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

Weakly-Relational Shapes for Numeric Abstractions: Improved Algorithms and Proofs of Correctness

Roberto Bagnara · Patricia M. Hill · Enea Zaffanella

Received: September 1, 2008

Abstract Weakly-relational numeric constraints provide a compromise between complexity and expressivity that is adequate for several applications in the field of formal analysis and verification of software and hardware systems. We address the problems to be solved for the construction of full-fledged, efficient and provably correct abstract domains based on such constraints. We first propose to work with *semantic* abstract domains, whose elements are *geometric shapes*, instead of the (more concrete) syntactic abstract domains of constraint networks and matrices on which the previous proposals are based. This allows to solve, once and for all, the problem whereby *closure by entailment*, a crucial operation for the realization of such domains, seemed to impede the realization of proper widening operators. In our approach, the implementation of widenings relies on the availability of an effective reduction procedure for the considered constraint description: one for the domain of *bounded difference shapes* already exists in the literature; we provide algorithms for the significantly more complex cases of rational and integer *octagonal shapes*. We also improve upon the state-of-the-art by presenting, along with their proof of correctness, closure by entailment algorithms of reduced complexity for domains based on rational and integer octagonal constraints. The consequences of implementing weakly-relational numerical domains with floating point numbers are also discussed.

This work has been partly supported by MURST projects "Constraint Based Verification of Reactive Systems" and "AIDA — Abstract Interpretation: Design and Applications," and by a Royal Society (UK) International Joint Project (ESEP) award.

R. Bagnara
Department of Mathematics, University of Parma
Viale G. P. Usberti 53/A, 43100 Parma, Italy
Tel.: +39-0521-906917, Fax: +39-0521-906950, E-mail: bagnara@cs.unipr.it

P. M. Hill
School of Computing, University of Leeds
Leeds LS2 9JT, ENGLAND
Tel.: +44-(0)113-3436807, Fax: +44-(0)113-3435468, E-mail: p.m.hill@leeds.ac.uk

E. Zaffanella
Department of Mathematics, University of Parma, Italy
Viale G. P. Usberti 53/A, 43100 Parma, Italy
Tel.: +39-0521-906963, Fax: +39-0521-906950, E-mail: zaffanella@cs.unipr.it

1 Introduction

Numerical properties are of great interest in the broad area of formal methods for their complete generality and for the crucial role they often play in the definition of static analyses and program verification techniques. In the field of abstract interpretation, classes of numerical properties are captured by numerical abstract domains. These have been and are widely used, either as the main abstraction for the application at hand, or as powerful ingredients to improve the precision of other abstract domains.

Among the wide spectrum of numerical abstractions proposed in the literature, probably the most famous are the (non-relational) abstract domain of intervals [1] and the (relational) abstract domain of convex polyhedra [2]. As far as the efficiency/precision trade-off is concerned, these domains occupy the opposite ends of the spectrum: on the one hand, the operations on convex polyhedra achieve a significant level of precision, which is however countered by a worst-case exponential time complexity that demands particular care with its use if scalability problems are to be avoided; on the other hand, the great efficiency of the corresponding operations on intervals is made unappealing by the fact that the obtained precision is often unsatisfactory. This well-known dichotomy (which does not impede that, for some applications, convex polyhedra or intervals are the right choices) has motivated recent studies on several abstract domains that lie somewhere between these two extremes, and can therefore be called *weakly-relational* abstract domains. Examples include domains based on constraint networks [3–5], the abstract domain of difference-bound matrices [6, 7], the octagon abstract domain [8], the 'two variables per inequality' abstract domain [9], the octahedron abstract domain [10], and the abstract domain of template constraint matrices [11]. Moreover, similar proposals that are not abstractions of the domain of convex polyhedra have been put forward, including the abstract domain of bounded quotients [3], the zone congruence abstract domain [12] and the one of *integer octagonal constraints*.

