Noname manuscript No.
(will be inserted by the editor)

Incrementally Closing Octagons

Aziem Chawdhary · **Ed Robbins** ·
Andy King

Received: date / Accepted: date

Abstract The octagon abstract domain is a widely used numeric abstract domain expressing relational information between variables whilst being both computationally efficient and simple to implement. Each element of the domain is a system of constraints where each constraint takes the restricted form $\pm x_i \pm x_j \leqslant c$. A key family of operations for the octagon domain are closure algorithms, which check satisfiability and provide a normal form for octagonal constraint systems. We present new quadratic incremental algorithms for closure, strong closure and integer closure and proofs of their correctness. We highlight the benefits and measure the performance of these new algorithms.

Keywords Abstract Interpretation · Octagons · Incremental Closure

1 Introduction

The view that simplicity is a virtue in competing scientific theories and that, other things being equal, simpler theories should be preferred to more complex ones, is widely advocated by scientists and engineers. Preferences for simpler theories are thought to have played a role in many episodes in science, and the field of abstract domain design is no exception. Abstract domains that have enduring appeal are typically those that are conceptually simple. Of all the weakly relational domains, for example, octagons [22] are arguably the most popular. One might claim that octagons are more elegant than, say, the two variable per inequality (TVPI) domain [32], and certainly they are easier to understand and implement. Yet one important operation for this popular domain has remained elusive: incremental closure.

Inequalities in the octagon domain take the restricted form of $\pm x_i \pm x_j \leqslant c$, where x_i and x_j are variables and c is a numerical constant. Difference bound matrices (DBMs) can be adapted to represent systems of octagonal constraints, for

This work was supported by EP/K031929/1 and EP/K032585/1 funded by GCHQ in association with EPSRC and the EPSRC/NRF UK/Singaporean research grant EP/N020243/1

Aziem Chawdhary, School of Computing, University of Kent, Canterbury, CT2 7NF, UK. Tel.: +44-1227-827911 Fax: +44-1227-762811 E-mail: a.a.chawdhary@kent.ac.uk

which a key family of operations is closure. Closure, in its various guises, provides normal forms for DBMs, allowing satisfiability to be observed and equality to be checked. Closure also underpins operations such as join and projection (the forget operator), hence the concept of closure is central to the design of the whole domain. Closure uses shortest path algorithms, such as Floyd-Warshall [13,36], to check for satisfiability. However, octagons can encode unary constraints, which require a stronger notion of closure, known as strong closure, to derive a normal form. Moreover, a refinement to strong closure, called integer closure, is required to detect whether octagonal constraints have an integral solution.

A frequent use-case in program analysis is adding a single new octagonal constraint to a closed DBM and then closing the augmented system. This is incremental closure. Incremental closure not only arises when an octagon for one line is adjusted to obtain an octagon for the next: incremental closure also occurs in integer wrapping [31] which involves repeatedly partitioning a space into two (by adding a single constraint), closing and then performing translation. Incremental closure is useful in access-based localisation [25], which analyses each procedure using abstractions defined over only those variables it accesses. One way to adapt localisation to octagons [5] is to introduce fresh variables, called anchors, that maintain the relationships which hold when a procedure is entered. One anchor is introduced for each variable that is accessed within the procedure. The body of the callee is analysed to capture how a variable changes relative to its anchor, and then this change is propagated into the caller. The abstraction of the callee is amalgamated with that of the caller by replacing the variables in the caller abstraction with their anchors, imposing the constraints from the callee abstraction, and then eliminating the anchors. If there are only a few non-redundant constraints in the callee [2] then incremental closure is attractive for combining caller and callee abstractions. Nevertheless, the experimental results focus on the use-case of adding a single constraint encountered on one line to an octagon that summaries the previous line.

In SMT solving, difference logic [24] is widely supported, suggesting that an incremental solver for the theory of octagons [28] would also be useful. Furthermore afield in constraint solving, relational and mixed integer-real abstract domains show promise for enhancing constraint solvers [26] and octagons have been deployed for solving continuous constraints [27]. In this context, a split operator is used to divide the solution space into two sub-spaces by adding opposing constraints such as $x_i - x_j \leqslant c$ and $x_j - x_i \leqslant -c$. Splitting is repeatedly applied until a set of octagons is derived that cover the entire solution space, within a given precision tolerance. Propagation is applied after every split, which suggests incremental closure, and a scheme in which incremental closure is applied whenever a propagator updates a variable. This use-case is also examined experimentally.

Closing an augmented DBM is less general than closing an arbitrary DBM and therefore one would expect incremental closure to be both efficient and conceptually simple. However the running time of the algorithm originally proposed for incremental closure [21, Section 4.3.4] is cubic in the number of variables (see Section 4.1 for an explanation of the impact of row and column swaps). The algorithms presented in this paper stem from the desire to understand incremental closure by providing correctness proofs that would, in turn, provide a pathway to mechanisation. Yet the act of restructuring the proofs for [10], exposed a degenerate form of propagation and revealed fresh algorithmic insights. The resulting family of closure algorithms includes: a new algorithm for increment closure; a new algorithm for strong closure that performs strengthening on-the-fly, rather than a separate pass over the whole DBM; a further refinement to strong closure applicable when the input DBM is strongly closed; and finally a new incremental closure algorithm for integer DBMs. All algorithms significantly outperform the incremental algorithm of Miné [21, Section 4.3.4], whilst entirely recovering closure, as is demonstrated from their deployment in an off-the-shelf abstract interpretation and a continuous constraint solver. The dramatic speedups underscore the importance of this domain operation.

1.1 Contributions

We summarise the contributions of our work as follows:

- Using new insights, we present new incremental algorithms for closure, strong closure and integer closure (Section 4, Section 5 and Section 6 respectively). We show how code hoisting can be applied to incremental closure and how strength reduction can be applied to strong incremental closure.
- We prove our algorithms correct and show how proofs for existing closure algorithms can be simplified, paving the way for mechanised formalisation. (To keep the length of the paper manageable, the proofs are relegated to Appendix A. The exception is Lemma 6.1 since the argument is itself a significant conceptual advance, hence is included in the body of the paper.)
- We give detailed proofs for in-place versions of our algorithms (Section 7).
- We implement these new algorithms which show significant performance improvements over existing closure algorithms in real-world setting (Section 8).

The paper is structured as follows: Section 2 contextualises this study and Section 3 provides the necessary preliminaries. Section 4 critiques the incremental algorithm of Miné, introduces a new incremental quadratic algorithm. Section 5 shows how to recover strong closure incrementally and do so, again, in a single DBM pass. Section 6 explains how to extend incrementally to integer closure. Section 7 suggest various optimisations to the incremental algorithms including in-place update. Experimental results are presented in Section 8 and Section 9 concludes.

2 Related Work

Since the thesis of Miné [21] and his subsequent magnum opus [22], algorithms for manipulating octagons, and even their representations, have continued to evolve. Early improvements showed how strengthening, the act of combining pairs of unary octagon constraints to improve binary octagon constraints, need not be applied repeatedly, but instead can be left to a single post-processing step [2]. This result, which was justified by an inventive correctness argument, led to a performance improvement of approximately 20% [2]. Showing that integer octagonal constraints admit polynomial satisfiability represented another significant advance [1], especially since dropping either the two variable or unary coefficient property makes the problem NP-complete [19].

Octagonal representations have come under recent scrutiny [18, Chapter 8]. In Coq, it is natural to realise DBMs as map from pairs of indices (represented as bit sequences) to matrix entries. Look-up becomes logarithmic in the dimension of the DBM, but the DBM itself can be sparse. Strengthening, which combine bounds on different variables, can populate a DBM with entries for binary constraints. Dropping strengthening thus improves sparsity, albeit at the cost of sacrificing a canonical representation. Join can be recovered by combining bounds during join itself, in effect, strengthening on-the-fly. Quite independently, sparse representations have recently been developed for differences [14]. Further field, $O(mn)$ decision procedures have been proposed for unit two variable per inequality (UTVPI) constraints [20] where m and n are the number of constraints and variables respectively. Subsequently an incremental version was proposed for UTVPI [30] with time complexity $O(m + n\log(n) + p)$ where p is the number of constraints tightened by the additional inequality. Certifying algorithms have also been devised for UTVPI constraints [34], supported by a graphical representation of these constraints, which aids the extraction of a certificate for validating unsatisfiability. DBMs, however, offer additional support for other operations that arise in program analysis such as join and projection. Moreover, there is no reason why each DBM entry could not be augmented with a pair of row and column coordinates which records how it was updated, allowing a proof for unsatisfiability to be extracted from a negative diagonal entry.

Other recent work [33] has proposed factoring octagons into independent subsystems, which reduces the size of the DBM. Domain operations are applied pointwise to the independent sub-matrices of the DBM, echoing [15]. The work also shows how the regular access patterns of DBMs enable vectorisation, the step beyond which is harnessing general purpose GPUs [3]. Packs [8] have also been proposed as a factoring device in which the set of programs variables is covered by a sets of variables called packs (or clusters). An octagon is computed for each pack to abstract the DBM as a set of low-dimensional DBMs. Recent work has even explored how packs can be introduced automatically using preanalysis and machine learning [16].

The alternative to simplifying the DBM representation is to assume that the DBM satisfies some prerequisites so that a domain operation need not be applied in full generality. Miné [21] showed that an incremental version of the closure could be derived by observing that a new constraint is independent of the first c variables of the DBM. This paper stems from an earlier work [10] that extends an incremental algorithm for disjunctive spatial constraints which originates in planning [4]. The work was motivated by the desire to augment [10] with conceptually simple correctness proofs, that revealed a deficiency in the propagation algorithm of [10] which prompted a more thorough study of incrementality.

Further afield, closure of octagons echos path consistency in temporal constraint networks [12], which also uses the Floyd-Warshall algorithm to tighten constraints. Furthermore, IncStrongClose, which processes key entries (staggered diagonal entries) first, tallies with how extremal values are first processed in constraint propagation [7]. Difference constraints can be generalised to Allen constraints [29] to express set theoretic properties, such as overlap. Solving Allen constraints is also polynomial, but each variable can be updated many times when calculating the fixpoint. By way of contrast, the restricted form of octagons means that each element in the DBM is updated at most once, which is key to the efficiency of incremental closure.

3 Preliminaries

This section serves as a self-contained introduction to the definitions and concepts required in subsequent sections. For more details, we invite the reader to consult both the seminal [21, 22] and subsequent [2, 10] works on the octagon abstract domain.

3.1 The Octagon Domain and its Representation

An octagonal constraint is a two variable inequality of the form $\pm x_i \pm x_j \leqslant d$ where x_i and x_j are variables and d is a constant. An octagon is a set of points satisfying a system of octagonal constraints. The octagon domain is the set of all octagons that can be defined over the variables $x_0, \ldots, x_{n-1}$.

Implementations of the octagon domain reuse the machinery developed for solving difference constraints of the form $x_i - x_j \leqslant d$. Miné [22] showed how to translate octagonal constraints to difference constraints over an extended set of variables $x'_0, \ldots, x'_{2n-1}$. A single octagonal constraint translates into a conjunction of one or more difference constraints as follows:

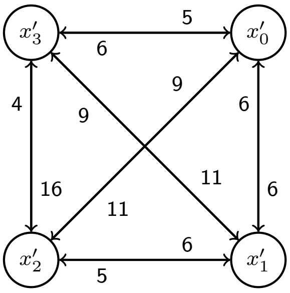
$$\begin{array}{rcl}
x_i - x_j \leqslant d & \rightsquigarrow & x'_{2i} - x'_{2j} \leqslant d \;\wedge\; x'_{2j+1} - x'_{2i+1} \leqslant d \\
x_i + x_j \leqslant d & \rightsquigarrow & x'_{2i} - x'_{2j+1} \leqslant d \;\wedge\; x'_{2j} - x'_{2i+1} \leqslant d \\
-x_i - x_j \leqslant d & \rightsquigarrow & x'_{2i+1} - x'_{2j} \leqslant d \;\wedge\; x'_{2j+1} - x'_{2i} \leqslant d \\
x_i \leqslant d & \rightsquigarrow & x'_{2i} - x'_{2i+1} \leqslant 2d \\
-x_i \leqslant d & \rightsquigarrow & x'_{2i+1} - x'_{2i} \leqslant 2d
\end{array}$$

A common representation for difference constraints is a difference bound matrix (DBM) which is a square matrix of dimension $n \times n$, where n is the number of variables in the difference system. The value of the entry $d = \mathbf{m}_{i,j}$ represents the constant d of the inequality $x_i - x_j \leqslant d$ where the indices $i, j \in \{0, \ldots, n-1\}$. An octagonal constraint system over n variables translates to a difference constraint system over $2n$ variables, hence a DBM representing an octagon has dimension $2n \times 2n$.

Example 1 Figure 1 serves as an example of how an octagon translates to a system of differences. The entries of the upper DBM correspond to the constants in the difference constraints. Note how differences which are (syntactically) absent from the system lead to entries which take a symbolic value of ∞. Observe too how that DBM represents an adjacency matrix for the illustrated graph where the weight of a directed edge abuts its arrow.

The interpretation of a DBM representing an octagon is different to a DBM representing difference constraints. Consequently there are two concretisations for DBMs: one for interpreting differences and another for interpreting octagons, although the latter is defined in terms of the former:

$$\begin{array}{r} x_0 \leqslant 3 \\ x_1 \leqslant 2 \\ x_0 + x_1 \leqslant 6 \\ -x_0 - x_1 \leqslant 5 \\ -x_0 \leqslant 3 \end{array} \qquad \begin{array}{l} x'_0 - x'_1 \leqslant 6 \\ x'_2 - x'_3 \leqslant 4 \\ x'_0 - x'_3 \leqslant 6 \\ x'_2 - x'_1 \leqslant 6 \\ x'_1 - x'_2 \leqslant 5 \\ x'_3 - x'_0 \leqslant 5 \\ x'_1 - x'_0 \leqslant 6 \end{array}$$



$$\begin{array}{c|cccc} & x'_0 & x'_1 & x'_2 & x'_3 \\ \hline x'_0 & \infty & 6 & \infty & 6 \\ x'_1 & 6 & \infty & 5 & \infty \\ x'_2 & \infty & 6 & \infty & 4 \\ x'_3 & 5 & \infty & \infty & \infty \end{array}$$

$$\begin{array}{c|cccc} & x'_0 & x'_1 & x'_2 & x'_3 \\ \hline x'_0 & 0 & 6 & 11 & 6 \\ x'_1 & 6 & 0 & 5 & 9 \\ x'_2 & 9 & 6 & 0 & 4 \\ x'_3 & 5 & 11 & 16 & 0 \end{array}$$

Fig. 1: Example of an octagonal system and its DBM representation

Definition 3.1 Concretisation for rational ($\mathbb{Q}^n$) solutions:

$$\begin{aligned} \gamma_{\text{diff}}(\mathbf{m}) &= \{\langle v_0, \ldots, v_{n-1}\rangle \in \mathbb{Q}^n \mid \forall i, j. v_i - v_j \leqslant \mathbf{m}_{i,j}\} \\ \gamma_{\text{oct}}(\mathbf{m}) &= \{\langle v_0, \ldots, v_{n-1}\rangle \in \mathbb{Q}^n \mid \langle v_0, -v_0, \ldots, v_{n-1}, -v_{n-1}\rangle \in \gamma_{\text{diff}}(\mathbf{m})\} \end{aligned}$$

where the concretisation for integer ($\mathbb{Z}^n$) solutions can be defined analogously.

Example 2 Since octagonal inequalities are modelled as two related differences, the upper DBM contains duplicated entries, for instance, $\mathbf{m}_{1,2} = \mathbf{m}_{3,0}$.

Operations on a DBM representing an octagon must maintain equality between the two entries that share the same constant of an octagonal inequality. This requirement leads to the definition of coherence:

Definition 3.2 (Coherence) A DBM $\mathbf{m}$ is coherent iff $\forall i.j.\mathbf{m}_{i,j} = \mathbf{m}_{\bar{j},\bar{\imath}}$ where $\bar{\imath} = i + 1$ if i is even and $i - 1$ otherwise.

Example 3 For the upper DBM observe $\mathbf{m}_{0,3} = 6 = \mathbf{m}_{2,1} = \mathbf{m}_{\bar{3},\bar{0}}$. Coherence holds in a degenerate way for unary inequalities, note $\mathbf{m}_{2,3} = 4 = \mathbf{m}_{2,3} = \mathbf{m}_{\bar{3},\bar{2}}$.

The bar operation can be realised without a branch using $\bar{\imath} = i$ **xor** 1 [21, Section 4.2.2]. Care should be taken to preserve coherence when manipulating DBMs, either by carefully designing algorithms or by using a data structure that enforces coherence [21, Section 4.5]. For clarity, we abstract away from the question of how to represent a DBM by presenting all algorithms for square matrices, rather than triangular matrices as introduced in [21, Section 4.5]. One final property is necessary for satisfiability:

Definition 3.3 (Consistency) A DBM $\mathbf{m}$ is consistent iff $\forall i.\mathbf{m}_{i,i} \geqslant 0$.

Intuitively, consistency means that there is not negative cycle in the DBM, which corresponds to unsatisfiability [6].

3.2 Definitions of Closure

Closure properties define canonical representations of DBMs, and can decide satisfiability and support operations such as join and projection. Bellman [6] showed

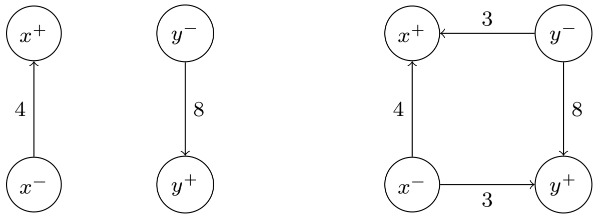


Fig. 2: Intuition behind strong closure: Two closed graphs representing the same octagon: $x \leqslant 2 \wedge y \leqslant 4$

that the satisfiability of a difference system can be decided using shortest path algorithms on a graph representing the differences. If the graph contains a negative cycle (a cycle whose edge weights sum to a negative value) then the difference system is unsatisfiable. The same applies for DBMs representing octagons. Closure propagates all the implicit (entailed) constraints in a system, leaving each entry in the DBM with the sharpest possible constraint entailed between the variables. Closure is formally defined below:

Definition 3.4 (Closure) A DBM $\mathbf{m}$ is closed iff

- $\forall i.\mathbf{m}_{i,i} = 0$
- $\forall i, j, k.\mathbf{m}_{i,j} \leqslant \mathbf{m}_{i,k} + \mathbf{m}_{k,j}$

Example 4 The top right DBM in Figure 1 is not closed. By running an all-pairs shortest path algorithm we get the following DBM:

$$\begin{array}{c|cccc|} & x'_0 & x'_1 & x'_2 & x'_3 \\ x'_0 & 11 & 6 & 11 & 6 \\ x'_1 & 6 & 11 & 5 & 9 \\ x'_2 & 9 & 6 & 11 & 4 \\ x'_3 & 5 & 11 & 16 & 11 \end{array}$$

Notice that the diagonal values have non-negative elements implying that the constraint system is satisfiable. Running shortest path closure algorithms propagates all constraints and makes every explicit all constraints implied by the original system. Once satisfiability has been established, we can set the diagonal values to zero to satisfy the definition of closure.

