1 0	0 1	0 1	0 1	1 0	1 0	
s_{1_2}	0 1 0 0	∘ ∘ 1 ∘ ∘ ∘ ∘ ∘	0 0 1 0	0 1	0 1	1 0	1 0	0 1	1 0	
s_{1_3}	0 0 0 1	∘ ∘ ∘ 1 ∘ ∘ ∘ ∘	0 0 0 1	1 0	0 1	1 0	1 0	1 0	0 1	
s_{1_4}	0 0 1 0	∘ ∘ ∘ ∘ 1 ∘ ∘ ∘	0 0 1 0	1 0	1 0	1 0	0 1	0 1	1 0	
s_{1_5}	1 0 0 0	∘ ∘ ∘ ∘ ∘ 1 ∘ ∘	0 0 0 1	0 1	1 0	1 0	0 1	1 0	0 1	
s_{1_6}	0 0 0 1	∘ ∘ ∘ ∘ ∘ ∘ 1 ∘	1 0 0 0	1 0	1 0	0 1	1 0	0 1	0 1	
s_{1_7}	0 1 0 0	∘ ∘ ∘ ∘ ∘ ∘ ∘ 1	0 1 0 0	0 1	1 0	0 1	1 0	1 0	1 0	
s_{2_0}	− 0 0 −	− 0 0 0 0 0 − 0	1 ∘ ∘ ∘	− −	− −	0 1	− −	0 1	0 1	$e \veebar \neg c \veebar \neg d = 0$
s_{2_1}	0 − − 0	0 − 0 0 0 0 0 −	∘ 1 ∘ ∘	− −	− −	0 1	− −	1 0	1 0	
s_{2_2}	0 − − 0	0 0 − 0 − 0 0 0	∘ ∘ 1 ∘	− −	− −	1 0	− −	0 1	1 0	
s_{2_3}	− 0 0 −	0 0 0 − 0 − 0 0	∘ ∘ ∘ 1	− −	− −	1 0	− −	1 0	0 1	
s_{3_0}	0 0 − −	0 − 0 − − 0 − 0	− − − −	1 ∘	− −	− −	− −	− −	− −	a
s_{3_1}	− − 0 0	− 0 − 0 0 − 0 −	− − − −	∘ 1	− −	− −	− −	− −	− −	$\neg a$
s_{4_0}	− − − −	0 0 0 0 − − − −	− − − −	− −	1 ∘	− −	− −	− −	− −	b
s_{4_1}	− − − −	− − − − 0 0 0 0	− − − −	− −	∘ 1	− −	− −	− −	− −	$\neg b$
s_{5_0}	− − − −	0 0 − − − − 0 0	0 0 − −	− −	− −	1 ∘	− −	− −	− −	e
s_{5_1}	− − − −	− − 0 0 0 0 − −	− − 0 0	− −	− −	∘ 1	− −	− −	− −	$\neg e$
s_{6_0}	0 − 0 −	0 0 − − 0 0 − −	− − − −	− −	− −	− −	1 ∘	− −	− −	f
s_{6_1}	− 0 − 0	− − 0 0 − − 0 0	− − − −	− −	− −	− −	∘ 1	− −	− −	$\neg f$
s_{7_0}	− − − −	0 − 0 − 0 − 0 −	0 − 0 −	− −	− −	− −	− −	1 ∘	− −	$\neg c$
s_{7_1}	− − − −	− 0 − 0 − 0 − 0	− 0 − 0	− −	− −	− −	− −	∘ 1	− −	c
s_{8_0}	0 − − 0	0 − − 0 − 0 0 −	0 − − 0	− −	− −	− −	− −	− −	1 ∘	$\neg d$
s_{8_1}	− 0 0 −	− 0 0 − 0 − − 0	− 0 0 −	− −	− −	− −	− −	− −	∘ 1	d

Figure 43: 3 XOR Gauss example - reformulated

12. 3-Regular Bipartite Graph Problem Example

An excerpt from a propositional problem derived from a 3-regular bipartite graph[JARV] is shown to demonstrate the visual information that can be gained from analyzing a propositional problem in structural logic.

The characteristic structure is already recognizable in the 3-state cell version in figure 44.

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	0 − −	1 0 0	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_1}	∘ 1 ∘	1 0 0	0 − −	1 0 0	− − 0	− − 0	− − −	− − −	− 0 −	− 0 −	− − −	− − −
s_{0_2}	∘ ∘ 1	1 0 0	0 − −	1 0 0	− 0 −	− 0 −	− − −	− − −	− − 0	− − 0	− − −	− − −
s_{1_0}	0 − −	1 ∘ ∘	0 − −	1 0 0	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{1_1}	1 0 0	∘ 1 ∘	1 0 0	0 − −	− − 0	− − 0	− − −	− − −	− − 0	− − 0	− − −	− − −
s_{1_2}	1 0 0	∘ ∘ 1	1 0 0	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	− − −
s_{2_0}	1 0 0	0 − −	1 ∘ ∘	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_1}	0 − −	1 0 0	∘ 1 ∘	1 0 0	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_2}	0 − −	1 0 0	∘ ∘ 1	1 0 0	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{3_0}	0 − −	1 0 0	0 − −	1 ∘ ∘	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{3_1}	1 0 0	0 − −	1 0 0	∘ 1 ∘	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{3_2}	1 0 0	0 − −	1 0 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{4_0}	− − −	− − −	− − −	− − −	1 ∘ ∘	0 − −	1 0 0	0 − −	− − −	− − −	− − −	− − −
s_{4_1}	− − 0	− − 0	− − −	− − −	∘ 1 ∘	1 0 0	0 − −	1 0 0	− − −	− − −	− − −	− − −
s_{4_2}	− 0 −	− 0 −	− − −	− − −	∘ ∘ 1	1 0 0	0 − −	1 0 0	− − −	− − −	− − −	− − −
s_{5_0}	− − −	− − −	− − −	− − −	0 − −	1 ∘ ∘	0 − −	1 0 0	− − −	− − −	− − −	− − −
s_{5_1}	− − 0	− − 0	− − −	− − −	1 0 0	∘ 1 ∘	1 0 0	0 − −	− − −	− − −	− − −	− − −
s_{5_2}	− 0 −	− 0 −	− − −	− − −	1 0 0	∘ ∘ 1	1 0 0	0 − −	− − −	− − −	− − −	− − −
s_{6_0}	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 ∘ ∘	0 − −	− − −	− − −	− − −	− − −
s_{6_1}	− − −	− − −	− − −	− − −	0 − −	1 0 0	∘ 1 ∘	1 0 0	− − −	− − −	− − −	− − −
s_{6_2}	− − −	− − −	− − −	− − −	0 − −	1 0 0	∘ ∘ 1	1 0 0	− − −	− − −	− − −	− − −
s_{7_0}	− − −	− − −	− − −	− − −	0 − −	1 0 0	0 − −	1 ∘ ∘	− − −	− − −	− − −	− − −
s_{7_1}	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 0 0	∘ 1 ∘	− − −	− − −	− − −	− − −
s_{7_2}	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 0 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{8_0}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	1 ∘ ∘	0 − −	1 0 0	0 − −
s_{8_1}	− 0 −	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1 ∘	1 0 0	0 − −	1 0 0
s_{8_2}	− − 0	− 0 −	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	1 0 0	0 − −	1 0 0
s_{9_0}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −	1 ∘ ∘	0 − −	1 0 0
s_{9_1}	− 0 −	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	1 0 0	∘ 1 ∘	1 0 0	0 − −
s_{9_2}	− − 0	− 0 −	− − −	− − −	− − −	− − −	− − −	− − −	1 0 0	∘ ∘ 1	1 0 0	0 − −
s_{10_0}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 ∘ ∘	0 − −
s_{10_1}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −	1 0 0	∘ 1 ∘	1 0 0
s_{10_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −	1 0 0	∘ ∘ 1	1 0 0
s_{11_0}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −	1 0 0	0 − −	1 ∘ ∘
s_{11_1}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 0 0	∘ 1 ∘
s_{11_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	1 0 0	0 − −	1 0 0	∘ ∘ 1

Figure 44: 3-state cell satoku matrix for bipartite problem (excerpt)

The 4-state cell satoku matrix shown in figure 45 was derived from merging the 3-state cells and reveals the XOR-structure of the problem entirely.

P	− − − −	− − − −	− − − −	− − − −	− − − −	− − − −
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	− 0 − 0	− − − −	− 0 − 0	− − − −
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	0 − 0 −	− − − −	0 − 0 −	− − − −
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	− 0 − 0	− − − −	0 − 0 −	− − − −
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	0 − 0 −	− − − −	− 0 − 0	− − − −
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	− − − −	− − − −	− − − −	− − − −
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	− − − −	− − − −	− − − −	− − − −
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	− − − −	− − − −	− − − −	− − − −
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	− − − −	− − − −	− − − −	− − − −
s_{2_0}	− 0 − 0	− − − −	1 ∘ ∘ ∘	− − 0 0	− − − −	− − − −
s_{2_1}	0 − 0 −	− − − −	∘ 1 ∘ ∘	− − 0 0	− − − −	− − − −
s_{2_2}	− 0 − 0	− − − −	∘ ∘ 1 ∘	0 0 − −	− − − −	− − − −
s_{2_3}	0 − 0 −	− − − −	∘ ∘ ∘ 1	0 0 − −	− − − −	− − − −
s_{3_0}	− − − −	− − − −	− − 0 0	1 ∘ ∘ ∘	− − − −	− − − −
s_{3_1}	− − − −	− − − −	− − 0 0	∘ 1 ∘ ∘	− − − −	− − − −
s_{3_2}	− − − −	− − − −	0 0 − −	∘ ∘ 1 ∘	− − − −	− − − −
s_{3_3}	− − − −	− − − −	0 0 − −	∘ ∘ ∘ 1	− − − −	− − − −
s_{4_0}	− 0 0 −	− − − −	− − − −	− − − −	1 ∘ ∘ ∘	− − 0 0
s_{4_1}	0 − − 0	− − − −	− − − −	− − − −	∘ 1 ∘ ∘	− − 0 0
s_{4_2}	− 0 0 −	− − − −	− − − −	− − − −	∘ ∘ 1 ∘	0 0 − −
s_{4_3}	0 − − 0	− − − −	− − − −	− − − −	∘ ∘ ∘ 1	0 0 − −
s_{5_0}	− − − −	− − − −	− − − −	− − − −	− − 0 0	1 ∘ ∘ ∘
s_{5_1}	− − − −	− − − −	− − − −	− − − −	− − 0 0	∘ 1 ∘ ∘
s_{5_2}	− − − −	− − − −	− − − −	− − − −	0 0 − −	∘ ∘ 1 ∘
s_{5_3}	− − − −	− − − −	− − − −	− − − −	0 0 − −	∘ ∘ ∘ 1

.

Figure 45: 4-state cell satoku matrix for bipartite problem (excerpt)

Note: It is possible to reconstruct the linear equation system from this information with simple heuristics. Even obfuscated versions of such problems can be deobfuscated. A major advantage of structural logic is its independence from encoding for these types of problems, while regular SAT-solvers with XOR-clause-detection for Gauss-elimination can easily be fooled (see section 13).

13. Construction of Desirable Encodings

Some SAT-solvers employ algorithms to detect XOR-clauses for Gaussian elminiation. However, when a propositional CNF problem is reencoded with direct encoding (especially, when leaving out the at-most-one constraints), the XOR structure is no longer detected.

In this case, structural logic can be used to construct a more suitable CNF encoding, that allows a SAT-solver to choose an optimal strategy.

The principle is shown as full proof with an 8-clause excerpt from a larger 3-regular bipartite graph problem (figure 46), mapped from the CNF formula in direct encoding:

$$
\begin{array}{ll}
(\; a \vee \; b \vee \; c \vee \; d) & \wedge \\
(\; e \vee \; f \vee \; g \vee \; h) & \wedge \\
(\neg a \vee \neg b) \wedge (\neg a \vee \neg c) \wedge (\neg a \vee \neg d) & \wedge \\
(\neg b \vee \neg c) \wedge (\neg b \vee \neg d) \wedge (\neg c \vee \neg d) & \wedge \\
(\neg e \vee \neg f) \wedge (\neg e \vee \neg g) \wedge (\neg e \vee \neg h) & \wedge \\
(\neg f \vee \neg g) \wedge (\neg f \vee \neg h) \wedge (\neg g \vee \neg h) & \wedge \\
(\neg a \vee \neg g) \wedge (\neg a \vee \neg h) & \wedge \\
(\neg b \vee \neg g) \wedge (\neg b \vee \neg h) & \wedge \\
(\neg c \vee \neg e) \wedge (\neg c \vee \neg f) & \wedge \\
(\neg d \vee \neg e) \wedge (\neg d \vee \neg f) &
\end{array}
$$

P	$- - - -$	$- - - -$
s_{0_0}	$1 \circ \circ \circ$	$- - 0\,0$
s_{0_1}	$\circ\, 1 \circ \circ$	$- - 0\,0$
s_{0_2}	$\circ \circ 1 \circ$	$0\,0 - -$
s_{0_3}	$\circ \circ \circ\, 1$	$0\,0 - -$
s_{1_0}	$- - 0\,0$	$1 \circ \circ \circ$
s_{1_1}	$- - 0\,0$	$\circ\, 1 \circ \circ$
s_{1_2}	$0\,0 - -$	$\circ \circ 1 \circ$
s_{1_3}	$0\,0 - -$	$\circ \circ \circ\, 1$

Figure 46: Propositional XOR and graph theoretic choice

Any 4-state cell in a satoku matrix can be represented by a 3-variable XOR. This has been done in figure 47 by adding the appropriate variable and DNF representations.

Note that the corresponding CNF encoding does not yet correctly produce the original satoku matrix, since the conflict relationships between the cells are not encoded.

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	1 0	1 0	1 0	− −	− −	− −
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	1 0	0 1	0 1	− −	− −	− −
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	0 1	1 0	0 1	− −	− −	− −
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	− −	− −	− −	1 0	1 0	1 0
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	− −	− −	− −	1 0	0 1	0 1
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	− −	− −	− −	0 1	1 0	0 1
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	− −	− −	− −	0 1	0 1	1 0
s_{2_0}	− − 0 0	− − − −	1 ∘	− −	− −	− −	− −	− −
s_{2_1}	0 0 − −	− − − −	∘ 1	− −	− −	− −	− −	− −
s_{3_0}	− 0 − 0	− − − −	− −	1 ∘	− −	− −	− −	− −
s_{3_1}	0 − 0 −	− − − −	− −	∘ 1	− −	− −	− −	− −
s_{4_0}	− 0 0 −	− − − −	− −	− −	1 ∘	− −	− −	− −
s_{4_1}	0 − − 0	− − − −	− −	− −	∘ 1	− −	− −	− −
s_{5_0}	− − − −	− − 0 0	− −	− −	− −	1 ∘	− −	− −
s_{5_1}	− − − −	0 0 − −	− −	− −	− −	∘ 1	− −	− −
s_{6_0}	− − − −	− 0 − 0	− −	− −	− −	− −	1 ∘	− −
s_{6_1}	− − − −	0 − 0 −	− −	− −	− −	− −	∘ 1	− −
s_{7_0}	− − − −	− 0 0 −	− −	− −	− −	− −	− −	1 ∘
s_{7_1}	− − − −	0 − − 0	− −	− −	− −	− −	− −	∘ 1

.

Figure 47: XOR/choice: Preliminary XOR variables

Single choice 2-state cells (at least one/at most one encoding) have been added in figure 48. These states correctly determine the original 4-state cells.

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −
$s0_0$	1 ∘ ∘ ∘	− − 0 0	1 0	1 0	1 0	− −	− −	− −	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1
$s0_1$	∘ 1 ∘ ∘	− − 0 0	1 0	0 1	0 1	− −	− −	− −	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1
$s0_2$	∘ ∘ 1 ∘	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1	− −	− −
$s0_3$	∘ ∘ ∘ 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1	− −	− −
$s1_0$	− − 0 0	1 ∘ ∘ ∘	− −	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1
$s1_1$	− − 0 0	∘ 1 ∘ ∘	− −	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1
$s1_2$	0 0 − −	∘ ∘ 1 ∘	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 0	0 1
$s1_3$	0 0 − −	∘ ∘ ∘ 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0
$s2_0$	− − 0 0	− − − −	1 ∘	− −	− −	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −
$s2_1$	0 0 − −	− − − −	∘ 1	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− −	− −
$s3_0$	− 0 − 0	− − − −	− −	1 ∘	− −	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −
$s3_1$	0 − 0 −	− − − −	− −	∘ 1	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −	− −
$s4_0$	− 0 0 −	− − − −	− −	− −	1 ∘	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− −
$s4_1$	0 − − 0	− − − −	− −	− −	∘ 1	− −	− −	− −	0 1	− −	− −	0 1	− −	− −	− −	− −
$s5_0$	− − − −	− − 0 0	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	− −	− −	− −	0 1	0 1
$s5_1$	− − − −	0 0 − −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −	0 1	0 1	− −	− −
$s6_0$	− − − −	− 0 − 0	− −	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	− −	0 1	− −	0 1
$s6_1$	− − − −	0 − 0 −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	0 1	− −	0 1	− −
$s7_0$	− − − −	− 0 0 −	− −	− −	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	0 1	0 1	− −
$s7_1$	− − − −	0 − − 0	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	0 1	− −	− −	0 1
$s8_0$	1 0 0 0	− − 0 0	1 0	1 0	1 0	− −	− −	− −	1 ∘	0 1	0 1	0 1	− −	− −	0 1	0 1
$s8_1$	0 − − −	− − − −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −	− −
$s9_0$	0 1 0 0	− − 0 0	1 0	0 1	0 1	− −	− −	− −	0 1	1 ∘	0 1	0 1	− −	− −	0 1	0 1
$s9_1$	− 0 − −	− − − −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −
$s10_0$	0 0 1 0	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 ∘	0 1	0 1	0 1	− −	− −
$s10_1$	− − 0 −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −
$s11_0$	0 0 0 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 ∘	0 1	0 1	− −	− −
$s11_1$	− − − 0	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −
$s12_0$	− − 0 0	1 0 0 0	− −	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 ∘	0 1	0 1	0 1
$s12_1$	− − − −	0 − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −
$s13_0$	− − 0 0	0 1 0 0	− −	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ∘	0 1	0 1
$s13_1$	− − − −	− 0 − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −
$s14_0$	0 0 − −	0 0 1 0	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 ∘	0 1
$s14_1$	− − − −	− − 0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −
$s15_0$	0 0 − −	0 0 0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ∘
$s15_1$	− − − −	− − − 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1

Figure 48: XOR/choice: single choice variables (at-most-one)

In figure 49, decisions for merging cells $c_{8_8}, c_{12_{12}}$.

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −
s_{0_0}	1 ○ ○ ○	− − 0 0	1 0	1 0	1 0	− −	− −	− −	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	− − − −
s_{0_1}	○ 1 ○ ○	− − 0 0	1 0	0 1	0 1	− −	− −	− −	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	− − − −
s_{0_2}	○ ○ 1 ○	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1	− −	− −	− − − −
s_{0_3}	○ ○ ○ 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1	− −	− −	− − − −
s_{1_0}	− − 0 0	1 ○ ○ ○	− −	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1	− − − −
s_{1_1}	− − 0 0	○ 1 ○ ○	− −	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1	− − − −
s_{1_2}	0 0 − −	○ ○ 1 ○	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 0	0 1	− − − −
s_{1_3}	0 0 − −	○ ○ ○ 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	− − − −
s_{2_0}	− − 0 0	− − − −	1 ○	− −	− −	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− − − −
s_{2_1}	0 0 − −	− − − −	○ 1	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− −	− −	− − − −
s_{3_0}	− 0 − 0	− − − −	− −	1 ○	− −	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −	− − − −
s_{3_1}	0 − 0 −	− − − −	− −	○ 1	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −	− −	− − − −
s_{4_0}	− 0 0 −	− − − −	− −	− −	1 ○	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− −	− − − −
s_{4_1}	0 − − 0	− − − −	− −	− −	○ 1	− −	− −	− −	0 1	− −	− −	0 1	− −	− −	− −	− −	− − − −
s_{5_0}	− − − −	− − 0 0	− −	− −	− −	1 ○	− −	− −	− −	− −	− −	− −	− −	− −	0 1	0 1	− − − −
s_{5_1}	− − − −	0 0 − −	− −	− −	− −	○ 1	− −	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− − − −
s_{6_0}	− − − −	− 0 − 0	− −	− −	− −	− −	1 ○	− −	− −	− −	− −	− −	− −	0 1	− −	0 1	− − − −
s_{6_1}	− − − −	0 − 0 −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	− −	0 1	− −	0 1	− −	− − − −
s_{7_0}	− − − −	− 0 0 −	− −	− −	− −	− −	− −	1 ○	− −	− −	− −	− −	− −	0 1	0 1	− −	− − − −
s_{7_1}	− − − −	0 − − 0	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	0 1	− −	− −	0 1	− − − −
s_{8_0}	1 0 0 0	− − 0 0	1 0	1 0	1 0	− −	− −	− −	1 ○	0 1	0 1	0 1	− −	− −	0 1	0 1	− − 0 0
s_{8_1}	0 − − −	− − − −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	− −	− −	− −	0 0 − −
s_{9_0}	0 1 0 0	− − 0 0	1 0	0 1	0 1	− −	− −	− −	0 1	1 ○	0 1	0 1	− −	− −	0 1	0 1	− − − −
s_{9_1}	− 0 − −	− − − −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	− −	− −	− − − −
s_{10_0}	0 0 1 0	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 ○	0 1	0 1	0 1	− −	− −	− − − −
s_{10_1}	− − 0 −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	− −	− − − −
s_{11_0}	0 0 0 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 ○	0 1	0 1	− −	− −	− − − −
s_{11_1}	− − − 0	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	− −	− − − −
s_{12_0}	− − 0 0	1 0 0 0	− −	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 ○	0 1	0 1	0 1	− 0 − 0
s_{12_1}	− − − −	0 − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− −	0 − 0 −
s_{13_0}	− − 0 0	0 1 0 0	− −	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ○	0 1	0 1	− − − −
s_{13_1}	− − − −	− 0 − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− −	− − − −
s_{14_0}	0 0 − −	0 0 1 0	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 ○	0 1	− − − −
s_{14_1}	− − − −	− − 0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− −	− − − −
s_{15_0}	0 0 − −	0 0 0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ○	− − − −
s_{15_1}	− − − −	− − − 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	○ 1	− − − −
s_{16_0}	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− 0	− −	− −	− −	− 0	− −	− −	− −	1 ○ ○ ○
s_{16_1}	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− 0	− −	− −	− −	0 −	− −	− −	− −	○ 1 ○ ○
s_{16_2}	− − − −	− − − −	− −	− −	− −	− −	− −	− −	0 −	− −	− −	− −	− 0	− −	− −	− −	○ ○ 1 ○
s_{16_3}	− − − −	− − − −	− −	− −	− −	− −	− −	− −	0 −	− −	− −	− −	0 −	− −	− −	− −	○ ○ ○ 1

Figure 49: XOR/choice

+ more CRs!