The latter constraints —that is, constraints of the form $ax + by \leq d$ where $a, b \in \{-1, 0, +1\}$, $d \in \mathbb{Z}$ and the variables x and y range over the integers— are also called *Unit Two Variables Per Inequality (UTVPI) integer constraints* and constitute an interesting subclass of linear integer constraints admitting polynomial solvability. The place which these constraints occupy in the complexity/expressivity spectrum is, in fact, peculiar. Concerning complexity, relaxing the restriction of (at most) two variables per constraint and/or relaxing the restriction on coefficients make the satisfiability problem NP-complete [13, 14]. Concerning expressivity, these constraints have been successfully used in a number of applications [15–18] in the field of formal analysis and verification of software and hardware systems. Besides that, integer octagonal constraints can be used for representing and solving many integer problems in the field of constraint programming, such as temporal reasoning and scheduling [13]. When (integer or rational) octagonal constraints are used to build abstract domains the most critical operation is *closure by entailment*.[1] This is the procedure whereby a set of octagonal constraints is augmented with (a finite representation of) all the octagonal constraints that can be inferred from it. The closure algorithms for rational octagonal constraints are sound but not complete for integer octagonal constraints. The latter require so-called *tight* closure algorithms that fully exploit the integrality of the variables.

In this paper, we address —providing the proofs of all the stated results— several issues that revolve around the closure by entailment operation: its interaction with

[1] In contrast, in constraint programming, satisfiability check is the crucial operation.

the *widening operators* that ensure fixpoint computations terminate after a reasonable number of iterations; its efficient computation, both on rational and integer domain; its correct realization when floating point numbers are used for the implementation.

1.1 Closure by Entailment and Widening Operators

For the abstract domain of convex polyhedra, all the widenings that have been proposed are variations of, and/or improvements to, what is commonly referred to as the *standard widening* [2,19]. This is based on the general widening principle "drop the unstable components" applied to constraints. Not surprisingly, most proposals for widening operators for the weakly relational domains are based on the same principle and analogous to the standard widening. For instance, for the domain of difference bound matrices mentioned above, an operator meant to match the standard widening is given in [7]. Unfortunately, as pointed out by A. Miné in [6,8] and also in his PhD thesis [18], this operator is not a widening, since it has no convergence guarantee. The reason is that *closure by entailment*, which is systematically performed so as to provide a canonical form for the elements and to improve the precision of several domain operations, has a negative interaction with the extrapolation operator of [7] that compromises the convergence guarantee. Intuitively, what can happen is that, while the extrapolation operator discards unstable constraints, the closure operation reinserts them (because they were redundant): failure to drop such unstable constraints can (and, in practice, quite often does) result in infinite upward iteration sequences. For this reason, it is proposed in [6,8,18] to apply the same operator given in [7] to the "syntactic" version of the same abstract domain, that is, where closure is only very carefully applied during the fixpoint computations.

We have taken a different approach and resolve the apparent conflict by considering a "semantic" abstract domain whose elements are the geometric shapes themselves. Since closure by entailment preserves the geometric shapes (even though it does not preserve their syntactic expressions), the approach is immune from the divergence problem described above. On the other hand, in order to use the standard widening as the basis of the proposed widening, it is important that we can compute *reduced* representations of the domain elements that encode non-redundant systems of constraints. Thus the implementations of any new widenings based on the semantic approach will need effective reduction procedures for the considered constraint description: here we provide such an algorithm for the domain of rational and integer *octagonal shapes*.

1.2 Computing the Closure by Entailment

Building upon the theoretical work required to verifying the correctness of the reduction algorithms mentioned above, we show that the algorithm for computing the strong closure of rational octagonal graphs as described in [6] can be simplified with a consequential improvement in its efficiency.

We also show that this result can be extended so as to apply to integer octagonal constraints. This enables us to present and fully justify an $\mathrm{O}(n^3)$ algorithm to compute the tight closure of a set of UTVPI integer constraints. In particular, this is the first time that an algorithm achieving such a complexity bound is provided with a proof of correctness.

Finally, we illustrate what happens when floating point numbers are used to implement the rational coefficients of weakly relational abstract domains.