Closure is not enough to provide a canonical form for DBMs representing octagons. Miné defined the notion of strong closure in [21,22] to do so:

Definition 3.5 (Strong closure) A DBM $\mathbf{m}$ is strongly closed iff

- $\mathbf{m}$ is closed
- $\forall i, j.\mathbf{m}_{i,j} \leqslant \mathbf{m}_{i,\bar{\imath}}/2 + \mathbf{m}_{\bar{\jmath},j}/2$

The strong closure of DBM $\mathbf{m}$ can be computed by $\textsc{Str}(\mathbf{m})$, the code for which is given in Figure 4. The algorithm propagates the property that if $x'_j - x'_{\bar{\jmath}} \leqslant c_1$ and $x'_{\bar{\imath}} - x'_i \leqslant c_2$ both hold then $x'_j - x'_i \leqslant (c_1 + c_2)/2$ also holds. This sharpens the bound on the difference $x'_j - x'_i$ using the two unary constraints encoded by $x'_j - x'_{\bar{\jmath}} \leqslant c_1$ and $x'_{\bar{\imath}} - x'_i \leqslant c_1$, namely, $2x'_j \leqslant c_1$ and $-2x'_i \leqslant c_2$. Note that this constraint propagation is not guaranteed to occur with a shortest path algorithm since there is not necessarily a path from a $\mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m}_{\bar{\jmath},j}$. An example in Figure 2 shows such a situation: the two graphs represent the octagon, but a shortest path algorithm will not propagate constraints on the left graph; hence strengthening is needed to bring the two graphs to the same normal form. Strong closure yields a canonical representation: there is a unique strongly closed DBM for any (non-empty) octagon [22]. Thus any semantically equivalent octagonal constraint systems are represented by the same strongly closed DBM. Strengthening is the act of computing strong closure.

Example 5 The lower right DBM in Figure 1 gives the strong closure of the upper right DBM. Strengthening is performed after the shortest path algorithm.

For octagonal constraints over integers, the strong closure property may result in non-integer values due to the division by two. The definition of strong closure for integer octagonal constraints thus needs to be refined. If x_i is integral then $x_i \leqslant c$ tightens to $x_i \leqslant \lfloor c \rfloor$. Since $x_i \leqslant c$ translates to the difference $x'_{2i} - x'_{2i+1} \leqslant 2c$, tightening the unary constraint is achieved by tightening the difference to $x'_{2i} - x'_{2i+1} \leqslant 2\lfloor c/2 \rfloor$.

Definition 3.6 (Tight closure) A DBM $\mathbf{m}$ is tightly closed iff

- $\mathbf{m}$ is strongly closed
- $\forall i . \mathbf{m}_{i,\bar{\imath}}$ is even

For the integer case, a tightening step is required before strengthening. Tightening a closed DBM results in a weaker form of closure, called weak closure. Strong closure can be recovered from weak closure by strengthening [1]. Note, however, that we introduce the property for completeness of exposition because our formalisation and proofs do not make use of this notion.

Definition 3.7 (Weak closure) A DBM $\mathbf{m}$ is weakly closed iff

- $\forall i . \mathbf{m}_{i,i} = 0$
- $\forall i, j, k . \mathbf{m}_{i,k} + \mathbf{m}_{k,j} \geqslant \min(\mathbf{m}_{i,j}, \mathbf{m}_{i,\bar{\imath}}/2 + \mathbf{m}_{\bar{\jmath},j}/2)$

3.3 High-level Overview

Figure 3 gives a high-level overview of closure calculation. First a closure algorithm is applied to a DBM. Next, consistency is checked by observing the diagonal has non-negative entries indicating the octagon is satisfiable. If satisfiable, then the DBM is strengthened, resulting in a strongly closed DBM. Note that consistency does not need to be checked again after strengthening. The dashed lines in the figure show the alternative path taken for integer problems: to ensure that the DBM entries are integral, a tightening step is applied which is then followed by an integer consistency check and strengthening.

Figure 4 shows how this architecture can be instantiated with algorithms for non-incremental strong closure. A Floyd-Warshall all-pairs shortest path algorithm [13,36] can be applied to a DBM to compute closure, which is cubic in n. The check for consistency involves a pass over the matrix diagonal to check for a strictly

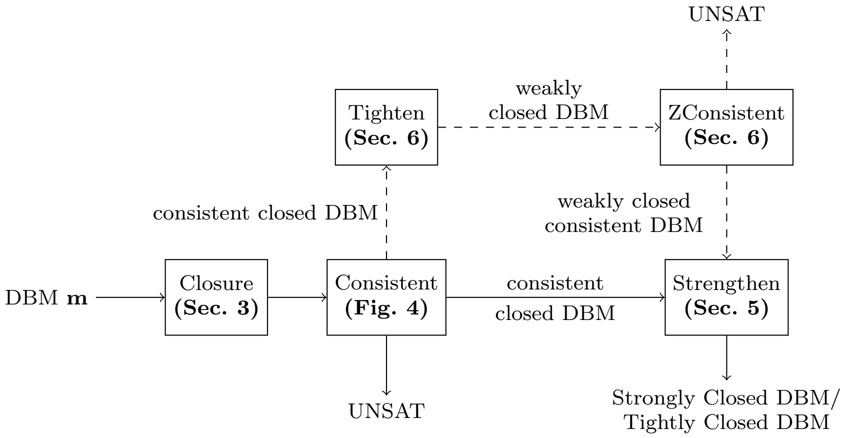


Fig. 3: High-Level Overview of Closure Algorithms for Octagons

```
function CLOSE($\mathbf{m}$)
    for $k \in \{0, \ldots, 2n-1\}$ do
        for $i \in \{0, \ldots, 2n-1\}$ do
            for $j \in \{0, \ldots, 2n-1\}$ do
                $\mathbf{m}'_{i,j} \leftarrow \min(\mathbf{m}_{i,j}, \mathbf{m}_{i,k} + \mathbf{m}_{k,j})$
            end for
        end for
    end for
    return $\mathbf{m}'$
end function
```

```
function CHECKCONSISTENT($\mathbf{m}$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        if $\mathbf{m}_{i,i} < 0$ then
            return false
        else
            $\mathbf{m}_{i,i} = 0$
        end if
    end for
    return true
end function
```

```
function STR($\mathbf{m}$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        for $j \in \{0, \ldots, 2n-1\}$ do
            $\mathbf{m}'_{i,j} \leftarrow \min(\mathbf{m}_{i,j}, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2)$
        end for
    end for
    return $\mathbf{m}'$
end function
```

Fig. 4: Non-incremental closure and strengthening

negative entry, as illustrated in the figure. (Note that CHECKCONSISTENT resets a strictly positive diagonal entry to zero as in [2,22], but the incremental algorithms presented in this paper never relax a zero diagonal entry to a strictly positive value. Hence the reset is actually redundant for the incremental algorithms that follow.) The consistency check is linear in n. Strong closure can be additionally obtained by following closure with a single call to STR, the code for which is also listed in the figure. This is quadratic in n.

4 Incremental Closure

We are interested in the specific use case of adding a new octagonal constraint to an existing closed octagon. Miné designed an incremental algorithm for this very task, which can be refactored into computing closure and then separately strengthening, as depicted in Figure 3. His incremental algorithm, and a refinement, are discussed in Section 4.1. Section 4.2 presents our new incremental algorithm with improved performance.

4.1 Classical Incremental Closure

Miné designed an incremental algorithm based on the observation that a new constraint will not affect all the variables of the octagon [21, Section 4.3.4]. Without loss of generality, suppose the inequality $x'_a - x'_b \leqslant d$ is added to the DBM (unary constraints are supported by putting $b = \bar{a}$). Adding $x'_a - x'_b \leqslant d$ implies that the equivalent constraint $x'_{\bar{b}} - x'_{\bar{a}} \leqslant d$ is added too, and the entries $\mathbf{m}_{a,b}$ and $\mathbf{m}_{\bar{b},\bar{a}}$ are updated to d to reflect this. Figure 5 presents a version of the incremental algorithm of Miné, specialised for adding $x'_a - x'_b \leqslant d$ to a closed DBM. The algorithm relies on the observation that updating $\mathbf{m}_{a,b}$ and $\mathbf{m}_{\bar{b},\bar{a}}$ will only (initially) mutate the rows and columns for the $x'_a, x'_b, x'_{\bar{a}}, x'_{\bar{b}}$ variables. Put $v = \min(a, b, \bar{a}, \bar{b})$. Since $\mathbf{m}$ was closed, despite the updates, it still follows that if $k < v$ then $\mathbf{m}_{i,j} \leqslant \mathbf{m}_{i,k} + \mathbf{m}_{k,j}$ for all $0 \leqslant i < 2n$ and $0 \leqslant j < 2n$. This is the inductive property which is established after the first v iterations of the outermost for loop of the standard Floyd-Warshall algorithm. Therefore, to restore closure it only necessary to apply the remaining $2n - v$ iterations of Floyd-Warshall, which leads to the algorithm of Figure 5.

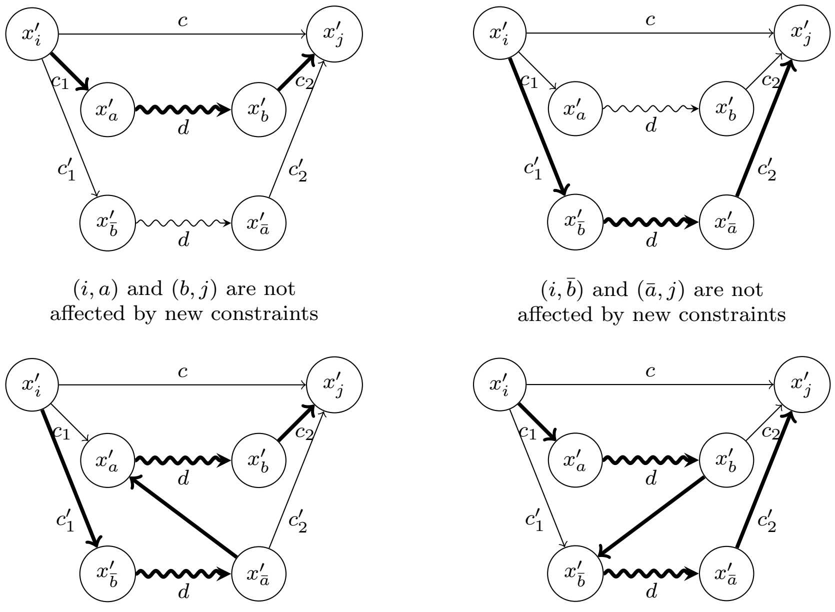
The incremental closure of Figure 5 reduces the number of min operations from $8n^3$ to $(2n-v)4n^2$ (notwithstanding those in Str). Prior to the updates, one could conceivably reconfigure the DBM by swapping rows and columns so that, say, $a = 2n - 4, \bar{a} = 2n - 3, b = 2n - 2, \bar{b} = 2n - 1$. Then $v = 2n - 4$ reducing incremental closure to $16n^2$. However, after closure, the rows and columns would need to be swapped back to maintain a consistent representation. Observe too that $x'_a - x'_b \leqslant d$ and $x'_e - x'_f \leqslant d$ can be added to the DBM simultaneously by putting $v = \min(a, b, \bar{a}, \bar{b}, e, f, \bar{e}, \bar{f})$ and then applying incremental closure once.

4.2 Improved Incremental Closure

To give the intuition behind our new incremental closure algorithm, consider adding the constraint $x'_a - x'_b \leqslant d$ and $x'_{\bar{b}} - x'_{\bar{a}} \leqslant d$ to the closed DBM $\mathbf{m}$. The four diagrams given in Figure 6 illustrate how the path between variables x'_i and x'_j can be shortened. The distance between x'_i and x'_j is c ($\mathbf{m}_{i,j} = c$), the distance between x'_i and x'_a is c_1 ($\mathbf{m}_{i,a} = c_1$), etc. The wavy lines denote the new constraints $x'_a - x'_b \leqslant d$ and $x'_{\bar{b}} - x'_{\bar{a}} \leqslant d$ and the heavy lines indicate short-circuiting paths between x'_i and x'_j. The bottom left path of the figure illustrates how the distance between x'_i and x'_a can be reduced from c_1 by the $x'_{\bar{b}} - x'_{\bar{a}} \leqslant d$ constraint. The same path illustrates how to shorten the distance between $x'_{\bar{a}}$ and x'_j from c'_2 using the $x'_a - x'_b \leqslant d$ constraint. The bottom right path of the figure gives two symmetric cases in which c'_1 and c_2 are sharpened by the addition of $x'_a - x'_b \leqslant d$

```
function MinéIncClose($\mathbf{m}$, $x'_a - x'_b \leqslant d$)
    $\mathbf{m}_{a,b} \leftarrow \min(\mathbf{m}_{a,b}, d)$
    $\mathbf{m}_{\bar{b},\bar{a}} \leftarrow \min(\mathbf{m}_{\bar{b},\bar{a}}, d)$
    $v \leftarrow \min(a, b, \bar{a}, \bar{b})$;
    for $k \in \{v, \dots, 2n-1\}$ do
        for $i \in \{0, \dots, 2n-1\}$ do
            for $j \in \{0, \dots, 2n-1\}$ do
                $\mathbf{m}_{i,j} \leftarrow \min(\mathbf{m}_{i,j}, \mathbf{m}_{i,k} + \mathbf{m}_{k,j})$
            end for
        end for
    end for
    return $\mathbf{m}$
end function
```

Fig. 5: Incremental Closure of Miné



(i, a) and (b, j) are not affected by new constraints

$(i, \bar{b})$ and $(\bar{a}, j)$ are not affected by new constraints

(i, a) shortened by $(i, \bar{b}) + d + (\bar{a}, a)$ or $(\bar{a}, j)$ shortened by $(\bar{a}, a) + d + (b, j)$

$(i, \bar{b})$ shortened by $(i, a) + d + (b, \bar{b})$ (b, j) shortened by $(b, \bar{b}) + d + (\bar{a}, j)$

Fig. 6: Four ways to reduce the distance between x'_i and x'_j

and $x'_{\bar{b}} - x'_{\bar{a}} \leqslant d$ respectively. Note that we cannot have the two paths from x'_i to x'_a and from x'_b to x'_j both shortened: at most one of them can change. The same holds for the two paths from x'_i to $x'_{\bar{b}}$ and $x'_{\bar{a}}$ to x'_j. These extra paths lead to the following strategy for updating $\mathbf{m'}_{i,j}$:

$$\mathbf{m'}_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix}$$

This leads to the incremental closure algorithm listed in top of Figure 7. Quintic min can be realised as four binary min operations, hence the total number of binary min operations required for IncClose is $16n^2$, which is quadratic in n. The listing in the bottom of the figure shows how commonality can be factored out so that each iteration of the inner loop requires a single ternary min to be computed. Factorisation reduces the number of binary min operations to $2n(2 + 4n) = 8n^2 + 4n$ in IncCloseHoist. Moreover, this form of code hoisting is also applicable algorithms that follow (though this optimisation is not elaborated in the sequel). Furthermore, like IncClose, IncCloseHoist is not sensitive to the specific traversal order of the DBM, hence has potential for parallelisation. In additional, both IncClose and IncCloseHoist do not incur any checks.

Example 6 To illustrate how the incremental closure algorithm of [10], from which the above is derived, omits a form of propagation, consider adding $x_0 - x_1 \leqslant 0$, or equivalently $x'_0 - x'_2 \leqslant 0$, to the system on the left

$$\begin{array}{l} x_0 \leqslant 7, \\ x_1 \leqslant 0, \\ x_0 - x_1 \leqslant 7, \\ x_0 + x_1 \leqslant 0 \end{array} \qquad \mathbf{m} = \begin{array}{cc} & \begin{array}{cccc} x'_0 & x'_1 & x'_2 & x'_3 \end{array} \\ \begin{array}{c} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{array} & \begin{bmatrix} 0 & 14 & 7 & 7 \\ \infty & 0 & \infty & \infty \\ \infty & 7 & 0 & 0 \\ \infty & 7 & \infty & 0 \end{bmatrix} \end{array}$$

whose DBM $\mathbf{m}$ is given on right. The system is illustrated spatially on the left hand side of Figure 8; the right hand side of the same figure shows the effect of adding the constraint $x_0 - x_1 \leqslant 0$. Adding $x_0 - x_1 \leqslant 0$ using the incremental closure algorithm from [10] gives the DBM $\mathbf{m'}$; IncClose gives the DBM $\mathbf{m''}$:

$$\mathbf{m'} = \begin{array}{cc} & \begin{array}{cccc} x'_0 & x'_1 & x'_2 & x'_3 \end{array} \\ \begin{array}{c} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{array} & \begin{bmatrix} 0 & 7 & 0 & 0 \\ \infty & 0 & \infty & \infty \\ \infty & 0 & 0 & 0 \\ \infty & 0 & \infty & 0 \end{bmatrix} \end{array} \qquad \mathbf{m''} = \begin{array}{cc} & \begin{array}{cccc} x'_0 & x'_1 & x'_2 & x'_3 \end{array} \\ \begin{array}{c} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ \infty & 0 & \infty & \infty \\ \infty & 0 & 0 & 0 \\ \infty & 0 & \infty & 0 \end{bmatrix} \end{array}$$

The DBM $\mathbf{m'}$ represents the constraint $x \leqslant \frac{7}{2}$ but $\mathbf{m''}$ encodes the tighter constraint $x \leqslant 0$. The reason for the discrepancy between entries $\mathbf{m'}_{0,1}$ and $\mathbf{m''}_{0,1}$ is shown by the following calculations:

$$\mathbf{m'}_{0,1} = \min \begin{pmatrix} \mathbf{m}_{0,1} \\ \mathbf{m}_{0,0} + 0 + \mathbf{m}_{2,1} \\ \mathbf{m}_{0,\bar{2}} + 0 + \mathbf{m}_{\bar{0},1} \end{pmatrix} = \min \begin{pmatrix} 14, \\ 0 + 0 + 7 \\ 7 + 0 + 0 \end{pmatrix} = 7$$

```
function IncClose($\mathbf{m}$, $x'_a - x'_b \leqslant d$)
    for $i \in \{0, \dots, 2n-1\}$ do
        for $j \in \{0, \dots, 2n-1\}$ do
            $\mathbf{m'}_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix}$
        end for
    end for
    if CheckConsistent($\mathbf{m'}$) then
        return $\mathbf{m'}$
    else
        return $false$
    end if
end function
```

```
function IncCloseHoist($\mathbf{m}$, $x'_a - x'_b \leqslant d$)
    $t_1 \leftarrow d + \mathbf{m}_{\bar{a},a} + d$;
    $t_2 \leftarrow d + \mathbf{m}_{b,\bar{b}} + d$;
    for $i \in \{0, \dots, 2n-1\}$ do
        $t_3 \leftarrow \min(\mathbf{m}_{i,a} + d, \mathbf{m}_{i,\bar{b}} + t_1)$;
        $t_4 \leftarrow \min(\mathbf{m}_{i,\bar{b}} + d, \mathbf{m}_{i,a} + t_2)$;
        for $j \in \{0, \dots, 2n-1\}$ do
            $\mathbf{m'}_{i,j} \leftarrow \min(\mathbf{m}_{i,j}, t_3 + \mathbf{m}_{b,j}, t_4 + \mathbf{m}_{\bar{a},j})$
        end for
        if $\mathbf{m'}_{i,i} < 0$ then
            return $false$
        end if
    end for
    return $\mathbf{m'}$
end function
```

Fig. 7: Incremental Closure (without and with code hoisting)

$$\mathbf{m''}_{0,1} = \min \begin{pmatrix} \mathbf{m}_{0,1} \\ \mathbf{m}_{0,0} + 0 + \mathbf{m}_{2,1} \\ \mathbf{m}_{0,\bar{2}} + 0 + \mathbf{m}_{\bar{0},1} \\ \mathbf{m}_{0,0} + 0 + \mathbf{m}_{2,\bar{2}} + 0 + \mathbf{m}_{\bar{0},1} \\ \mathbf{m}_{0,\bar{2}} + 0 + \mathbf{m}_{\bar{0},0} + 0 + \mathbf{m}_{2,1} \end{pmatrix} = \min \begin{pmatrix} 14 \\ 0 + 0 + 7 \\ 7 + 0 + 0 \\ 0 + 0 + 0 + 0 + 0 \\ 7 + 0 + \infty + 0 + 7 \end{pmatrix} = 0$$

The entry at $\mathbf{m'}_{0,1}$ is calculated using $\mathbf{m}_{2,1}$, but this entry will itself reduce to 0; $\mathbf{m'}_{0,1}$ must take into account the change that occurs to $\mathbf{m}_{2,1}$. More generally, when calculating $\mathbf{m'}_{i,j}$, the min expression of [10] overlooks how the added constraint can tighten $\mathbf{m}_{i,a}$, $\mathbf{m}_{i,b}$, $\mathbf{m}_{i,\bar{b}}$ or $\mathbf{m}_{\bar{a},j}$. □