Consolidation produces the *decided* conflict relationships $r_{8_{0_5}}$ and the propagated decision in $r_{0_{0_5}}$. Decisions for merging cells $c_{9_9}, c_{13_{13}}$ have been added (see figure 50).

P	– – – –	– – – –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– – – –	– – – –
s_{0_0}	1 ∘ ∘ ∘	– – 0 0	1 0	1 0	1 0	1 0	– –	– –	1 0	0 1	0 1	0 1	– –	– –	0 1	0 1	– – 0 0	– – – –
s_{0_1}	∘ 1 ∘ ∘	– – 0 0	1 0	0 1	0 1	– –	– –	– –	0 1	1 0	0 1	0 1	– –	– –	0 1	0 1	0 0 – –	– – – –
s_{0_2}	∘ ∘ 1 ∘	0 0 – –	0 1	1 0	0 1	– –	– –	– –	0 1	0 1	1 0	0 1	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{0_3}	∘ ∘ ∘ 1	0 0 – –	0 1	0 1	1 0	– –	– –	– –	0 1	0 1	0 1	1 0	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{1_0}	– – 0 0	1 ∘ ∘ ∘	1 0	– –	– –	1 0	1 0	1 0	– –	– –	0 1	0 1	1 0	0 1	0 1	0 1	– 0 – 0	– – – –
s_{1_1}	– – 0 0	∘ 1 ∘ ∘	1 0	– –	– –	1 0	0 1	0 1	– –	– –	0 1	0 1	0 1	1 0	0 1	0 1	0 – 0 –	– – – –
s_{1_2}	0 0 – –	∘ ∘ 1 ∘	– –	– –	– –	0 1	1 0	0 1	0 1	0 1	– –	– –	0 1	0 1	1 0	0 1	0 0 0 1	– – – –
s_{1_3}	0 0 – –	∘ ∘ ∘ 1	– –	– –	– –	0 1	0 1	1 0	0 1	0 1	– –	– –	0 1	0 1	0 1	1 0	0 0 0 1	– – – –
s_{2_0}	– – 0 0	– – – –	1 ∘	– –	– –	– –	– –	– –	– –	– –	0 1	0 1	– –	– –	– –	– –	– – – –	– – – –
s_{2_1}	0 0 – –	0 0 – –	∘ 1	– –	– –	– –	– –	– –	0 1	0 1	– –	– –	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{3_0}	– 0 – 0	– – – –	– –	1 ∘	– –	– –	– –	– –	– –	0 1	– –	0 1	– –	– –	– –	– –	– – 0 –	– – – –
s_{3_1}	0 – 0 –	– – – –	– –	∘ 1	– –	– –	– –	– –	0 1	– –	0 1	– –	– –	– –	– –	– –	0 0 – –	– – – –
s_{4_0}	– 0 0 –	– – – –	– –	– –	1 ∘	– –	– –	– –	– –	0 1	0 1	– –	– –	– –	– –	– –	– – 0 –	– – – –
s_{4_1}	0 – – 0	– – – –	– –	– –	∘ 1	– –	– –	– –	0 1	– –	– –	0 1	– –	– –	– –	– –	0 0 – –	– – – –
s_{5_0}	– – – –	– – 0 0	– –	– –	– –	1 ∘	– –	– –	– –	– –	– –	– –	– –	– –	0 1	0 1	– – – –	– – – –
s_{5_1}	0 – – –	0 0 – –	– –	– –	– –	∘ 1	– –	– –	0 1	– –	– –	– –	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{6_0}	– – – –	– 0 – 0	– –	– –	– –	– –	1 ∘	– –	– –	– –	– –	– –	– –	0 1	– –	0 1	– 0 – –	– – – –
s_{6_1}	– – – –	0 – 0 –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	– –	0 1	– –	0 1	– –	0 – 0 –	– – – –
s_{7_0}	– – – –	– 0 0 –	– –	– –	– –	– –	– –	1 ∘	– –	– –	– –	– –	– –	0 1	0 1	– –	– 0 – –	– – – –
s_{7_1}	– – – –	0 – – 0	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	0 1	– –	– –	0 1	0 – 0 –	– – – –
s_{8_0}	1 0 0 0	– – 0 0	1 0	1 0	1 0	1 0	– –	– –	1 ∘	0 1	0 1	0 1	– –	– –	0 1	0 1	– – 0 0	– – – –
s_{8_1}	0 – – –	– – – –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	– –	– –	– –	0 0 – –	– – – –
s_{9_0}	0 1 0 0	– – 0 0	1 0	0 1	0 1	– –	– –	– –	0 1	1 ∘	0 1	0 1	– –	– –	0 1	0 1	0 0 – –	– – 0 0
s_{9_1}	– 0 – –	– – – –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	– –	– –	– – 0 –	0 0 – –
s_{10_0}	0 0 1 0	0 0 – –	0 1	1 0	0 1	– –	– –	– –	0 1	0 1	1 ∘	0 1	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{10_1}	– – 0 –	– – – –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	– –	– – – –	– – – –
s_{11_0}	0 0 0 1	0 0 – –	0 1	0 1	1 0	– –	– –	– –	0 1	0 1	0 1	1 ∘	0 1	0 1	– –	– –	0 0 0 1	– – – –
s_{11_1}	– – – 0	– – – –	– –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	– –	– – – –	– – – –
s_{12_0}	– – 0 0	1 0 0 0	1 0	– –	– –	1 0	1 0	1 0	– –	– –	0 1	0 1	1 ∘	0 1	0 1	0 1	– 0 – 0	– – – –
s_{12_1}	– – – –	0 – – –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– –	0 – 0 –	– – – –
s_{13_0}	– – 0 0	0 1 0 0	1 0	– –	– –	1 0	0 1	0 1	– –	– –	0 1	0 1	0 1	1 ∘	0 1	0 1	0 – 0 –	– 0 – 0
s_{13_1}	– – – –	– 0 – –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– –	– 0 – –	0 – 0 –
s_{14_0}	0 0 – –	0 0 1 0	– –	– –	– –	0 1	1 0	0 1	0 1	0 1	– –	– –	0 1	0 1	1 ∘	0 1	0 0 0 1	– – – –
s_{14_1}	– – – –	– – 0 –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– –	– – – –	– – – –
s_{15_0}	0 0 – –	0 0 0 1	– –	– –	– –	0 1	0 1	1 0	0 1	0 1	– –	– –	0 1	0 1	0 1	1 ∘	0 0 0 1	– – – –
s_{15_1}	– – – –	– – – 0	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	– –	∘ 1	– – – –	– – – –
s_{16_0}	1 0 0 0	1 0 0 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 ∘ ∘ ∘	– – – –
s_{16_1}	1 0 0 0	0 1 0 0	1 0	1 0	1 0	1 0	0 1	0 1	1 0	0 1	0 1	0 1	0 1	1 0	0 1	0 1	∘ 1 ∘ ∘	– – – –
s_{16_2}	0 1 0 0	1 0 0 0	1 0	0 1	0 1	1 0	1 0	1 0	0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 1	∘ ∘ 1 ∘	– – – –
s_{16_3}	0 – – –	0 – – –	– –	– –	– –	– –	– –	– –	0 1	– –	– –	– –	0 1	– –	– –	– –	∘ ∘ ∘ 1	– – – –
s_{17_0}	– – – –	– – – –	– –	– –	– –	– –	– –	– –	– –	– 0	– –	– –	– –	– 0	– –	– –	– – – –	1 ∘ ∘ ∘
s_{17_1}	– – – –	– – – –	– –	– –	– –	– –	– –	– –	– –	– 0	– –	– –	– –	0 –	– –	– –	– – – –	∘ 1 ∘ ∘
s_{17_2}	– – – –	– – – –	– –	– –	– –	– –	– –	– –	– –	0 –	– –	– –	– –	– 0	– –	– –	– – – –	∘ ∘ 1 ∘
s_{17_3}	– – – –	– – – –	– –	– –	– –	– –	– –	– –	– –	0 –	– –	– –	– –	0 –	– –	– –	– – – –	∘ ∘ ∘ 1

Figure 50: XOR/choice

Consolidation produces the *decided* conflict relationships $r_{9_{0_5}}$ and the propagated decision in $r_{0_{1_5}}$. Decisions for merging cells c_{2_2}, c_{5_5} have been added (see figure 51).

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −	− − − −	− − − −
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	1 0	1 0	1 0	1 0	− −	− −	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	− − 0 0	0 0 − −	− − − −
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	1 0	0 1	0 1	1 0	− −	− −	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 0 − −	− − 0 0	− − − −
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	− − − −
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	− − − −
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	1 0	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 0	0 1	0 1	0 1	− 0 − 0	0 − 0 −	− − − −
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	1 0	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	0 1	0 1	0 − 0 −	− 0 − 0	− − − −
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 0	0 1	0 0 0 1	0 0 0 1	− − − −
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	0 0 0 1	0 0 0 1	− − − −
s_{2_0}	− − 0 0	− − − −	1 ∘	− −	− −	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− − − −	− − − −	− − 0 0
s_{2_1}	0 0 − −	0 0 − −	∘ 1	− −	− −	− −	− −	− −	0 1	0 1	− −	− −	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	0 0 − −
s_{3_0}	− 0 − 0	− − − −	− −	1 ∘	− −	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −	− − 0 −	0 0 − −	− − − −
s_{3_1}	0 − 0 −	− − − −	− −	∘ 1	− −	− −	− −	− −	0 1	− −	0 1	− −	− −	− −	− −	− −	0 0 − −	− − 0 −	− − − −
s_{4_0}	− 0 0 −	− − − −	− −	− −	1 ∘	− −	− −	− −	− −	0 1	0 1	− −	− −	− −	− −	− −	− − 0 −	0 0 − −	− − − −
s_{4_1}	0 − − 0	− − − −	− −	− −	∘ 1	− −	− −	− −	0 1	− −	− −	0 1	− −	− −	− −	− −	0 0 − −	− − 0 −	− − − −
s_{5_0}	− − − −	− − 0 0	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	− −	− −	− −	0 1	0 1	− − − −	− − − −	− 0 − 0
s_{5_1}	0 0 − −	0 0 − −	− −	− −	− −	∘ 1	− −	− −	0 1	0 1	− −	− −	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	0 − 0 −
s_{6_0}	− − − −	− 0 − 0	− −	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	− −	0 1	− −	0 1	− 0 − −	0 − 0 −	− − − −
s_{6_1}	− − − −	0 − 0 −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	0 1	− −	0 1	− −	0 − 0 −	− 0 − −	− − − −
s_{7_0}	− − − −	− 0 0 −	− −	− −	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	0 1	0 1	− −	− 0 − −	0 − 0 −	− − − −
s_{7_1}	− − − −	0 − − 0	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	0 1	− −	− −	0 1	0 − 0 −	− 0 − −	− − − −
s_{8_0}	1 0 0 0	− − 0 0	1 0	1 0	1 0	1 0	− −	− −	1 ∘	0 1	0 1	0 1	− −	− −	0 1	0 1	− − 0 0	0 0 − −	− − − −
s_{8_1}	0 − − −	− − − −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −	− −	0 0 − −	− − 0 −	− − − −
s_{9_0}	0 1 0 0	− − 0 0	1 0	0 1	0 1	1 0	− −	− −	0 1	1 ∘	0 1	0 1	− −	− −	0 1	0 1	0 0 − −	− − 0 0	− − − −
s_{9_1}	− 0 − −	− − − −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −	− − 0 −	0 0 − −	− − − −
s_{10_0}	0 0 1 0	0 0 − −	0 1	1 0	0 1	− −	− −	− −	0 1	0 1	1 ∘	0 1	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	− − − −
s_{10_1}	− − 0 −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− − − −	− − − −	− − − −
s_{11_0}	0 0 0 1	0 0 − −	0 1	0 1	1 0	− −	− −	− −	0 1	0 1	0 1	1 ∘	0 1	0 1	− −	− −	0 0 0 1	0 0 0 1	− − − −
s_{11_1}	− − − 0	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	− −	− − − −	− − − −	− − − −
s_{12_0}	− − 0 0	1 0 0 0	1 0	− −	− −	1 0	1 0	1 0	− −	− −	0 1	0 1	1 ∘	0 1	0 1	0 1	− 0 − 0	0 − 0 −	− − − −
s_{12_1}	− − − −	0 − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −	0 − 0 −	− 0 − −	− − − −
s_{13_0}	− − 0 0	0 1 0 0	1 0	− −	− −	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ∘	0 1	0 1	0 − 0 −	− 0 − 0	− − − −
s_{13_1}	− − − −	− 0 − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− 0 − −	0 − 0 −	− − − −
s_{14_0}	0 0 − −	0 0 1 0	− −	− −	− −	0 1	1 0	0 1	0 1	0 1	− −	− −	0 1	0 1	1 ∘	0 1	0 0 0 1	0 0 0 1	− − − −
s_{14_1}	− − − −	− − 0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− − − −	− − − −	− − − −
s_{15_0}	0 0 − −	0 0 0 1	− −	− −	− −	0 1	0 1	1 0	0 1	0 1	− −	− −	0 1	0 1	0 1	1 ∘	0 0 0 1	0 0 0 1	− − − −
s_{15_1}	− − − −	− − − 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− − − −	− − − −	− − − −
s_{16_0}	1 0 0 0	1 0 0 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 ∘ ∘ ∘	0 0 0 1	− − − −
s_{16_1}	1 0 0 0	0 1 0 0	1 0	1 0	1 0	1 0	0 1	0 1	1 0	0 1	0 1	0 1	0 1	1 0	0 1	0 1	∘ 1 ∘ ∘	0 0 1 0	− − − −
s_{16_2}	0 1 0 0	1 0 0 0	1 0	0 1	0 1	1 0	1 0	1 0	0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 1	∘ ∘ 1 ∘	0 1 0 0	− − − −
s_{16_3}	0 − − −	0 − − −	− −	− −	− −	− −	− −	− −	0 1	− −	− −	− −	0 1	− −	− −	− −	∘ ∘ ∘ 1	− 0 0 −	− − − −
s_{17_0}	0 1 0 0	0 1 0 0	1 0	0 1	0 1	1 0	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 0	0 1	0 1	0 0 0 1	1 ∘ ∘ ∘	− − − −
s_{17_1}	0 1 0 0	1 0 0 0	1 0	0 1	0 1	1 0	1 0	1 0	0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 1	0 0 1 0	∘ 1 ∘ ∘	− − − −
s_{17_2}	1 0 0 0	0 1 0 0	1 0	1 0	1 0	1 0	0 1	0 1	1 0	0 1	0 1	0 1	0 1	1 0	0 1	0 1	0 1 0 0	∘ ∘ 1 ∘	− − − −
s_{17_3}	− 0 − −	− 0 − −	− −	− −	− −	− −	− −	− −	− −	0 1	− −	− −	− −	0 1	− −	− −	− 0 0 −	∘ ∘ ∘ 1	− − − −
s_{18_0}	− − − −	− − − −	− 0	− −	− −	− 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −	− − − −	1 ∘ ∘ ∘
s_{18_1}	− − − −	− − − −	− 0	− −	− −	0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −	− − − −	∘ 1 ∘ ∘
s_{18_2}	− − − −	− − − −	0 −	− −	− −	− 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −	− − − −	∘ ∘ 1 ∘
s_{18_3}	− − − −	− − − −	0 −	− −	− −	0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− − − −	− − − −	∘ ∘ ∘ 1

Figure 51: XOR/choice

add description

Consolidation reveals in figure 52, that states of cells c_{2_2}, c_{5_5} are actually equivalent.

P	−−−−	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{0_0}	1○○○	−−00	10	10	10	10	−−	−−	10	01	01	01	−−	−−	01	01	−−00	00−−	1000
s_{0_1}	○1○○	−−00	10	01	01	10	−−	−−	01	10	01	01	−−	−−	01	01	00−−	−−00	1000
s_{0_2}	○○1○	00−−	01	10	01	01	−−	−−	01	01	10	01	01	01	−−	−−	0001	0001	0001
s_{0_3}	○○○1	00−−	01	01	10	01	−−	−−	01	01	01	10	01	01	−−	−−	0001	0001	0001
s_{1_0}	−−00	1○○○	10	−−	−−	10	10	10	−−	−−	01	01	10	01	01	01	−0−0	0−0−	1000
s_{1_1}	−−00	○1○○	10	−−	−−	10	01	01	−−	−−	01	01	01	10	01	01	0−0−	−0−0	1000
s_{1_2}	00−−	○○1○	01	−−	−−	01	10	01	01	01	−−	−−	01	01	10	01	0001	0001	0001
s_{1_3}	00−−	○○○1	01	−−	−−	01	01	10	01	01	−−	−−	01	01	01	10	0001	0001	0001
s_{2_0}	−−00	−−00	1○	−−	−−	10	−−	−−	−−	−−	01	01	−−	−−	01	01	−−−−	−−−−	1000
s_{2_1}	00−−	00−−	○1	−−	−−	01	−−	−−	01	01	−−	−−	01	01	−−	−−	0001	0001	0001
s_{3_0}	−0−0	−−−−	−−	1○	−−	−−	−−	−−	−−	01	−−	01	−−	−−	−−	−−	−−0−	00−−	−00−
s_{3_1}	0−0−	−−−−	−−	○1	−−	−−	−−	−−	01	−−	01	−−	−−	−−	−−	−−	00−−	−−0−	−00−
s_{4_0}	−00−	−−−−	−−	−−	1○	−−	−−	−−	−−	01	01	−−	−−	−−	−−	−−	−−0−	00−−	−00−
s_{4_1}	0−−0	−−−−	−−	−−	○1	−−	−−	−−	01	−−	−−	01	−−	−−	−−	−−	00−−	−−0−	−00−
s_{5_0}	−−00	−−00	10	−−	−−	1○	−−	−−	−−	−−	01	01	−−	−−	01	01	−−−−	−−−−	1000
s_{5_1}	00−−	00−−	01	−−	−−	○1	−−	−−	01	01	−−	−−	01	01	−−	−−	0001	0001	0001
s_{6_0}	−−−−	−0−0	−−	−−	−−	−−	1○	−−	−−	−−	−−	−−	−−	01	−−	01	−0−−	0−0−	−00−
s_{6_1}	−−−−	0−0−	−−	−−	−−	−−	○1	−−	−−	−−	−−	−−	01	−−	01	−−	0−0−	−0−−	−00−
s_{7_0}	−−−−	−00−	−−	−−	−−	−−	−−	1○	−−	−−	−−	−−	−−	01	01	−−	−0−−	0−0−	−00−
s_{7_1}	−−−−	0−−0	−−	−−	−−	−−	−−	○1	−−	−−	−−	−−	01	−−	−−	01	0−0−	−0−−	−00−
s_{8_0}	1000	−−00	10	10	10	10	−−	−−	1○	01	01	01	−−	−−	01	01	−−00	00−−	1000
s_{8_1}	0−−−	−−−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−−	−−	−−	−−	−−	00−−	−−0−	−00−
s_{9_0}	0100	−−00	10	01	01	10	−−	−−	01	1○	01	01	−−	−−	01	01	00−−	−−00	1000
s_{9_1}	−0−−	−−−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−−	−−	−−	−−	−−0−	00−−	−00−
s_{10_0}	0010	00−−	01	10	01	01	−−	−−	01	01	1○	01	01	01	−−	−−	0001	0001	0001
s_{10_1}	−−0−	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{11_0}	0001	00−−	01	01	10	01	−−	−−	01	01	01	1○	01	01	−−	−−	0001	0001	0001
s_{11_1}	−−−0	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{12_0}	−−00	1000	10	−−	−−	10	10	10	−−	−−	01	01	1○	01	01	01	−0−0	0−0−	1000
s_{12_1}	−−−−	0−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−−	0−0−	−0−−	−00−
s_{13_0}	−−00	0100	10	−−	−−	10	01	01	−−	−−	01	01	01	1○	01	01	0−0−	−0−0	1000
s_{13_1}	−−−−	−0−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−	−0−−	0−0−	−00−
s_{14_0}	00−−	0010	01	−−	−−	01	10	01	01	01	−−	−−	01	01	1○	01	0001	0001	0001
s_{14_1}	−−−−	−−0−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−	−−−−	−−−−	−00−
s_{15_0}	00−−	0001	01	−−	−−	01	01	10	01	01	−−	−−	01	01	01	1○	0001	0001	0001
s_{15_1}	−−−−	−−−0	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	○1	−−−−	−−−−	−00−
s_{16_0}	1000	1000	10	10	10	10	10	10	10	01	01	01	10	01	01	01	1○○○	0001	1000
s_{16_1}	1000	0100	10	10	10	10	01	01	10	01	01	01	01	10	01	01	○1○○	0010	1000
s_{16_2}	0100	1000	10	01	01	10	10	10	01	10	01	01	10	01	01	01	○○1○	0100	1000
s_{16_3}	0−−−	0−−−	−−	−−	−−	−−	−−	−−	01	−−	−−	−−	01	−−	−−	−−	○○○1	−00−	−00−
s_{17_0}	0100	0100	10	01	01	10	01	01	01	10	01	01	01	10	01	01	0001	1○○○	1000
s_{17_1}	0100	1000	10	01	01	10	10	10	01	10	01	01	10	01	01	01	0010	○1○○	1000
s_{17_2}	1000	0100	10	10	10	10	01	01	10	01	01	01	01	10	01	01	0100	○○1○	1000
s_{17_3}	−0−−	−0−−	−−	−−	−−	−−	−−	−−	−−	01	−−	−−	−−	01	−−	−−	−00−	○○○1	−00−
s_{18_0}	−−00	−−00	10	−−	−−	10	−−	−−	−−	−−	01	01	−−	−−	01	01	−−−−	−−−−	1○○○
s_{18_1}	0000	0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0000	0000	○○○○
s_{18_2}	0000	0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0000	0000	○○○○
s_{18_3}	00−−	00−−	01	−−	−−	01	−−	−−	01	01	−−	−−	01	01	−−	−−	0001	0001	○○○1

Figure 52: XOR/choice

After redundancy removal (figure 53), the 5 2-state cells $c_{2_2} - c_{6_6}$ can be mapped to propositional variables and a conjunction of DNF clauses can be directly derived from the conflict relations in $r_{0_{xy}}, r_{1_{xy}}, x = (0,1,2,3), y = (2,3,4,5,6)$.