1.3 Plan of the Paper

This paper which combines, improves and extends the work presented in [20,21] and [22], is structured as follows: Section 2 recalls the required concepts and notations; Section 3 presents domains based on bounded differences and the solution of widening-closure conflict on these, simpler numerical abstractions; Section 4 extends such results to domains based on rational octagonal constraints; Section 5 presents the improved algorithm for computing closure by entailment of a system of rational octagonal constraints; Section 6 generalizes the algorithms for closure and reduction to the case of integer octagonal constraints; floating point computations are discussed in Section 7; related work is presented in Section 8; and Section 9 concludes.

2 Preliminaries

The reader is assumed to be familiar with the fundamental concepts of lattice theory [23] and abstract interpretation theory [24,25]. We recall here the definition of *widening operator*, providing a general mechanism for enforcing or accelerating termination of an abstract interpretation. In its simplest form, a widening operator on a poset $\langle P, \sqsubseteq \rangle$ is a partial function $\nabla \colon P \times P \rightarrowtail P$ satisfying:

1. for all $x, y \in P$, if $x \nabla y$ is defined then $x \sqsubseteq x \nabla y$ and $y \sqsubseteq x \nabla y$;
2. for all increasing chains $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$, if the increasing chain $x_0 := y_0$ and $x_{i+1} := x_i \nabla y_{i+1}$ is defined for all $i \in \mathbb{N}$, then it is not strictly increasing.

We refer the reader to the classical works on the numeric domains of intervals [1] and convex polyhedra [2] for the specification of the corresponding widening operators.

Let $\mathbb{Q}_\infty := \mathbb{Q} \cup \{+\infty\}$ be totally ordered by the extension of '<' such that $d < +\infty$ for each $d \in \mathbb{Q}$. A *rational-weighted directed graph* (graph, for short) G in the finite set of *nodes* $\mathcal{N}$ is a pair $(\mathcal{N}, w)$, where $w \colon \mathcal{N} \times \mathcal{N} \to \mathbb{Q}_\infty$ is the weight function for G. A pair $(n_i, n_j) \in \mathcal{N} \times \mathcal{N}$ is an *arc* of G if $w(n_i, n_j) < +\infty$; the arc is *proper* if $n_i \neq n_j$. A *path* $\pi = n_0 \cdots n_p$ in G is a non-empty and finite sequence of nodes such that (n_{i-1}, n_i) is an arc of G, for all $i = 1, \ldots, p$. Each node n_i where $i = 0, \ldots, p$ and each arc (n_{i-1}, n_i) where $i = 1, \ldots, p$ is said to be *in* the path π. The *length* of the path π is the number p of occurrences of arcs in π and denoted by $\|\pi\|$; the *weight* of the path π is $\sum_{i=1}^{p} w(n_{i-1}, n_i)$ and denoted by $w(\pi)$. The path π is *simple* if each node occurs at most once in π; it is *proper* if all the arcs in it are proper; it is a *proper cycle* if it is a proper path, $n_0 = n_p$ and $p \geq 2$; it is a *zero-cycle* if it is a proper cycle and $w(\pi) = 0$. A graph is *zero-cycle free* if it has no zero-cycles. If $\pi_1 = n_0 \cdots n_h$ and $\pi_2 = n_h \cdots n_p$ are paths, where $0 \leq h \leq p$, then the path concatenation $\pi = n_0 \cdots n_h \cdots n_p$ of π_1 and π_2 is denoted by $\pi_1 :: \pi_2$; if $\pi_1 = n_0 n_1$ (so that $h = 1$), then $\pi_1 :: \pi_2$ will also be denoted by $n_0 \cdot \pi_2$. Note that path concatenation is not the same as sequence concatenation.

A graph $(\mathcal{N}, w)$ can be interpreted as the system of *potential constraints*

$$\mathcal{C} := \{\, n_i - n_j \leq w(n_i, n_j) \mid n_i, n_j \in \mathcal{N} \,\}.$$

The graph $(\mathcal{N}, w)$ is *consistent* if and only if the system of constraints it represents is satisfiable in $\mathbb{Q}$, i.e., there exists a rational valuation $\rho\colon \mathcal{N} \to \mathbb{Q}$ such that, for each constraint $(n_i - n_j \leq d) \in \mathcal{C}$, the relation $\rho(n_i) - \rho(n_j) \leq d$ holds. It is well-known that a graph is consistent if and only if it has no negative weight cycles (see [26, Section 25.5] and [27]).