The new incremental algorithm is justified by Theorem 4.1 which, in turn, is supported by the following lemma:

Lemma 4.1 *Suppose* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m'} = \text{IncClose}(\mathbf{m}, o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then* $\mathbf{m'}$ *is consistent if and only if:*

- $\mathbf{m}_{b,a} + d \geqslant 0$
- $\mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$

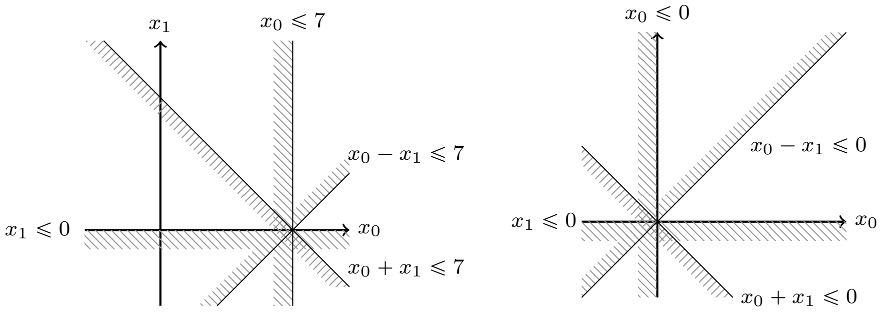


Fig. 8: Before and after adding $x_0 - x_1 \leqslant 0$

- $\mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d \geqslant 0$

Theorem 4.1 (Correctness of IncClose) *Suppose* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m}' =$ IncClose$(\mathbf{m}, o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then* $\mathbf{m}'$ *is either closed or it is not consistent.*

Note that unsatisfiability can be detected without applying any min operations at all, though for brevity this is omitted in the presentation of the algorithms. Fast unsatisfiability checking is justified by the following corollary of Lemma 4.1:

Corollary 4.1 *Suppose* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m}' =$ IncClose$(\mathbf{m}, o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. If*

- $\mathbf{m}_{b,a} + d < 0$ *or*
- $\mathbf{m}_{\bar{a},\bar{b}} + d < 0$ *or*
- $\mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d < 0$

then $\mathbf{m}'$ *is not consistent.*

4.3 Properties of Incremental Closure

By design IncClose recovers closure, but it should also be natural for the algorithm to preserve and enforce other properties too. These properties are not just interesting within themselves; they provide scaffolding for results that follow:

Proposition 4.1 *Suppose* $\mathbf{m} \leqslant \mathbf{m}'$ *(pointwise) and* $o = (x'_a - x'_b \leqslant d)$*. Then* IncClose$(\mathbf{m}, o) \leqslant$ IncClose$(\mathbf{m}', o)$*.*

Proposition 4.2 *Suppose* $\mathbf{m}$ *is coherent,* $\mathbf{m}' =$ IncClose$(\mathbf{m}, o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then* $\mathbf{m}'$ *is coherent.*

An important property of IncClose is idempotence: it formalises the idea that an octagon should not change shape if it is repeatedly intersected with the same inequality. If idempotence did not hold then there would exist $\mathbf{m}' =$ IncClose$(\mathbf{m}, o)$ and $\mathbf{m}'' =$ IncClose$(\mathbf{m}', o)$ for which $\mathbf{m}' \neq \mathbf{m}''$. This would suggest that IncClose did not properly tighten $\mathbf{m}$ using the inequality o, but overlooked some propagation, which is the form of suboptimal behaviour we are aiming to avoid.

Proposition 4.3 *Suppose that* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m}' = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}'' = \textsc{IncClose}(\mathbf{m}', o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then either* $\mathbf{m}'$ *is consistent and* $\mathbf{m}'' = \mathbf{m}'$ *or* $\mathbf{m}''$ *is not consistent.*

5 Incremental Strong Closure

We now turn our attention from recovering closure to recovering strong closure, which generates a canonical representation for any (non-empty) octagon.

5.1 Classical Strong Closure

The classical strong closure by Miné repeatedly invokes Str within the main Floyd-Warshall loop, but it was later shown by Bagnara et al. [2] that this was equivalent to applying Str just once after the main loop. The following theorem [2, Theorem 3] justifies this tactic, though the proofs we present have been revisited and streamlined:

Theorem 5.1 *Suppose* $\mathbf{m}$ *is a closed, coherent DBM and* $\mathbf{m}' = \textsc{Str}(\mathbf{m})$*. Then* $\mathbf{m}'$ *is a strongly closed DBM.*

5.2 Properties of Strong Closure

We establish a number of properties about Str which will be useful when we prove in-place versions of our incremental strong (and tight) closure algorithms.

Proposition 5.1 *Suppose* $\mathbf{m}$ *be a DBM and* $\mathbf{m}' = \textsc{Str}(\mathbf{m})$*. Then* $\mathbf{m}' = \textsc{Str}(\mathbf{m}')$*.*

Proposition 5.2 *Suppose* $\mathbf{m}^1 \leqslant \mathbf{m}^2$ *(pointwise). Then* $\textsc{Str}(\mathbf{m}^1) \leqslant \textsc{Str}(\mathbf{m}^2)$*.*

Proposition 5.3 *Suppose* $\mathbf{m}$ *is a DBM and* $\mathbf{m}' = \textsc{Str}(\mathbf{m})$*. Then* $\mathbf{m}' \leqslant \mathbf{m}$*.*

Proposition 5.4 *Suppose* $\mathbf{m}$ *is a closed, coherent DBM. Then* $\mathbf{m}' = \textsc{Str}(\mathbf{m})$ *is a coherent DBM.*

5.3 Incremental Strong Closure

Theorem 5.1 states that a strongly closed DBM can be obtained by calculating closure and then strengthening. This is realised by calling IncClose, from Figure 7, followed by a call to Str. Although this is conventional wisdom, it incurs two passes over the DBM: one by IncClose and the other by Str. The two passes can be unified by observing that strengthening $\mathbf{m}'$ critically depends on the entries $\mathbf{m}'_{i,\bar{\imath}}$ where $i \in \{0, \ldots, 2n-1\}$. Furthermore, these entries, henceforth called key entries, are themselves not changed by strengthening because:

$$\min(\mathbf{m}'_{i,\bar{\imath}}, (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\bar{\imath}},\bar{\imath}})/2) = \min(\mathbf{m}'_{i,\bar{\imath}}, (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{i,\bar{\imath}})/2) = \mathbf{m}'_{i,\bar{\imath}}$$

This suggests precomputing the key entries up front and then using them in the main loop of IncClose to strengthen on-the-fly. This insight leads to the algorithm

```
 1: function $\textsc{IncStrongClose}(\mathbf{m}, x'_a - x'_b \leqslant d)$
 2:     for $i \in \{0, \ldots, 2n-1\}$ do
 3:         $\mathbf{m}'_{i,\bar{\imath}} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix}$
 4:     end for
 5:     for $i \in \{0, \ldots, 2n-1\}$ do
 6:         for $j \in \{0, \ldots, 2n-1\}$ do
 7:             if $j \neq \bar{\imath}$ then
 8:                 $\mathbf{m}'_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j})/2 \end{pmatrix}$
 9:             end if
10:         end for
11:         if $\mathbf{m}'_{i,i} < 0$ then
12:             return $false$
13:         end if
14:     end for
15:     return $\mathbf{m}'$
16: end function
```

Fig. 9: Incremental Strong Closure

listed in Figure 9. Line 3 generates the key entries which are closed by construction and unchanged by strengthening. Once the key entries are computed, the algorithm iterates over the rest of the DBM, closing and simultaneously strengthening each entry $\mathbf{m}_{i,j}$ at line 8.

The total number of binary min operations required for IncStrongClose is $8n + 10n(2n-1) = 20n^2 - 2n$, which improves on following IncClose by Str, which requires $16n^2 + 4n^2 = 20n^2$. Furthermore, since $\mathbf{m}$ is coherent $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}} = \mathbf{m}_{\bar{a},\bar{\imath}} + d + \mathbf{m}_{i,\bar{b}} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}$ so that the quintic min on line 4 becomes quartic, reducing the min count for IncClose to $20n^2 - 4n$. Furthermore, the entry $\mathbf{m}_{i,\bar{\imath}}$ can be cached in a linear array $\mathbf{a}_i$ of dimension $2n$ and the expression $(\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j})/2$ in line 8 can be replaced with $(\mathbf{a}_i + \mathbf{a}_{\bar{\jmath}})/2$, thereby avoiding two lookups in a two-dimensional matrix. We omit the algorithm using array caching for space reasons as this is a simple change to Figure 9.

The following theorem justifies the correctness of the new incremental strong closure algorithm:

Theorem 5.2 (Correctness of IncStrongClose) *Suppose* $\mathbf{m}$ *is a DBM,* $\mathbf{m}' = \textsc{IncStrongClose}(\mathbf{m}, o)$*,* $\mathbf{m}^\dagger = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}^* = \textsc{Str}(\mathbf{m}^\dagger)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then* $\mathbf{m}' = \mathbf{m}^*$*.*

Code is duplicated in IncStrongClose in the assignments of $\mathbf{m}'_{i,\bar{\imath}}$ and $\mathbf{m}'_{i,j}$ on lines 3 and 8 respectively. Fig 10 shows how this can be factored out in that line 3 of IncStrongCloseMotion need only consider updates stemming from $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$. Moreover, the guard on line 7 of Fig 9 is eliminated but moving the remainder of the $\mathbf{m}'_{i,\bar{\imath}}$ calculation into the main loop. This increases the min count by $2n$ but reduces code size. This can potentially be a good exchange because min is itself essentially a check (though it can be implemented as straight-line code for machine integers [35]), and eliminating the guard from the main loop avoids $4n^2$ checks, giving a saving overall. However, putting asymptotic arguments aside, whether IncStrongCloseMotion outperforms IncStrongClose depends on the relative cost of the integer comparison on line 7 of Fig 9 to the comparison implicit in line 3 of Fig 10, which is performed in the underlying number system. The following result justifies this form of code motion:

Theorem 5.3 (Correctness of IncStrongCloseMotion) *Suppose* $\mathbf{m}$ *is a strongly closed, coherent DBM and let* $\mathbf{m}^* = \textsc{IncStrongClose}(\mathbf{m}, o)$ *where* $o = (x'_a - x'_b \leqslant d)$ *and*

$$\mathbf{m}''_{i,j} = \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2, \\ (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j})/2 \end{pmatrix}$$

Then either $\mathbf{m}^* = \mathbf{m}''$ *or* $\mathbf{m}^*$ *is not consistent and* $\mathbf{m}''$ *is not inconsistent.*

The force of the above result is that $\mathbf{m}'_{i,j}$ is only affected by a change to $\mathbf{m}'_{i,\bar{\imath}}$ via $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ or a change to $\mathbf{m}'_{\bar{\jmath},j}$ via $\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}$. Thus the initial loop on line 3, need only check whether $\mathbf{m}_{i,\bar{\imath}}$ is shortened by $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ in order to correctly update an arbitrary entry $\mathbf{m}_{i,j}$ in the loop on line 8. Note that $\mathbf{m}$ is not just required to be closed, but also strongly closed and coherent.

6 Incremental Tight Closure

The strong closure algorithms previously presented have to be modified to support integer octagonal constraints. If x_i is integral then $x_i \leqslant c$ can be tightened to $x_i \leqslant \lfloor c \rfloor$. Since $x_i \leqslant c$ is represented as the difference $x'_{2i} - x'_{2i+1} \leqslant 2c$, tightening is achieved by sharpening the difference to $x'_{2i} - x'_{2i+1} \leqslant 2\lfloor c/2 \rfloor$, so that the constant $2\lfloor c/2 \rfloor$ is even. This is achieved by applying Tighten($\mathbf{m}$), the code for which is given in Figure 11. As suggested by Figure 3, closure does not need to be reapplied after tightening to check for consistency; it is sufficient to check that $\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\imath},i} < 0$ [2], which is the role of CheckZConsistent($\mathbf{m}$). One subtlety that is worthy of note is that after running tighten($\mathbf{m}$) on a closed DBM $\mathbf{m}$, the resulting DBM will not necessarily be closed but will instead satisfy a weaker property, namely weak closure. Strong closure can be recovered from weak closure, however, by strengthening [2]. However, we do not use this approach in the sequel: instead we use tightening and strengthening together to avoid having to work with weakly closed DBMs. First we prove that tightening followed by strengthening will return a closed DBM when the resulting system is satisfiable:

```
function IncStrongCloseMotion($\mathbf{m}, x'_a - x'_b \leqslant d$)
    for $i \in \{0, \dots, 2n-1\}$ do
        $\mathbf{m}'_{i,\bar{\imath}} \leftarrow \min\begin{pmatrix}\mathbf{m}_{i,\bar{\imath}},\\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}\end{pmatrix}$
    end for
    for $i \in \{0, \dots, 2n-1\}$ do
        for $j \in \{0, \dots, 2n-1\}$ do
            $\mathbf{m}'_{i,j} \leftarrow \min\begin{pmatrix}\mathbf{m}_{i,j},\\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j},\\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j},\\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j},\\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j},\\ (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j})/2\end{pmatrix}$
        end for
        if $\mathbf{m}'_{i,i} < 0$ then
            return $false$
        end if
    end for
    return $\mathbf{m}'$
end function
```

Fig. 10: Incremental Strong Closure with code motion

Lemma 6.1 *Suppose* $\mathbf{m}$ *is a closed, coherent integer DBM. Let* $\mathbf{m}'$ *be defined as follows:*

$$\mathbf{m}'_{i,j} = \min(\mathbf{m}_{i,j}, \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor)$$

Then $\mathbf{m}'$ *is either closed or it is not consistent.*

Proof Suppose $\mathbf{m}'$ is consistent. Because $\mathbf{m}$ is closed $\mathbf{m}'_{i,i} \leqslant \mathbf{m}_{i,i} = 0$ and since $\mathbf{m}'$ is consistent $0 \leqslant \mathbf{m}'_{i,i}$ hence $\mathbf{m}'_{i,i} = 0$. Now to show $\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} \geqslant \mathbf{m}'_{i,j}$.

1. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Because $\mathbf{m}$ is closed:

$$\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} = \mathbf{m}_{i,k} + \mathbf{m}_{k,j} \geqslant \mathbf{m}_{i,j} \geqslant \mathbf{m}'_{i,j}$$

2. Suppose $\mathbf{m}'_{i,k} \neq \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$.
   (a) Suppose $\mathbf{m}_{\bar{k},k}$ is even. Because $\mathbf{m}$ is closed and coherent:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \mathbf{m}_{k,j} = \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{\bar{k},k} + 2\mathbf{m}_{k,j}}{2}\\ &\geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{\bar{k},j} + \mathbf{m}_{k,j}}{2} = \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{\bar{\jmath},k} + \mathbf{m}_{k,j}}{2}\\ &\geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{j,\bar{\jmath}}}{2} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{j,\bar{\jmath}}}{2} \right\rfloor \geqslant \mathbf{m}'_{i,j}\end{aligned}$$

   (b) Suppose $\mathbf{m}_{\bar{k},k}$ is odd. Then

$$\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} = \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \mathbf{m}_{k,j} = \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{(\mathbf{m}_{\bar{k},k} - 1) + 2\mathbf{m}_{k,j}}{2}$$

   Because $\mathbf{m}$ is closed and coherent:

$$\frac{(\mathbf{m}_{\bar{k},k} - 1) + 2\mathbf{m}_{k,j}}{2} \geqslant \frac{\mathbf{m}_{\bar{k},j} + \mathbf{m}_{k,j} - 1}{2} = \frac{\mathbf{m}_{\bar{\jmath},k} + \mathbf{m}_{k,j} - 1}{2} \geqslant \frac{\mathbf{m}_{\bar{\jmath},j} - 1}{2}$$

i. Suppose $\mathbf{m}_{\bar{k},k} + 2\mathbf{m}_{k,j} = \mathbf{m}_{\bar{j},j}$. Since $\mathbf{m}_{\bar{k},k}$ is odd $\mathbf{m}_{\bar{j},j}$ is odd thus

$$\frac{\mathbf{m}_{\bar{j},j} - 1}{2} = \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor \text{ and } \mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor \geqslant \mathbf{m}'_{i,j}$$

ii. Suppose $\mathbf{m}_{\bar{k},k} + 2\mathbf{m}_{k,j} > \mathbf{m}_{\bar{j},j}$. Thus $(\mathbf{m}_{\bar{k},k} - 1) + 2\mathbf{m}_{k,j} \geqslant \mathbf{m}_{\bar{j},j}$

$$\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{\bar{j},j}}{2} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor \geqslant \mathbf{m}'_{i,j}$$

3. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} \neq \mathbf{m}_{k,j}$. Symmetric to the previous case.
4. Suppose $\mathbf{m}'_{i,k} \neq \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} \neq \mathbf{m}_{k,j}$. Then

$$\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} = \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{k,\bar{k}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor$$

Since $\mathbf{m}$ is closed and $\mathbf{m}'$ is consistent:

$$0 \leqslant \mathbf{m}'_{\bar{k},\bar{k}} = \min(\mathbf{m}_{\bar{k},\bar{k}}, \left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{k,\bar{k}}}{2} \right\rfloor) = \min(0, \left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{k,\bar{k}}}{2} \right\rfloor)$$

Therefore

$$\left\lfloor \frac{\mathbf{m}_{\bar{k},k}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{k,\bar{k}}}{2} \right\rfloor \geqslant 0 \text{ and } \mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor \geqslant \mathbf{m}'_{i,j}$$

□

It should be noted that the above proof by-passes the notion of weak closure which was previously thought to be necessary [2, pages 28–31] greatly simplifying the proofs. Using the proof that tighten and strengthening gives a closed DBM, it can now be shown that the resulting DBM is also tightly closed:

Theorem 6.1 *([2, Theorem 4]) Suppose* $\mathbf{m}$ *is a closed, coherent integer DBM. Let* $\mathbf{m}'$ *be defined as follows:*

$$\mathbf{m}'_{i,j} = \min(\mathbf{m}_{i,j}, \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{j},j}}{2} \right\rfloor)$$

Then $\mathbf{m}'$ *is either tightly closed or it is not consistent.*

Notice that the proof of tight closure does not use the concept of weak closure as advocated in [2]. The above proof goes directly from a closed DBM to a tightly closed DBM relying only on simple algebra; it is not based on showing that tightening gives a weakly closed (intermediate) DBM which can be subsequently strengthen to give a tightly closed DBM (see Figure 3).

Tight closure requires the key entries, and only these, to be tightened. This suggests tightening the key entries on-the-fly immediately after they have been computed by closure. This leads to the algorithm given in Figure 12 which coincides with IncStrongClose($\mathbf{m}$) except in one crucial detail: line 4 tightens the key entries as they are computed. Moreover the key entries are strengthened, with the other entries of the DBM, in the main loop in tandem with the closure calculation, thereby ensuring strong closure. Thus tightening can be accommodated, almost effortlessly, within incremental strong closure.

Theorem 6.2 (Correctness of IncZClose) *Suppose* $\mathbf{m}$ *is an integer DBM and* $\mathbf{m}' =$ IncZClose$(\mathbf{m}, o)$ *where* $o = x'_a - x'_b \leqslant d$*. Let* $\mathbf{m}^{\dagger} =$ IncClose$(\mathbf{m}, o)$*,* $\mathbf{m}^{\ddagger} =$ Tighten$(\mathbf{m}^{\dagger})$ *and* $\mathbf{m}^{*} =$ Str$(\mathbf{m}^{\ddagger})$*. Then* $\mathbf{m}^{*} = \mathbf{m}'$*.*

```
function TIGHTEN($\mathbf{m}$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        $\mathbf{m}_{i,\bar{\imath}} \leftarrow 2\lfloor \mathbf{m}_{i,\bar{\imath}}/2 \rfloor$
    end for
end function
```

```
function TIGHTCLOSE($\mathbf{m}$)
    SHORTESTPATHCLOSURE($\mathbf{m}$)
    if CHECKCONSISTENT($\mathbf{m}$) then
        $\mathbf{m} \leftarrow$ TIGHTEN($\mathbf{m}$)
        if CHECKZCONSISTENT($\mathbf{m}$) then
            return STR($\mathbf{m}$)
        else
            return false
        end if
    else
        return false
    end if
end function
```

```
function CHECKZCONSISTENT($\mathbf{m}$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        if $\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\imath},i} < 0$ then
            return false
        end if
    end for
    return true
end function
```

Fig. 11: Tight Closure

6.1 Properties of Tight Closure

We prove a number of properties about TIGHTEN which will be useful when we justify the in-place versions of our incremental tight closure algorithm.