P	−−−−	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{0_0}	1∘∘∘	−−00	10	10	10	−−	−−	10	01	01	01	−−	−−	01	01	−−00	00−−	1000
s_{0_1}	∘1∘∘	−−00	10	01	01	−−	−−	01	10	01	01	−−	−−	01	01	00−−	−−00	1000
s_{0_2}	∘∘1∘	00−−	01	10	01	−−	−−	01	01	10	01	01	01	−−	−−	0001	0001	0001
s_{0_3}	∘∘∘1	00−−	01	01	10	−−	−−	01	01	01	10	01	01	−−	−−	0001	0001	0001
s_{1_0}	−−00	1∘∘∘	10	−−	−−	10	10	−−	−−	01	01	10	01	01	01	−0−0	0−0−	1000
s_{1_1}	−−00	∘1∘∘	10	−−	−−	01	01	−−	−−	01	01	01	10	01	01	0−0−	−0−0	1000
s_{1_2}	00−−	∘∘1∘	01	−−	−−	10	01	01	01	−−	−−	01	01	10	01	0001	0001	0001
s_{1_3}	00−−	∘∘∘1	01	−−	−−	01	10	01	01	−−	−−	01	01	01	10	0001	0001	0001
s_{2_0}	−−00	−−00	1∘	−−	−−	−−	−−	−−	−−	01	01	−−	−−	01	01	−−−−	−−−−	1000
s_{2_1}	00−−	00−−	∘1	−−	−−	−−	−−	01	01	−−	−−	01	01	−−	−−	0001	0001	0001
s_{3_0}	−0−0	−−−−	−−	1∘	−−	−−	−−	−−	01	−−	01	−−	−−	−−	−−	−−0−	00−−	−00−
s_{3_1}	0−0−	−−−−	−−	∘1	−−	−−	−−	01	−−	01	−−	−−	−−	−−	−−	00−−	−−0−	−00−
s_{4_0}	−00−	−−−−	−−	−−	1∘	−−	−−	−−	01	01	−−	−−	−−	−−	−−	−−0−	00−−	−00−
s_{4_1}	0−−0	−−−−	−−	−−	∘1	−−	−−	01	−−	−−	01	−−	−−	−−	−−	00−−	−−0−	−00−
s_{5_0}	−−−−	−0−0	−−	−−	−−	1∘	−−	−−	−−	−−	−−	−−	01	−−	01	−0−−	0−0−	−00−
s_{5_1}	−−−−	0−0−	−−	−−	−−	∘1	−−	−−	−−	−−	−−	01	−−	01	−−	0−0−	−0−−	−00−
s_{6_0}	−−−−	−00−	−−	−−	−−	−−	1∘	−−	−−	−−	−−	−−	01	01	−−	−0−−	0−0−	−00−
s_{6_1}	−−−−	0−−0	−−	−−	−−	−−	∘1	−−	−−	−−	−−	01	−−	−−	01	0−0−	−0−−	−00−
s_{7_0}	1000	−−00	10	10	10	−−	−−	1∘	01	01	01	−−	−−	01	01	−−00	00−−	1000
s_{7_1}	0−−−	−−−−	−−	−−	−−	−−	−−	∘1	−−	−−	−−	−−	−−	−−	−−	00−−	−−0−	−00−
s_{8_0}	0100	−−00	10	01	01	−−	−−	01	1∘	01	01	−−	−−	01	01	00−−	−−00	1000
s_{8_1}	−0−−	−−−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−	−−	−−	−−	−−	−−0−	00−−	−00−
s_{9_0}	0010	00−−	01	10	01	−−	−−	01	01	1∘	01	01	01	−−	−−	0001	0001	0001
s_{9_1}	−−0−	−−−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{10_0}	0001	00−−	01	01	10	−−	−−	01	01	01	1∘	01	01	−−	−−	0001	0001	0001
s_{10_1}	−−−0	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−	−−	−−	−−−−	−−−−	−00−
s_{11_0}	−−00	1000	10	−−	−−	10	10	−−	−−	01	01	1∘	01	01	01	−0−0	0−0−	1000
s_{11_1}	−−−−	0−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−	−−	0−0−	−0−−	−00−
s_{12_0}	−−00	0100	10	−−	−−	01	01	−−	−−	01	01	01	1∘	01	01	0−0−	−0−0	1000
s_{12_1}	−−−−	−0−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−	−0−−	0−0−	−00−
s_{13_0}	00−−	0010	01	−−	−−	10	01	01	01	−−	−−	01	01	1∘	01	0001	0001	0001
s_{13_1}	−−−−	−−0−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−	−−−−	−−−−	−00−
s_{14_0}	00−−	0001	01	−−	−−	01	10	01	01	−−	−−	01	01	01	1∘	0001	0001	0001
s_{14_1}	−−−−	−−−0	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	∘1	−−−−	−−−−	−00−
s_{15_0}	1000	1000	10	10	10	10	10	10	01	01	01	10	01	01	01	1∘∘∘	0001	1000
s_{15_1}	1000	0100	10	10	10	01	01	10	01	01	01	01	10	01	01	∘1∘∘	0010	1000
s_{15_2}	0100	1000	10	01	01	10	10	01	10	01	01	10	01	01	01	∘∘1∘	0100	1000
s_{15_3}	0−−−	0−−−	−−	−−	−−	−−	−−	01	−−	−−	−−	01	−−	−−	−−	∘∘∘1	−00−	−00−
s_{16_0}	0100	0100	10	01	01	01	01	01	10	01	01	01	10	01	01	0001	1∘∘∘	1000
s_{16_1}	0100	1000	10	01	01	10	10	01	10	01	01	10	01	01	01	0010	∘1∘∘	1000
s_{16_2}	1000	0100	10	10	10	01	01	10	01	01	01	01	10	01	01	0100	∘∘1∘	1000
s_{16_3}	−0−−	−0−−	−−	−−	−−	−−	−−	−−	01	−−	−−	−−	01	−−	−−	−00−	∘∘∘1	−00−
s_{17_0}	−−00	−−00	10	−−	−−	−−	−−	−−	−−	01	01	−−	−−	01	01	−−−−	−−−−	1∘∘∘
s_{17_1}	0000	0000	00	00	00	00	00	00	00	00	00	00	00	00	00	0000	0000	∘∘∘∘
s_{17_2}	0000	0000	00	00	00	00	00	00	00	00	00	00	00	00	00	0000	0000	∘∘∘∘
s_{17_3}	00−−	00−−	01	−−	−−	−−	−−	01	01	−−	−−	01	01	−−	−−	0001	0001	∘∘∘1

Figure 53: XOR/choice

The resulting conjunction of DNF clauses:

$$
\begin{array}{lll}
(& (\;\; a \wedge \;\; b \wedge \;\; c) & \vee \\
 & (\;\; a \wedge \neg b \wedge \neg c) & \vee \\
 & (\neg a \wedge \;\; b \wedge \neg c) & \vee \\
 & (\neg a \wedge \neg b \wedge \;\; c) & \quad) \wedge \\
(& (\;\; a \wedge \;\; d \wedge \;\; e) & \vee \\
 & (\;\; a \wedge \neg d \wedge \neg e) & \vee \\
 & (\neg a \wedge \;\; d \wedge \neg e) & \vee \\
 & (\neg a \wedge \neg d \wedge \;\; e) & \quad)
\end{array}
$$

can be transformed to a CNF formula with the well-known equivalences of XOR logic[9.]:

$$
\begin{array}{ll}
(\;\; a \vee \;\; b \vee \;\; c) & \wedge \\
(\;\; a \vee \neg b \vee \neg c) & \wedge \\
(\neg a \vee \;\; b \vee \neg c) & \wedge \\
(\neg a \vee \neg b \vee \;\; c) & \wedge \\
(\;\; a \vee \;\; d \vee \;\; e) & \wedge \\
(\;\; a \vee \neg d \vee \neg e) & \wedge \\
(\neg a \vee \;\; d \vee \neg e) & \wedge \\
(\neg a \vee \neg d \vee \;\; e) &
\end{array}
$$

The CNF formula is more suitable for SAT-solvers with Gaussian elimination than the original formula in direct encoding.

Note that once this deduction rule is proved, it can be directly applied, without constructing the choice variables or merging cells.

9. I prefer multiplying out the DNF clauses, which does not involve a research for the proof, but only truth tables. For the life of me, I cannot seem to remember, where all of these tautologies can be found.

13.1 Substituting Gaussion Elimination for Determining Satisfiability

When the at-most-one clauses from the above direct encoding are omitted:

$$
\begin{array}{ll}
(\ \ a \vee \ \ b \vee \ \ c \vee \ \ d) & \wedge \\
(\ \ e \vee \ \ f \vee \ \ g \vee \ \ h) & \wedge \\
(\neg a \vee \neg g) \wedge (\neg a \vee \neg h) & \wedge \\
(\neg b \vee \neg g) \wedge (\neg b \vee \neg h) & \wedge \\
(\neg c \vee \neg e) \wedge (\neg c \vee \neg f) & \wedge \\
(\neg d \vee \neg e) \wedge (\neg d \vee \neg f) &
\end{array}
$$

the semantics of the problem change to “one or more” of the choices can be made. However, in structural logic, the problem presents the same relevant structure (see figure 54).

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −
s_{0_0}	1 ∘ ∘ ∘	− − 0 0	0 1	− −	− −	− −	1 0	− −	− −	− −	− −	− −	0 1	0 1
s_{0_1}	∘ 1 ∘ ∘	− − 0 0	1 0	0 1	− −	− −	0 1	1 0	− −	− −	− −	− −	0 1	0 1
s_{0_2}	∘ ∘ 1 ∘	0 0 − −	1 0	1 0	0 1	− −	0 1	0 1	1 0	− −	0 1	0 1	− −	− −
s_{0_3}	∘ ∘ ∘ 1	0 0 − −	1 0	1 0	1 0	0 1	0 1	0 1	0 1	1 0	0 1	0 1	− −	− −
s_{1_0}	− − 0 0	1 ∘ ∘ ∘	− −	− −	1 0	1 0	− −	− −	0 1	0 1	1 0	− −	− −	− −
s_{1_1}	− − 0 0	∘ 1 ∘ ∘	− −	− −	1 0	1 0	− −	− −	0 1	0 1	0 1	1 0	− −	− −
s_{1_2}	0 0 − −	∘ ∘ 1 ∘	1 0	1 0	− −	− −	0 1	0 1	− −	− −	0 1	0 1	1 0	− −
s_{1_3}	0 0 − −	∘ ∘ ∘ 1	1 0	1 0	− −	− −	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0
s_{2_0}	0 − − −	− − − −	1 ∘	− −	− −	− −	0 1	− −	− −	− −	− −	− −	− −	− −
s_{2_1}	1 0 0 0	− − 0 0	∘ 1	− −	− −	− −	1 0	− −	− −	− −	− −	− −	0 1	0 1
s_{3_0}	− 0 − −	− − − −	− −	1 ∘	− −	− −	− −	0 1	− −	− −	− −	− −	− −	− −
s_{3_1}	− − 0 0	− − 0 0	− −	∘ 1	− −	− −	− −	1 0	− −	− −	− −	− −	0 1	0 1
s_{4_0}	− − 0 −	− − − −	− −	− −	1 ∘	− −	− −	− −	0 1	− −	− −	− −	− −	− −
s_{4_1}	− − − 0	0 0 − −	− −	− −	∘ 1	− −	− −	− −	1 0	− −	0 1	0 1	− −	− −
s_{5_0}	− − − 0	− − − −	− −	− −	− −	1 ∘	− −	− −	− −	0 1	− −	− −	− −	− −
s_{5_1}	− − − −	0 0 − −	− −	− −	− −	∘ 1	− −	− −	− −	1 0	0 1	0 1	− −	− −
s_{6_0}	1 0 0 0	− − 0 0	0 1	− −	− −	− −	1 ∘	− −	− −	− −	− −	− −	0 1	0 1
s_{6_1}	0 − − −	− − − −	1 0	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −	− −
s_{7_0}	− − 0 0	− − 0 0	− −	0 1	− −	− −	− −	1 ∘	− −	− −	− −	− −	0 1	0 1
s_{7_1}	− 0 − −	− − − −	− −	1 0	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −	− −
s_{8_0}	− − − 0	0 0 − −	− −	− −	0 1	− −	− −	− −	1 ∘	− −	0 1	0 1	− −	− −
s_{8_1}	− − 0 −	− − − −	− −	− −	1 0	− −	− −	− −	∘ 1	− −	− −	− −	− −	− −
s_{9_0}	− − − −	0 0 − −	− −	− −	− −	0 1	− −	− −	− −	1 ∘	0 1	0 1	− −	− −
s_{9_1}	− − − 0	− − − −	− −	− −	− −	1 0	− −	− −	− −	∘ 1	− −	− −	− −	− −
s_{10_0}	− − 0 0	1 0 0 0	− −	− −	1 0	1 0	− −	− −	0 1	0 1	1 ∘	− −	− −	− −
s_{10_1}	− − − −	0 − − −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −	− −
s_{11_0}	− − 0 0	− − 0 0	− −	− −	1 0	1 0	− −	− −	0 1	0 1	− −	1 ∘	− −	− −
s_{11_1}	− − − −	− 0 − −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −	− −
s_{12_0}	0 0 − −	− − − 0	1 0	1 0	− −	− −	0 1	0 1	− −	− −	− −	− −	1 ∘	− −
s_{12_1}	− − − −	− − 0 −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1	− −
s_{13_0}	0 0 − −	− − − −	1 0	1 0	− −	− −	0 1	0 1	− −	− −	− −	− −	− −	1 ∘
s_{13_1}	− − − −	− − − 0	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	− −	∘ 1

Figure 54: XOR/choice

After redundancy removal and making the problem *strictly provable* by merging, it presents the possible solutions in row $c_{10_{10}}$ of figure 55.

P	− − − −	− − − −	− −	− −	− −	− −	− −	− −	− −	− −	− − − − − − − −
s_{0_0}	1 ○ ○ ○	− − 0 0	1 0	− −	− −	− −	− −	− −	0 1	0 1	− − 0 0 0 0 0 0
s_{0_1}	○ 1 ○ ○	− − 0 0	0 1	1 0	− −	− −	− −	− −	0 1	0 1	0 0 − − 0 0 0 0
s_{0_2}	○ ○ 1 ○	0 0 − −	0 1	0 1	1 0	− −	0 1	0 1	− −	− −	0 0 0 0 − − 0 0
s_{0_3}	○ ○ ○ 1	0 0 − −	0 1	0 1	0 1	1 0	0 1	0 1	− −	− −	0 0 0 0 0 0 − −
s_{1_0}	− − 0 0	1 ○ ○ ○	− −	− −	0 1	0 1	1 0	− −	− −	− −	− 0 − 0 0 0 0 0
s_{1_1}	− − 0 0	○ 1 ○ ○	− −	− −	0 1	0 1	0 1	1 0	− −	− −	0 − 0 − 0 0 0 0
s_{1_2}	0 0 − −	○ ○ 1 ○	0 1	0 1	− −	− −	0 1	0 1	1 0	− −	0 0 0 0 − 0 − 0
s_{1_3}	0 0 − −	○ ○ ○ 1	0 1	0 1	− −	− −	0 1	0 1	0 1	1 0	0 0 0 0 0 − 0 −
s_{2_0}	1 0 0 0	− − 0 0	1 ○	− −	− −	− −	− −	− −	0 1	0 1	− − 0 0 0 0 0 0
s_{2_1}	0 − − −	− − − −	○ 1	− −	− −	− −	− −	− −	− −	− −	0 0 − − − − − −
s_{3_0}	− − 0 0	− − 0 0	− −	1 ○	− −	− −	− −	− −	0 1	0 1	− − − − 0 0 0 0
s_{3_1}	− 0 − −	− − − −	− −	○ 1	− −	− −	− −	− −	− −	− −	− − 0 0 − − − −
s_{4_0}	− − − 0	0 0 − −	− −	− −	1 ○	− −	0 1	0 1	− −	− −	0 0 0 0 − − 0 0
s_{4_1}	− − 0 −	− − − −	− −	− −	○ 1	− −	− −	− −	− −	− −	− − − − 0 0 − −
s_{5_0}	− − − −	0 0 − −	− −	− −	− −	1 ○	0 1	0 1	− −	− −	0 0 0 0 − − − −
s_{5_1}	− − − 0	− − − −	− −	− −	− −	○ 1	− −	− −	− −	− −	− − − − − − 0 0
s_{6_0}	− − 0 0	1 0 0 0	− −	− −	0 1	0 1	1 ○	− −	− −	− −	− 0 − 0 0 0 0 0
s_{6_1}	− − − −	0 − − −	− −	− −	− −	− −	○ 1	− −	− −	− −	0 − 0 − − − − −
s_{7_0}	− − 0 0	− − 0 0	− −	− −	0 1	0 1	− −	1 ○	− −	− −	− − − − 0 0 0 0
s_{7_1}	− − − −	− 0 − −	− −	− −	− −	− −	− −	○ 1	− −	− −	− 0 − 0 − − − −
s_{8_0}	0 0 − −	− − − 0	0 1	0 1	− −	− −	− −	− −	1 ○	− −	0 0 0 0 − 0 − 0
s_{8_1}	− − − −	− − 0 −	− −	− −	− −	− −	− −	− −	○ 1	− −	− − − − 0 − 0 −
s_{9_0}	0 0 − −	− − − −	0 1	0 1	− −	− −	− −	− −	− −	1 ○	0 0 0 0 − − − −
s_{9_1}	− − − −	− − − 0	− −	− −	− −	− −	− −	− −	− −	○ 1	− − − − − 0 − 0
s_{10_0}	1 0 0 0	1 0 0 0	1 0	− −	0 1	0 1	1 0	− −	0 1	0 1	1 ○ ○ ○ ○ ○ ○ ○
s_{10_1}	1 0 0 0	0 1 0 0	1 0	− −	0 1	0 1	0 1	1 0	0 1	0 1	○ 1 ○ ○ ○ ○ ○ ○
s_{10_2}	0 1 0 0	1 0 0 0	0 1	1 0	0 1	0 1	1 0	− −	0 1	0 1	○ ○ 1 ○ ○ ○ ○ ○
s_{10_3}	0 1 0 0	0 1 0 0	0 1	1 0	0 1	0 1	0 1	1 0	0 1	0 1	○ ○ ○ 1 ○ ○ ○ ○
s_{10_4}	0 0 1 0	0 0 1 0	0 1	0 1	1 0	− −	0 1	0 1	1 0	− −	○ ○ ○ ○ 1 ○ ○ ○
s_{10_5}	0 0 1 0	0 0 0 1	0 1	0 1	1 0	− −	0 1	0 1	0 1	1 0	○ ○ ○ ○ ○ 1 ○ ○
s_{10_6}	0 0 0 1	0 0 1 0	0 1	0 1	0 1	1 0	0 1	0 1	1 0	− −	○ ○ ○ ○ ○ ○ 1 ○
s_{10_7}	0 0 0 1	0 0 0 1	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 0	○ ○ ○ ○ ○ ○ ○ 1

Figure 55: XOR/choice

When comparing the previous result with the *strictly provable* problem having the at-most-one conflicts in figure 56, it becomes clear, that the relevant decisions in cells c_{10_0} and c_{10_1} are necessarily equivalent. The set of possible solutions is only determined by the mapped conflict relationships of the original propositional variables which allow for more combinations in the case of "at least one" compared to the case of "at most one".