The set of consistent graphs in $\mathcal{N}$ is denoted by $\mathbb{G}$. This set is partially ordered by the relation '$\trianglelefteq$' defined, for all $G_1 = (\mathcal{N}, w_1)$ and $G_2 = (\mathcal{N}, w_2)$, by

$$G_1 \trianglelefteq G_2 \qquad \Longleftrightarrow \qquad \forall i, j \in \mathcal{N} : w_1(i, j) \leq w_2(i, j).$$

When augmented with a bottom element $\bot$ representing inconsistency, this partially ordered set becomes a non-complete lattice $\mathbb{G}_\bot = \langle \mathbb{G} \cup \{\bot\}, \trianglelefteq, \sqcap, \sqcup \rangle$, where '$\sqcap$' and '$\sqcup$' denote the finitary greatest lower bound and least upper bound operators, respectively. A graph $G' = (\mathcal{N}, w')$ is a *subgraph* of $G = (\mathcal{N}, w)$ if, for all $i, j \in \mathcal{N}$, we have $w'(i, j) = w(i, j)$ or $w'(i, j) = +\infty$; the subgraph is *proper* if $G' \neq G$. If G' is a subgraph of G, then $G \trianglelefteq G'$ (the converse does not necessarily hold).

Definition 1 (Closed graph.) A consistent graph $G = (\mathcal{N}, w)$ is *closed* if the following properties hold:

$$\forall i \in \mathcal{N} : w(i, i) = 0; \tag{1}$$

$$\forall i, j, k \in \mathcal{N} : w(i, j) \leq w(i, k) + w(k, j). \tag{2}$$

The *(shortest-path) closure* of a consistent graph G in $\mathcal{N}$ is

$$\mathrm{closure}(G) := \bigsqcup \bigl\{\, G' \in \mathbb{G} \bigm| G' \trianglelefteq G \text{ and } G' \text{ is closed} \,\bigr\}.$$

Although the lattice of rational graphs is not complete, it will include all the infinite least upper bounds defining the closures of rational graphs. Informally, this must hold since the weights of closure(G) must be linear combinations of the rational weights of G and hence are also rational.

When trivially extended so as to behave as the identity function on the bottom element $\bot$, shortest-path closure is a kernel operator (monotonic, idempotent and reductive) on the lattice $\mathbb{G}_\bot$, therefore providing a canonical form.

3 Bounded Difference Graphs and Shapes

The typical way to simplify the domain of convex polyhedra is by restricting attention to particular subclasses of linear inequalities. One possibility, which has a long tradition in computer science [28], is to only consider *potential constraints*, also known as *bounded differences*: these are restricted to take the form $v_i - v_j \leq d$ or $\pm v_i \leq d$, where d is a constant. Systems of bounded differences have been used by the artificial intelligence community as a way to reason about temporal quantities [29,30], as well as by the model checking community as an efficient yet precise way to model and propagate timing requirements during the verification of various kinds of concurrent systems [31, 32]. In the abstract interpretation field, the idea of using an abstract domain of bounded differences was put forward in [3].

3.1 Bounded Difference Graphs

A finite system $\mathcal{C}$ of bounded differences on variables $\mathcal{V} = \{v_0, \dots, v_{n-1}\}$ can be represented by a weighted directed graph $G = (\mathcal{N}_{\mathbf{0}}, w)$ where $\mathcal{N}_{\mathbf{0}} = \{\mathbf{0}\} \cup \mathcal{V}$, $\mathbf{0} \notin \mathcal{V}$ is the *special variable*, and the weight function w is defined, for each $v_i, v_j \in \mathcal{N}_{\mathbf{0}}$, by