Proposition 6.1 *Suppose* $\mathbf{m}$ *is a DBM and* $\mathbf{m}' = \text{TIGHTEN}(\mathbf{m})$*. Then* $\mathbf{m}' = \text{TIGHTEN}(\mathbf{m}')$*.*

Proposition 6.2 *Suppose* $\mathbf{m}^1 \leqslant \mathbf{m}^2$ *(pointwise). Then* $\text{TIGHTEN}(\mathbf{m}^1) \leqslant \text{TIGHTEN}(\mathbf{m}^2)$*.*

Proposition 6.3 *Suppose* $\mathbf{m}$ *is a DBM and* $\mathbf{m}' = \text{TIGHTEN}(\mathbf{m})$*. Then* $\mathbf{m}' \leqslant \mathbf{m}$*.*

Proposition 6.4 *Let* $\mathbf{m}$ *be a coherent DBM and* $\mathbf{m}' = \text{TIGHTEN}(\mathbf{m})$*. Then* $\mathbf{m}'$ *is coherent.*

7 In-place Update

Closure algorithms are traditionally formulated in a way that is simple to reason about mathematically (see [21, Def 3.3.2]), typically using a series of intermediate DBMs and then present the algorithm itself using in-place update (see [21, Def 3.3.3]). An operation on a DBM will conceptually calculate an output DBM from the input DBM. Since this requires two DBMs, the input and the output, to be stored simultaneously, it is attractive to mutate the input DBM to derive the output DBM. This is called in-place update. The subtlety of in-place update, in the context of a DBM operation, is that one element can be calculated in terms of others, some of which may have already been updated. The question of equivalence between the mathematical formulation and the practical in-place implementation is arguably not given the space it should. Miné, in his magnus opus [21], merely states that equivalence can be shown by using an argument for the Floyd-Warshall algorithm [11, Section 26.2]. However that in-place argument

```
function IncZClose($\mathbf{m}$, $x'_a - x'_b \leqslant d$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        $\mathbf{m'}_{i,\bar{\imath}} \leftarrow 2 \left\lfloor \min \begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix} /2 \right\rfloor$
    end for
    if CheckZConsistent($\mathbf{m'}$) then
        for $i \in \{0, \ldots, 2n-1\}$ do
            for $j \in \{0, \ldots, 2n-1\}$ do
                if $j \neq \bar{\imath}$ then
                    $\mathbf{m'}_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2 \end{pmatrix}$
                end if
            end for
            if $\mathbf{m'}_{i,i} < 0$ then
                return $false$
            end if
        end for
    else
        return $false$
    end if
    return $\mathbf{m'}$
end function
```

Fig. 12: Incremental Tight Closure

is itself informal. Later editions of the book do not help, leaving the proof as an exercise for the reader. But the question of equivalence is more subtle again for incremental closure. Correctness is therefore argued for incremental closure in Section 7.1, incremental strong closure in Section 7.2 and incremental tight closure in Section 7.3, one correctness argument extending another.

7.1 In-place Incremental Closure

Figure 13 gives an in-place version of IncClose algorithm listed in Figure 7. At first glance one might expect that mutating the entries $\mathbf{m}_{i,a}$, $\mathbf{m}_{b,\bar{\imath}}$, $\mathbf{m}_{i,\bar{b}}$, $\mathbf{m}_{\bar{a},\bar{\imath}}$, $\mathbf{m}_{\bar{a},a}$ or $\mathbf{m}_{b,\bar{b}}$ could potentially perturb those entries of $\mathbf{m}$ which are updated later. The following theorem asserts that this is not so. Correctness follows from Corollary 7.1 which is stated below:

Corollary 7.1 *Suppose that* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m'} = \text{IncClose}(\mathbf{m}, o)$, $o = (x'_a - x'_b \leqslant d)$ *and* $\mathbf{m'}$ *is consistent. Then the following hold:*

- $\mathbf{m'}_{i,j} \leqslant \mathbf{m'}_{i,a} + d + \mathbf{m'}_{b,j}$
- $\mathbf{m'}_{i,j} \leqslant \mathbf{m'}_{i,\bar{b}} + d + \mathbf{m'}_{\bar{a},j}$
- $\mathbf{m'}_{i,j} \leqslant \mathbf{m'}_{i,\bar{b}} + d + \mathbf{m'}_{\bar{a},a} + d + \mathbf{m'}_{b,j}$
- $\mathbf{m'}_{i,j} \leqslant \mathbf{m'}_{i,a} + d + \mathbf{m'}_{b,\bar{b}} + d + \mathbf{m'}_{\bar{a},j}$

```
function InplaceIncClose($\mathbf{m}$, $x'_a - x'_b \leqslant d$)
    for $i \in \{0,\ldots,2n-1\}$ do
        for $j \in \{0,\ldots,2n-1\}$ do
            $\mathbf{m}_{i,j} \leftarrow \min\begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix}$
        end for
        if $\mathbf{m}_{i,i} < 0$ then
            return $false$
        end if
    end for
    return m
end function
```

Fig. 13: In-place Incremental Closure

The following theorem asserts that in-place update does not compromise correctness. It is telling that the correctness argument does not refer to the entries $\mathbf{m}_{i,a}$, $\mathbf{m}_{b,\bar{\imath}}$, $\mathbf{m}_{i,\bar{b}}$, $\mathbf{m}_{\bar{a},\bar{\imath}}$, $\mathbf{m}_{\bar{a},a}$ or $\mathbf{m}_{b,\bar{b}}$ at all. This is because the corollary on which the theorem is founded follows from the high-level property of idempotence. Notice too that the theorem is parameterised by the traversal order over **m** and therefore is independent of it.

Theorem 7.1 (Correctness of InplaceIncClose) *Suppose* $\rho : \{0,\ldots,2n-1\}^2 \to \{0,\ldots,4n^2-1\}$ *is a bijective map,* $\mathbf{m}$ *is a closed DBM,* $\mathbf{m}' = \text{IncClose}(\mathbf{m}, o)$, $o = (x'_a - x'_b \leqslant d)$, $\mathbf{m}^0 = \mathbf{m}$ *and*

$$\mathbf{m^{k+1}}_{i,j} = \begin{cases} \mathbf{m^k}_{i,j} & \text{if } \rho(i,j) \neq k \\ \min\begin{pmatrix} \mathbf{m^k}_{i,j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j} \end{pmatrix} & \text{if } \rho(i,j) = k \end{cases}$$

Then either $\mathbf{m}'$ *is consistent and*

- $\forall 0 \leqslant \ell < k . \mathbf{m^k}_{\rho^{-1}(\ell)} = \mathbf{m}'_{\rho^{-1}(\ell)}$
- $\forall k \leqslant \ell < 4n^2 . \mathbf{m^k}_{\rho^{-1}(\ell)} = \mathbf{m}_{\rho^{-1}(\ell)}$

or $\mathbf{m^{4n^2}}$ *is inconsistent.*

7.2 In-place Incremental Strong Closure

The in-place version of the incremental strong closure algorithm is presented in Figure 14. The following lemma shows that running incremental closure followed by strengthening refines the entries in the DBM to their tightest possible value with respect to the new octagonal constraint.

```
function InplaceIncStrongClose($\mathbf{m}, x'_a - x'_b \leqslant d$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        $\mathbf{m}_{i,\bar{\imath}} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix}$
    end for
    for $i \in \{0, \ldots, 2n-1\}$ do
        for $j \in \{0, \ldots, 2n-1\}$ do
            if $j \neq \bar{\imath}$ then
                $\mathbf{m}_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{j},j})/2 \end{pmatrix}$
            end if
        end for
        if $\mathbf{m}_{i,i} < 0$ then
            return $false$
        end if
    end for
    return m
end function
```

Fig. 14: In-place Incremental Strong Closure

Lemma 7.1 *Suppose* $\mathbf{m}$ *is a closed, coherent DBM and* $\mathbf{m}' = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}'' = \textsc{Str}(\mathbf{m}')$*,* $\mathbf{m}''' = \textsc{IncClose}(\mathbf{m}'', o)$ *and* $o = (x'_a - x'_b \leqslant d)$*. Then either* $\mathbf{m}'$ *is consistent and* $\mathbf{m}''' = \mathbf{m}''$ *or* $\mathbf{m}'''$ *is not consistent.*

Now we move onto the theorem showing the correctness of InplaceIncStrongClose. We show that the in-place version of the algorithm produces the same DBM as the non-in place version of the algorithm. A bijective map used in the proof to process key entries first before processing non-key entries: the condition $\forall 0 \leqslant i < 2n.\rho(i,\bar{\imath}) < 2n$ ensures this property. Note that this is the only caveat on the order produced by the map: the order in which key entries themselves are ordered is irrelevant and similarly for non-key entries.

Theorem 7.2 (Correctness of InplaceIncStrongClose) *Suppose* $\mathbf{m}$ *is a closed, coherent DBM,* $\mathbf{m}' = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}'' = \textsc{Str}(\mathbf{m}')$*,* $o = (x'_a - x'_b \leqslant d)$*,* $\rho : \{0, \ldots, 2n-1\}^2 \to \{0, \ldots, 4n^2-1\}$ *is a bijective map with* $\forall 0 \leqslant i < 2n.\rho(i,\bar{\imath}) < 2n$*,*

$\mathbf{m}^0 = \mathbf{m}$ *and*

$$\mathbf{m^{k+1}}_{i,j} = \begin{cases} \mathbf{m^k}_{i,j} & \textit{if } \rho(i,j) \neq k \\ \min \begin{pmatrix} \mathbf{m^k}_{i,\bar{\imath}}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{\imath}}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},\bar{\imath}}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,\bar{\imath}}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},\bar{\imath}} \end{pmatrix} & \textit{if } \rho(i,j) = k \wedge j = \bar{\imath} \\ \min \begin{pmatrix} \mathbf{m^k}_{i,j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j}, \\ (\mathbf{m^k}_{i,\bar{\imath}} + \mathbf{m^k}_{\bar{\jmath},j})/2 \end{pmatrix} & \textit{if } \rho(i,j) = k \wedge j \neq \bar{\imath} \end{cases}$$

Then either $\mathbf{m}'$ *is consistent and*

- $\forall 0 \leqslant \ell < k . \mathbf{m}^{\mathbf{k}}_{\rho^{-1}(\ell)} = \mathbf{m}''_{\rho^{-1}(\ell)}$
- $\forall k \leqslant \ell < 4n^2 . \mathbf{m}^{\mathbf{k}}_{\rho^{-1}(\ell)} = \mathbf{m}_{\rho^{-1}(\ell)}$

or $\mathbf{m^{4n^2}}$ *is inconsistent.*

7.3 In-place Incremental Tight Closure

The in-place version of the incremental tight closure algorithm is presented in Figure 14, the only difference with incremental strong closure is that for key entries we also run a tightening step (line 3). As in the previous section, we have a helper lemma for the main theorem, showing that incremental closure followed by tightening and strengthening refines the entries in the DBM to the tightest value with respect to the new octagonal constraint.

Lemma 7.2 *Suppose* $\mathbf{m}$ *is a closed, coherent DBM and* $\mathbf{m}' = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}'' = \textsc{Tighten}(\mathbf{m}', o)$*,* $\mathbf{m}''' = \textsc{Str}(\mathbf{m}'', o)$*,* $\mathbf{m}^* = \textsc{IncClose}(\mathbf{m}''', o)$ *and* $\mathbf{m}^* = \mathbf{m}'''$ *or* $\mathbf{m}^*$ *is inconsistent.*

The following theorem is analogous to the theorem for in-place strong closure:

Theorem 7.3 (Correctness of InplaceIncZClose) *Suppose* $\mathbf{m}$ *is a closed, coherent DBM,* $\mathbf{m}' = \textsc{IncClose}(\mathbf{m}, o)$*,* $\mathbf{m}'' = \textsc{Tighten}(\mathbf{m}')$*,* $\mathbf{m}''' = \textsc{Str}(\mathbf{m}')$*,* $o = (x'_a - x'_b \leqslant d)$*, that* $\rho : \{0, \ldots, 2n-1\}^2 \to \{0, \ldots, 4n^2 - 1\}$ *is a bijective map with*

```
function InplaceIncZClose($\mathbf{m}, x'_a - x'_b \leqslant d$)
    for $i \in \{0, \ldots, 2n-1\}$ do
        $\mathbf{m}_{i,\bar{\imath}} \leftarrow 2 \left\lfloor \min \begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix} /2 \right\rfloor$
    end for
    if CheckZConsistent($\mathbf{m}'$) then
        for $i \in \{0, \ldots, 2n-1\}$ do
            for $j \in \{0, \ldots, 2n-1\}$ do
                if $j \neq \bar{\imath}$ then
                    $\mathbf{m}_{i,j} \leftarrow \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2 \end{pmatrix}$
                end if
            end for
            if $\mathbf{m}_{i,i} < 0$ then
                return $false$
            end if
        end for
    else
        return $false$
    end if
    return m
end function
```

Fig. 15: In-place Incremental Tight Closure

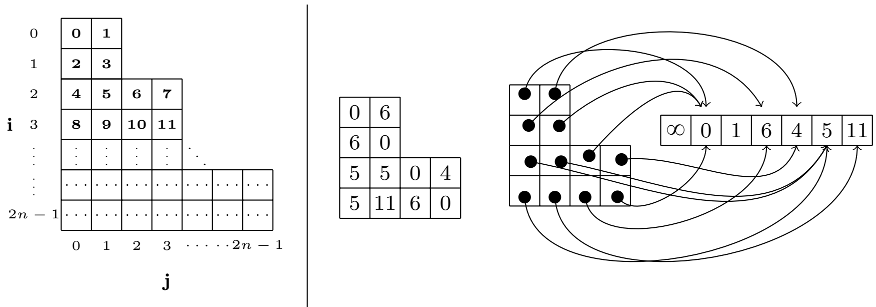
$\forall 0 \leqslant i < 2n.\rho(i, \bar{\imath}) < 2n$, $\mathbf{m}^0 = \mathbf{m}$ and

$$\mathbf{m^{k+1}}_{i,j} = \begin{cases} \mathbf{m^k}_{i,j} & \textit{if } \rho(i,j) \neq k \\ 2 \left\lfloor \min \begin{pmatrix} \mathbf{m^k}_{i,\bar{\imath}}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{\imath}}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},\bar{\imath}}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,\bar{\imath}}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},\bar{\imath}} \end{pmatrix} /2 \right\rfloor & \textit{if } \rho(i,j) = k \wedge j = \bar{\imath} \\ \min \begin{pmatrix} \mathbf{m^k}_{i,j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j}, \\ (\mathbf{m^k}_{i,\bar{\imath}} + \mathbf{m^k}_{\bar{\jmath},j})/2 \end{pmatrix} & \textit{if } \rho(i,j) = k \wedge j \neq \bar{\imath} \end{cases}$$

Then either $\mathbf{m}'$ is consistent and

- $\forall 0 \leqslant \ell < k.\mathbf{m}^{\mathbf{k}}_{\rho^{-1}(\ell)} = \mathbf{m}'''_{\rho^{-1}(\ell)}$
- $\forall k \leqslant \ell < 4n^2.\mathbf{m}^{\mathbf{k}}_{\rho^{-1}(\ell)} = \mathbf{m}_{\rho^{-1}(\ell)}$

or $\mathbf{m^{4n^2}}$ is inconsistent.



(1) (2)

Fig. 16: (1) Representing a DBM as an array; (2) Representing a DBM as a CoDBM

8 Experimental Evaluation

For a fair and robust evaluation, the algorithms were implemented using machinery provided in the Apron library [17]. The library provides implementations of the box, polyhedra and octagon abstract domains, the latter used for purposes of comparison. Apron is realised in C, and provides bindings for OCaml, C++ and Java. IncClose and IncStrongClose where then compared against the optimised implementation of incremental closure provided by Apron. Three sets of experiments were performed. First, the closure algorithms were applied to randomly generated DBMs, subject to various size constraints, to systematically exercise the algorithms on a range of problem size. Henceforth these randomly generated problems will be referred to as the micro-benchmarks. Second, to investigate the performance of the algorithms in a real-world setting, the algorithms were integrated into Frama-C, which is an industrial-strength static analysis tool for C code. The tool was then applied to a collection of C programs drawn from the Frama-C benchmarks repository. Third, the algorithms were integrated into AbSolute solver [26] and evaluated against benchmarks drawn from continuous constraint programming.

All experiments were performed on a 32-core Intel Xeon workstation with 128GB of memory.

8.1 Apron Library

The Apron library [17] supports various number systems, such as single precision floating-point numbers and GNU multiple-precision (GMP) rationals. The default number system for the OCaml bindings is rationals, but it must be appreciated that the computational overhead of allocating memory for the rationals dominates the runtime, potentially masking the benefits of IncStrongClose over IncClose. (Recall that IncStrongClose saves a separate pass over the DBM relative to IncClose, avoiding counter increments and integer comparisons.)

In Apron, numbers are represented by a type `bound_t`, which depending on compile-time options, will select a specific header file containing concrete implementations of operations involving numbers extended to the symbolic values of $-\infty$ and $+\infty$. Every `bound_t` object has to be initialised via a call to `bound_init`, which in the case of rationals will invoke malloc and initialise space for the rational number. DBMs are stored taking advantage of the half-matrix nature of octagonal DBMs which follows by the definition of coherence. An array of bound_t objects is then used to represent the half-matrix, as shown in figure 16, subfigure (1). If $i \geqslant j$ or $i = \bar{\jmath}$ then the entry at (i, j) in the DBM is stored at index $j + \lfloor i^2/2 \rfloor$ in the array. Otherwise (i, j) is stored at the array element reserved for entry $(\bar{\jmath}, \bar{\imath})$. A DBM of size n requires an array of size $2n(n + 1)$ which gives a significant space reduction over a naive representation of size $4n^2$.

8.2 Compact DBMs

Unexpectedly, initial experiments with Frama-C suggested that much of its runtime was spent in memory management rather than the domain operations themselves. Further investigation using Callgrind showed that 36% of all function calls emanated from malloc-like routines. In response, the underlying DBM data structure was refactored to ensure that this undesirable memory management feature did not artificially perturb the experiments. The refactoring is fully described in a separate work [9], but to keep the paper self-contained the main idea is summarised in the following paragraph.

The DBM representation was changed from a matrix storing numbers to a matrix storing pointers to numbers stored in a cache. This reduces the amount of memory used by a DBM as shown in figure 16. The modified data structure has been dubbed a compact DBM or CoDBM for short. The cache is an array initialised to contain ∞ as its first entry, augmented with an ordered table which enables the pointer for any given number to be found (if it exists) in the cache using the bisection search method. As new numbers are created they are added to the cache, and the table is extended in sync. This representation which, crucially, factors out the overhead of storing a number repeatedly, has a significant impact on the memory usage of the Apron library. It also rebalances the proportion of time spend in domain operations. Further performance debugging of Frama-C, for instance to speed up parsing, would only increase the fraction of time spend on the domain operations and closure in particular.