P	−−−−	−−−−	−−	−−	−−	−−	−−	−−	−−	−−	−−−−−−−−
s_{0_0}	1 ∘ ∘ ∘	−−0 0	1 0	0 1	0 1	0 1	−−	−−	0 1	0 1	−−0 0 0 0 0 0
s_{0_1}	∘ 1 ∘ ∘	−−0 0	0 1	1 0	0 1	0 1	−−	−−	0 1	0 1	0 0−−0 0 0 0
s_{0_2}	∘ ∘ 1 ∘	0 0−−	0 1	0 1	1 0	0 1	0 1	0 1	−−	−−	0 0 0 0−−0 0
s_{0_3}	∘ ∘ ∘ 1	0 0−−	0 1	0 1	0 1	1 0	0 1	0 1	−−	−−	0 0 0 0 0 0−−
s_{1_0}	−−0 0	1 ∘ ∘ ∘	−−	−−	0 1	0 1	1 0	0 1	0 1	0 1	−0−0 0 0 0 0
s_{1_1}	−−0 0	∘ 1 ∘ ∘	−−	−−	0 1	0 1	0 1	1 0	0 1	0 1	0−0−0 0 0 0
s_{1_2}	0 0−−	∘ ∘ 1 ∘	0 1	0 1	−−	−−	0 1	0 1	1 0	0 1	0 0 0 0−0−0
s_{1_3}	0 0−−	∘ ∘ ∘ 1	0 1	0 1	−−	−−	0 1	0 1	0 1	1 0	0 0 0 0 0−0−
s_{2_0}	1 0 0 0	−−0 0	1 ∘	0 1	0 1	0 1	−−	−−	0 1	0 1	−−0 0 0 0 0 0
s_{2_1}	0−−−	−−−−	∘ 1	−−	−−	−−	−−	−−	−−	−−	0 0−−−−−−
s_{3_0}	0 1 0 0	−−0 0	0 1	1 ∘	0 1	0 1	−−	−−	0 1	0 1	0 0−−0 0 0 0
s_{3_1}	−0−−	−−−−	−−	∘ 1	−−	−−	−−	−−	−−	−−	−−0 0−−−−
s_{4_0}	0 0 1 0	0 0−−	0 1	0 1	1 ∘	0 1	0 1	0 1	−−	−−	0 0 0 0−−0 0
s_{4_1}	−−0−	−−−−	−−	−−	∘ 1	−−	−−	−−	−−	−−	−−−−0 0−−
s_{5_0}	0 0 0 1	0 0−−	0 1	0 1	0 1	1 ∘	0 1	0 1	−−	−−	0 0 0 0 0 0−−
s_{5_1}	−−−0	−−−−	−−	−−	−−	∘ 1	−−	−−	−−	−−	−−−−−−0 0
s_{6_0}	−−0 0	1 0 0 0	−−	−−	0 1	0 1	1 ∘	0 1	0 1	0 1	−0−0 0 0 0 0
s_{6_1}	−−−−	0−−−	−−	−−	−−	−−	∘ 1	−−	−−	−−	0−0−−−−−
s_{7_0}	−−0 0	0 1 0 0	−−	−−	0 1	0 1	0 1	1 ∘	0 1	0 1	0−0−0 0 0 0
s_{7_1}	−−−−	−0−−	−−	−−	−−	−−	−−	∘ 1	−−	−−	−0−0−−−−
s_{8_0}	0 0−−	0 0 1 0	0 1	0 1	−−	−−	0 1	0 1	1 ∘	0 1	0 0 0 0−0−0
s_{8_1}	−−−−	−−0−	−−	−−	−−	−−	−−	−−	∘ 1	−−	−−−−0−0−
s_{9_0}	0 0−−	0 0 0 1	0 1	0 1	−−	−−	0 1	0 1	0 1	1 ∘	0 0 0 0 0−0−
s_{9_1}	−−−−	−−−0	−−	−−	−−	−−	−−	−−	−−	∘ 1	−−−−−0−0
s_{10_0}	1 0 0 0	1 0 0 0	1 0	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 ∘ ∘ ∘ ∘ ∘ ∘ ∘
s_{10_1}	1 0 0 0	0 1 0 0	1 0	0 1	0 1	0 1	0 1	1 0	0 1	0 1	∘ 1 ∘ ∘ ∘ ∘ ∘ ∘
s_{10_2}	0 1 0 0	1 0 0 0	0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 1	∘ ∘ 1 ∘ ∘ ∘ ∘ ∘
s_{10_3}	0 1 0 0	0 1 0 0	0 1	1 0	0 1	0 1	0 1	1 0	0 1	0 1	∘ ∘ ∘ 1 ∘ ∘ ∘ ∘
s_{10_4}	0 0 1 0	0 0 1 0	0 1	0 1	1 0	0 1	0 1	0 1	1 0	0 1	∘ ∘ ∘ ∘ 1 ∘ ∘ ∘
s_{10_5}	0 0 1 0	0 0 0 1	0 1	0 1	1 0	0 1	0 1	0 1	0 1	1 0	∘ ∘ ∘ ∘ ∘ 1 ∘ ∘
s_{10_6}	0 0 0 1	0 0 1 0	0 1	0 1	0 1	1 0	0 1	0 1	1 0	0 1	∘ ∘ ∘ ∘ ∘ ∘ 1 ∘
s_{10_7}	0 0 0 1	0 0 0 1	0 1	0 1	0 1	1 0	0 1	0 1	0 1	1 0	∘ ∘ ∘ ∘ ∘ ∘ ∘ 1

Figure 56: XOR/choice

This is consistent with the fact that if an XORSAT problem is satisfiable, the corresponding SAT problem is also satisfiable.

This can be used in this case to solve the easier equisatisfiable XORSAT problem instead of the harder SAT problem, if the exact number of solutions does not matter.

start section hardness

Regarding hardness of problems, regular XORSAT problems have an interesting property, in that the number of decisions required by DPLL is exactly the same as the number of partitions that must be made to reduce the satoku matrix to a 2-SAT instance.

14. Constraint Satisfaction Example

This example shows some of the effects that encoding has on the hardness of a problem.

14.1 Direct Encoding Without At-Most-One Constraints

The example given encodes a sudoku block. Figure 57 shows the problem in direct encoding without at-most-one constraints. Although it is still not possible to assign a value twice within the block, a solving strategy may be forced to (re-) detect the missing constraints very late (figure 59).

P	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––
s_{0_0}	1○○○○○○○○	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––
s_{0_1}	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{0_2}	○○1○○○○○○	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––
s_{0_3}	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{0_4}	○○○○1○○○○	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––
s_{0_5}	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––
s_{0_6}	○○○○○○1○○	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––
s_{0_7}	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–
s_{0_8}	○○○○○○○○1	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0
s_{1_0}	0––––––––	1○○○○○○○○	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––
s_{1_1}	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{1_2}	––0––––––	○○1○○○○○○	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––
s_{1_3}	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{1_4}	––––0––––	○○○○1○○○○	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––
s_{1_5}	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––
s_{1_6}	––––––0––	○○○○○○1○○	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––
s_{1_7}	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–
s_{1_8}	––––––––0	○○○○○○○○1	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0
s_{2_0}	0––––––––	0––––––––	1○○○○○○○○	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––
s_{2_1}	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{2_2}	––0––––––	––0––––––	○○1○○○○○○	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––
s_{2_3}	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{2_4}	––––0––––	––––0––––	○○○○1○○○○	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––
s_{2_5}	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––
s_{2_6}	––––––0––	––––––0––	○○○○○○1○○	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––
s_{2_7}	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–
s_{2_8}	––––––––0	––––––––0	○○○○○○○○1	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0
s_{3_0}	0––––––––	0––––––––	0––––––––	1○○○○○○○○	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––
s_{3_1}	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{3_2}	––0––––––	––0––––––	––0––––––	○○1○○○○○○	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––
s_{3_3}	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{3_4}	––––0––––	––––0––––	––––0––––	○○○○1○○○○	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––
s_{3_5}	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––
s_{3_6}	––––––0––	––––––0––	––––––0––	○○○○○○1○○	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––
s_{3_7}	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–
s_{3_8}	––––––––0	––––––––0	––––––––0	○○○○○○○○1	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0
s_{4_0}	0––––––––	0––––––––	0––––––––	0––––––––	1○○○○○○○○	0––––––––	0––––––––	0––––––––	0––––––––
s_{4_1}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{4_2}	––0––––––	––0––––––	––0––––––	––0––––––	○○1○○○○○○	––0––––––	––0––––––	––0––––––	––0––––––
s_{4_3}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{4_4}	––––0––––	––––0––––	––––0––––	––––0––––	○○○○1○○○○	––––0––––	––––0––––	––––0––––	––––0––––
s_{4_5}	–––––0–––	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––	–––––0–––
s_{4_6}	––––––0––	––––––0––	––––––0––	––––––0––	○○○○○○1○○	––––––0––	––––––0––	––––––0––	––––––0––
s_{4_7}	–––––––0–	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–	–––––––0–
s_{4_8}	––––––––0	––––––––0	––––––––0	––––––––0	○○○○○○○○1	––––––––0	––––––––0	––––––––0	––––––––0
s_{5_0}	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	1○○○○○○○○	0––––––––	0––––––––	0––––––––
s_{5_1}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––	–0–––––––
s_{5_2}	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	○○1○○○○○○	––0––––––	––0––––––	––0––––––
s_{5_3}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––	–––0–––––
s_{5_4}	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	○○○○1○○○○	––––0––––	––––0––––	––––0––––
s_{5_5}	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––	–––––0–––
s_{5_6}	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	○○○○○○1○○	––––––0––	––––––0––	––––––0––
s_{5_7}	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–	–––––––0–
s_{5_8}	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	○○○○○○○○1	––––––––0	––––––––0	––––––––0
s_{6_0}	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	1○○○○○○○○	0––––––––	0––––––––
s_{6_1}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––	–0–––––––
s_{6_2}	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	○○1○○○○○○	––0––––––	––0––––––
s_{6_3}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––	–––0–––––
s_{6_4}	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	○○○○1○○○○	––––0––––	––––0––––
s_{6_5}	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––	–––––0–––
s_{6_6}	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	○○○○○○1○○	––––––0––	––––––0––
s_{6_7}	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–	–––––––0–
s_{6_8}	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	○○○○○○○○1	––––––––0	––––––––0
s_{7_0}	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	1○○○○○○○○	0––––––––
s_{7_1}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○	–0–––––––
s_{7_2}	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	○○1○○○○○○	––0––––––
s_{7_3}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○	–––0–––––
s_{7_4}	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	○○○○1○○○○	––––0––––
s_{7_5}	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○	–––––0–––
s_{7_6}	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	○○○○○○1○○	––––––0––
s_{7_7}	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○	–––––––0–
s_{7_8}	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	○○○○○○○○1	––––––––0
s_{8_0}	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	0––––––––	1○○○○○○○○
s_{8_1}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––	○1○○○○○○○
s_{8_2}	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	––0––––––	○○1○○○○○○
s_{8_3}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––0–––––	○○○1○○○○○
s_{8_4}	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	––––0––––	○○○○1○○○○
s_{8_5}	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	–––––0–––	○○○○○1○○○
s_{8_6}	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	––––––0––	○○○○○○1○○
s_{8_7}	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	–––––––0–	○○○○○○○1○
s_{8_8}	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	––––––––0	○○○○○○○○1

Figure 57: Sudoku block in direct encoding, implicit constraints omitted

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ◦ ◦	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	◦ 1 ◦	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −
s_{0_2}	◦ ◦ 1	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0
s_{1_0}	0 − −	1 ◦ ◦	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	− 0 −	◦ 1 ◦	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −
s_{1_2}	− − 0	◦ ◦ 1	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0
s_{2_0}	0 − −	0 − −	1 ◦ ◦	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 −	− 0 −	◦ 1 ◦	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −
s_{2_2}	− − 0	− − 0	◦ ◦ 1	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0
s_{3_0}	0 − −	0 − −	0 − −	1 ◦ ◦	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− 0 −	◦ 1 ◦	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −
s_{3_2}	− − 0	− − 0	− − 0	◦ ◦ 1	− − 0	− − 0	− − 0	− − 0	− − 0
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ◦ ◦	0 − −	0 − −	0 − −	0 − −
s_{4_1}	− 0 −	− 0 −	− 0 −	− 0 −	◦ 1 ◦	− 0 −	− 0 −	− 0 −	− 0 −
s_{4_2}	− − 0	− − 0	− − 0	− − 0	◦ ◦ 1	− − 0	− − 0	− − 0	− − 0
s_{5_0}	0 − −	0 − −	0 − −	0 − −	0 − −	1 ◦ ◦	0 − −	0 − −	0 − −
s_{5_1}	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	◦ 1 ◦	− 0 −	− 0 −	− 0 −
s_{5_2}	− − 0	− − 0	− − 0	− − 0	− − 0	◦ ◦ 1	− − 0	− − 0	− − 0
s_{6_0}	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	1 ◦ ◦	0 − −	0 − −
s_{6_1}	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	◦ 1 ◦	− 0 −	− 0 −
s_{6_2}	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	◦ ◦ 1	− − 0	− − 0
s_{7_0}	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	1 ◦ ◦	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	◦ 1 ◦	− 0 −
s_{7_2}	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	◦ ◦ 1	− − 0
s_{8_0}	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	0 − −	1 ◦ ◦
s_{8_1}	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	− 0 −	◦ 1 ◦
s_{8_2}	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	− − 0	◦ ◦ 1

Figure 58: Sudoku block in direct encoding, too many *impossible* states removed

P	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	− 0 −
s_{0_0}	◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{0_1}	◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{1_0}	0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{1_1}	0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{1_2}	0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_0}	0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_1}	0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_0}	0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_1}	0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_0}	0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_2}	0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0	0 0 0
s_{5_0}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0
s_{5_1}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0
s_{5_2}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0	0 0 0
s_{6_0}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0
s_{6_1}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0
s_{6_2}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0	0 0 0
s_{7_0}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0
s_{7_1}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0
s_{7_2}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦	0 0 0
s_{8_0}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦
s_{8_1}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦
s_{8_2}	0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	◦ ◦ ◦

Figure 59: Sudoku block in direct encoding, contradiction detected

A good solving strategy is to follow strict pair-wise requirements, which better manages the pitfalls of excluding too many states.

P	0 0 0 0 0 0 0 – –	– – – – – – – – –	– – – – – – – – –
s_{0_0}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_1}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_2}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_3}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_4}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_5}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_6}	○ ○ ○ ○ ○ ○ ○ ○ ○	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
s_{0_7}	○ ○ ○ ○ ○ ○ ○ 1 ○	– – – – – – – 0 –	– – – – – – – 0 –
s_{0_8}	○ ○ ○ ○ ○ ○ ○ ○ 1	– – – – – – – – 0	– – – – – – – – 0
s_{1_0}	0 0 0 0 0 0 0 – –	1 ○ ○ ○ ○ ○ ○ ○ ○	0 – – – – – – – –
s_{1_1}	0 0 0 0 0 0 0 – –	○ 1 ○ ○ ○ ○ ○ ○ ○	– 0 – – – – – – –
s_{1_2}	0 0 0 0 0 0 0 – –	○ ○ 1 ○ ○ ○ ○ ○ ○	– – 0 – – – – – –
s_{1_3}	0 0 0 0 0 0 0 – –	○ ○ ○ 1 ○ ○ ○ ○ ○	– – – 0 – – – – –
s_{1_4}	0 0 0 0 0 0 0 – –	○ ○ ○ ○ 1 ○ ○ ○ ○	– – – – 0 – – – –
s_{1_5}	0 0 0 0 0 0 0 – –	○ ○ ○ ○ ○ 1 ○ ○ ○	– – – – – 0 – – –
s_{1_6}	0 0 0 0 0 0 0 – –	○ ○ ○ ○ ○ ○ 1 ○ ○	– – – – – – 0 – –
s_{1_7}	0 0 0 0 0 0 0 0 1	○ ○ ○ ○ ○ ○ ○ 1 ○	– – – – – – – 0 0
s_{1_8}	0 0 0 0 0 0 0 1 0	○ ○ ○ ○ ○ ○ ○ ○ 1	– – – – – – – 0 0
s_{2_0}	0 0 0 0 0 0 0 – –	0 – – – – – – – –	1 ○ ○ ○ ○ ○ ○ ○ ○
s_{2_1}	0 0 0 0 0 0 0 – –	– 0 – – – – – – –	○ 1 ○ ○ ○ ○ ○ ○ ○
s_{2_2}	0 0 0 0 0 0 0 – –	– – 0 – – – – – –	○ ○ 1 ○ ○ ○ ○ ○ ○
s_{2_3}	0 0 0 0 0 0 0 – –	– – – 0 – – – – –	○ ○ ○ 1 ○ ○ ○ ○ ○
s_{2_4}	0 0 0 0 0 0 0 – –	– – – – 0 – – – –	○ ○ ○ ○ 1 ○ ○ ○ ○
s_{2_5}	0 0 0 0 0 0 0 – –	– – – – – 0 – – –	○ ○ ○ ○ ○ 1 ○ ○ ○
s_{2_6}	0 0 0 0 0 0 0 – –	– – – – – – 0 – –	○ ○ ○ ○ ○ ○ 1 ○ ○
s_{2_7}	0 0 0 0 0 0 0 0 1	– – – – – – – 0 0	○ ○ ○ ○ ○ ○ ○ 1 ○
s_{2_8}	0 0 0 0 0 0 0 1 0	– – – – – – – 0 0	○ ○ ○ ○ ○ ○ ○ ○ 1

(a) step 1

P	– –	– –	– –	– –	– –	– –	– –	– –	1 0
s_{0_0}	1 ○	0 1	– –	– –	– –	– –	– –	– –	1 0
s_{0_1}	○ 1	1 0	– –	– –	– –	– –	– –	– –	1 0
s_{1_0}	0 1	1 ○	– –	– –	– –	– –	– –	– –	1 0
s_{1_1}	1 0	○ 1	– –	– –	– –	– –	– –	– –	1 0
s_{2_0}	– –	– –	1 ○	0 1	– –	– –	– –	– –	1 0
s_{2_1}	– –	– –	○ 1	1 0	– –	– –	– –	– –	1 0
s_{3_0}	– –	– –	0 1	1 ○	– –	– –	– –	– –	1 0
s_{3_1}	– –	– –	1 0	○ 1	– –	– –	– –	– –	1 0
s_{4_0}	– –	– –	– –	– –	1 ○	0 1	– –	– –	1 0
s_{4_1}	– –	– –	– –	– –	○ 1	1 0	– –	– –	1 0
s_{5_0}	– –	– –	– –	– –	0 1	1 ○	– –	– –	1 0
s_{5_1}	– –	– –	– –	– –	1 0	○ 1	– –	– –	1 0
s_{6_0}	– –	– –	– –	– –	– –	– –	1 ○	0 1	1 0
s_{6_1}	– –	– –	– –	– –	– –	– –	○ 1	1 0	1 0
s_{7_0}	– –	– –	– –	– –	– –	– –	0 1	1 ○	1 0
s_{7_1}	– –	– –	– –	– –	– –	– –	1 0	○ 1	1 0
s_{8_0}	– –	– –	– –	– –	– –	– –	– –	– –	1 ○
s_{8_1}	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	○ ○

(b) result

Figure 60: Sudoku block, strict 2-state reduction

The research as to what extent missing constraints (see section 14.2) can be reconstructed is still ongoing. E.g., for the sudoku example, the deduction of the missing constraints is very simple.

Constructing additional cells with states that are mutually exclusive, but cannot be forced to require each other, reveals the structure of the missing constraints as shown in section 14.2.

The extra state "or-none-of-the-above" can be eliminated by reasoning, that there are 9 unique states for 9 cells available, so none of them is optional.

14.2 Direct Encoding With All Constraints

Figure 61 shows an excerpt of the same sudoku block problem in direct encoding with all constraints fully specified. The constraints are sufficient for structural logic to detect and resolve naked/hidden singles/pairs/triples/quads immediately by consolidation alone.

P	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––	–––––––––
s_{0_0}	1∘∘∘∘∘∘∘∘	0––––––––	0––––––––	0––––––––	100000000	0––––––––	0––––––––	0––––––––
s_{0_1}	∘1∘∘∘∘∘∘∘	–0–––––––	–0–––––––	–0–––––––	0––––––––	100000000	0––––––––	0––––––––
s_{0_2}	∘∘1∘∘∘∘∘∘	––0––––––	––0––––––	––0––––––	0––––––––	0––––––––	100000000	0––––––––
s_{0_3}	∘∘∘1∘∘∘∘∘	–––0–––––	–––0–––––	–––0–––––	0––––––––	0––––––––	0––––––––	100000000
s_{0_4}	∘∘∘∘1∘∘∘∘	––––0––––	––––0––––	––––0––––	0––––––––	0––––––––	0––––––––	0––––––––
s_{0_5}	∘∘∘∘∘1∘∘∘	–––––0–––	–––––0–––	–––––0–––	0––––––––	0––––––––	0––––––––	0––––––––
s_{0_6}	∘∘∘∘∘∘1∘∘	––––––0––	––––––0––	––––––0––	0––––––––	0––––––––	0––––––––	0––––––––
s_{0_7}	∘∘∘∘∘∘∘1∘	–––––––0–	–––––––0–	–––––––0–	0––––––––	0––––––––	0––––––––	0––––––––
s_{0_8}	∘∘∘∘∘∘∘∘1	––––––––0	––––––––0	––––––––0	0––––––––	0––––––––	0––––––––	0––––––––
s_{1_0}	0––––––––	1∘∘∘∘∘∘∘∘	0––––––––	0––––––––	010000000	–0–––––––	–0–––––––	–0–––––––
s_{1_1}	–0–––––––	∘1∘∘∘∘∘∘∘	–0–––––––	–0–––––––	–0–––––––	010000000	–0–––––––	–0–––––––
s_{1_2}	––0––––––	∘∘1∘∘∘∘∘∘	––0––––––	––0––––––	–0–––––––	–0–––––––	010000000	–0–––––––
s_{1_3}	–––0–––––	∘∘∘1∘∘∘∘∘	–––0–––––	–––0–––––	–0–––––––	–0–––––––	–0–––––––	010000000
s_{1_4}	––––0––––	∘∘∘∘1∘∘∘∘	––––0––––	––––0––––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{1_5}	–––––0–––	∘∘∘∘∘1∘∘∘	–––––0–––	–––––0–––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{1_6}	––––––0––	∘∘∘∘∘∘1∘∘	––––––0––	––––––0––	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{1_7}	–––––––0–	∘∘∘∘∘∘∘1∘	–––––––0–	–––––––0–	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{1_8}	––––––––0	∘∘∘∘∘∘∘∘1	––––––––0	––––––––0	–0–––––––	–0–––––––	–0–––––––	–0–––––––
s_{2_0}	0––––––––	0––––––––	1∘∘∘∘∘∘∘∘	0––––––––	001000000	––0––––––	––0––––––	––0––––––
s_{2_1}	–0–––––––	–0–––––––	∘1∘∘∘∘∘∘∘	–0–––––––	––0––––––	001000000	––0––––––	––0––––––
s_{2_2}	––0––––––	––0––––––	∘∘1∘∘∘∘∘∘	––0––––––	––0––––––	––0––––––	001000000	––0––––––
s_{2_3}	–––0–––––	–––0–––––	∘∘∘1∘∘∘∘∘	–––0–––––	––0––––––	––0––––––	––0––––––	001000000
s_{2_4}	––––0––––	––––0––––	∘∘∘∘1∘∘∘∘	––––0––––	––0––––––	––0––––––	––0––––––	––0––––––
s_{2_5}	–––––0–––	–––––0–––	∘∘∘∘∘1∘∘∘	–––––0–––	––0––––––	––0––––––	––0––––––	––0––––––
s_{2_6}	––––––0––	––––––0––	∘∘∘∘∘∘1∘∘	––––––0––	––0––––––	––0––––––	––0––––––	––0––––––
s_{2_7}	–––––––0–	–––––––0–	∘∘∘∘∘∘∘1∘	–––––––0–	––0––––––	––0––––––	––0––––––	––0––––––
s_{2_8}	––––––––0	––––––––0	∘∘∘∘∘∘∘∘1	––––––––0	––0––––––	––0––––––	––0––––––	––0––––––
s_{3_0}	0––––––––	0––––––––	0––––––––	1∘∘∘∘∘∘∘∘	000100000	–––0–––––	–––0–––––	–––0–––––
s_{3_1}	–0–––––––	–0–––––––	–0–––––––	∘1∘∘∘∘∘∘∘	–––0–––––	000100000	–––0–––––	–––0–––––
s_{3_2}	––0––––––	––0––––––	––0––––––	∘∘1∘∘∘∘∘∘	–––0–––––	–––0–––––	000100000	–––0–––––
s_{3_3}	–––0–––––	–––0–––––	–––0–––––	∘∘∘1∘∘∘∘∘	–––0–––––	–––0–––––	–––0–––––	000100000
s_{3_4}	––––0––––	––––0––––	––––0––––	∘∘∘∘1∘∘∘∘	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{3_5}	–––––0–––	–––––0–––	–––––0–––	∘∘∘∘∘1∘∘∘	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{3_6}	––––––0––	––––––0––	––––––0––	∘∘∘∘∘∘1∘∘	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{3_7}	–––––––0–	–––––––0–	–––––––0–	∘∘∘∘∘∘∘1∘	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{3_8}	––––––––0	––––––––0	––––––––0	∘∘∘∘∘∘∘∘1	–––0–––––	–––0–––––	–––0–––––	–––0–––––
s_{4_0}	100000000	0––––––––	0––––––––	0––––––––	1∘∘∘∘∘∘∘∘	0––––––––	0––––––––	0––––––––
s_{4_1}	0––––––––	100000000	0––––––––	0––––––––	∘1∘∘∘∘∘∘∘	–0–––––––	–0–––––––	–0–––––––
s_{4_2}	0––––––––	0––––––––	100000000	0––––––––	∘∘1∘∘∘∘∘∘	––0––––––	––0––––––	––0––––––
s_{4_3}	0––––––––	0––––––––	0––––––––	100000000	∘∘∘1∘∘∘∘∘	–––0–––––	–––0–––––	–––0–––––
s_{4_4}	0––––––––	0––––––––	0––––––––	0––––––––	∘∘∘∘1∘∘∘∘	––––0––––	––––0––––	––––0––––
s_{4_5}	0––––––––	0––––––––	0––––––––	0––––––––	∘∘∘∘∘1∘∘∘	–––––0–––	–––––0–––	–––––0–––
s_{4_6}	0––––––––	0––––––––	0––––––––	0––––––––	∘∘∘∘∘∘1∘∘	––––––0––	––––––0––	––––––0––
s_{4_7}	0––––––––	0––––––––	0––––––––	0––––––––	∘∘∘∘∘∘∘1∘	–––––––0–	–––––––0–	–––––––0–
s_{4_8}	0––––––––	0––––––––	0––––––––	0––––––––	∘∘∘∘∘∘∘∘1	––––––––0	––––––––0	––––––––0
s_{5_0}	010000000	–0–––––––	–0–––––––	–0–––––––	0––––––––	1∘∘∘∘∘∘∘∘	0––––––––	0––––––––
s_{5_1}	–0–––––––	010000000	–0–––––––	–0–––––––	–0–––––––	∘1∘∘∘∘∘∘∘	–0–––––––	–0–––––––
s_{5_2}	–0–––––––	–0–––––––	010000000	–0–––––––	––0––––––	∘∘1∘∘∘∘∘∘	––0––––––	––0––––––
s_{5_3}	–0–––––––	–0–––––––	–0–––––––	010000000	–––0–––––	∘∘∘1∘∘∘∘∘	–––0–––––	–––0–––––
s_{5_4}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	––––0––––	∘∘∘∘1∘∘∘∘	––––0––––	––––0––––
s_{5_5}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–––––0–––	∘∘∘∘∘1∘∘∘	–––––0–––	–––––0–––
s_{5_6}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	––––––0––	∘∘∘∘∘∘1∘∘	––––––0––	––––––0––
s_{5_7}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	–––––––0–	∘∘∘∘∘∘∘1∘	–––––––0–	–––––––0–
s_{5_8}	–0–––––––	–0–––––––	–0–––––––	–0–––––––	––––––––0	∘∘∘∘∘∘∘∘1	––––––––0	––––––––0
s_{6_0}	001000000	––0––––––	––0––––––	––0––––––	0––––––––	0––––––––	1∘∘∘∘∘∘∘∘	0––––––––
s_{6_1}	––0––––––	001000000	––0––––––	––0––––––	–0–––––––	–0–––––––	∘1∘∘∘∘∘∘∘	–0–––––––
s_{6_2}	––0––––––	––0––––––	001000000	––0––––––	––0––––––	––0––––––	∘∘1∘∘∘∘∘∘	––0––––––
s_{6_3}	––0––––––	––0––––––	––0––––––	001000000	–––0–––––	–––0–––––	∘∘∘1∘∘∘∘∘	–––0–––––
s_{6_4}	––0––––––	––0––––––	––0––––––	––0––––––	––––0––––	––––0––––	∘∘∘∘1∘∘∘∘	––––0––––
s_{6_5}	––0––––––	––0––––––	––0––––––	––0––––––	–––––0–––	–––––0–––	∘∘∘∘∘1∘∘∘	–––––0–––
s_{6_6}	––0––––––	––0––––––	––0––––––	––0––––––	––––––0––	––––––0––	∘∘∘∘∘∘1∘∘	––––––0––
s_{6_7}	––0––––––	––0––––––	––0––––––	––0––––––	–––––––0–	–––––––0–	∘∘∘∘∘∘∘1∘	–––––––0–
s_{6_8}	––0––––––	––0––––––	––0––––––	––0––––––	––––––––0	––––––––0	∘∘∘∘∘∘∘∘1	––––––––0
s_{7_0}	000100000	–––0–––––	–––0–––––	–––0–––––	0––––––––	0––––––––	0––––––––	1∘∘∘∘∘∘∘∘
s_{7_1}	–––0–––––	000100000	–––0–––––	–––0–––––	–0–––––––	–0–––––––	–0–––––––	∘1∘∘∘∘∘∘∘
s_{7_2}	–––0–––––	–––0–––––	000100000	–––0–––––	––0––––––	––0––––––	––0––––––	∘∘1∘∘∘∘∘∘
s_{7_3}	–––0–––––	–––0–––––	–––0–––––	000100000	–––0–––––	–––0–––––	–––0–––––	∘∘∘1∘∘∘∘∘
s_{7_4}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	––––0––––	––––0––––	––––0––––	∘∘∘∘1∘∘∘∘
s_{7_5}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––––0–––	–––––0–––	–––––0–––	∘∘∘∘∘1∘∘∘
s_{7_6}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	––––––0––	––––––0––	––––––0––	∘∘∘∘∘∘1∘∘
s_{7_7}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	–––––––0–	–––––––0–	–––––––0–	∘∘∘∘∘∘∘1∘
s_{7_8}	–––0–––––	–––0–––––	–––0–––––	–––0–––––	––––––––0	––––––––0	––––––––0	∘∘∘∘∘∘∘∘1

Figure 61: Sudoku block in enhanced encoding (excerpt)

Figure 62 shows the satoku matrix just before a hidden pair is generated in cells c_{2_2}, c_{3_3} by removing 2 states from each of the other cells.

P	−−−−−−−	−−−−−−−−	−−−−−−−−−	−−−−−−−−−	−−−−−−−−−	−−−−−−−−−	−−−−−−−−−	−−−−−−−−−
s_{0_0}	1○○○○○○	−0−−−−−−	−−0−−−−−−	−−0−−−−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−
s_{0_1}	○1○○○○○	−−0−−−−−	−−−0−−−−−	−−−0−−−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−
s_{0_2}	○○1○○○○	−−−0−−−−	−−−−0−−−−	−−−−0−−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−
s_{0_3}	○○○1○○○	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	000001000	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−
s_{0_4}	○○○○1○○	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−−−−−0−−−	000001000	−−−−−0−−−	−−−−−0−−−
s_{0_5}	○○○○○1○	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−−−0−−−	−−−−−0−−−	000001000	−−−−−0−−−
s_{0_6}	○○○○○○1	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−	000001000
s_{1_0}	−−−−−−−	1○○○○○○○	−0−−−−−−−	−0−−−−−−−	−−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−−−−−0−
s_{1_1}	0−−−−−−	○1○○○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_2}	−0−−−−−	○○1○○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_3}	−−0−−−−	○○○1○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_4}	−−−0−−−	○○○○1○○○	−−0000000	−−0000000	000000100	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_5}	−−−−0−−	○○○○○1○○	−−0000000	−−0000000	−−−−−−000	000000100	−−−−−−000	−−−−−−000
s_{1_6}	−−−−−0−	○○○○○○1○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	000000100	−−−−−−000
s_{1_7}	−−−−−−0	○○○○○○○1	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	000000100
s_{2_0}	−−−−−−−	−−−−−−−−	1○○○○○○○○	0−−−−−−−−	−−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−−−−−0−
s_{2_1}	−−−−−−−	0−−−−−−−	○1○○○○○○○	100000000	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{2_2}	0−−−−−−	10000000	○○1○○○○○○	100000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{2_3}	−0−−−−−	10000000	○○○1○○○○○	100000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{2_4}	−−0−−−−	10000000	○○○○1○○○○	100000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{2_5}	−−−0−−−	10000000	○○○○○1○○○	100000000	000000010	−−−−−−000	−−−−−−000	−−−−−−000
s_{2_6}	−−−−0−−	10000000	○○○○○○1○○	100000000	−−−−−−000	000000010	−−−−−−000	−−−−−−000
s_{2_7}	−−−−−0−	10000000	○○○○○○○1○	100000000	−−−−−−000	−−−−−−000	000000010	−−−−−−000
s_{2_8}	−−−−−−0	10000000	○○○○○○○○1	100000000	−−−−−−000	−−−−−−000	−−−−−−000	000000010
s_{3_0}	−−−−−−−	−−−−−−−−	0−−−−−−−−	1○○○○○○○○	−−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−−−−−−−0
s_{3_1}	−−−−−−−	0−−−−−−−	100000000	○1○○○○○○○	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{3_2}	0−−−−−−	10000000	100000000	○○1○○○○○○	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{3_3}	−0−−−−−	10000000	100000000	○○○1○○○○○	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{3_4}	−−0−−−−	10000000	100000000	○○○○1○○○○	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{3_5}	−−−0−−−	10000000	100000000	○○○○○1○○○	000000001	−−−−−−000	−−−−−−000	−−−−−−000
s_{3_6}	−−−−0−−	10000000	100000000	○○○○○○1○○	−−−−−−000	000000001	−−−−−−000	−−−−−−000
s_{3_7}	−−−−−0−	10000000	100000000	○○○○○○○1○	−−−−−−000	−−−−−−000	000000001	−−−−−−000
s_{3_8}	−−−−−−0	10000000	100000000	○○○○○○○○1	−−−−−−000	−−−−−−000	−−−−−−000	000000001
s_{4_0}	−−−0−−−	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	1○○○○○○○○	0−−−−−−−−	0−−−−−−−−	0−−−−−−−−
s_{4_1}	−−−0−−−	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○1○○○○○○○	−0−−−−−−−	−0−−−−−−−	−0−−−−−−−
s_{4_2}	−−−0−−−	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○○1○○○○○○	−−0−−−−−−	−−0−−−−−−	−−0−−−−−−
s_{4_3}	−−−0−−−	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○○○1○○○○○	−−−0−−−−−	−−−0−−−−−	−−−0−−−−−
s_{4_4}	−−−0−−−	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○○○○1○○○○	−−−−0−−−−	−−−−0−−−−	−−−−0−−−−
s_{4_5}	0001000	−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○○○○○1○○○	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−
s_{4_6}	−−−0−−−	00001000	−−0000000	−−0000000	○○○○○○1○○	−−−−−−000	−−−−−−000	−−−−−−000
s_{4_7}	−−−0−−−	10000000	000001000	100000000	○○○○○○○1○	−−−−−−000	−−−−−−000	−−−−−−000
s_{4_8}	−−−0−−−	10000000	100000000	000001000	○○○○○○○○1	−−−−−−000	−−−−−−000	−−−−−−000
s_{5_0}	−−−−0−−	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	0−−−−−−−−	1○○○○○○○○	0−−−−−−−−	0−−−−−−−−
s_{5_1}	−−−−0−−	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−0−−−−−−−	○1○○○○○○○	−0−−−−−−−	−0−−−−−−−
s_{5_2}	−−−−0−−	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−−0−−−−−−	○○1○○○○○○	−−0−−−−−−	−−0−−−−−−
s_{5_3}	−−−−0−−	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−−−0−−−−−	○○○1○○○○○	−−−0−−−−−	−−−0−−−−−
s_{5_4}	−−−−0−−	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−−−−0−−−−	○○○○1○○○○	−−−−0−−−−	−−−−0−−−−
s_{5_5}	0000100	−−−−−0−−	−−−−−−0−−	−−−−−−0−−	−−−−−0−−−	○○○○○1○○○	−−−−−0−−−	−−−−−0−−−
s_{5_6}	−−−−0−−	00000100	−−0000000	−−0000000	−−−−−−000	○○○○○○1○○	−−−−−−000	−−−−−−000
s_{5_7}	−−−−0−−	10000000	000000100	100000000	−−−−−−000	○○○○○○○1○	−−−−−−000	−−−−−−000
s_{5_8}	−−−−0−−	10000000	100000000	000000100	−−−−−−000	○○○○○○○○1	−−−−−−000	−−−−−−000
s_{6_0}	−−−−−0−	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	0−−−−−−−−	0−−−−−−−−	1○○○○○○○○	0−−−−−−−−
s_{6_1}	−−−−−0−	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−0−−−−−−−	−0−−−−−−−	○1○○○○○○○	−0−−−−−−−
s_{6_2}	−−−−−0−	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−0−−−−−−	−−0−−−−−−	○○1○○○○○○	−−0−−−−−−
s_{6_3}	−−−−−0−	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−0−−−−−	−−−0−−−−−	○○○1○○○○○	−−−0−−−−−
s_{6_4}	−−−−−0−	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−−0−−−−	−−−−0−−−−	○○○○1○○○○	−−−−0−−−−
s_{6_5}	0000010	−−−−−−0−	−−−−−−−0−	−−−−−−−0−	−−−−−0−−−	−−−−−0−−−	○○○○○1○○○	−−−−−0−−−
s_{6_6}	−−−−−0−	00000010	−−0000000	−−0000000	−−−−−−000	−−−−−−000	○○○○○○1○○	−−−−−−000
s_{6_7}	−−−−−0−	10000000	000000010	100000000	−−−−−−000	−−−−−−000	○○○○○○○1○	−−−−−−000
s_{6_8}	−−−−−0−	10000000	100000000	000000010	−−−−−−000	−−−−−−000	○○○○○○○○1	−−−−−−000
s_{7_0}	−−−−−−0	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	0−−−−−−−−	0−−−−−−−−	0−−−−−−−−	1○○○○○○○○
s_{7_1}	−−−−−−0	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−0−−−−−−−	−0−−−−−−−	−0−−−−−−−	○1○○○○○○○
s_{7_2}	−−−−−−0	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−0−−−−−−	−−0−−−−−−	−−0−−−−−−	○○1○○○○○○
s_{7_3}	−−−−−−0	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−−0−−−−−	−−−0−−−−−	−−−0−−−−−	○○○1○○○○○
s_{7_4}	−−−−−−0	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−−−0−−−−	−−−−0−−−−	−−−−0−−−−	○○○○1○○○○
s_{7_5}	0000001	−−−−−−−0	−−−−−−−−0	−−−−−−−−0	−−−−−0−−−	−−−−−0−−−	−−−−−0−−−	○○○○○1○○○
s_{7_6}	−−−−−−0	00000001	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	○○○○○○1○○
s_{7_7}	−−−−−−0	10000000	000000001	100000000	−−−−−−000	−−−−−−000	−−−−−−000	○○○○○○○1○
s_{7_8}	−−−−−−0	10000000	100000000	000000001	−−−−−−000	−−−−−−000	−−−−−−000	○○○○○○○○1

.

Figure 62: Sudoku block in enhanced encoding, 1 step before hidden pair detection in c_{2_2}, c_{3_3}

Figure 63 shows the satoku matrix with a hidden pair in cells c_{2_2}, c_{3_3}.

P	−−−−−−−	−−−−−−−	−−0000000	−−0000000	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{0_0}	1○○○○○○	0−−−−−−	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	−−−−−0−00	−−−−−0−00
s_{0_1}	○1○○○○○	−0−−−−−	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	−−−−−0−00	−−−−−0−00
s_{0_2}	○○1○○○○	−−0−−−−	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	−−−−−0−00	−−−−−0−00
s_{0_3}	○○○1○○○	−−−0−−−	−−0000000	−−0000000	000001000	−−−−−0−00	−−−−−0−00	−−−−−0−00
s_{0_4}	○○○○1○○	−−−−0−−	−−0000000	−−0000000	−−−−−0−00	000001000	−−−−−0−00	−−−−−0−00
s_{0_5}	○○○○○1○	−−−−−0−	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	000001000	−−−−−0−00
s_{0_6}	○○○○○○1	−−−−−−0	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	−−−−−0−00	000001000
s_{1_0}	0−−−−−−	1○○○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_1}	−0−−−−−	○1○○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_2}	−−0−−−−	○○1○○○○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_3}	−−−0−−−	○○○1○○○	−−0000000	−−0000000	000000100	−−−−−−000	−−−−−−000	−−−−−−000
s_{1_4}	−−−−0−−	○○○○1○○	−−0000000	−−0000000	−−−−−−000	000000100	−−−−−−000	−−−−−−000
s_{1_5}	−−−−−0−	○○○○○1○	−−0000000	−−0000000	−−−−−−000	−−−−−−000	000000100	−−−−−−000
s_{1_6}	−−−−−−0	○○○○○○1	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	000000100
s_{2_0}	−−−−−−−	−−−−−−−	1○○○○○○○○	010000000	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{2_1}	−−−−−−−	−−−−−−−	○1○○○○○○○	100000000	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{2_2}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_3}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_4}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_5}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_6}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_7}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{2_8}	0000000	0000000	○○○○○○○○○	000000000	000000000	000000000	000000000	000000000
s_{3_0}	−−−−−−−	−−−−−−−	010000000	1○○○○○○○○	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{3_1}	−−−−−−−	−−−−−−−	100000000	○1○○○○○○○	−−−−−−−00	−−−−−−−00	−−−−−−−00	−−−−−−−00
s_{3_2}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_3}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_4}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_5}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_6}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_7}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{3_8}	0000000	0000000	000000000	○○○○○○○○○	000000000	000000000	000000000	000000000
s_{4_0}	−−−0−−−	−−−0−−−	−−0000000	−−0000000	1○○○○○○○○	0−−−−−−00	0−−−−−−00	0−−−−−−00
s_{4_1}	−−−0−−−	−−−0−−−	−−0000000	−−0000000	○1○○○○○○○	−0−−−−−00	−0−−−−−00	−0−−−−−00
s_{4_2}	−−−0−−−	−−−0−−−	−−0000000	−−0000000	○○1○○○○○○	−−0−−−−00	−−0−−−−00	−−0−−−−00
s_{4_3}	−−−0−−−	−−−0−−−	−−0000000	−−0000000	○○○1○○○○○	−−−0−−−00	−−−0−−−00	−−−0−−−00
s_{4_4}	−−−0−−−	−−−0−−−	−−0000000	−−0000000	○○○○1○○○○	−−−−0−−00	−−−−0−−00	−−−−0−−00
s_{4_5}	0001000	−−−0−−−	−−0000000	−−0000000	○○○○○1○○○	−−−−−0−00	−−−−−0−00	−−−−−0−00
s_{4_6}	−−−0−−−	0001000	−−0000000	−−0000000	○○○○○○1○○	−−−−−−000	−−−−−−000	−−−−−−000
s_{4_7}	0000000	0000000	000000000	000000000	○○○○○○○○○	000000000	000000000	000000000
s_{4_8}	0000000	0000000	000000000	000000000	○○○○○○○○○	000000000	000000000	000000000
s_{5_0}	−−−−0−−	−−−−0−−	−−0000000	−−0000000	0−−−−−−00	1○○○○○○○○	0−−−−−−00	0−−−−−−00
s_{5_1}	−−−−0−−	−−−−0−−	−−0000000	−−0000000	−0−−−−−00	○1○○○○○○○	−0−−−−−00	−0−−−−−00
s_{5_2}	−−−−0−−	−−−−0−−	−−0000000	−−0000000	−−0−−−−00	○○1○○○○○○	−−0−−−−00	−−0−−−−00
s_{5_3}	−−−−0−−	−−−−0−−	−−0000000	−−0000000	−−−0−−−00	○○○1○○○○○	−−−0−−−00	−−−0−−−00
s_{5_4}	−−−−0−−	−−−−0−−	−−0000000	−−0000000	−−−−0−−00	○○○○1○○○○	−−−−0−−00	−−−−0−−00
s_{5_5}	0000100	−−−−0−−	−−0000000	−−0000000	−−−−−0−00	○○○○○1○○○	−−−−−0−00	−−−−−0−00
s_{5_6}	−−−−0−−	0000100	−−0000000	−−0000000	−−−−−−000	○○○○○○1○○	−−−−−−000	−−−−−−000
s_{5_7}	0000000	0000000	000000000	000000000	000000000	○○○○○○○○○	000000000	000000000
s_{5_8}	0000000	0000000	000000000	000000000	000000000	○○○○○○○○○	000000000	000000000
s_{6_0}	−−−−−0−	−−−−−0−	−−0000000	−−0000000	0−−−−−−00	0−−−−−−00	1○○○○○○○○	0−−−−−−00
s_{6_1}	−−−−−0−	−−−−−0−	−−0000000	−−0000000	−0−−−−−00	−0−−−−−00	○1○○○○○○○	−0−−−−−00
s_{6_2}	−−−−−0−	−−−−−0−	−−0000000	−−0000000	−−0−−−−00	−−0−−−−00	○○1○○○○○○	−−0−−−−00
s_{6_3}	−−−−−0−	−−−−−0−	−−0000000	−−0000000	−−−0−−−00	−−−0−−−00	○○○1○○○○○	−−−0−−−00
s_{6_4}	−−−−−0−	−−−−−0−	−−0000000	−−0000000	−−−−0−−00	−−−−0−−00	○○○○1○○○○	−−−−0−−00
s_{6_5}	0000010	−−−−−0−	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	○○○○○1○○○	−−−−−0−00
s_{6_6}	−−−−−0−	0000010	−−0000000	−−0000000	−−−−−−000	−−−−−−000	○○○○○○1○○	−−−−−−000
s_{6_7}	0000000	0000000	000000000	000000000	000000000	000000000	○○○○○○○○○	000000000
s_{6_8}	0000000	0000000	000000000	000000000	000000000	000000000	○○○○○○○○○	000000000
s_{7_0}	−−−−−−0	−−−−−−0	−−0000000	−−0000000	0−−−−−−00	0−−−−−−00	0−−−−−−00	1○○○○○○○○
s_{7_1}	−−−−−−0	−−−−−−0	−−0000000	−−0000000	−0−−−−−00	−0−−−−−00	−0−−−−−00	○1○○○○○○○
s_{7_2}	−−−−−−0	−−−−−−0	−−0000000	−−0000000	−−0−−−−00	−−0−−−−00	−−0−−−−00	○○1○○○○○○
s_{7_3}	−−−−−−0	−−−−−−0	−−0000000	−−0000000	−−−0−−−00	−−−0−−−00	−−−0−−−00	○○○1○○○○○
s_{7_4}	−−−−−−0	−−−−−−0	−−0000000	−−0000000	−−−−0−−00	−−−−0−−00	−−−−0−−00	○○○○1○○○○
s_{7_5}	0000001	−−−−−−0	−−0000000	−−0000000	−−−−−0−00	−−−−−0−00	−−−−−0−00	○○○○○1○○○
s_{7_6}	−−−−−−0	0000001	−−0000000	−−0000000	−−−−−−000	−−−−−−000	−−−−−−000	○○○○○○1○○
s_{7_7}	0000000	0000000	000000000	000000000	000000000	000000000	000000000	○○○○○○○○○
s_{7_8}	0000000	0000000	000000000	000000000	000000000	000000000	000000000	○○○○○○○○○