8.3 Micro-benchmarks

Each micro-benchmark suite was a collection of 10 problems, each problem consisting of a random octagon and a randomly generated octagonal constraint. Each random octagon was generated from a prescribed number of octagonal constraints, so as to always contain the origin, for a given number for variables. Each octagon was then closed. A single randomly generated octagonal constraint, not necessarily containing the origin, was then added to the closed octagon using incremental closure. IncClose and IncStrongClose where then timed and compared against the Apron version for DBMs over rationals. The resulting DBMs were then all checked for equality against Close. All timings were averaged over 10 runs and, moreover, all algorithms were exercised on exactly the same collection of problems. Fig. 17 presents timings for the micro-benchmark suites. The results show

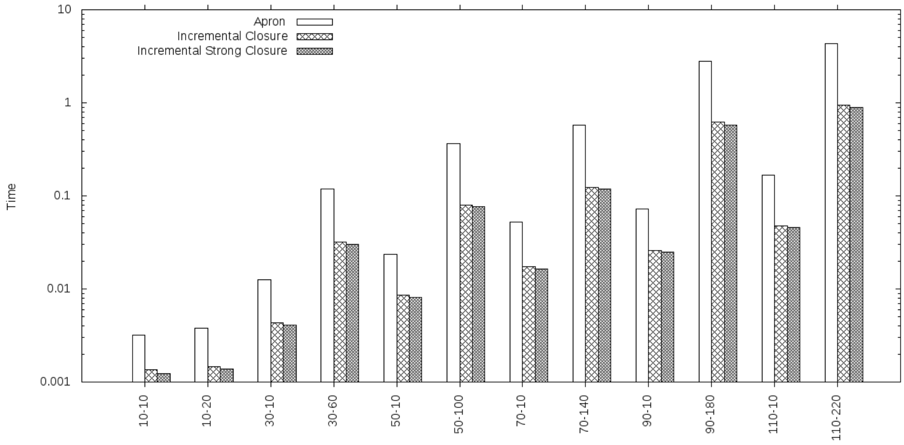


Fig. 17: Micro-benchmark timings for rationals

Name	Benchmark	LOC	Description
lev	levenstein	187	Levenstein string distance library
sol	solitaire	334	card cipher
2048	2048	435	2048 game
kh	khash	652	hash code from klib C library
taes	Tiny-AES	813	portable AES-128 implementation
qlz	qlz	1168	fast compression library
mod	libmodbus	7685	library to interact with Modbus protocol
mgmp	mini-gmp	11787	subset of GMP library
unq	unqlite	64795	embedded NoSQL DB
bzip	bzip-single-file	74017	bzip single file for static analysis benchmarking

Table 1: Benchmark suite of C programs

that IncClose outperforms the original Apron implementation by a factor of 3–4 and IncStrongClose offers an additional 4–9% speedup over IncClose.

8.4 Frama-C Benchmarks

The EVA plugin of Frama-C implements an abstract interpreter over the internal intermediate language used by Frama-C. The plugin uses the Apron library to perform an octagon domain analysis, and so by modifying Apron, Frama-C can make direct use of IncClose and IncStrongClose (and specially their IncCloseHoist and IncStrongCloseMotion variants). Table 1 lists the benchmark programs passed to EVA to interpret the programs over the octagon domain. It should be noted that EVA is a prototype (which may explain its memory behaviour) and as such does not use widely used heuristics and optimisations such as variable packing [8, 16] or localisation techniques [5, 25] to enable the analysis to scale. Nonetheless, the octagon analysis successfully terminated over the selected benchmarks.

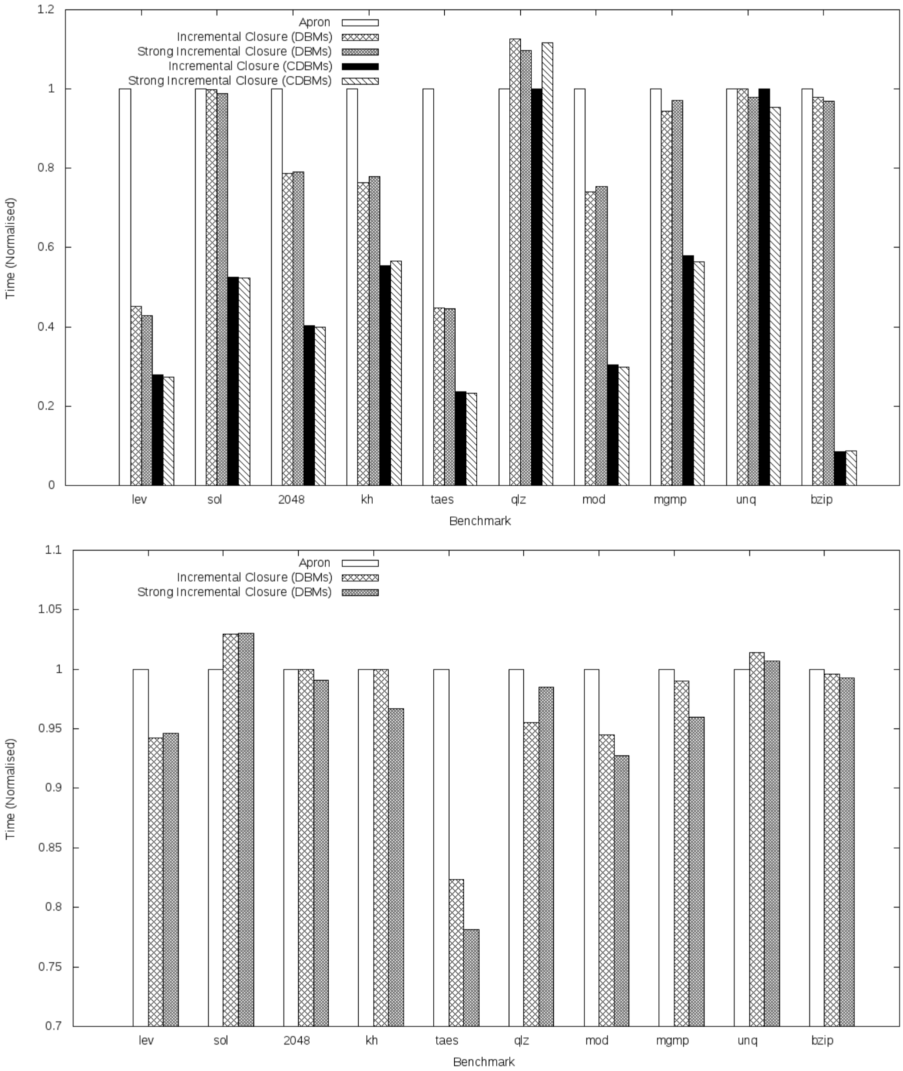


Fig. 18: Normalised timings of Frama-C for rationals (above) and floating point (below)

Figure 18 gives the timings of benchmarks for rational (above) and floating point arithmetic (below), normalised to the time required by the Apron implementation. For rationals, normalised timings are given for both DBMs and CoDBMs. The relative speedup obtained from deploying IncClose and IncStrongClose over Apron algorithm is variable, ranging from a large speedup for taes to a modest slowdown for qlz. Table 2 amplifies the relative timings presented in the bar chart, giving the exact timings in seconds. The table shows that the longest running analyses (which correspond to those employing the largest DBMs) are best served by IncClose and IncStrongClose.

		DBM		CoDBM	
Benchmark	Apron	IncClose	IncStrongClose	IncClose	IncStrongClose
lev	33.16	14.98	14.21	9.23	9.05
sol	49.80	49.76	49.19	26.17	26.03
2048	33.16	26.10	26.26	13.39	13.23
kh	1.80	1.37	1.40	1.00	1.02
taes	1817.91	814.77	810.00	430.60	421.32
qlz	1.08	1.21	1.18	1.08	1.20
mod	463.46	343.05	349.62	141.17	138.60
mgmp	2.09	1.97	2.03	1.21	1.18
unq	1.49	1.49	1.46	1.49	1.42
bzip	621.53	607.88	602.78	53.51	52.63
cumulative	3025.48	1862.58	1858.13	678.85	665.68

		DBM	
Benchmark	Apron	IncClose	IncStrongClose
lev	2.61	2.46	2.47
sol	12.62	12.99	13.00
2048	4.48	4.48	4.44
kh	0.60	0.60	0.58
taes	113.26	93.26	88.47
qlz	1.35	1.29	1.33
mod	57.59	54.43	53.41
mgmp	1.00	0.99	0.96
unq	1.44	1.46	1.45
bzip	22.69	22.60	22.52
cumulative	217.64	194.56	188.63

Table 2: Absolute timings of Frama-C for rationals (above) and floating point (below)

Cachegrind [23] profiling sheds light on qlz: some of the refined incremental algorithms actually increase the number of first-level data cache misses, giving a net slowdown. This cache anomaly might arise because the DBMs generated by qlz are tiny. Cachegrind also suggests this is the exception, revealing that the large speedup on bzip, mod and taes for CoDBMs over DBMs stems from a reduction in the number of misses to level 3 unified data and instruction cache. In fact, for bzip, mod and taes, the number of level 3 cache misses is reduced to zero. This validates the CoDBM data-structure. It also illustrates that optimisations which match the architecture can have surprising impact.

Floating point arithmetic is much faster than rationals, so the proportion of the overall execution time spent in closure is decreased, hence one would expect the relative speedup from IncClose and IncStrongClose over Apron to be likewise reduced. Figure 18 and table 2 shows that this is the general pattern. CoDBMs timings are not given for floats because floats have a much denser representation than GMP rationals. Nevertheless, the longest running analysis, which arises on taes, significantly benefits from both IncClose and IncStrongClose.

8.5 AbSolute Constraint Solver Benchmarks

The AbSolute constraint solver [26] applies principles from abstract interpretation to continuous constraint programming. Continuous constraint programming uses interval approximations to approximate solutions to continuous constraints: in

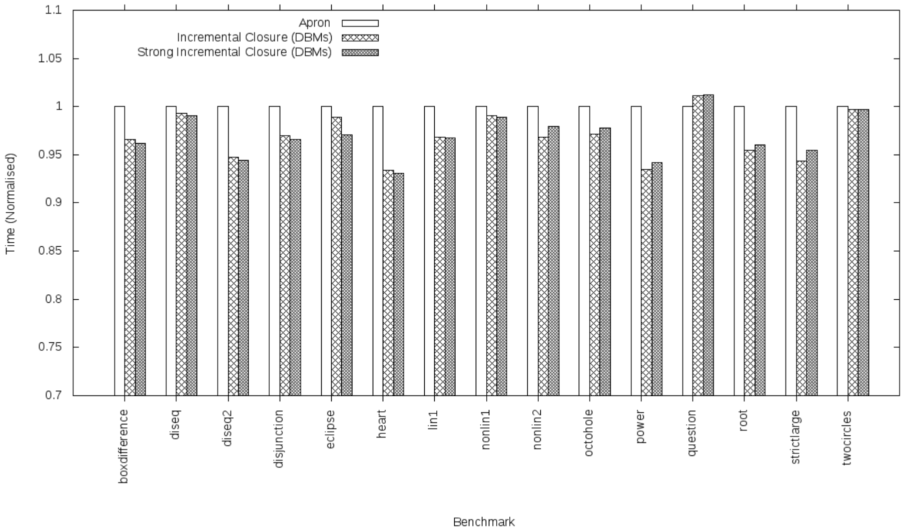


Fig. 19: Normalised timings for the AbSolute constraint solver (doubles)

Benchmark	Apron	IncClose	IncStrongClose
boxdifference	8.72	8.42	8.39
diseq	18.25	18.11	18.07
diseq2	15.62	14.80	14.75
disjunction	4.30	4.17	4.15
eclipse	42.39	41.91	41.14
heart	1014.13	947.32	944.04
lin1	12.15	11.77	11.76
nonlin1	2.38	2.35	2.35
nonlin2	4.05	3.92	3.97
octo_hole	2.11	2.05	2.06
power	24.57	22.97	23.13
question	5.30	5.36	5.37
root	13.53	12.92	13.00
strict_large	4.52	4.26	4.31
two_circles	11.99	11.95	11.95
cumulative	1192.38	1112.28	1108.44

Table 3: Absolute timing for the AbSolute constraint solver (doubles)

essence a solution enclosed by a single interval is successively refined to a set of intervals covering the solution (provided one exists).

The AbSolute solver deploys octagons rather than intervals to obtain a more precise and scalable solver. It uses Apron to implement its abstract domain operations, working over floats rather than rationals. The benchmarks selected to exercise AbSolute are a strict subset of those contained in the AbSolute repository (some problems fail to parse while others contain trigonometric functions not supported by the Apron library).

Figure 19 summaries the relative performance of Apron, IncClose and IncStrongClose; Table 3 gives the exact timings in seconds. All but one benchmarks show an improvement with IncClose and IncStrongClose, even though the size of the DBMs are small compared to those that arise in the Frama-C benchmarks.

9 Concluding Discussion

The octagon domain is used for many applications due to its expressiveness and its easy of implementation, relative to other relational abstract domains. Yet the elegance of their domain operations is at odds with the subtlety of the underlying ideas, and the reasoning needed to justify refinements that appear to be straightforward, such as tightening and in-place update.

This paper has presented novel algorithms to incrementally update an octagonal constraint system. More specifically, we have developed new incremental algorithms for closure, strong closure and integer closure, and their in-place variants. Experimental results with a prototype implementation demonstrate significant speedups over existing closure algorithms. We leave as future work the generalisation of the in-place update results to parallel evaluation and the application of our incremental algorithms for modelling machine arithmetic [31] in binary analysis which, incidentally, was the problem that motivated this thread of work.

Acknowledgements We thank Jacques-Henri Jourdan for stimulating discussions on Verasco at POPL'16 in St. Petersburg and the "Verified Trustworthy Software Systems" Royal Society event in London.

A Proof Appendix

A.1 Proofs for the Correctness of Incremental Closure

Proof (for lemma 4.1) We first prove the if case: since $\mathbf{m}'$ is consistent $\mathbf{m}'_{\bar{a},\bar{a}} \geqslant 0$ hence

$$\min \begin{pmatrix} \mathbf{m}_{\bar{a},\bar{a}}, \\ \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{a}}, \\ \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{a}}, \\ \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{a}}, \\ \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{a}} \end{pmatrix} = \mathbf{m}'_{\bar{a},\bar{a}} \geqslant 0$$

Therefore $\mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{a}} \geqslant 0$ and $\mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{a}} \geqslant 0$. Since $\mathbf{m}$ is closed $\mathbf{m}_{\bar{a},\bar{a}} = 0$ hence $\mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$ and $\mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d \geqslant 0$.

Repeating the argument $\mathbf{m}'_{b,b} \geqslant 0$ hence

$$\min \begin{pmatrix} \mathbf{m}_{b,b}, \\ \mathbf{m}_{b,a} + d + \mathbf{m}_{b,b}, \\ \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},b}, \\ \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,b}, \\ \mathbf{m}_{b,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},b} \end{pmatrix} = \mathbf{m}'_{b,b} \geqslant 0$$

Therefore $\mathbf{m}_{b,a} + d + \mathbf{m}_{b,b} \geqslant 0$. Since $\mathbf{m}_{b,b} = 0$ it follows that $\mathbf{m}_{b,a} + d \geqslant 0$.
Now suppose that $\mathbf{m}_{b,a} + d \geqslant 0$, $\mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$ and $\mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d \geqslant 0$. To show consistency we need to show that $\forall i . \mathbf{m}'_{i,i} \geqslant 0$. Pick an arbitrary i, then:

$$\mathbf{m}'_{i,i} = \min \begin{pmatrix} \mathbf{m}_{i,i}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,i}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},i}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,i}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},i} \end{pmatrix}$$

We will show that $\mathbf{m'}_{i,i} \geqslant 0$. Recall that $\mathbf{m}$ is closed, and thus the second line above simplifies to: $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,i} \geqslant \mathbf{m}_{b,a} + d \geqslant 0$. Similarly the third line: $\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},i} \geqslant \mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$, the fourth line :$\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,i} \geqslant \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d$ and the fifth line: $\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},i} \geqslant \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d \geqslant 0$. Thus every entry in the min expression is greater than 0 and thus $\forall i.\mathbf{m}_{i,i} \geqslant 0$ as required. □

Proof (for theorem 4.1) Suppose $\mathbf{m'}$ is consistent. Because $\mathbf{m}$ is closed $0 = \mathbf{m}_{i,i} \geqslant \mathbf{m'}_{i,i} \geqslant 0$ hence $\mathbf{m'}_{i,i} = 0$. It therefore remains to show $\forall i, j, k.\mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} \geqslant A$ where

$$A = \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix}$$

There are 5 cases for $\mathbf{m'}_{i,k}$ and 5 for $\mathbf{m'}_{k,j}$ giving 25 in total:

1-1. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,j}$. Because $\mathbf{m}$ is closed:

$$\mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} = \mathbf{m}_{i,k} + \mathbf{m}_{k,j} \geqslant \mathbf{m}_{i,j} \geqslant A$$

1-2. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed:

$$\mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} = \mathbf{m}_{i,k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j} \geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \geqslant A$$

1-3. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed:

$$\mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} = \mathbf{m}_{i,k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A$$

1-4. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed:

$$\begin{aligned} \mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} &= \mathbf{m}_{i,k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \geqslant A \end{aligned}$$

1-5. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed:

$$\begin{aligned} \mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} &= \mathbf{m}_{i,k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A \end{aligned}$$

2-1. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,j}$. Symmetric to case 1-2.

2-2. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned} \mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} &= \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,a} + d + \mathbf{m}_{b,j} \geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \geqslant A \end{aligned}$$

2-3. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed:

$$\begin{aligned} \mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} &= \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A \end{aligned}$$

2-4. Suppose $\mathbf{m'}_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m'}_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned} \mathbf{m'}_{i,k} + \mathbf{m'}_{k,j} &= \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \geqslant A \end{aligned}$$

2-5. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A\end{aligned}$$

3-1. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Symmetric to case 1-3.

3-2. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j}$. Symmetric to case 2-3.

3-3. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A\end{aligned}$$

3-4. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \geqslant A\end{aligned}$$

3-5. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A\end{aligned}$$

4-1. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Symmetric to case 1-4.

4-2. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j}$. Symmetric to case 2-4.

4-3. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Symmetric to case 3-4.

4-4. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \geqslant A\end{aligned}$$

4-5. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\\ &\geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A\end{aligned}$$

5-1. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Symmetric to case 1-5.

5-2. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,j}$. Symmetric to case 2-5.

5-3. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Symmetric to case 3-5.

5-4. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Symmetric to case 4-5.

5-5. Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Because $\mathbf{m}$ is closed and by Lemma 4.1:

$$\begin{aligned}\mathbf{m}'_{i,k} + \mathbf{m}'_{k,j} &= \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},k} + \mathbf{m}_{k,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ &\geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant A\end{aligned}$$

□

A.2 Proofs for Properties of Incremental Closure

Proof (for proposition 4.2)

- Suppose $\mathbf{m}'_{i,j} = \mathbf{m}_{i,j}$. Because $\mathbf{m}$ is coherent $\mathbf{m}'_{i,j} = \mathbf{m}_{\bar{\jmath},\bar{\imath}} \geqslant \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$.
- Suppose $\mathbf{m}'_{i,j} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is coherent $\mathbf{m}'_{i,j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \geqslant \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$.
- Suppose $\mathbf{m}'_{i,j} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Similar to the previous case.
- Suppose $\mathbf{m}'_{i,j} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Because $\mathbf{m}$ is coherent $\mathbf{m}'_{i,j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}} \geqslant \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$.
- Suppose $\mathbf{m}'_{i,j} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. Similar to the previous case.