Figure 63: Sudoku block in enhanced encoding, hidden pair detected in c_{2_2}, c_{3_3}

Rintanen writes[RINTANEN]: “The most primitive non-trivial invariant has the form $\neg a \vee \neg b$, saying that a and b cannot be true simultaneously. Adding this type of constraints in the SAT encodings of planning is often critical for its efficiency.”

The sudoku example shows, that this is not only often, but always.

15. Constructing Variable Sets

E.g. exp/genalea-40-171-force-conflict/genalea-40-171-350409699.x.v-004-opt.fca

16. Identifying Relevant Problem

Merge dense cells.

For sparse cells: construct conflict cell and move to non-essential section.

If result is in problem, delete from problem.

E.g.,

```
exp/x1n/x1n3-1in1.x.v-002.mtx
exp/x1n/x1n3-2in2.x.v-002.mtx
exp/x1n/x1n3-3in3-spread.x.v-003.mtx
exp/x1n/x1n3-3in3.x.v-005.mtx
```

See also:

exp/ex-bipartite-3cage-resolve-cfl/ex-bipartite-3cage-resolve-cfl-002.mtx

17. Multi-value Logic Loops

Strategy:

- construct conflict cells to determine core problem
- separate “or-none” state
- fully merge non-excluded states referenced by “or-none” state
- 2-state partition

The hardest problems are multi-value problems as introduced with the sudoku example, that are not fully constrained (ambiguous) and contain unsatisfiable loops.

Example: hgen2-v450-s41511877.shuffled-as.sat03-1682.used-as.sat04-816.cnf from SAT-competition 2003.

Quote from the generator:

> generate 2-clauses expressing $\sum_{i \in I} x_i \leq 1$ for disjoint I's $1..DDD, DDD+1..2*DDD$, ...; then generate random clauses of length L (defined below) expressing $\sum i \in J \neg x_i \geq 1$. Currenly, DDD is set to 5. L is defined so that the latter clauses express $\sum_{all} x_i \geq M+1$ while the former give $\sum ... \leq M$. Note that for certain DDD and L that would result in PHP.

It is still fun to watch a CDCL solver running into all of the terminal impossible states without ever learning anything substantial.

Note: As soon as the number of irredundant clauses reads 54 instead of 1575, I will know, that somebody has discovered structural logic.

```
c Lingeling SAT Solver
c
c Version azd 0d997521ad2e7d4e94f5d74a4665455b91309b62
c
c Copyright (C) 2010-2014 Armin Biere JKU Linz Austria.
c All rights reserved.
c
c released Wed Oct 29 15:03:13 CET 2014
...
c reading input file hgen2-v450-s41511877.shuffled-as.sat03-1682.used-as.sat04-816.cn
c no embedded options
c found 'p cnf 450 1575' header
c read 450 variables, 1575 clauses, 4725 literals in 0.00 seconds
```

```
c
c  seconds         irredundant          redundant clauses agility   height
c          variables clauses conflicts large ternary binary    glue         MB
c
c S    0.0     450     1575         0       0      0      0   0  0.0   0.0   0
c S    1.0     450     1575     37712    3898      0      0  29 20.5  34.9   2
c S    2.0     450     1575     69502    5842      0      0  30 20.4  34.8   2
c S    5.0     450     1575    154369    7724      0      0  29 20.1  34.3   2
c S   10.0     450     1575    278802   14700      0      0  30 20.3  34.7   4
c S   20.0     450     1575    444808   26735      0      0  29 20.5  34.9   4
c S   30.0     450     1575    667548   17164      0      0  29 20.8  35.3   3
c S   40.0     450     1575    843123   22394      0      0  29 20.8  35.3   3
c S   50.0     450     1575    976735   15355      0      0  29 20.8  35.4   3
c S   60.0     450     1575   1125056   29763      0      0  29 20.8  35.3   6
c S  120.0     450     1575   1742517   50713      0      0  28 20.8  35.3   8
c S  180.0     450     1575   2453422   27218      0      0  29 20.6  34.9   4
c S  240.1     450     1575   3325062   44995      0      0  28 21.0  35.3   8
c S  300.0     450     1575   3879415   59214      0      0  29 21.2  35.3  10
c S  600.1     450     1575   6265267  103649      0      0  28 21.4  35.2  21
c S  900.2     450     1575   7770967  120941      0      0  28 21.6  35.3  24
c S 1800.0     450     1575  14553134   77217      0      0  29 22.0  36.3  11
c S 2700.1     450     1575  18670695   68006      0      0  29 22.2  36.6   9
c S 3600.1     450     1575  22700104  149509      0      0  29 22.4  37.1  28
c S 4500.2     450     1575  25304933  173846      0      0  28 22.6  37.3  29
c S 5400.1     450     1575  27526480  223181      0      0  28 22.6  37.4  42
c S 6300.4     450     1575  29565854  194097      0      0  29 22.7  37.5  32
c S 7200.4     450     1575  31400601  231189      0      0  29 22.7  37.5  40
c
c  seconds         irredundant          redundant clauses agility   height
c          variables clauses conflicts large ternary binary    glue         MB
c
c S 10800.1     450     1575  40396499   79772      0      0  29 22.5  37.1  11
c S 14400.0     450     1575  55618031   98857      0      0  29 22.7  37.4  13
c S 18000.3     450     1575  65189102  189649      0      0  28 22.7  37.3  27
c S 21600.8     450     1575  71635072  314341      0      0  28 22.6  37.1  58
c S 25200.4     450     1575  80144153  207753      0      0  28 22.6  37.0  33
c S 28800.7     450     1575  86764505  283561      0      0  28 22.6  36.8  49
c S 32400.2     450     1575  92031841  304383      0      0  28 22.5  36.7  51
...
```

18. Schaefer's Dichotomy Theorem

Schaefer's Dichotomy Theorem[9] (SDT) states:

> ... the problem $SAT(S)$ is viewed as a constraint satisfaction problem over the Boolean domain. In this area, it is standard to denote the set of relations by Γ and the decision problem defined by Γ as $CSP(\Gamma)$.
>
> An operation $f : D^m \to D$ is a polymorphism of a relation $R \subseteq D^k$ if, for any choice of m tuples $(t_{11}, \ldots, t_{1k}), \ldots, (t_{m1}, \ldots, t_{mk})$ from R, it holds that the tuple obtained from these m tuples by applying f coordinate-wise, i.e. $(f(t_{11}, \ldots, t_{m1}), \ldots, f(t_{1k}, \ldots, t_{mk}))$, is in R. That is, an operation f is a polymorphism of R if R is closed under f: applying f to any tuples in R yields another tuple inside R. A set of relations Γ is said to have a polymorphism f if every relation in Γ has f as a polymorphism.

The practical disadvantage of SDT is, that it fully depends on the encoding of a satisfiability problem. Adding one clause, $(a \vee b \vee c)$, to a problem Γ, which is otherwise decidable in polynomial time, makes Γ NP-complete, since there is no longer a polymorphism f for **every** relation in Γ.

SDT still holds, since P $\subseteq$ NP, but it is no longer "easy to check if any of the tractability conditions hold".

Translating an XORSAT problem Γ_X to a CNF problem Γ_C preserving satisfiability in the usual manner:

$$\begin{array}{ll}
\Gamma_X = & (a \oplus b \oplus c) \\
\Gamma_X \mapsto \Gamma_C : & \\
(\neg a \vee \neg b \vee c) & \wedge \\
(\neg a \vee b \vee \neg c) & \wedge \\
(a \vee \neg b \vee \neg c) & \wedge \\
(a \vee b \vee c) &
\end{array}$$

also makes Γ_X NP-complete for decision algorithms. This is the incentive for adding XOR-clause detection to CDCL SAT solvers. However, that is no remedy, since XOR detection again depends on the "proper" encoding.

When the structural decomposition is taken one step further by encoding Γ_C with direct encoding to Γ_{X1} preserving satisfiability (at-most-one clauses omitted):

$$
\begin{array}{l}
\Gamma_C \mapsto \Gamma_{X1}: \\
(\ s0 \vee \ s1 \vee \ s2) \quad \wedge \\
(\ t0 \vee \ t1 \vee \ t2) \quad \wedge \\
(\ u0 \vee \ u1 \vee \ u2) \quad \wedge \\
(\ v0 \vee \ v1 \vee \ v2) \quad \wedge \\
\ldots \\
(\neg s0 \vee \neg u0) \quad \wedge \\
(\neg s0 \vee \neg v0) \quad \wedge \\
(\neg s1 \vee \neg t1) \quad \wedge \\
(\neg s1 \vee \neg v1) \quad \wedge \\
(\neg s2 \vee \neg t2) \quad \wedge \\
(\neg s2 \vee \neg u2) \quad \wedge \\
(\neg t0 \vee \neg u0) \quad \wedge \\
(\neg t0 \vee \neg v0) \quad \wedge \\
(\neg t1 \vee \neg u1) \quad \wedge \\
(\neg t2 \vee \neg v2) \quad \wedge \\
(\neg u1 \vee \neg v1) \quad \wedge \\
(\neg u2 \vee \neg v2),
\end{array}
$$

XOR-clause detection based on CNF encoding fails and CDCL solvers indeed confirm SDT in that regard by taking up exponentially more time to determine unsatisfiabiliy.

What SDT does not explain is the following effect. Translate Γ_C to Γ_{CM}, by applying the tautology

$$(p \vee q \vee r) = ((p) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q \wedge r))$$

to each clause of Γ_C:

$$
\begin{array}{lll}
 & \Gamma_C \mapsto \Gamma_{CM}: & \\
(& (\neg a) \quad \vee & \\
 & (\ a \wedge \neg b) \quad \vee & \\
 & (\ a \wedge \ b \wedge \ c) &) \ \wedge \\
(& (\neg a) \quad \vee & \\
 & (\ a \wedge \ b) \quad \vee & \\
 & (\ a \wedge \neg b \wedge \neg c) &) \ \wedge \\
(& (\ a) \quad \vee & \\
 & (\neg a \wedge \neg b) \quad \vee & \\
 & (\neg a \wedge \ b \wedge \neg c) &) \ \wedge \\
(& (\ a) \quad \vee & \\
 & (\neg a \wedge \ b) \quad \vee & \\
 & (\neg a \wedge \neg b \wedge \ c) &).
\end{array}
$$

Translating Γ_{CM} to Γ_{X1M} with direct encoding, preserving satisfiability (at-most-one clauses omitted):

$$
\begin{array}{ll}
\Gamma_{CM} \mapsto \Gamma_{X1M}: & \\
(\ \ s0 \vee \ \ s1 \vee \ \ s2) & \wedge \\
(\ \ t0 \vee \ \ t1 \vee \ \ t2) & \wedge \\
(\ \ u0 \vee \ \ u1 \vee \ \ u2) & \wedge \\
(\ \ v0 \vee \ \ v1 \vee \ \ v2) & \wedge \\
\ldots & \\
(\neg s0 \vee \neg t1) & \wedge \\
(\neg s0 \vee \neg t2) & \wedge \\
(\neg s0 \vee \neg u0) & \wedge \\
(\neg s0 \vee \neg v0) & \wedge \\
(\neg s1 \vee \neg t0) & \wedge \\
(\neg s1 \vee \neg t1) & \wedge \\
(\neg s1 \vee \neg u1) & \wedge \\
(\neg s1 \vee \neg u2) & \wedge \\
(\neg s1 \vee \neg v1) & \wedge \\
(\neg s1 \vee \neg v2) & \wedge \\
(\neg s2 \vee \neg t0) & \wedge \\
(\neg s2 \vee \neg t2) & \wedge \\
(\neg s2 \vee \neg u1) & \wedge \\
(\neg s2 \vee \neg u2) & \wedge \\
(\neg s2 \vee \neg v1) & \wedge \\
(\neg s2 \vee \neg v2) & \wedge \\
(\neg t0 \vee \neg u0) & \wedge \\
(\neg t0 \vee \neg v0) & \wedge \\
(\neg t1 \vee \neg u1) & \wedge \\
(\neg t1 \vee \neg u2) & \wedge \\
(\neg t1 \vee \neg v1) & \wedge \\
(\neg t1 \vee \neg v2) & \wedge \\
(\neg t2 \vee \neg u1) & \wedge \\
(\neg t2 \vee \neg u2) & \wedge \\
(\neg t2 \vee \neg v1) & \wedge \\
(\neg t2 \vee \neg v2) & \wedge \\
(\neg u0 \vee \neg v1) & \wedge \\
(\neg u0 \vee \neg v2) & \wedge \\
(\neg u1 \vee \neg v0) & \wedge \\
(\neg u1 \vee \neg v1) & \wedge \\
(\neg u2 \vee \neg v0) & \wedge \\
(\neg u2 \vee \neg v2), &
\end{array}
$$

shows that there are significantly more conflict clauses than for Γ_{X1}. Γ_{X1M} becomes significantly "easier" for CDCL solvers than Γ_{X1}.

The encoding effects can be easily verfied by applying the encodings to any random CNF problem (however, not CSP problems). The most significant effects can be seen with unsatisfiable instances of 3-XORSAT and SAT solvers with Gausssian elimintation, e.g., mod2-3cage-unsat-9-10.cnf from http://www.cs.helsinki.fi/u/mjarvisa/benchmarks/:

	Lingeling Version ats 57807c8f410a9e676816984a0ad0c410e485bcae
Γ_C	c found 'p cnf 87 232' header
	c 33531 decisions, 197182.0 decisions/sec
	c 0.2 seconds, 1.9 MB
Γ_{X1}	c found 'p cnf 696 2320' header
	c S 36000.1 464 1392 229422256 242874 2076 131 32 20.0 26.5 96
	interrrupted after 10 hours
Γ_{X1M}	c found 'p cnf 696 6688' header
	c 8647016 decisions, 68950.9 decisions/sec
	c 125.4 seconds, 16.3 MB

Since all encodings are derived from the same problem Γ_X, it appears strange, that their "hardness" for decision algorithms varies to such a great extent.

Similar observations were made by:

Formalizing Dangerous SAT Encodings
http://dx.doi.org/10.1007/978-3-540-72788-0_18
Comes closest to giving an explanation for the encoding sensitivity of DPLL solvers.

Efficient CNF Encoding of Boolean Cardinality Constraints
http://dx.doi.org/10.1007/978-3-540-45193-8_8
Demonstrates that problem encoding is essential for DPLL solvers.

Bai, Yun, and Yan Zhang. "Program Completion as Constraint Satisfaction: Tight Logic Programs Revisited."
Elaborates on the fact, that SDT is impractical to determine tractability.

The Order Encoding: From Tractable CSP to Tractable SAT
http://dx.doi.org/10.1007/978-3-642-21581-0_34
Emphasizes the importance and especially limits of using SDT to evaluate an encoding.

But none of them gives an explanation other than "arc-consistency" for DPLL solvers. This effect is found in structural logic as "missing at-least-one constraints" (section 20).

For par16-2-c.cnf:

Γ_C:

```
c found 'p cnf 349 1392' header
c
c    0.028  43% simplifying
c    0.037  57% search
c ====================================
c    0.065 100% all
c
c 2622 decisions, 40378.8 decisions/sec
c 2412 conflicts, 37144.8 conflicts/sec
c 127364 propagations, 2.0 megaprops/sec
c 0.1 seconds, 0.3 MB
```

Γ_{X1}:

```
c found 'p cnf 4054 72899' header
c
c    3.228  35% simplifying
c    6.037  65% search
c ====================================
c    9.266 100% all
c
c 164466 decisions, 17749.2 decisions/sec
c 68105 conflicts, 7349.9 conflicts/sec
c 24110566 propagations, 2.6 megaprops/sec
c 9.3 seconds, 7.5 MB
```

Γ_{X1M}:

```
c found 'p cnf 2348 56195' header
c
c    0.345  73% simplifying
c    0.131  27% search
c ====================================
c    0.476 100% all
c
c 6160 decisions, 12941.5 decisions/sec
c 5569 conflicts, 11699.9 conflicts/sec
c 629612 propagations, 1.3 megaprops/sec
c 0.5 seconds, 2.3 MB
```

19. Partial Distributive Expansion

Instead of actually performing distributive expansion, only conflicts are propagated in a round based algorithm until no new conflicts are found:

$$
\begin{array}{llll}
 & (a \vee b) & (\neg a \vee c) & (\neg c \vee d) \\
= & ((a \wedge c) \vee b) & ((\neg a \wedge b) \vee (c \wedge d)) & ((\neg c \wedge \neg a) \vee d) \\
= & ((a \wedge c \wedge d) \vee b) & ((\neg a \wedge b) \vee (c \wedge d)) & ((\neg c \wedge \neg a \wedge b) \vee d)
\end{array}
$$

When performed in a matrix, where each literal is mapped to its maximal length, the complexity of the algorithm is determined by number of literals $l = k \cdot m$ and space requirements l^2. Complexity of consolidation is determined by comparisons per round l^3, worst case l^P.

20. Hardness - Propositional Argument

The reason for the difference in hardness of problem structure between 2-state and 3-state problems is prominently visible in propositional logic.

As culprits of hardness, we have identified the relationship of “parity” XOR, to “exactly-one” X1, and the ambiguous indirect loops that can be constructed with them. Also the multi-value encoding without explicit “at-least-one” constraints, leading to implicit “at-most-one” X1N, where the “or-none” alternative has to be proved false.

Mapping a 2-state disjunction of conjunctions to a 2-state X1 is the primary mapping of 2-SAT problems to a satoku matrix:

$$
\begin{array}{ll}
 & \mathrm{OR}(p,q) \\
= & (\quad p \vee q \quad) \\
= & (\quad (p) \vee (\neg p \wedge \quad q) \quad) \\
= & (\quad (q) \vee (\neg q \wedge \quad p) \quad)
\end{array}
$$

p	q	XOR	X1	X1N	AND	OR
0	0	0	0	1	0	0
0	1	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	0	0	1	1

Table 5: 2-state truth tables XOR, X1, X1N

Table 5 shows, that XOR and X1 are equivalent in 2-state problems, so a 2-state XOR can be expressed by a 2-state X1:

$$\begin{array}{rl} & \mathrm{XOR}(p,q) \\ = & (\quad p \oplus q \quad) \\ = & (\neg p \vee \neg q) \wedge (p \vee q) \\ = & ((\neg p \wedge q) \vee (p \wedge \neg q)) \end{array}$$

2-state X1N maps to $\neg(p \wedge q)$ which is resolved to a 2-state OR $(\neg p \vee \neg q)$. There are various ways to reduce a 2-state X1N to a 2-state X1:

$$\begin{array}{rl} & \mathrm{X1N}(p,q) \\ = & ((\neg p \wedge \neg q) \vee (\neg p \wedge q) \vee (p \wedge \neg q)) \\ = & (\neg p \vee (p \wedge \neg q)) \\ = & (\neg q \vee (q \wedge \neg p)) \\ = & (\neg p \vee \neg q) \end{array}$$

2-state cells limit the maximum multi-value representation to 2 values, which is the same as the number of states already used. This is already obvious from the mapping of 2-state X1N to 2-state X1.

So there is no room for amibiguity in 2-SAT problems, hence there is no hardness.

Since graph theory is the domain of X1 (independent set) and X1N (edge cover), the existence of polynomial time algorithms for 2-SAT problems are simply a deductible necessity.

Table 6 shows, that there is no such tight relation between 3-state XOR, X1, X1N. Only 3-state OR still maps nicely to a 3-state X1 as a disjunction of conjunctions.

p	q	r	XOR	X1	X1N	AND	OR
0	0	0	0	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	1	0	0	0	0	1
1	0	0	1	1	1	0	1
1	0	1	0	0	0	0	1
1	1	0	0	0	0	0	1
1	1	1	1	0	0	1	1

Table 6: 3-state truth tables XOR, X1, X1N

Although XOR represention is inherently exponential (3-state XOR $\mapsto$ 4-state X1, 5-state XOR $\mapsto$ 8-state X1, 7-state XOR $\mapsto$ 16-state X1, ...), it can be broken down into 3-state parts due to the symmetric properties of XOR, which makes XOR detection and XOR loop detection quite simple in the X1 satoku matrix.

A k-state X1N, $k \geq 3$, can only be represented by a $(k+1)$-state X1, which is enough to make detection of missing "at-least-one" constraints for multi-value problems as hard as solving the problem itself for a x-state, $x \geq 4$, multi-value problem.

20.1 Hardness and Complexity

n-complexity is simply useless, as there is no single exclusive set of variables to represent a propositional problem.

m-complexity for 2-state reduction is the worst case upper bound for a decision algorithm.

m-complexity over CNF-clauses alone does not account for redundant clauses or loops, and therefore cannot be an accurate measure of hardness.

E.g., hgen2-v450-s41511877.shuffled-as.sat03-1682.used-as.sat04-816.cnf:

There are 54 4-state cells. Worst case 2-state reduction m-complexity is therefore $O(2^{54})$. Since contradictions are detected earlier — just like with the XORSAT example — the depth of 37 is expected and can be verified in a satoku matrix by 2-state reduction (depth 24).

While XORSAT loops are prominently visible and do not have redundant constraints, they seem quite hard. But with Gauss-Jordan elimination as loop detection tool, they are managable.

However, multi-value problems do not have such nice properties. As shown, the entire set of "at-least-one" constraints can be left out and must be (re-) proved from the ambiguous "at-most-one" constraints. Since that is a problem of equivalent complexity as proving the incomplete source problem, there is not much that can be done about it.

21. The Laws of Logic

When the laws of logic are interpreted from the perspective of structural logic, it is important to understand, that *provability* does not necessarily mean, that anything **must** be actually decided. So, while the satoku matrix is still *undecided* not all the laws of logic are satisfied, which is a little bit reminiscent of dialethism and constructive logic. However in a *decided* satoku matrix all the laws of logic hold.

The mapping of 2 propositional variables p, q as $(p \vee \neg p) \wedge (q \vee \neg q)$ with the conflict relationship $(\neg p \vee \neg q)$ into 2-state cells results in a satoku matrix as shown in figure 64.

P	$--$	$--$	
s_{0_0}	$1 \circ$	$0\ 1$	p
s_{0_1}	$\circ\ 1$	$--$	$\neg p$
s_{1_0}	$0\ 1$	$1 \circ$	q
s_{1_1}	$--$	$\circ\ 1$	$\neg q$

.

Figure 64: $\neg p$ or $\neg q$

The variable state $p = \mathtt{T}$ is represented by atomic state $s_{0_{0_{0_0}}}$, The variable state $p = \mathtt{F}$ is represented by atomic state $s_{0_{1_{0_1}}}$.

- The law of excluded middle (LEM): "either A or ˜A", holds since a third state would make the represented states *impossible* which would cause a *contradiction*.

- Law of identity (LI): "A is A, and A is not ˜A", holds since both states are represented and *mutually exclusive*. Each state has its own row and column and there exists no transformation, that exchanges only parts of these columns[10].

- Law of non-contradiction (LNC): "not (both A and ˜A)", holds as soon as the cell representing the variable states is decided. However, as long as the cell is not decided, both states are still *possible*. The same holds for the conflict relationship row $r_{1_{1_0}}$, which only signifies that both states are possible. However, when $r_{1_{1_0}}$ becomes decided, only one of the states can be *required*.

10. When rephrasing LI as "A is A, and ˜A is ˜A" and interpreting an atomic state with its conflict relations as a state that requires itself, the propositional variable representation of two atomic states fits perfectly. But that may be actually entering the twilight zone.

22. Summary

While structural logic will not become a contender in the next SAT race and outright cannot be handled by a human without the aid of a computer, it can certainly provide theoretical insight into the structure of propositional problems.

Structural logic can also be used for real applications, e.g., to construct more desirable encodings for SAT-solvers.

To test the hypotheses of structural logic, the author conducted an experiment, by repeatedly reencoding a sudoku matrix in direct encoding, each time adding the new redundant variables and therefore convoluting the problem without actually making it harder. The findings were, that SAT-solvers (miniSAT, CryptoMiniSat, march_rw, walksat, Lingeling) spend increasingly more time checking these insignificant variables that cannot be easily identified as such in a one-dimensional environment.

In the context of structural logic, such 2-state constructs are simply ignored, since they are irrelevant for determining *provability*.

In other experiments, a significant decrease in decisions was found, when the problem was transformed with conflict maximization and re-encoded in direct encoding.

List of Tables

List of Figures

List of Algorithms

List of Theorems

List of Equations

References

[MOUNT] David M. Mount, CMSC 451 Lecture Notes, Fall 2012, pp. 92, URL = http://www.cs.umd.edu/class/fall2012/cmsc451/Lects/cmsc451-lects.pdf

[BROWN] Brown, Frank Markham, Boolean Reasoning: The Logic of Boolean Equations, Kluwer Academic Publishers, Boston, 1990. Second edition, Dover Publications, Mineola, 2003. URL = zbMATH review.

[JARV] Matti Järvisalo, Further investigations into regular XORSAT, Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06), pages 1873–1874, AAAI Press, 2006, URL = http://www.cs.helsinki.fi/u/mjarvisa/benchmarks/, accessed 2015-01-28

[RINTANEN] Rintanen, Jussi, Planning as Satisfiability: state of the art. http://users.ics.aalto.fi/rintanen/jussi/satplan.html, accessed 2015-02-06.

[HERTEL] Hertel, Alexander and Hertel, Philipp and Urquhart, Alasdair, Formalizing Dangerous SAT Encodings, Theory and Applications of Satisfiability Testing – SAT 2007, Vol 4501, Lecture Notes in Computer Science, pages 159–172, Springer Berlin Heidelberg, 2007 URL = http://dx.doi.org/10.1007/978-3-540-72788-0_18

[sep-para] Priest, Graham, Tanaka, Koji and Weber, Zach, "Paraconsistent Logic", The Stanford Encyclopedia of Philosophy (Fall 2013 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/fall2013/entries/logic-paraconsistent/.

[sep-dia] Priest, Graham and Berto, Francesco, "Dialetheism", The Stanford Encyclopedia of Philosophy (Summer 2013 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/sum2013/entries/dialetheism/.

[sep-proof] von Plato, Jan, The Development of Proof Theory, The Stanford Encyclopedia of Philosophy (Summer 2013 Edition), URL = http://plato.stanford.edu/archives/sum2013/entries/proof-theory-development/, accessed 2015-01-28.

[9] Various authors, Schaefer's dichotomy theorem - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Schaefer%27s_dichotomy_theorem, accessed 2015-02-11.

[wiki-logic] Various authors, Logic - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Logic, accessed 2015-01-28.

[wiki-ml] Various authors, Mathematical logic - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Mathematical_logic, accessed 2015-01-28.

[wiki-graph] Various authors, Graph theory - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Graph_theory, accessed 2015-01-28.

[wiki-prop] Various authors, Propositional calculus - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Propositional_calculus, accessed 2015-01-28.

[wiki-seq] Various authors, Sequent calculus - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Sequent_calculus, accessed 2015-01-28.

[wiki-mvl] Various authors, Many-valued logic - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Many-valued_logic, accessed 2015-01-28.

[wiki-cl] Various authors, Intuitionistic logic - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Intuitionistic_logic, accessed 2015-01-28.

[wiki-am] Various authors, Adjacency matrix - Wikipedia, the free encyclopedia.
URL = http://en.wikipedia.org/wiki/Adjacency_matrix, accessed 2015-01-28.

[SCHPDE] Wolfgang Scherer, Generalization of CNF and Consequences for DNF of Implicants under Distributive Expansion.
http://sw-amt.ws/satoku/doc/doc-cnf-cdf-pde/satoku-cnf-cdf-pde.pdf, accessed 2015-02-01.

[SCH2SAT] Wolfgang Scherer, 2-SAT Algorithm For Complete Set of Solutions.
http://sw-amt.ws/jsat/2-SAT-satoku-matrix.pdf, accessed 2015-02-01.

[SCHSUD] Wolfgang Scherer, World's Hardest Sudoku.
http://sw-amt.ws/sudoku/worlds-hardest-sudoku/, accessed 2015-02-01.

[SCHCDCL] Wolfgang Scherer, CDCL and Direct Encoding.
http://sw-amt.ws/satoku/doc/doc-experiments/README.html,
accessed 2017-05-12.

Appendix A. Examples

A.1. Examples for Proof of Advance Decisions

Here are some examples to show that it is not possible to construct a state row s_{x_y}, that changes *provability* of a satoku matrix $\mathbb{S}$, when a state row s_{i_j} is a superset of another state row s_{e_f}.

State row s_{1_1} in figure 65 is a superset of state row s_{0_1}. When s_{1_1} is modified to require $s_{0_{1_{0_1}}}$, provability of satoku matrix $\mathbb{S}$ would only change, if it was possible that $s_{1_{1_{1_1}}}$ was required in another state row s_{x_y}, while CFR $s_{x_{y_{0_1}}}$ was impossible.

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− − −	− − −	− − −	− − −
s_{0_1}	∘ 1 ∘	− − −	0 − −	− − −	− − −
s_{0_2}	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	− − −
s_{1_1}	− − −	∘ 1 ∘	0 − −	− 0 −	− − −
s_{1_2}	− − −	∘ ∘ 1	− − −	− − −	− − −
s_{2_0}	− 0 −	− 0 −	1 ∘ ∘	− − −	− − −
s_{2_1}	− − −	− − −	∘ 1 ∘	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	− − −
s_{3_1}	− − −	− 0 −	− − −	∘ 1 ∘	− − −
s_{3_2}	− − −	− − −	− − −	∘ ∘ 1	− − −
s_{4_0}	− − −	− − −	− − −	− − −	1 ∘ ∘
s_{4_1}	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1

Figure 65: conflict-superset-000

This situation is constructed in s_{4_0} of figure 66a. The *impossible* CFR $s_{4_{0_{0_1}}}$ has caused its mirror state $s_{0_{1_{4_0}}}$ to become *impossible* too, which in turn breaks the superset relation of s_{1_1} to s_{0_1}. To restore the superset relation, $s_{1_{1_{4_0}}}$ must also become *impossible* as shown in figure 66b. The consequence is, that the mirror state $s_{4_{0_{1_1}}}$ also becomes *impossible*. However, this renders cell row $r_{4_{0_1}}$ *impossible*, so that the entire status row s_{4_0} becomes *impossible*.

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ◦ ◦	− − −	− − −	− − −	− − −
s_{0_1}	◦ 1 ◦	− − −	0 − −	− − −	0 − −
s_{0_2}	◦ ◦ 1	− − −	− − −	− − −	− − −
s_{1_0}	− − −	1 ◦ ◦	− − −	− − −	0 − −
s_{1_1}	− − −	◦ 1 ◦	0 − −	− 0 −	− − −
s_{1_2}	− − −	◦ ◦ 1	− − −	− − −	0 − −
s_{2_0}	− 0 −	− 0 −	1 ◦ ◦	− − −	− − −
s_{2_1}	− − −	− − −	◦ 1 ◦	− − −	− − −
s_{2_2}	− − −	− − −	◦ ◦ 1	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ◦ ◦	− − −
s_{3_1}	− − −	− 0 −	− − −	◦ 1 ◦	− − −
s_{3_2}	− − −	− − −	− − −	◦ ◦ 1	− − −
s_{4_0}	− 0 −	0 − 0	− − −	− − −	1 ◦ ◦
s_{4_1}	− − −	− − −	− − −	− − −	◦ 1 ◦
s_{4_2}	− − −	− − −	− − −	− − −	◦ ◦ 1

(a) conflict-superset-direct-000

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ◦ ◦	− − −	− − −	− − −	− − −
s_{0_1}	◦ 1 ◦	− − −	0 − −	− − −	0 − −
s_{0_2}	◦ ◦ 1	− − −	− − −	− − −	− − −
s_{1_0}	− − −	1 ◦ ◦	− − −	− − −	0 − −
s_{1_1}	− − −	◦ 1 ◦	0 − −	− 0 −	0 − −
s_{1_2}	− − −	◦ ◦ 1	− − −	− − −	0 − −
s_{2_0}	− 0 −	− 0 −	1 ◦ ◦	− − −	− − −
s_{2_1}	− − −	− − −	◦ 1 ◦	− − −	− − −
s_{2_2}	− − −	− − −	◦ ◦ 1	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ◦ ◦	− − −
s_{3_1}	− − −	− 0 −	− − −	◦ 1 ◦	− − −
s_{3_2}	− − −	− − −	− − −	◦ ◦ 1	− − −
s_{4_0}	− 0 −	0 0 0	− − −	− − −	1 ◦ ◦
s_{4_1}	− − −	− − −	− − −	− − −	◦ 1 ◦
s_{4_2}	− − −	− − −	− − −	− − −	◦ ◦ 1

(b) conflict-superset-direct-001

Figure 66: conflict-superset-direct

When the *impossible* CFR for the subset state row s_{g_h} is required indirectly, the superset property of state row s_{i_j} is not violated, like the condition in state row s_{4_0} of figure 67a requiring state row s_{2_0} shows. However, consolidation will merge $s_{2_{0_{1_1}}}$ into $s_{4_{0_{1_1}}}$ and therefore cell row $r_{4_{0_1}}$ will become *impossible*.

To avoid this, $s_{2_{0_{1_1}}}$ would have to be *possible* (see figure 67a), which would also make $s_{1_{1_{2_0}}}$ *possible*, thus again violating the necessary superset property of s_{1_1}.

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ◦ ◦	− − −	− − −	− − −	− − −
s_{0_1}	◦ 1 ◦	− − −	0 − −	− − −	− − −
s_{0_2}	◦ ◦ 1	− − −	− − −	− − −	− − −
s_{1_0}	− − −	1 ◦ ◦	− − −	− − −	0 − −
s_{1_1}	− − −	◦ 1 ◦	0 − −	− 0 −	− − −
s_{1_2}	− − −	◦ ◦ 1	− − −	− − −	0 − −
s_{2_0}	− 0 −	− 0 −	1 ◦ ◦	− − −	− − −
s_{2_1}	− − −	− − −	◦ 1 ◦	− − −	0 − −
s_{2_2}	− − −	− − −	◦ ◦ 1	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ◦ ◦	− − −
s_{3_1}	− − −	− 0 −	− − −	◦ 1 ◦	− − −
s_{3_2}	− − −	− − −	− − −	◦ ◦ 1	− − −
s_{4_0}	− − −	0 − 0	− 0 0	− − −	1 ◦ ◦
s_{4_1}	− − −	− − −	− − −	− − −	◦ 1 ◦
s_{4_2}	− − −	− − −	− − −	− − −	◦ ◦ 1

(a) conflict-superset-indirect-001

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ◦ ◦	− − −	− − −	− − −	− − −
s_{0_1}	◦ 1 ◦	− − −	0 − −	− − −	− − −
s_{0_2}	◦ ◦ 1	− − −	− − −	− − −	− − −
s_{1_0}	− − −	1 ◦ ◦	− − −	− − −	0 − −
s_{1_1}	− − −	◦ 1 ◦	− − −	− 0 −	− − −
s_{1_2}	− − −	◦ ◦ 1	− − −	− − −	0 − −
s_{2_0}	− 0 −	− − −	1 ◦ ◦	− − −	− − −
s_{2_1}	− − −	− − −	◦ 1 ◦	− − −	0 − −
s_{2_2}	− − −	− − −	◦ ◦ 1	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ◦ ◦	− − −
s_{3_1}	− − −	− 0 −	− − −	◦ 1 ◦	− − −
s_{3_2}	− − −	− − −	− − −	◦ ◦ 1	− − −
s_{4_0}	− − −	0 − 0	− 0 0	− − −	1 ◦ ◦
s_{4_1}	− − −	− − −	− − −	− − −	◦ 1 ◦
s_{4_2}	− − −	− − −	− − −	− − −	◦ ◦ 1

(b) conflict-superset-indirect-002

Figure 67: conflict-superset-indirect

It is, however, perfectly possible for a condition to exist in a *consolidated* satoku matrix $\mathbb{S}$, that requires a state row s_{g_h} and is *mutually exclusive* with a state row s_{i_j}, when state row s_{g_h} is a true subset of state row s_{i_j}.

In figure 68a, state row s_{0_1} is a true subset of state row s_{1_1}, and state row s_{4_0} requires state $s_{0_{1_{0_1}}}$ and is mutually exclusive with state $s_{1_{1_{1_1}}}$. After consolidation in figure 68b, the condition still holds.

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− − −	− − −	− − −	0 − −
s_{0_1}	∘ 1 ∘	− − −	0 − −	− − −	− − −
s_{0_2}	∘ ∘ 1	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	− − −
s_{1_1}	− − −	∘ 1 ∘	0 − −	− 0 −	0 − −
s_{1_2}	− − −	∘ ∘ 1	− − −	− − −	− − −
s_{2_0}	− 0 −	− 0 −	1 ∘ ∘	− − −	− − −
s_{2_1}	− − −	− − −	∘ 1 ∘	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	− − −
s_{3_1}	− − −	− 0 −	− − −	∘ 1 ∘	− − −
s_{3_2}	− − −	− − −	− − −	∘ ∘ 1	− − −
s_{4_0}	0 − 0	− 0 −	− − −	− − −	1 ∘ ∘
s_{4_1}	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1

(a) Subset *required*, superset *impossible*

P	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− − −	− − −	− − −	0 − −
s_{0_1}	∘ 1 ∘	− − −	0 − −	− − −	− − −
s_{0_2}	∘ ∘ 1	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	− − −
s_{1_1}	− − −	∘ 1 ∘	0 − −	− 0 −	0 − −
s_{1_2}	− − −	∘ ∘ 1	− − −	− − −	− − −
s_{2_0}	− 0 −	− 0 −	1 ∘ ∘	− − −	0 − −
s_{2_1}	− − −	− − −	∘ 1 ∘	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	− − −
s_{3_1}	− − −	− 0 −	− − −	∘ 1 ∘	− − −
s_{3_2}	− − −	− − −	− − −	∘ ∘ 1	− − −
s_{4_0}	0 1 0	− 0 −	0 − −	− − −	1 ∘ ∘
s_{4_1}	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1

(b) satoku matrix $\mathbb{S}$ *consolidated*

Figure 68: True subset s_{0_1} *required*, superset s_{1_1} *impossible*

A.2. Example: conflict detection by advance decision

With a 2-variable *contradiction* polynomially expanded to 3-SAT:

$$
\begin{aligned}
&(\neg p \vee \neg q \vee \neg a) \wedge \\
&(\neg p \vee \neg q \vee a) \wedge \\
&(\neg p \vee q \vee \neg b) \wedge \\
&(\neg p \vee q \vee b) \wedge \\
&(p \vee \neg q \vee \neg c) \wedge \\
&(p \vee \neg q \vee c) \wedge \\
&(p \vee q \vee \neg d) \wedge \\
&(p \vee q \vee d)
\end{aligned}
$$

The advance decision algorithm (see section 8.1) determines unsatisfiability during the first decision.

Figure 69a shows an advance decision: if $s_{0_{0_{0_0}}}$ is selected, then $s_{1_{0_{1_0}}}$ is also selected, and vice versa.