Since $\mathbf{m}'_{i,j} \geqslant \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$ it follows $\mathbf{m}'_{\bar{\jmath},\bar{\imath}} \geqslant \mathbf{m}'_{i,j}$ hence $\mathbf{m}'_{i,j} = \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$ as required. □

Proof (for proposition 4.3) Suppose $\mathbf{m}'$ is consistent. By Lemma 4.1 it follows that $\mathbf{m}_{b,a} + d \geqslant 0$, $\mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$, $\mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d \geqslant 0$ and $\mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d \geqslant 0$. Therefore

$$\mathbf{m}'_{\bar{a},a} = \min\begin{pmatrix}\mathbf{m}_{\bar{a},a}, \\ \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,a}, \\ \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},a} \\ \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} \\ \mathbf{m}_{\bar{a},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,a}\end{pmatrix} = \mathbf{m}_{\bar{a},a}$$

Likewise $\mathbf{m}'_{b,\bar{b}} = \mathbf{m}_{b,\bar{b}}$. Using the same inequalities it follows

$$\begin{aligned}\mathbf{m}'_{i,a} &= \min\left(\mathbf{m}_{i,a}, \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a}\right) & \mathbf{m}'_{b,j} &= \min\left(\mathbf{m}_{b,j}, \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\right) \\ \mathbf{m}'_{i,\bar{b}} &= \min\left(\mathbf{m}_{i,\bar{b}}, \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}}\right) & \mathbf{m}'_{\bar{a},j} &= \min\left(\mathbf{m}_{\bar{a},j}, \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\right)\end{aligned}$$

Therefore

$$\begin{aligned}\mathbf{m}'_{i,a} + d + \mathbf{m}'_{b,j} &= \min\begin{pmatrix}\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\end{pmatrix} \\ &\geqslant \min\begin{pmatrix}\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}\end{pmatrix} \geqslant \mathbf{m}'_{i,j}\end{aligned}$$

Likewise $\mathbf{m}'_{i,\bar{b}} + d + \mathbf{m}'_{\bar{a},j} \geqslant \mathbf{m}'_{i,j}$. Now consider

$$\begin{aligned}&\mathbf{m}'_{i,a} + d + \mathbf{m}'_{b,\bar{b}} + d + \mathbf{m}'_{\bar{a},j} \\ &= \min\begin{pmatrix}\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\end{pmatrix} \\ &\geqslant \min\begin{pmatrix}\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}\end{pmatrix} \geqslant \mathbf{m}'_{i,j}\end{aligned}$$

Likewise $\mathbf{m}'_{i,\bar{b}} + d + \mathbf{m}'_{\bar{a},a} + d + \mathbf{m}'_{b,j} \geqslant \mathbf{m}'_{i,j}$. Thus $\mathbf{m}''_{i,j} = \mathbf{m}'_{i,j}$. Now suppose $\mathbf{m}'$ is not consistent. Since $\mathbf{m}''_{i,i} \leqslant \mathbf{m}'_{i,i}$ then $\mathbf{m}''$ is not consistent. □

A.3 Proofs for Incremental Strong Closure

Proof (for theorem 5.1) Observe that $\mathbf{m}'_{i,\bar{\imath}} = \min(\mathbf{m}_{i,\bar{\imath}}, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{i,\bar{\imath}})/2) = \mathbf{m}_{i,\bar{\imath}}$ and likewise $\mathbf{m}'_{j,\bar{\jmath}} = \mathbf{m}_{j,\bar{\jmath}}$. Therefore

$$\frac{\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j}}{2} = \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j}}{2} \geqslant \min\begin{pmatrix} \mathbf{m}_{i,j} \\ \frac{\mathbf{m}_{i,\bar{\imath}}+\mathbf{m}_{\bar{\jmath},j}}{2} \end{pmatrix} = \mathbf{m}'_{i,j}$$

Because $\mathbf{m}$ is closed $0 = \mathbf{m}_{i,i} \leqslant \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\imath},i}$ and thus

$$\mathbf{m}'_{i,i} = \min(\mathbf{m}_{i,i}, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\imath},i})/2) = \min(0, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\imath},i})/2) = 0$$

To show $\mathbf{m}'_{i,j} \leqslant \mathbf{m}'_{i,k} + \mathbf{m}'_{k,j}$ we proceed by case analysis:

- Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Because $\mathbf{m}$ is closed:

$$\mathbf{m}'_{i,j} \leqslant \mathbf{m}_{i,j} \leqslant \mathbf{m}_{i,k} + \mathbf{m}_{k,j} = \mathbf{m}'_{i,k} + \mathbf{m}'_{k,j}$$

- Suppose $\mathbf{m}'_{i,k} \neq \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} = \mathbf{m}_{k,j}$. Because $\mathbf{m}$ is closed and coherent:

$$\begin{aligned} 2\mathbf{m}'_{i,k} + 2\mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{k},k} + 2\mathbf{m}_{k,j} \geqslant \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{k},j} + \mathbf{m}_{k,j} \\ &= \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},k} + \mathbf{m}_{k,j} \geqslant \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j} \geqslant 2\mathbf{m}'_{i,j} \end{aligned}$$

- Suppose $\mathbf{m}'_{i,k} = \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} \neq \mathbf{m}_{k,j}$. Symmetric to the previous case.
- Suppose $\mathbf{m}'_{i,k} \neq \mathbf{m}_{i,k}$ and $\mathbf{m}'_{k,j} \neq \mathbf{m}_{k,j}$. Because $\mathbf{m}$ is closed:

$$\begin{aligned} 2\mathbf{m}'_{i,k} + 2\mathbf{m}'_{k,j} &= \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{k},k} + \mathbf{m}_{k,\bar{k}} + \mathbf{m}_{\bar{\jmath},j} \\ &\geqslant \mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{k},\bar{k}} + \mathbf{m}_{\bar{\jmath},j} = \mathbf{m}_{i,\bar{\imath}} + 0 + \mathbf{m}_{\bar{\jmath},j} \geqslant 2\mathbf{m}'_{i,j} \end{aligned}$$

□

Proof (for proposition 5.1) Let $\mathbf{m}'' = \textsc{Str}(\mathbf{m}')$. Observe $\mathbf{m}'_{i,\bar{\imath}} = \min(\mathbf{m}_{i,\bar{\imath}}, (\mathbf{m}_{i,\bar{\imath}}+\mathbf{m}_{i,\bar{\imath}})/2) = \mathbf{m}_{i,\bar{\imath}}$ and likewise $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. Therefore

$$\begin{aligned} \mathbf{m}''_{i,j} &= \min(\mathbf{m}'_{i,j}, (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j})/2) \\ &= \min(\min(\mathbf{m}_{i,j}, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2), (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2) \\ &= \min(\mathbf{m}_{i,j}, (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2) = \mathbf{m}'_{i,j} \end{aligned}$$

□

Proof (for proposition 5.2)

$$\begin{aligned} \textsc{Str}(\mathbf{m^2}_{i,j}) &= \min(\mathbf{m^2}_{i,j}, \tfrac{\mathbf{m^2}_{i,\bar{\imath}}+\mathbf{m^2}_{\bar{\jmath},j}}{2}) \\ &\geqslant \min(\mathbf{m^1}_{i,j}, \tfrac{\mathbf{m^1}_{i,\bar{\imath}}+\mathbf{m^1}_{\bar{\jmath},j}}{2}) = \textsc{Str}(\mathbf{m^1}_{i,j}) \end{aligned}$$

□

Proof (for proposition 5.3)
$\mathbf{m}'_{i,j} = \min(\mathbf{m}_{i,j}, \frac{\mathbf{m}_{i,\bar{\imath}}+\mathbf{m}_{\bar{\jmath},j}}{2}) \leqslant \mathbf{m}_{i,j}$ □

Proof (for proposition 5.4)

$$\mathbf{m}'_{i,j} = \min(\mathbf{m}_{i,j}, \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j}}{2}) = \min(\mathbf{m}_{\bar{\jmath},\bar{\imath}}, \frac{\mathbf{m}_{\bar{\jmath},j} + \mathbf{m}_{i,\bar{\imath}}}{2}) = \mathbf{m}'_{\bar{\jmath},\bar{\imath}}$$

□

Proof (for theorem 5.2) We prove that $\forall i, j. \mathbf{m}'_{i,j} = \mathbf{m^*}_{i,j}$. Pick some i, j.

– Suppose $j = \bar{\imath}$. Then

$$\begin{aligned}
\mathbf{m^*}_{i,\bar{\imath}} &= \min(\mathbf{m^\dagger}_{i,\bar{\imath}}, \mathbf{m^\dagger}_{i,\bar{\imath}}/2 + \mathbf{m^\dagger}_{i,\bar{\imath}}/2) = \mathbf{m^\dagger}_{i,\bar{\imath}} \\
&= \min\begin{pmatrix}
\mathbf{m}_{i,\bar{\imath}}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}
\end{pmatrix} = \mathbf{m'}_{i,\bar{\imath}}
\end{aligned}$$

– Suppose $j \neq \bar{\imath}$. Then

$$\begin{aligned}
\mathbf{m^*}_{i,j} &= \min(\mathbf{m^\dagger}_{i,j}, \mathbf{m^\dagger}_{i,\bar{\imath}}/2 + \mathbf{m^\dagger}_{j,\bar{\jmath}}/2) \\
&= \min(\mathbf{m^\dagger}_{i,j}, \mathbf{m'}_{i,\bar{\imath}}/2 + \mathbf{m'}_{j,\bar{\jmath}}/2) \\
&= \min\begin{pmatrix}
\mathbf{m}_{i,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
(\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2
\end{pmatrix} = \mathbf{m'}_{i,j}
\end{aligned}$$

□

Proof (Proof for Theorem 5.3) Suppose $\mathbf{m}_{a,b} + d \geqslant 0$. Then it is sufficient to show that:

$$\min\begin{pmatrix}
\mathbf{m}_{i,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
(\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2
\end{pmatrix} = \min\begin{pmatrix}
\mathbf{m}_{i,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\
(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j})/2, \\
(\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j})/2
\end{pmatrix}$$

where

$$\mathbf{m'}_{i,\bar{\imath}} = \min\begin{pmatrix}
\mathbf{m}_{i,\bar{\imath}}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\
\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\
\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}},
\end{pmatrix} \quad \mathbf{m'}_{\bar{\jmath},j} = \min\begin{pmatrix}
\mathbf{m}_{\bar{\jmath},j}, \\
\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\
\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j},
\end{pmatrix}$$

Using the above, $(\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2$ expands into one of the following cases:

1-1 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. By strong closure $\frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j}}{2} \geqslant \mathbf{m}_{i,j}$. Thus this case is redundant.

1-2 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}$. This case is not redundant.

1-3 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. By strong closure and coherence:

$$\begin{aligned}
&\frac{\mathbf{m}_{i,\bar{\imath}} + (\mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} = \\
&\quad \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{a},a}}{2} + \frac{2d + \mathbf{m}_{\bar{\jmath},\bar{b}} + \mathbf{m}_{b,j}}{2} \geqslant \mathbf{m}_{i,a} + \frac{2d + 2\mathbf{m}_{b,j}}{2} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}
\end{aligned}$$

1-4 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. By strong closure and coherence:

$$\begin{aligned}
&\frac{\mathbf{m}_{i,\bar{\imath}} + (\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2} = \\
&\quad \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{b,\bar{b}}}{2} + \frac{2d + \mathbf{m}_{\bar{\jmath},a} + \mathbf{m}_{\bar{a},j}}{2} \geqslant \mathbf{m}_{i,\bar{b}} + \frac{2d + 2\mathbf{m}_{\bar{a},j}}{2} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}
\end{aligned}$$

2-1 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. This case is not redundant.

2-2 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}$. Observe that if $x \leqslant y$ then $x \leqslant (x+y)/2 \leqslant y$ and if $y \leqslant x$ then $y \leqslant (x+y)/2 \leqslant x$. Thus $(x+y)/2 \geqslant \min(x,y)$ hence

$$\frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j})}{2} \geqslant \min(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j})$$

Thus this case is redundant.

2-3 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. By coherence and using $(x+y)/2 \geqslant \min(x,y)$:

$$\begin{aligned}
&\frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} \\
&= \frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}) + (\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} \\
&\geqslant \min(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})
\end{aligned}$$

Thus this case is redundant.

2-4 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. By coherence and using $(x+y)/2 \geqslant \min(x,y)$:

$$\begin{aligned}
&\frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2} \\
&= \frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}) + (\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2} \\
&\geqslant \min(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j})
\end{aligned}$$

Thus this case is redundant.

3-1 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. Symmetric to 1-3.

3-2 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}$. Symmetric to case 2-3.

3-3 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. Then

$$\begin{aligned}
&\frac{(\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} \\
&= \frac{(\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}) + (\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} \\
&= \frac{(\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}) + (\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2} \\
&= \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}
\end{aligned}$$

Thus this case is redundant.

3-4 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. By coherence, strong closure and because $\mathbf{m}_{b,a} + d \geqslant 0$:

$$\begin{aligned}
&\frac{(\mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2} \\
&= \frac{\mathbf{m}_{\bar{a},a} + \mathbf{m}_{b,\bar{b}}}{2} + \frac{4d + 2\mathbf{m}_{i,\bar{b}} + 2\mathbf{m}_{\bar{a},j}}{2} \geqslant \mathbf{m}_{\bar{a},\bar{b}} + 2d + \mathbf{m}_{i,\bar{b}} + \mathbf{m}_{\bar{a},j} \\
&= (\mathbf{m}_{b,a} + d) + \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \geqslant \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}
\end{aligned}$$

Thus this case is redundant.

4-1 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. Symmetric to case 1-4.

4-2 Suppose $\mathbf{m}'_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}$ and $\mathbf{m}'_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,j}$. Symmetric to case 2-4.

4-3 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}$. By coherence, strong closure and because $\mathbf{m}_{\bar{a},\bar{b}} + d \geqslant 0$:

$$\begin{aligned}&\frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j})}{2}\\ &= \frac{\mathbf{m}_{\bar{a},a} + \mathbf{m}_{b,\bar{b}}}{2} + \frac{4d + 2\mathbf{m}_{i,a} + 2\mathbf{m}_{b,j}}{2} \geqslant \mathbf{m}_{\bar{a},\bar{b}} + 2d + \mathbf{m}_{i,a} + \mathbf{m}_{b,j}\\ &= (\mathbf{m}_{b,a} + d) + \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j} \geqslant \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}\end{aligned}$$

Thus this case is redundant.

4-4 Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}$. By coherence:

$$\begin{aligned}&\frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}) + (\mathbf{m}_{\bar{\jmath},a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2}\\ &= \frac{(\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}) + (\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})}{2}\\ &= (\mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j})\end{aligned}$$

Now suppose that $\mathbf{m}_{a,b} + d < 0$. By corollary 4.1 IncClose($\mathbf{m}, o$) is not consistent and since $\mathbf{m^*} \leqslant$ IncClose($\mathbf{m}, o$) and $\mathbf{m''} \leqslant$ IncClose($\mathbf{m}, o$) it follows that both $\mathbf{m^*}$ and $\mathbf{m''}$ are not consistent as required. □

A.4 Proofs for Incremental Tight Closure

Proof (for lemma 6.1) Suppose $\mathbf{m'}$ is consistent. By lemma 6.1 it follows that $\mathbf{m'}$ is closed. We will now show that $\mathbf{m'}$ is strongly closed i.e $\forall i, j.\mathbf{m'}_{i,j} \leqslant \mathbf{m'}_{i,\bar{\imath}}/2 + \mathbf{m'}_{\bar{\jmath},j}/2$.

- Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. Then

$$\frac{\mathbf{m'}_{i,\bar{\imath}}}{2} + \frac{\mathbf{m'}_{\bar{\jmath},j}}{2} = \frac{\mathbf{m}_{i,\bar{\imath}}}{2} + \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor \geqslant \mathbf{m'}_{i,j}$$

- Suppose $\mathbf{m'}_{i,\bar{\imath}} \neq \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} = \mathbf{m}_{\bar{\jmath},j}$. Then

$$\begin{aligned}\frac{\mathbf{m'}_{i,\bar{\imath}}}{2} + \frac{\mathbf{m'}_{\bar{\jmath},j}}{2} &= \frac{\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \rfloor + \lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \rfloor}{2} + \frac{\mathbf{m}_{\bar{\jmath},j}}{2}\\ &= \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \geqslant \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor \geqslant \mathbf{m}_{i,j} = \mathbf{m'}_{i,j}\end{aligned}$$

- Suppose $\mathbf{m'}_{i,\bar{\imath}} = \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} \neq \mathbf{m}_{\bar{\jmath},j}$. Symmetric to the previous case.
- Suppose $\mathbf{m'}_{i,\bar{\imath}} \neq \mathbf{m}_{i,\bar{\imath}}$ and $\mathbf{m'}_{\bar{\jmath},j} \neq \mathbf{m}_{\bar{\jmath},j}$. Then

$$\begin{aligned}\frac{\mathbf{m'}_{i,\bar{\imath}}}{2} + \frac{\mathbf{m'}_{\bar{\jmath},j}}{2} &= \frac{\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor}{2} + \frac{\left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor}{2}\\ &= \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{\bar{\jmath},j}}{2} \right\rfloor \geqslant \mathbf{m'}_{i,j}\end{aligned}$$

Thus, if $\mathbf{m'}$ is consistent, it is strongly closed. It remains to show that $\forall i.\mathbf{m'}_{i,\bar{\imath}}$ is even. Observe that:

$$\mathbf{m'}_{i,\bar{\imath}} = \min(\mathbf{m}_{i,\bar{\imath}}, \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor + \left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor) = \min(\mathbf{m}_{i,\bar{\imath}}, 2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor)$$

- Suppose $\mathbf{m}_{i,\bar{\imath}}$ is even. Then $2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor = \mathbf{m}_{i,\bar{\imath}} = \mathbf{m'}_{i,\bar{\imath}}$ which is even.
- Suppose $\mathbf{m}_{i,\bar{\imath}}$ is odd. Then $2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor = \mathbf{m}_{i,\bar{\imath}} - 1 = \mathbf{m'}_{i,\bar{\imath}}$ which is even. □

Proof (for theorem 6) We prove that $\forall i,j.\mathbf{m}_{i,j} = \mathbf{m'}_{i,j}$. Pick some i,j.

- Suppose $j = \bar{\imath}$. Then

$$\begin{aligned}
\mathbf{m^*}_{i,\bar{\imath}} &= \min(\mathbf{m^\ddagger}_{i,\bar{\imath}}, \mathbf{m^\ddagger}_{i,\bar{\imath}}/2 + \mathbf{m^\ddagger}_{i,\bar{\imath}}/2) = \mathbf{m^\ddagger}_{i,\bar{\imath}} = 2\lfloor \mathbf{m^\dagger}_{i,\bar{\imath}}/2 \rfloor \\
&= 2\left\lfloor \min\begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix} /2 \right\rfloor = \mathbf{m'}_{i,\bar{\imath}}
\end{aligned}$$

- Suppose $j \neq \bar{\imath}$. Then

$$\begin{aligned}
\mathbf{m^*}_{i,j} &= \min(\mathbf{m^\ddagger}_{i,j}, \mathbf{m^\ddagger}_{i,\bar{\imath}}/2 + \mathbf{m^\ddagger}_{\bar{j},j}/2) = \min(\mathbf{m^\dagger}_{i,j}, \mathbf{m'}_{i,\bar{\imath}}/2 + \mathbf{m'}_{\bar{j},j}/2) \\
&= \min\begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{j},j})/2 \end{pmatrix} = \mathbf{m'}_{i,j}
\end{aligned}$$

□

Proof (for proposition 6.1) Let $\mathbf{m''} = \textsc{Tighten}(\mathbf{m'})$.

- Suppose $j \neq \bar{\imath}$. Then $\mathbf{m''}_{i,j} = \mathbf{m'}_{i,j}$.
- Suppose $j = \bar{\imath}$. Then $\mathbf{m''}_{i,\bar{\imath}} = 2\left\lfloor \frac{\mathbf{m'}_{i,\bar{\imath}}}{2} \right\rfloor = 2\left\lfloor \frac{2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor}{2} \right\rfloor = 2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor = \mathbf{m'}_{i,\bar{\imath}}$

□

Proof (for proposition 6.2)

- Suppose $j \neq \bar{\imath}$. Then $\textsc{Tighten}(\mathbf{m^2}_{i,j}) = \mathbf{m^2}_{i,j} \geqslant \mathbf{m^1}_{i,j} = \textsc{Tighten}(\mathbf{m^1}_{i,j})$.
- Suppose $j = \bar{\imath}$. Then $\textsc{Tighten}(\mathbf{m^2}_{i,\bar{\imath}}) = 2\left\lfloor \frac{\mathbf{m^2}_{i,\bar{\imath}}}{2} \right\rfloor \geqslant 2\left\lfloor \frac{\mathbf{m^1}_{i,\bar{\imath}}}{2} \right\rfloor = \textsc{Tighten}(\mathbf{m^1}_{i,\bar{\imath}})$

□

Proof (for proposition 6.3)

- Suppose $j = \bar{\imath}$. Then $\mathbf{m'}_{i,j} = \mathbf{m'}_{i,\bar{\imath}} \leqslant 2\left\lfloor \frac{\mathbf{m}_{i,\bar{\imath}}}{2} \right\rfloor = \mathbf{m}_{i,\bar{\imath}} = \mathbf{m}_{i,j}$.
- Suppose $j \neq \bar{\imath}$. Then $\mathbf{m'}_{i,j} = \mathbf{m}_{i,j}$.

□

Proof (for proposition 6.4)

- Suppose $j = \bar{\imath}$. Then $\mathbf{m'}_{\bar{j},\bar{\imath}} = 2\left\lfloor \frac{\mathbf{m}_{\bar{j},\bar{\imath}}}{2} \right\rfloor = 2\left\lfloor \frac{\mathbf{m}_{i,j}}{2} \right\rfloor = \mathbf{m'}_{i,j}$.
- Suppose $j \neq \bar{\imath}$. Then $\mathbf{m'}_{\bar{j},\bar{\imath}} = \mathbf{m}_{\bar{j},\bar{\imath}} = \mathbf{m}_{i,j} = \mathbf{m'}_{i,j}$.

□

A.5 Proofs for In-place Update

Proof (for corollary 7.1) By Proposition 4.3 it follows $\mathbf{m'} = \textsc{IncClose}(\mathbf{m'}, o)$. The result then follows from Theorem 4.1.

Proof (for theorem 7.1) Suppose $\mathbf{m'}$ is consistent.

Let $k = 0$. It vacuously follows that $\forall 0 \leqslant \ell < k.\mathbf{m^k}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m'}_{\rho^{-\mathbf{1}}(\ell)}$. Moreover $\forall k \leqslant \ell < 4n^2.\mathbf{m^k}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}_{\rho^{-\mathbf{1}}(\ell)}$ since $\mathbf{m}^0 = \mathbf{m}$.

Now let $k > 0$ and suppose $\rho(i,j) = k$ and consider

$$\mathbf{m^{k+1}}_{i,j} = \min \begin{pmatrix} \mathbf{m^k}_{i,j} \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j} \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j} \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j} \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j} \end{pmatrix}$$

If $\rho^{-1}(i,a) < k$ then $\mathbf{m^k}_{i,a} = \mathbf{m'}_{i,a}$ whereas if $\rho^{-1}(i,a) \geqslant k$ then $\mathbf{m^k}_{i,a} = \mathbf{m}_{i,a} \geqslant \mathbf{m'}_{i,a}$. Thus $\mathbf{m^k}_{i,a} \geqslant \mathbf{m'}_{i,a}$ and likewise $\mathbf{m^k}_{b,j} \geqslant \mathbf{m'}_{b,j}$. By Corollary 7.1 it follows $\mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j} \geqslant \mathbf{m'}_{i,a} + d + \mathbf{m'}_{b,j} \geqslant \mathbf{m'}_{i,j}$. By a similar argument $\mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j} \geqslant \mathbf{m'}_{i,j}$, $\mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j} \geqslant \mathbf{m'}_{i,j}$ and likewise $\mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j} \geqslant \mathbf{m'}_{i,j}$.

Since $\mathbf{m^k}_{i,j} = \mathbf{m}_{i,j} \geqslant \mathbf{m'}_{i,j}$ it follows $\mathbf{m^{k+1}}_{i,j} \geqslant \mathbf{m'}_{i,j}$. But $\mathbf{m^k} \leqslant \mathbf{m}$ and by Proposition 4.1 $\mathbf{m^{k+1}}_{i,j} \leqslant \mathbf{m'}_{i,j}$ hence $\mathbf{m^{k+1}}_{i,j} = \mathbf{m'}_{i,j}$. Hence it follows $\forall 0 \leqslant \ell < k+1 . \mathbf{m^{k+1}}_{\boldsymbol{\rho^{-1}}(\ell)} = \mathbf{m'}_{\boldsymbol{\rho^{-1}}(\ell)}$. Moreover $\forall k+1 \leqslant \ell < 4n^2 . \mathbf{m^{k+1}}_{\boldsymbol{\rho^{-1}}(\ell)} = \mathbf{m}_{\boldsymbol{\rho^{-1}}(\ell)}$.

Suppose $\mathbf{m'}$ is inconsistent hence $\mathbf{m'}_{i,i} < 0$. Put $k = \rho(i,i)$. But $\mathbf{m^k} \leqslant \mathbf{m}$ and by Proposition 4.1 $\mathbf{m^{4n^2}}_{i,i} = \mathbf{m^{k+1}}_{i,i} \leqslant \mathbf{m'}_{i,i} < 0$ as required. □

Proof (for lemma 7.1) Suppose $\mathbf{m'}$ is consistent. By Proposition 4.2 $\mathbf{m'}$ is coherent.

1. To show $\mathbf{m''}_{i,j} \leqslant \mathbf{m''}_{i,a} + d + \mathbf{m''}_{b,j}$.
   - Suppose $\mathbf{m''}_{i,a} = \mathbf{m'}_{i,a}$ and $\mathbf{m''}_{b,j} = \mathbf{m'}_{b,j}$. Because $\mathbf{m'}$ is consistent by Corollary 7.1 it follows:
     $$\mathbf{m''}_{i,a} + d + \mathbf{m''}_{b,j} = \mathbf{m'}_{i,a} + d + \mathbf{m'}_{b,j} \geqslant \mathbf{m'}_{i,j} \geqslant \mathbf{m''}_{i,j}$$
   - Suppose $\mathbf{m''}_{i,a} = (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},a})/2$ and $\mathbf{m''}_{b,j} = \mathbf{m'}_{b,j}$. Because $\mathbf{m'}$ is consistent by Corollary 7.1 it follows $\mathbf{m'}_{\bar{a},j} \leqslant \mathbf{m'}_{\bar{a},a} + d + \mathbf{m'}_{b,j}$ and $\mathbf{m'}_{\bar{\jmath},j} \leqslant \mathbf{m'}_{\bar{\jmath},a} + d + \mathbf{m'}_{b,j}$. Hence
     $$\begin{aligned} \mathbf{m''}_{i,a} + d + \mathbf{m''}_{b,j} &= (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},a} + 2d + 2\mathbf{m'}_{b,j})/2 \\ &\geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},j} + d + \mathbf{m'}_{b,j})/2 \\ &\geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2 \geqslant \mathbf{m''}_{i,j} \end{aligned}$$
   - Suppose $\mathbf{m''}_{i,a} = \mathbf{m'}_{i,a}$ and $\mathbf{m''}_{b,j} = (\mathbf{m'}_{b,\bar{b}} + \mathbf{m'}_{\bar{\jmath},j})/2$. Symmetric to the previous case.
   - Suppose $\mathbf{m''}_{i,a} = (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},a})/2$ and $\mathbf{m''}_{b,j} = (\mathbf{m'}_{b,\bar{b}} + \mathbf{m'}_{\bar{\jmath},j})/2$. Because $\mathbf{m'}$ is consistent by Corollary 7.1 it follows $\mathbf{m'}_{\bar{a},\bar{b}} \leqslant \mathbf{m'}_{\bar{a},a} + d + \mathbf{m'}_{b,\bar{b}}$ and $\mathbf{m'}_{i,a} \leqslant \mathbf{m'}_{i,a} + d + \mathbf{m'}_{b,a}$ thus $0 \leqslant d + \mathbf{m'}_{b,a}$. Hence
     $$\begin{aligned} \mathbf{m''}_{i,a} + d + \mathbf{m''}_{b,j} &= (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},a} + 2d + \mathbf{m'}_{b,\bar{b}} + \mathbf{m'}_{\bar{\jmath},j})/2 \\ &\geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{a},\bar{b}} + d + \mathbf{m'}_{\bar{\jmath},j})/2 \\ &\geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2 \geqslant \mathbf{m'}_{i,j} \end{aligned}$$
2. To show $\mathbf{m''}_{i,j} \leqslant \mathbf{m''}_{i,\bar{b}} + d + \mathbf{m''}_{\bar{a},j}$. Analogous to the previous case.
3. To show $\mathbf{m''}_{i,j} \leqslant \mathbf{m''}_{i,\bar{b}} + d + \mathbf{m''}_{\bar{a},a} + d + \mathbf{m''}_{b,j}$.
   - Suppose $\mathbf{m''}_{i,\bar{b}} = \mathbf{m'}_{i,\bar{b}}$ and $\mathbf{m''}_{b,j} = \mathbf{m'}_{b,j}$. Since $\mathbf{m''}_{\bar{a},a} = \mathbf{m'}_{\bar{a},a}$ and because $\mathbf{m'}$ is consistent by Corollary 7.1 it follows
     $$\begin{aligned} &\mathbf{m''}_{i,\bar{b}} + d + \mathbf{m''}_{\bar{a},a} + d + \mathbf{m''}_{b,j} \\ &\quad = \mathbf{m'}_{i,\bar{b}} + d + \mathbf{m'}_{\bar{a},a} + d + \mathbf{m'}_{b,j} \geqslant \mathbf{m'}_{i,j} \geqslant \mathbf{m''}_{i,j} \end{aligned}$$
   - Suppose $\mathbf{m''}_{i,\bar{b}} = (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{b,\bar{b}})/2$ and $\mathbf{m''}_{b,j} = \mathbf{m'}_{b,j}$. Because $\mathbf{m'}$ is consistent by Corollary 7.1 it follows $\mathbf{m'}_{\bar{a},j} \leqslant \mathbf{m'}_{\bar{a},a} + d + \mathbf{m'}_{b,j}$, $\mathbf{m'}_{\bar{\jmath},j} \leqslant \mathbf{m'}_{\bar{\jmath},a} + d + \mathbf{m'}_{b,j}$, $\mathbf{m'}_{b,a} \leqslant \mathbf{m'}_{b,\bar{b}} + d + \mathbf{m'}_{\bar{a},a}$ and $0 \leqslant d + \mathbf{m'}_{b,a}$. Therefore
     $$\begin{aligned} &\mathbf{m''}_{i,\bar{b}} + d + \mathbf{m''}_{\bar{a},a} + d + \mathbf{m''}_{b,j} \\ &\quad = (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{b,\bar{b}} + 2\mathbf{m'}_{\bar{a},a} + 4d + 2\mathbf{m'}_{b,j})/2 \\ &\quad \geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{b,\bar{b}} + \mathbf{m'}_{\bar{a},a} + 3d + \mathbf{m'}_{\bar{a},j} + \mathbf{m'}_{b,j})/2 \\ &\quad \geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{b,\bar{b}} + \mathbf{m'}_{\bar{a},a} + 2d + \mathbf{m'}_{\bar{\jmath},j})/2 \\ &\quad \geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{b,a} + d + \mathbf{m'}_{\bar{\jmath},j})/2 \\ &\quad \geqslant (\mathbf{m'}_{i,\bar{\imath}} + \mathbf{m'}_{\bar{\jmath},j})/2 \geqslant \mathbf{m''}_{i,j} \end{aligned}$$

- Suppose $\mathbf{m}''_{i,\bar{b}} = \mathbf{m}'_{i,\bar{b}}$ and $\mathbf{m}''_{b,j} = (\mathbf{m}'_{b,\bar{b}} + \mathbf{m}'_{\bar{\jmath},j})/2$. Symmetric to the previous case.
- Suppose $\mathbf{m}''_{i,\bar{b}} = (\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{b,\bar{b}})/2$ and $\mathbf{m}''_{b,j} = (\mathbf{m}'_{b,\bar{b}} + \mathbf{m}'_{\bar{\jmath},j})/2$. Because $\mathbf{m}'$ is consistent by Corollary 7.1 it follows $\mathbf{m}'_{b,a} \leqslant \mathbf{m}'_{b,\bar{b}} + d + \mathbf{m}'_{\bar{a},a}$ and $0 \leqslant d + \mathbf{m}'_{b,a}$. Therefore

$$\begin{aligned}
&\mathbf{m}''_{i,\bar{b}} + d + \mathbf{m}''_{\bar{a},a} + d + \mathbf{m}''_{b,j} \\
&\quad = (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{b,\bar{b}} + 4d + 2\mathbf{m}''_{\bar{a},a} + 2\mathbf{m}'_{b,\bar{b}} + \mathbf{m}'_{\bar{\jmath},j})/2 \\
&\quad \geqslant (\mathbf{m}'_{i,\bar{\imath}} + 2\mathbf{m}'_{b,a} + 2d + \mathbf{m}'_{\bar{\jmath},j})/2 \\
&\quad \geqslant (\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j})/2 \geqslant \mathbf{m}''_{i,j}
\end{aligned}$$

4. To show $\mathbf{m}''_{i,j} \leqslant \mathbf{m}''_{i,a} + d + \mathbf{m}''_{b,\bar{b}} + d + \mathbf{m}''_{\bar{a},j}$. Analogous to the previous case.

It therefore follows that $\mathbf{m}''' = \mathbf{m}''$. Now suppose $\mathbf{m}'$ is not consistent. Hence $\mathbf{m}''$ is not consistent thus $\mathbf{m}'''$ is not consistent. □

Proof (for theorem 7.2) Suppose $\mathbf{m}'$ is consistent.

Let $k = 0$. It vacuously follows that $\forall 0 \leqslant \ell < k.\mathbf{m}^{\mathbf{k}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}''_{\rho^{-\mathbf{1}}(\ell)}$. Moreover $\forall k \leqslant \ell < 4n^2.\mathbf{m}^{\mathbf{k}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}_{\rho^{-\mathbf{1}}(\ell)}$ since $\mathbf{m}^0 = \mathbf{m}$.

Suppose $0 < k$ and $\rho(i,j) = k$. Now suppose $j = \bar{\imath}$. Then

$$\mathbf{m}^{\mathbf{k+1}}_{i,\bar{\imath}} = \min \begin{pmatrix} \mathbf{m}^{\mathbf{k}}_{i,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},\bar{\imath}} \end{pmatrix}$$

If $\rho(i,a) < k$ then $\mathbf{m}^{\mathbf{k}}_{i,j} = \mathbf{m}''_{i,a}$ otherwise $\rho(i,a) \geqslant k$ then $\mathbf{m}^{\mathbf{k}}_{i,a} = \mathbf{m}_{i,a} \geqslant \mathbf{m}''_{i,a}$ which implies $\mathbf{m}^{\mathbf{k}}_{i,a} \geqslant \mathbf{m}''_{i,a}$. Likewise $\mathbf{m}^{\mathbf{k}}_{b,\bar{\imath}} \geqslant \mathbf{m}''_{b,\bar{\imath}}$. By Lemma 7.1 and Corollary 7.1 it follows $\mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,j} \geqslant \mathbf{m}''_{i,a} + d + \mathbf{m}''_{b,j} \geqslant \mathbf{m}''_{i,j}$. By a similar argument $\mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j} \geqslant \mathbf{m}''_{i,j}$, $\mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j} \geqslant \mathbf{m}''_{i,j}$ and likewise $\mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}_{b,j} \geqslant \mathbf{m}''_{i,j}$. Thus $\mathbf{m}^{\mathbf{k+1}}_{i,j} \geqslant \mathbf{m}''_{i,j}$. Now to show $\mathbf{m}''_{i,j} \geqslant \mathbf{m}^{\mathbf{k+1}}_{i,j}$. Observe

$$\begin{aligned}
\mathbf{m}''_{i,\bar{\imath}} = \mathbf{m}'_{i,\bar{\imath}} &= \min \begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix} \\
&\geqslant \min \begin{pmatrix} \mathbf{m}^{\mathbf{k}}_{i,j}, \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,j}, \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j}, \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}_{b,j}, \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j} \end{pmatrix} = \mathbf{m}^{\mathbf{k+1}}_{i,\bar{\imath}}
\end{aligned}$$

Hence $\forall 0 \leqslant \ell < k.\mathbf{m}^{k}_{\rho^{-1}(\ell)} = \mathbf{m}''_{\rho^{-1}(\ell)}$. Moreover $\forall k + 1 \leqslant \ell < 4n^2.\mathbf{m}^{\mathbf{k+1}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}_{\rho^{-\mathbf{1}}(\ell)}$ follows from the inductive hypothesis and the definition of $\mathbf{m}^{\mathbf{k+1}}_{i,j}$.

Now suppose that $j \neq \bar{\imath}$. Then $2n < \rho(i,j)$ and consider

$$\mathbf{m}^{\mathbf{k+1}}_{i,j} = \min \begin{pmatrix} \mathbf{m}^{\mathbf{k}}_{i,j} \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,j} \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j} \\ \mathbf{m}^{\mathbf{k}}_{i,a} + d + \mathbf{m}^{\mathbf{k}}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},j} \\ \mathbf{m}^{\mathbf{k}}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}_{b,j}, \\ (\mathbf{m}^{\mathbf{k}}_{i,\bar{\imath}} + \mathbf{m}^{\mathbf{k}}_{\bar{\jmath},j})/2 \end{pmatrix}$$

Notice that $\mathbf{m}^{\mathbf{k}}_{i,\bar{\imath}} + \mathbf{m}^{\mathbf{k}}_{\bar{\jmath},j}/2 = \mathbf{m}''_{i,\bar{\imath}} + \mathbf{m}''_{\bar{\jmath},j}/2$, since $\rho(i,\bar{\imath}) < 2n \leqslant \rho(i,j) = k$ and $\rho(\bar{\jmath},j) < \rho(i,j) = k$. By

Lemma 5.1, $\mathbf{m}''_{i,\bar{\imath}} + \mathbf{m}''_{\bar{\jmath},j}/2 \geqslant \mathbf{m}''_{i,j}$. Repeating the argument above it follows that $\mathbf{m^k}_{i,j} \geqslant \mathbf{m}''_{i,j}$ Hence $\forall 0 \leqslant \ell < k.\mathbf{m}^k_{\rho^{-1}(\ell)} = \mathbf{m}''_{\rho^{-1}(\ell)}$. Now to show $\mathbf{m}''_{i,j} \geqslant \mathbf{m^{k+1}}_{i,j}$. Observe that:

$$\begin{aligned}
\mathbf{m}''_{i,j} &= \min(\mathbf{m}'_{i,j}, \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j}}{2}) \\
&= \min\left(\min\left(\begin{array}{l} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{array}\right), \frac{\mathbf{m}_{i,\bar{\imath}} + \mathbf{m}_{\bar{\jmath},j}}{2}\right) \\
&\geqslant \min\left(\min\left(\begin{array}{l} \mathbf{m^k}_{i,j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j}, \\ \mathbf{m^k}_{i,\bar{b}} + d + \mathbf{m^k}_{\bar{a},a} + d + \mathbf{m^k}_{b,j}, \\ \mathbf{m^k}_{i,a} + d + \mathbf{m^k}_{b,\bar{b}} + d + \mathbf{m^k}_{\bar{a},j} \end{array}\right), \frac{\mathbf{m}'_{i,\bar{\imath}} + \mathbf{m}'_{\bar{\jmath},j}}{2}\right) = \mathbf{m^{k+1}}_{i,j}
\end{aligned}$$

Hence it follows $\forall 0 \leqslant \ell < k{+}1.\mathbf{m^{k+1}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}''_{\rho^{-\mathbf{1}}(\ell)}$. Note $\forall k{+}1 \leqslant \ell < 4n^2.\mathbf{m^{k+1}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}_{\rho^{-\mathbf{1}}(\ell)}$ follows from the inductive hypothesis and the definition of $\mathbf{m^{k+1}}_{i,j}$.