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− 0 0	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{0_2}	∘ ∘ 1	0 − 0	− − −	− − −	− − −	− − −	− − −	− − −
s_{1_0}	− 0 0	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_2}	0 − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1

(a) Request $s_{0_{0_{1_0}}}, s_{1_{0_{0_0}}}$

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	1 0 0	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_0}	1 0 0	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_2}	0 − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 0	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 0	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −
s_{6_1}	− 0 0	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘
s_{7_1}	− 0 0	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1

(b) Satisfy $s_{0_{0_{1_0}}}, s_{1_{0_{0_0}}}, s_{0_{2_{1_1}}}$

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	1 0 0	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_0}	1 0 0	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_2}	0 1 0	∘ ∘ 1	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	1 0 0	1 0 0	∘ 1 ∘	− − −	0 0 1	0 0 0	0 − −	0 − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 0	− 0 0	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 0 −	0 − −	1 ∘ ∘	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −
s_{5_0}	0 − −	0 − −	0 0 −	0 − −	− − −	1 ∘ ∘	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −
s_{5_2}	− − −	− − −	− 0 −	− − −	− − 0	∘ ∘ 1	− − −	− − −
s_{6_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	1 ∘ ∘	− − −
s_{6_1}	− 0 0	− 0 0	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0
s_{7_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	− − −	1 ∘ ∘
s_{7_1}	− 0 0	− 0 0	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1

(c) Satisfy $s_{1_{2_{0_1}}}, s_{2_{1_{0_0}}}, s_{2_{1_{4_2}}} \rightarrow \neg r_{2_{1_5}}$

P	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	1 0 0	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_2}	0 1 0	∘ ∘ 1	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	1 0 0	1 0 0	− 0 −	∘ 1 ∘	0 0 1	0 0 0	0 − −	0 − −
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 0 −	0 0 −	1 ∘ ∘	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −
s_{4_2}	− − −	− − −	− 0 −	− − −	∘ ∘ 1	− − 0	− − −	− − −
s_{5_0}	0 − −	0 − −	0 0 −	0 0 −	− − −	1 ∘ ∘	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	− − 0	∘ ∘ 1	− − −	− − −
s_{6_0}	0 − −	0 − −	0 0 −	0 0 −	− − −	− − −	1 ∘ ∘	− − −
s_{6_1}	− 0 0	− 0 0	− 0 −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −
s_{6_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	∘ ∘ 1	− − 0
s_{7_0}	0 − −	0 − −	0 0 −	0 0 −	− − −	− − −	− − −	1 ∘ ∘
s_{7_1}	− 0 0	− 0 0	− 0 −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − 0	∘ ∘ 1

(d) Satisfy $s_{3_{1_{0_0}}}, s_{3_{1_{4_2}}} \rightarrow \neg r_{3_{1_5}}$

Figure 69: 2-variable contradiction - pre-decision - stage 1

P	− − −	− − −	− 0 −	− 0 −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	1 0 0	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{1_2}	0 1 0	∘ ∘ 1	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	1 0 0	0 − −	0 − −	0 − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 0 0	0 0 −	1 ∘ ∘	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	− − 0	− − −	− − −
s_{5_0}	0 − −	0 − −	0 0 0	0 0 −	− − −	1 ∘ ∘	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	− − 0	∘ ∘ 1	− − −	− − −
s_{6_0}	0 − −	0 − −	0 0 0	0 0 −	− − −	− − −	1 ∘ ∘	− − −
s_{6_1}	− 0 0	− 0 0	− 0 −	− 0 −	− 0 −	− 0 −	∘ 1 ∘	− − −
s_{6_2}	− − −	− − −	− 0 −	− 0 −	− − −	− − −	∘ ∘ 1	− − 0
s_{7_0}	0 − −	0 − −	0 0 0	0 0 −	− − −	− − −	− − −	1 ∘ ∘
s_{7_1}	− 0 0	− 0 0	− 0 −	− 0 −	− 0 −	− 0 −	− − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− 0 −	− 0 −	− − −	− − −	− − 0	∘ ∘ 1

(a) Satisfy $s_{2_{2_{3_0}}} \rightarrow \neg\, r_{x_{0_2}}, x = (4, 5, 6, 7)$

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{0_0}	1 ∘ ∘	1 0 0	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 −
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 −
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 −
s_{1_2}	0 1 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	1 0 0	0 − −	0 − −	0 − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	0 − −	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	0 0 −	0 0 −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	0 − −	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	0 0 −	0 0 −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	0 0 −	0 − −
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{6_1}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 1	0 0 0	∘ 1 ∘	0 − −
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	0 − 0
s_{7_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{7_1}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 −	0 0 −	0 − −	∘ 1 ∘
s_{7_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 − 0	∘ ∘ 1

(b) Satisfy $s_{6_{1_{4_2}}} \rightarrow \neg\, r_{6_{1_5}} \rightarrow \neg\, r_{7_{2_6}}$

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 1	0 1 0
s_{0_0}	1 ∘ ∘	1 0 0	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − 0
s_{0_1}	∘ 1 ∘	0 − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0
s_{0_2}	∘ ∘ 1	0 1 0	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − 0
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0
s_{1_2}	0 1 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 0 −	0 − 0
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	1 0 0	0 − −	0 − −	0 0 −	0 − 0
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 0 −	0 − 0
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	0 0 −	0 − 0
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	0 0 −	0 0 0
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	0 0 −	0 − 0
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	0 0 −	0 0 0
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	0 0 −	0 − 0
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	0 − 0
s_{7_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{7_1}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 −	0 0 −	0 0 −	∘ 1 ∘
s_{7_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘

(c) $\rightarrow \neg\, r_{0_{i_7}}, \neg\, r_{1_{i_7}}, i = (1, 2), \neg\, r_{4_{1_7}}, \neg\, r_{5_{1_7}}$

P	1 0 0	1 0 0	− 0 −	− 0 −	0 0 1	0 0 1	0 0 1	0 1 0
s_{0_0}	1 ∘ ∘	1 0 0	− 0 −	− 0 −	0 0 −	0 0 −	0 0 −	0 − 0
s_{0_1}	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{0_2}	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{1_0}	1 0 0	1 ∘ ∘	− 0 −	− 0 −	0 0 −	0 0 −	0 0 −	0 − 0
s_{1_1}	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{1_2}	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_0}	− 0 0	− 0 0	1 ∘ ∘	− 0 −	0 0 −	0 0 −	0 0 −	0 − 0
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− 0 0	− 0 0	∘ ∘ 1	1 0 0	0 0 −	0 0 −	0 0 −	0 − 0
s_{3_0}	− 0 0	− 0 0	− 0 −	1 ∘ ∘	0 0 −	0 0 −	0 0 −	0 − 0
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− 0 0	− 0 0	− 0 0	∘ ∘ 1	0 0 −	0 0 −	0 0 −	0 − 0
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{4_1}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{4_2}	− 0 0	− 0 0	− 0 −	− 0 −	∘ ∘ 1	0 0 0	0 0 −	0 − 0
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{5_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{5_2}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 0	∘ ∘ 1	0 0 −	0 − 0
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{6_2}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 −	0 0 −	∘ ∘ 1	0 − 0
s_{7_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{7_1}	− 0 0	− 0 0	− 0 −	− 0 −	0 0 −	0 0 −	0 0 −	∘ 1 ∘
s_{7_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘

(d) $\rightarrow \neg\, r_{5_{2_4}} \rightarrow \neg\, c_{5_4} \rightarrow$ CTR

Figure 70: 2-variable contradiction - pre-decision - stage 2

A.3. Example: Stepwise conflict detection

Stepwise indirect conflict detection of a 2-variable *contradiction* polynomially expanded to 3-SAT.

$$
\begin{array}{l}
(\neg p \lor \neg q \lor \neg a)\land \\
(\neg p \lor \neg q \lor a)\land \\
(\neg p \lor q \lor \neg b)\land \\
(\neg p \lor q \lor b)\land \\
(p \lor \neg q \lor \neg c)\land \\
(p \lor \neg q \lor c)\land \\
(p \lor q \lor \neg d)\land \\
(p \lor q \lor d)
\end{array}
$$

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− − −	− − −	− − −	0 − −	0 − −	0 − −	0 − −	− 0 −
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − 0
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −	− − −
s_{2_1}	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	− − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	− − −
s_{3_1}	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −	− − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −	− − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −	− − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘	− − −
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− − −
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −
s_{8_0}	− 0 0	0 − 0	− − −	− − −	− − −	− − −	− − −	− − −	1 ∘ ∘
s_{8_1}	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{8_2}	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1

(a) Request $s_{8_{0_{0_0}}}, s_{8_{0_{1_1}}}$

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{0_0}	1 ∘ ∘	− 0 −	− − −	− − −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 1 0
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	0 − −
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	0 − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	0 − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −	0 − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	0 − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	0 − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	0 − −
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{8_1}	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{8_2}	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1

(b) Consolidation reveals indirect conflict

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− 0 −	− − −	− − −	0 − −	0 − −	0 − −	0 − −	− 0 −
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −	− − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− − 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	− − −
s_{3_1}	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −	0 − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− − −
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −
s_{8_0}	1 0 0	− 0 −	0 1 0	− − −	0 0 −	0 0 −	0 − −	0 − −	1 ∘ ∘
s_{8_1}	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1

(c) Request and satisfy $s_{8_{0_{0_0}}}, s_{8_{0_{2_1}}}$

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{0_0}	1 ∘ ∘	− 0 −	− − −	− − −	0 − −	0 − −	0 − −	0 − −	− 0 −
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− − −	− − −	0 − −	0 − −	0 − −	0 − −	− − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −	− − −	− − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	− − −	− − −	− − 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	− − −
s_{3_1}	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	− − −	− − −	− − −
s_{3_2}	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	− − −	− − −	− − −
s_{4_0}	0 − −	0 − −	0 − −	0 − −	1 ∘ ∘	− − −	− − −	− − −	0 − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −	− − −	− − −
s_{5_0}	0 − −	0 − −	0 − −	0 − −	− − −	1 ∘ ∘	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	1 ∘ ∘	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	− − −	− − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1	− − 0	− − −
s_{7_0}	0 − −	0 − −	0 − −	0 − −	− − −	− − −	− − −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	∘ 1 ∘	− − −
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	∘ ∘ 1	− − −
s_{8_0}	1 0 0	− 0 −	0 1 0	− − −	0 0 1	0 0 0	0 − −	0 − −	1 ∘ ∘
s_{8_1}	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	∘ ∘ 1

(d) Satisfy $s_{8_{0_{4_2}}} \to$ *impossible* $r_{8_{0_5}}$

Figure 71: 2-variable contradiction polynomially expanded to 3-SAT - stage 1

P	− − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{0_0}	1 ○ ○	− 0 −	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	○ 1 ○	− − −	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	○ ○ 1	− − 0	− − −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ○ ○	− − −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	○ 1 ○	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	○ ○ 1	− 0 −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ○ ○	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 1	1 0 0	○ 1 ○	− − −	0 0 1	0 0 0	0 − −	0 − −	0 − 0
s_{2_2}	− − −	− − −	○ ○ 1	− − 0	− − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− − −	1 ○ ○	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− − −	○ 1 ○	− 0 −	− 0 −	− − −	− − −	0 − −
s_{3_2}	− − −	− − −	− − 0	○ ○ 1	− − −	− − −	− − −	− − −	0 − −
s_{4_0}	0 − −	0 − −	0 0 −	0 − −	1 ○ ○	− − −	− − −	− − −	0 − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	○ 1 ○	− − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− − −	− − −	○ ○ 1	− − 0	− − −	− − −	0 − −
s_{5_0}	0 − −	0 − −	0 0 −	0 − −	− − −	1 ○ ○	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	− − −	○ 1 ○	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− − −	− − 0	○ ○ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	1 ○ ○	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	○ 1 ○	− − −	0 − −
s_{6_2}	− − −	− − −	− − −	− − −	− − −	− − −	○ ○ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	− − −	1 ○ ○	0 − −
s_{7_1}	− 0 −	− 0 −	− − −	− − −	− 0 −	− 0 −	− − −	○ 1 ○	0 − −
s_{7_2}	− − −	− − −	− − −	− − −	− − −	− − −	− − 0	○ ○ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○
s_{8_1}	0 − −	− − −	− − −	− − −	− − −	− − −	− − −	− − −	○ 1 ○
s_{8_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	○ ○ 1

(a) Satisfy $s_{2_{1_{0_2}}}, s_{2_{1_{1_0}}}, s_{2_{1_{4_2}}} \rightarrow \neg\, r_{2_{1_5}}$

P	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{0_0}	1 ○ ○	− 0 −	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	○ 1 ○	− − −	− 0 −	− 0 −	0 − −	− − −	− 0 −	− 0 −	0 − −
s_{0_2}	○ ○ 1	− − 0	− 0 −	− − −	0 − −	− − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ○ ○	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −	− − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	○ ○ 1	− 0 −	− − −	− − −	− − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ○ ○	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	○ ○ 1	− − 0	− − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ○ ○	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− 0 −	○ 1 ○	− 0 −	− 0 −	− − −	− − −	0 − −
s_{3_2}	− − −	− − −	− 0 0	○ ○ 1	0 − −	− − −	− − −	− − −	0 − −
s_{4_0}	0 0 0	0 0 1	0 0 1	0 1 0	1 ○ ○	− 0 −	− − −	− − −	0 − −
s_{4_1}	− − −	− − −	− 0 −	− 0 −	○ 1 ○	− − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− 0 −	− − −	○ ○ 1	− − 0	− − −	− − −	0 − −
s_{5_0}	0 − −	0 − −	0 0 −	0 − −	− − −	1 ○ ○	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− − −	− − 0	○ ○ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	1 ○ ○	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− 0 −	− − −	− 0 −	− 0 −	○ 1 ○	− − −	0 − −
s_{6_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	○ ○ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 − −	− − −	− − −	− − −	1 ○ ○	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− − −	− 0 −	− 0 −	− − −	○ 1 ○	0 − −
s_{7_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − 0	○ ○ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○
s_{8_1}	0 − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	○ 1 ○
s_{8_2}	− − −	− − −	− 0 −	− − −	− − −	− − −	− − −	− − −	○ ○ 1

(b) Satisfy $s_{4_{0_{2_2}}}, s_{4_{0_{4_1}}}, s_{4_{0_{2_2}}} \rightarrow \neg\, r_{4_{0_0}}$

P	− − −	− − −	− 0 −	− − −	0 − −	− − −	− − −	− − −	0 − −
s_{0_0}	1 ○ ○	− 0 −	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	○ 1 ○	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{0_2}	○ ○ 1	− − 0	− 0 −	− − −	0 − −	0 − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ○ ○	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	○ ○ 1	− 0 −	− − −	0 − −	− − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ○ ○	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	○ ○ 1	− − 0	0 − −	− − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ○ ○	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− 0 −	○ 1 ○	0 0 −	− 0 −	− − −	− − −	0 − −
s_{3_2}	− − −	− − −	− 0 0	○ ○ 1	0 − −	0 − −	− − −	− − −	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	○ 1 ○	0 − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− 0 −	− − −	○ ○ 1	− − 0	− − −	− − −	0 − −
s_{5_0}	0 0 0	0 0 1	0 0 1	0 1 0	0 0 −	1 ○ ○	− − −	− − −	0 − −
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− − −	0 − 0	○ ○ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 −	0 − −	0 − −	− − −	1 ○ ○	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− 0 −	− − −	0 0 −	− 0 −	○ 1 ○	− − −	0 − −
s_{6_2}	− − −	− − −	− 0 −	− − −	0 − −	− − −	○ ○ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 − −	0 − −	− − −	− − −	1 ○ ○	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− − −	0 0 −	− 0 −	− − −	○ 1 ○	0 − −
s_{7_2}	− − −	− − −	− 0 −	− − −	0 − −	− − −	− − 0	○ ○ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○
s_{8_1}	0 − −	− − −	− 0 −	− − −	0 − −	− − −	− − −	− − −	○ 1 ○
s_{8_2}	− − −	− − −	− 0 −	− − −	0 − −	− − −	− − −	− − −	○ ○ 1

(c) Satisfy $s_{5_{0_{2_2}}}, s_{5_{0_{4_1}}}, s_{5_{0_{2_2}}} \rightarrow \neg\, r_{5_{0_0}}$

P	− − −	− − −	− 0 −	− − −	0 − −	0 − −	− − −	− − −	0 − −
s_{0_0}	1 ○ ○	− 0 −	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	○ 1 ○	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{0_2}	○ ○ 1	− − 0	− 0 −	− − −	0 − −	0 − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ○ ○	− 0 −	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	○ ○ 1	− 0 −	− − −	0 − −	0 − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ○ ○	− − −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	○ ○ 1	− − 0	0 − −	0 − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ○ ○	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	− 0 −	− 0 −	− 0 −	○ 1 ○	0 0 0	0 0 1	− − −	− − −	0 − −
s_{3_2}	− − −	− − −	− 0 0	○ ○ 1	0 − −	0 − −	− − −	− − −	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	○ 1 ○	0 − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	○ ○ 1	0 − 0	− − −	− − −	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○	0 0 0	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	○ 1 ○	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− − −	0 − 0	○ ○ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 −	0 − −	0 − −	0 − −	1 ○ ○	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− 0 −	− − −	0 0 −	0 0 −	○ 1 ○	− − −	0 − −
s_{6_2}	− − −	− − −	− 0 −	− − −	0 − −	0 − −	○ ○ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 − −	0 − −	0 − −	− − −	1 ○ ○	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− − −	0 0 −	0 0 −	− − −	○ 1 ○	0 − −
s_{7_2}	− − −	− − −	− 0 −	− − −	0 − −	0 − −	− − 0	○ ○ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	○ ○ ○
s_{8_1}	0 − −	− − −	− 0 −	− − −	0 − −	0 − −	− − −	− − −	○ 1 ○
s_{8_2}	− − −	− − −	− 0 −	− − −	0 − −	0 − −	− − −	− − −	○ ○ 1

(d) Satisfy $s_{3_{1_{5_2}}} \rightarrow$ *impossible* $r_{3_{1_4}}$

Figure 72: 2-variable contradiction polynomially expanded to 3-SAT - stage 2

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− − −	− − −	0 − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −	0 0 1
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− 0 −	− 0 −	0 − −	0 − −	− − −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	− − −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 − −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− 0 0	0 − −	0 − −	− − −	− − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 − −	0 − −	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	− − −	− − −	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	− 0 −	− − −	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	− − −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 −	0 0 −	0 − −	0 − −	1 ∘ ∘	− − −	0 − −
s_{6_1}	− 0 −	− 0 −	− 0 −	− 0 −	0 0 0	0 0 1	∘ 1 ∘	− − −	0 − −
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 0 −	0 − −	0 − −	− − −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− 0 −	0 0 −	0 0 −	− − −	∘ 1 ∘	0 − −
s_{7_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− − 0	∘ ∘ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{8_1}	0 − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− − −	− − −	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− − −	− − −	∘ ∘ 1

(a) Satisfy $s_{6_{1_{5_2}}} \rightarrow$ *impossible* $r_{6_{1_4}}$

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− − −	0 − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − −	0 0 1
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− − −	0 − −
s_{1_0}	− − −	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − −	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− 0 −	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− − −	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 0 −	0 − −	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− 0 0	0 − −	0 − −	0 0 −	− − −	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 0 −	0 − −	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	− 0 −	− − −	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	− 0 −	− 0 −	0 − −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	− 0 −	− − −	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	− 0 −	− − −	0 − −
s_{6_0}	0 − −	0 − −	0 0 0	0 0 1	0 − −	0 − −	1 ∘ ∘	− − −	0 − −
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 0 −	0 − −	0 − −	− 0 −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− 0 −	0 0 −	0 0 −	− 0 −	∘ 1 ∘	0 − −
s_{7_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 0	∘ ∘ 1	0 − −
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{8_1}	0 − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− − −	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	− 0 −	− − −	∘ ∘ 1

(b) Satisfy $s_{6_{0_{3_2}}} \rightarrow \neg\, r_{6_{0_2}} \rightarrow \neg\, r_{7_{2_6}}$

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 1	− − 0	0 − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − 0	0 0 1
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− − 0	0 − −
s_{1_0}	− − −	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 − 0	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− − 0	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 0 −	0 − 0	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− 0 0	0 − −	0 − −	0 0 −	− − 0	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 0 −	0 − 0	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	0 0 −	− − 0	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	0 0 −	− 0 0	0 − −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	0 0 −	− 0 0	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	0 0 −	− 0 0	0 − −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	0 0 −	− − 0	0 − −
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	− − 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 −	0 0 −	0 − −	0 − −	0 0 −	1 ∘ ∘	0 − −
s_{7_1}	− 0 −	− 0 −	− 0 −	− 0 −	0 0 0	0 0 1	0 0 −	∘ 1 ∘	0 − −
s_{7_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{8_1}	0 − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− − 0	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− − 0	∘ ∘ 1

(c) Satisfy $s_{7_{1_{5_2}}} \rightarrow$ *impossible* $r_{7_{1_4}}$

P	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 1	1 0 0	0 − −
s_{0_0}	1 ∘ ∘	− 0 −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0	0 0 1
s_{0_1}	∘ 1 ∘	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{0_2}	∘ ∘ 1	− − 0	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{1_0}	− − −	1 ∘ ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	0 0 0	0 − −
s_{1_1}	0 − −	∘ 1 ∘	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{1_2}	− − 0	∘ ∘ 1	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{2_0}	− − −	− − −	1 ∘ ∘	− 0 −	0 − −	0 − −	0 0 −	0 0 0	0 − −
s_{2_1}	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{2_2}	− − −	− − −	∘ ∘ 1	− 0 0	0 − −	0 − −	0 0 −	− 0 0	0 − −
s_{3_0}	− − −	− − −	− 0 −	1 ∘ ∘	0 − −	0 − −	0 0 −	0 0 0	0 − −
s_{3_1}	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
s_{3_2}	− − −	− − −	− 0 0	∘ ∘ 1	0 − −	0 − −	0 0 −	0 0 0	0 − −
s_{4_0}	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0	0 0 0
s_{4_1}	− − −	− − −	− 0 −	− 0 −	∘ 1 ∘	0 − −	0 0 −	− 0 0	0 − −
s_{4_2}	− − −	− − −	− 0 −	− 0 −	∘ ∘ 1	0 − 0	0 0 −	− 0 0	0 − −
s_{5_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0	0 0 0
s_{5_1}	− − −	− − −	− 0 −	− 0 −	0 − −	∘ 1 ∘	0 0 −	− 0 0	0 − −
s_{5_2}	− − −	− − −	− 0 −	− 0 −	0 − 0	∘ ∘ 1	0 0 −	− 0 0	0 − −
s_{6_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{6_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0	0 0 0
s_{6_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	∘ ∘ 1	− 0 0	0 − −
s_{7_0}	0 − −	0 − −	0 0 1	0 0 0	0 − −	0 − −	0 0 −	1 ∘ ∘	0 − −
s_{7_1}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{7_2}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘	0 0 0
s_{8_0}	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	∘ ∘ ∘
s_{8_1}	0 − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	∘ 1 ∘
s_{8_2}	− − −	− − −	− 0 −	− 0 −	0 − −	0 − −	0 0 −	− 0 0	∘ ∘ 1

(d) Satisfy $s_{7_{0_{2_2}}} \rightarrow \neg\, r_{7_{0_3}} \rightarrow \neg\, c_{7_3} \rightarrow$ CTR

Figure 73: 2-variable contradiction polynomially expanded to 3-SAT - stage 3