Suppose $\mathbf{m}'$ is inconsistent hence $\mathbf{m}'_{i,i} < 0$. Put $k = \rho(i,i)$. But $\mathbf{m^k} \leqslant \mathbf{m}$ and by Proposition 4.1 $\mathbf{m^{4n^2}}_{i,i} = \mathbf{m^{k+1}}_{i,i} \leqslant \mathbf{m}'_{i,i} < 0$ as required. □

Proof (for lemma 7.2) Suppose $\mathbf{m}'''$ is consistent. By Proposition 5.3 $\mathbf{m}''' \leqslant \mathbf{m}''$ and by Proposition 6.3 $\mathbf{m}'' \leqslant \mathbf{m}'$ thus $\mathbf{m}'$ is consistent. By Theorem 4.1 $\mathbf{m}'$ is closed hence $\mathbf{m}'_{a,a} = \mathbf{m}'_{b,b} = \mathbf{m}'_{\bar{a},\bar{a}} = \mathbf{m}'_{\bar{b},\bar{b}} = 0$. By Corollary 7.1 it follows that $\mathbf{m}'_{a,b} \leqslant \mathbf{m}'_{a,a} + d + \mathbf{m}'_{b,b} = d$ and $\mathbf{m}'_{\bar{b},\bar{a}} \leqslant \mathbf{m}'_{\bar{b},\bar{b}} + d + \mathbf{m}'_{\bar{a},\bar{a}} = d$ therefore $\mathbf{m}'''_{a,b} \leqslant d$ and $\mathbf{m}'''_{\bar{b},\bar{a}} \leqslant d$. By Proposition 4.2 $\mathbf{m}'$ is coherent hence $\mathbf{m}'''$ is closed by Lemma 5.1.

– To show $\mathbf{m}'''_{i,a} + d + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,j}$. Since $\mathbf{m}'''$ is closed it follows

$$\mathbf{m}'''_{i,a} + d + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,a} + \mathbf{m}'''_{a,b} + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,b} + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,j}$$

– To show $\mathbf{m}'''_{i,\bar{b}} + d + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,j}$. Since $\mathbf{m}'''$ is closed it follows

$$\mathbf{m}'''_{i,\bar{b}} + d + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,\bar{b}} + \mathbf{m}'''_{\bar{b},\bar{a}} + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,\bar{a}} + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,j}$$

– To show $\mathbf{m}'''_{i,a} + d + \mathbf{m}'''_{b,\bar{b}} + d + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,j}$. Since $\mathbf{m}'''$ is closed

$$\begin{aligned}
\mathbf{m}'''_{i,a} + d + \mathbf{m}'''_{b,\bar{b}} + d + \mathbf{m}'''_{\bar{a},j} &\geqslant \mathbf{m}'''_{i,a} + \mathbf{m}'''_{a,b} + \mathbf{m}'''_{b,\bar{b}} + \mathbf{m}'''_{\bar{b},\bar{a}} + \mathbf{m}'''_{\bar{a},j} \\
&\geqslant \mathbf{m}'''_{i,b} + \mathbf{m}'''_{b,\bar{b}} + \mathbf{m}'''_{\bar{b},\bar{a}} + \mathbf{m}'''_{\bar{a},j} \\
&\geqslant \mathbf{m}'''_{i,\bar{b}} + \mathbf{m}'''_{\bar{b},\bar{a}} + \mathbf{m}'''_{\bar{a},j} \\
&\geqslant \mathbf{m}'''_{i,\bar{a}} + \mathbf{m}'''_{\bar{a},j} \geqslant \mathbf{m}'''_{i,j}
\end{aligned}$$

– To show $\mathbf{m}'''_{i,\bar{b}} + d + \mathbf{m}'''_{\bar{a},a} + d + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,j}$. Since $\mathbf{m}'''$ is closed

$$\begin{aligned}
\mathbf{m}'''_{i,\bar{b}} + d + \mathbf{m}'''_{\bar{a},a} + d + \mathbf{m}'''_{b,j} &\geqslant \mathbf{m}'''_{i,\bar{b}} + \mathbf{m}'''_{\bar{b},\bar{a}} + \mathbf{m}'''_{\bar{a},a} + \mathbf{m}'''_{a,b} + \mathbf{m}'''_{b,j} \\
&\geqslant \mathbf{m}'''_{i,\bar{a}} + \mathbf{m}'''_{\bar{a},a} + \mathbf{m}'''_{a,b} + \mathbf{m}'''_{b,j} \\
&\geqslant \mathbf{m}'''_{i,a} + \mathbf{m}'''_{a,b} + \mathbf{m}'''_{b,j} \\
&\geqslant \mathbf{m}'''_{i,b} + \mathbf{m}'''_{b,j} \geqslant \mathbf{m}'''_{i,j}
\end{aligned}$$

By Proposition 4.3 it follows that $\mathbf{m^*} = \mathbf{m}'''$. □

Proof (for theorem 7.3) Suppose $\mathbf{m}'$ is consistent.

Let $k = 0$. It vacuously follows that $\forall 0 \leqslant \ell < k.\mathbf{m}^{\mathbf{k}}_{\rho^{\mathbf{-1}}(\ell)} = \mathbf{m}''_{\rho^{\mathbf{-1}}(\ell)}$. Moreover $\forall k \leqslant \ell < 4n^2.\mathbf{m}^{\mathbf{k}}_{\rho^{\mathbf{-1}}(\ell)} = \mathbf{m}_{\rho^{\mathbf{-1}}(\ell)}$ since $\mathbf{m}^0 = \mathbf{m}$. Now let $k > 0$ and suppose $\rho(i,j) = k$. Now suppose that $j = \bar{\imath}$. Then

$$\mathbf{m}^{\mathbf{k+1}}{}_{i,j} = 2\left\lfloor \min\begin{pmatrix} \mathbf{m}^{\mathbf{k}}{}_{i,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},\bar{\imath}} \end{pmatrix}/2 \right\rfloor$$

If $\rho^{-1}(i,a) < k$ then $\mathbf{m}^{\mathbf{k}}{}_{i,a} = \mathbf{m}'''{}_{i,a}$ whereas if $\rho^{-1}(i,a) \geqslant k$ then $\mathbf{m}^{\mathbf{k}}{}_{i,a} = \mathbf{m}_{i,a} \geqslant \mathbf{m}'''{}_{i,a}$: this implies that $\mathbf{m}^{\mathbf{k}}{}_{i,a} \geqslant \mathbf{m}'''{}_{i,a}$ and likewise $\mathbf{m}^{\mathbf{k}}{}_{b,j} \geqslant \mathbf{m}_{b,j}$. By Lemma 7.2 and Corollary 7.1 it follows that $\mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j} \geqslant \mathbf{m}'''{}_{i,a} + d + \mathbf{m}'''{}_{b,j}$. By a similar argument $\mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j} \geqslant \mathbf{m}'''{}_{i,j}$, $\mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j} \geqslant \mathbf{m}'''{}_{i,j}$ and likewise $\mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j} \geqslant \mathbf{m}'''{}_{i,j}$. Moreover $(\mathbf{m}''{}_{i,\bar{\imath}} + \mathbf{m}''{}_{\bar{\jmath},j})/2 \geqslant \min(\mathbf{m}''{}_{i,j}, (\mathbf{m}''{}_{i,\bar{\imath}} + \mathbf{m}''{}_{\bar{\jmath},j})/2) = \mathbf{m}'''{}_{i,j}$. Thus $\mathbf{m}^{\mathbf{k}}{}_{i,j} \geqslant \mathbf{m}'''{}_{i,j}$. Now to show $\mathbf{m}'''{}_{i,j} \geqslant \mathbf{m}^{\mathbf{k+1}}{}_{i,j}$.

$$\begin{aligned} \mathbf{m}'''{}_{i,\bar{\imath}} = \mathbf{m}''{}_{i,\bar{\imath}} = 2\left\lfloor \mathbf{m}'{}_{i,\bar{\imath}}/2 \right\rfloor &= 2\left\lfloor \min\begin{pmatrix} \mathbf{m}_{i,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,\bar{\imath}}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},\bar{\imath}} \end{pmatrix}/2 \right\rfloor \\ &\geqslant 2\left\lfloor \min\begin{pmatrix} \mathbf{m}^{\mathbf{k}}{}_{i,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{\imath}}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},\bar{\imath}} \end{pmatrix}/2 \right\rfloor \\ &= \mathbf{m}^{\mathbf{k+1}}{}_{i,\bar{\imath}} \end{aligned}$$

Hence it follows $\forall 0 \leqslant \ell < k + 1.\mathbf{m}^{\mathbf{k+1}}_{\rho^{\mathbf{-1}}(\ell)} = \mathbf{m}''_{\rho^{\mathbf{-1}}(\ell)}$. Moreover $\forall k + 1 \leqslant \ell < 4n^2.\mathbf{m}^{\mathbf{k+1}}_{\rho^{\mathbf{-1}}(\ell)} = \mathbf{m}_{\rho^{\mathbf{-1}}(\ell)}$ follows from the inductive hypothesis and the definition of $\mathbf{m}^{\mathbf{k+1}}{}_{i,j}$.

Now suppose that $j \neq \bar{\imath}$ and consider

$$\mathbf{m}^{\mathbf{k+1}}{}_{i,j} = \min\begin{pmatrix} \mathbf{m}^{\mathbf{k}}{}_{i,j} \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j} \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j} \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j} \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j}, \\ (\mathbf{m}^{\mathbf{k}}{}_{i,\bar{\imath}} + \mathbf{m}^{\mathbf{k}}{}_{\bar{\jmath},j})/2 \end{pmatrix}$$

Notice that $(\mathbf{m}^{\mathbf{k}}{}_{i,\bar{\imath}} + \mathbf{m}^{\mathbf{k}}{}_{\bar{\jmath},j})/2 \geqslant (\mathbf{m}'''{}_{i,\bar{\imath}} + \mathbf{m}'''{}_{\bar{\jmath},j})/2$ since $\rho(i,\bar{\imath}) < 2n \leqslant \rho(i,j) = k$ and similarly $\rho(\bar{\jmath},j) < \rho(i,j) = k$. By Lemma 7.2 $\mathbf{m}'''{}_{i,\bar{\imath}} + \mathbf{m}'''{}_{\bar{\jmath},j}/2 \geqslant \mathbf{m}'''{}_{i,j}$ and thus $(\mathbf{m}^{\mathbf{k}}{}_{i,\bar{\imath}} + \mathbf{m}^{\mathbf{k}}{}_{\bar{\jmath},j})/2 \geqslant \mathbf{m}'''{}_{i,j}$. Repeating the argument above it follows that $\mathbf{m}^{\mathbf{k+1}}{}_{i,j} \geqslant \mathbf{m}'''{}_{i,j}$. Now to show $\mathbf{m}'''{}_{i,j} \geqslant \mathbf{m}^{\mathbf{k+1}}{}_{i,j}$ observe:

$$\begin{aligned} \mathbf{m}'''{}_{i,j} = \mathbf{m}''{}_{i,j} &= \min\left(\mathbf{m}'{}_{i,j}, \frac{\mathbf{m}'{}_{i,\bar{\imath}} + \mathbf{m}'{}_{\bar{\jmath},j}}{2}\right) \\ &= \min\left(\min\begin{pmatrix} \mathbf{m}_{i,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},j}, \\ \mathbf{m}_{i,\bar{b}} + d + \mathbf{m}_{\bar{a},a} + d + \mathbf{m}_{b,j}, \\ \mathbf{m}_{i,a} + d + \mathbf{m}_{b,\bar{b}} + d + \mathbf{m}_{\bar{a},j} \end{pmatrix}, \frac{\mathbf{m}'{}_{i,\bar{\imath}} + \mathbf{m}'{}_{\bar{\jmath},j}}{2}\right) \\ &= \min\left(\min\begin{pmatrix} \mathbf{m}^{\mathbf{k}}{}_{i,j}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,j}, \\ \mathbf{m}^{\mathbf{k}}{}_{i,a} + d + \mathbf{m}^{\mathbf{k}}{}_{b,\bar{b}} + d + \mathbf{m}^{\mathbf{k}}{}_{\bar{a},j} \end{pmatrix}, \frac{\mathbf{m}'{}_{i,\bar{\imath}} + \mathbf{m}'{}_{\bar{\jmath},j}}{2}\right) = \mathbf{m}^{\mathbf{k+1}}{}_{i,j} \end{aligned}$$

Hence it follows $\forall 0 \leqslant \ell < k{+}1 . \mathbf{m}^{\mathbf{k+1}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}''_{\rho^{-\mathbf{1}}(\ell)}$. Note $\forall k{+}1 \leqslant \ell < 4n^2 . \mathbf{m}^{\mathbf{k+1}}_{\rho^{-\mathbf{1}}(\ell)} = \mathbf{m}_{\rho^{-\mathbf{1}}(\ell)}$ follows by inductive hypothesis and definition of $\mathbf{m^{k+1}}_{i,j}$.

Suppose $\mathbf{m}'$ is inconsistent hence $\mathbf{m}'_{i,i} < 0$. Put $k = \rho(i,i)$. But $\mathbf{m^k} \leqslant \mathbf{m}$ and by Proposition 4.1 $\mathbf{m^{4n^2}}_{i,i} = \mathbf{m^{k+1}}_{i,i} \leqslant \mathbf{m}'_{i,i} < 0$ as required. □

References

1. Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. An Improved Tight Closure Algorithm for Integer Octagonal Constraints. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, number 4905 in LNCS, pages 8–21. Springer, 2008.
2. Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. Weakly-relational Shapes for Numeric Abstractions: Improved Algorithms and Proofs of Correctness. *Formal Methods in System Design*, 35(3):279–323, 2009.
3. Francesco Banterle and Roberto Giacobazzi. A Fast Implementation of the Octagon Abstract Domain on Graphics Hardware. In *Static Analysis Symposium*, volume 4634 of *LNCS*, pages 315–335. Springer, 2007.
4. Can A. Baykan and Mark S. Fox. Spatial Synthesis by Disjunctive Constraint Satisfaction. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 11(4):245–262, 1997.
5. Eva Beckschulze, Stefan Kowalewski, and Jörg Brauer. Access-Based Localization for Octagons. *Electronic Notes in Theoretical Computer Science*, 287:29–40, 2012.
6. Richard Bellman. On a Routing Problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
7. Christian Bessiere. Constraint Propagation. In Francesca Rossi, Peter van Beek, and Toby Walsh, editors, *Handbook of Constraint Programming*, pages 39–81. Elsevier, 2006.
8. Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. A Static Analyzer for Large Safety-Critical Software. In *Programming Language Design and Implementation*, pages 196–207, 2003.
9. Aziem Chawdhary and Andy King. Compact Difference Bound Matrices. In *Asian Symposium on Programming Languages and Systems*, volume 10695 of *LNCS*, pages 471–490. Springer, 2017.
10. Aziem Chawdhary, Ed Robbins, and Andy King. Simple and Efficient Algorithms for Octagons. In *Asian Symposium on Programming Languages and Systems*, volume 8858 of *LNCS*, pages 296–313. Springer, 2014.
11. Thomas. H. Cormen, Charles E. Leiserson, and Roland. L Rivest. *Introduction to Algorithms.* The MIT Press, 1990.
12. Rina Dechter, Itay Meiri, and Judea Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49:61–95, 1991.
13. Robert W. Floyd. Algorithm 97: Shortest Path. *Communications of the ACM*, 5(6):345–345, 1962.
14. Graeme Gange, Jorge A. Navas, Peter Schachte, Harold Søndergaard, and Peter J. Stuckey. Exploiting Sparsity in Difference-bound Matrices. In *Static Analysis Symposium*, volume 9837 of *LNCS*, pages 189–211, 2016.
15. Nicolas Halbwachs, David Merchat, and Laure Gonnord. Some ways to Reduce the Space Dimension in Polyhedra Computations. *Formal Methods in System Design*, 29:79–95, 2006.
16. Kihong Heo, Hakjoo Oh, and Hongseok Yang. Learning a Variable-Clustering Strategy for Octagon From Labeled Data Generated by a Static Analysis. In *Static Analysis Symposium*, volume 9837 of *LNCS*, pages 237–256, 2016.
17. Bertrand Jeannet and Antoine Miné. Apron: A Library of Numerical Abstract Domains for Static Analysis. In *Computer Aided Verification*, volume 5643 of *LNCS*, pages 661–667. Springer, 2009.
18. Jacques-Henri Jourdan. *Verasco: a Formally Verified C Static Analyzer.* PhD thesis, Université Paris Diderot (Paris 7) Sorbonne Paris Cité, May 2016. `https://jhjourdan.mketjh.fr/thesis_jhjourdan.pdf`.

19. Jeffrey C. Lagarias. The Computational Complexity of Simultaneous Diophantine Approximation Problems. *SIAM Journal on Computing*, 14(1):196–209, 1985.
20. Shuvendu Lahiri and Madan Musuvathi. An Efficient Decision Procedure for UTVPI Constraints. In *Frontiers of Combining Systems*, volume 3717 of *LNAI*, pages 168–183. Springer, 2005.
21. Anotine Miné. *Weakly Relational Numerical Abstract Domains.* PhD thesis, École Polytechnique En Informatique, 2004. `https://www-apr.lip6.fr/~mine/these/these-color.pdf`.
22. Antoine Miné. The Octagon Abstract Domain. *Higher-Order and Symbolic Programming*, 19(1):31–100, 2006.
23. Nicholas Nethercote. *Dynamic Binary Analysis and Instrumentation.* PhD thesis, Trinity College, University of Cambridge, 2004.
24. Robert Nieuwenhuis and Albert Oliveras. DPLL(T) with Exhaustive Theory Propagation and its Application to Difference Logic. In *Computer Aided Verification*, volume 3576 of *LNCS*, pages 321–334. Springer, 2005.
25. Hakjoo Oh, Lucas Brutschy, and Kwangkeun Yi. Access Analysis-Based Tight Localization of Abstract Memories. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, volume 6538 of *LNCS*, pages 356–370, 2011.
26. Marie Pelleau, Antoine Miné, Charlotte Truchet, and Frédéric Benhamou. A Constraint Solver Based on Abstract Domains. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, volume 7737 of *LNCS*, pages 434–454. Springer, 2013.
27. Marie Pelleau, Charlotte Truchet, and Frédéric Benhamou. The Octagon Abstract Domain for Continuous Constraints. *Constraints*, 19(3):309–337, 2014.
28. Ed Robbins, Jacob M. Howe, and Andy King. Theory Propagation and Reification. *Science of Computer Programming*, 111(1):3–22, 2015.
29. Pierre Roy, Guillaume Perez, Jean-Charles Régin, Alexandre Papadopoulos, François Pachet, and Marco Marchini. Enforcing Structure on Temporal Sequences: The Allen Constraint. In *Principles and Practice of Constraint Programming*, volume 9892 of *LNCS*, pages 786–801, 2016.
30. Andreas Schutt and Peter J. Stuckey. Incremental Satisfiability and Implication for UTVPI Constraints. *INFORMS Journal on Computing*, 22(4):514–527, 2010.
31. Axel Simon and Andy King. Taming the Wrapping of Integer Arithmetic. In *Static Analysis Symposium*, volume 4634 of *LNCS*, pages 121–136. Springer, 2007.
32. Axel Simon, Andy King, and Jacob M. Howe. The Two Variable Per Inequality Abstract Domain. *Higher-Order and Symbolic Programming*, 31(1):182–196, 2010. `http://kar.kent.ac.uk/30678`.
33. Gagandeep Singh, Markus Püschel, and Martin Vechev. Making Numerical Program Analysis Fast. In *Programming Language Design and Implementation*, pages 303–313. ACM Press, 2015.
34. K. Subramani and Piotr Wojciechowski. A Graphical Theorem of the Alternaive for UTVPI Constraints. In *ICTAC*, volume 9399 of *LNCS*, pages 328–345. Springer, 2015.
35. Henry S. Warren. *Hacker's Delight*. Addison-Wesley, 2002.
36. Stephen Warshall. A Theorem on Boolean Matrices. *Journal of the ACM*, 9(1):11–12, 1962.