$$w(v_i, v_j) := \begin{cases} \min\bigl\{\, d \in \mathbb{Q} \bigm| (v_i - v_j \le d) \in \mathcal{C} \,\bigr\}, & \text{if } v_i \neq \mathbf{0} \text{ and } v_j \neq \mathbf{0}; \\ \min\bigl\{\, d \in \mathbb{Q} \bigm| (v_i \le d) \in \mathcal{C} \,\bigr\}, & \text{if } v_i \neq \mathbf{0} \text{ and } v_j = \mathbf{0}; \\ \min\bigl\{\, d \in \mathbb{Q} \bigm| (-v_j \le d) \in \mathcal{C} \,\bigr\}, & \text{if } v_i = \mathbf{0} \text{ and } v_j \neq \mathbf{0}; \\ 0, & \text{if } v_i = v_j = \mathbf{0}. \end{cases}$$

Notice that we assume that $\min \varnothing = +\infty$; moreover, unary constraints are encoded by means of the special variable, which is meant to always have value 0. A possible representation of (the weight function of) the graph G is by means of a matrix-like data structure called *Difference-Bound Matrix* (DBM) [28]. However, this representation provides no conceptual advantage over the isomorphic graph (or *constraint network* [30]) representation. For this reason we will consistently adopt the terminology and notation introduced in Section 2 for weighted directed graphs. In particular, a graph encoding a consistent system of bounded differences will be called a *Bounded Difference Graph* (BDG).

The first fully developed application of bounded differences in the field of abstract interpretation can be found in [7], where an abstract domain of closed BDGs is defined. In this case, the shortest-path closure requirement was meant as a simple and well understood way to obtain a canonical form for the domain elements by abstracting away from the syntactic details; basically, it corresponds to the closure by entailment of the encoded system of bounded differences with respect to the *transitivity* inference rule:

$$\frac{i - k \le d_1 \qquad k - j \le d_2}{i - j \le d_1 + d_2} \tag{3}$$

In [7] the specification of all the required abstract semantics operators is provided, including an operator that is meant to match the widening operators defined on more classical numeric domains. This operator can be interpreted either as a generalization for closed BDGs of the widening operator defined on the abstract domain of intervals [1], or as a restriction on the domain of closed BDGs of the standard widening defined on the abstract domain of convex polyhedra [2,19]: its implementation is based on the following upper bound operator on the set of consistent graph representations.

Definition 2 (Widening graphs.) Let $G_1 = (\mathcal{N}, w_1)$ and $G_2 = (\mathcal{N}, w_2)$ be consistent graphs. Then $G_1 \nabla G_2 := (\mathcal{N}, w)$, where the weight function w is defined, for each $i, j \in \mathcal{N}$, by

$$w(i, j) := \begin{cases} w_1(i, j), & \text{if } w_1(i, j) \ge w_2(i, j); \\ +\infty, & \text{otherwise.} \end{cases}$$

Unfortunately, as pointed out in [6,8], when used in conjunction with shortest-path closure, this extrapolation operator does not provide a convergence guarantee for fixpoint computations, hence it is not a widening. The reason is that, whereas the closure operation adds redundant constraints to the input BDG, a key requirement in the specification of the standard widening is that the first argument polyhedron must be described by a non-redundant system of constraints.[2] Thus we have a "conflict of interest" between the use of a convenient canonical form for the abstract domain —a form that also allows for increased precision of several domain operations— and the requirements of the widening.

The abstract domain of BDGs has been reconsidered in [6]. Differently from [7], in [6] BDGs are not required to be closed. In this more concrete, syntactic domain, the shortest-path closure operator maps each domain element into the smallest BDG encoding the same geometric shape. Closure is typically used as a preprocessing step before the application of most, though not all, of the abstract semantic operators, allowing for improved accuracy in the results of the abstract computation. The same widening operator proposed in [7] is also used in [6]; however, it is observed that this widening "could have intriguing interactions" with shortest-path closure, therefore identifying the divergence issue faced in [7]. This observation led the author of [6] to the adoption of the syntactic domain of BDGs, where closure is not enforced.

3.2 Bounded Difference Shapes

While the analysis of the divergence problem is absolutely correct, the solution identified in [6] is sub-optimal since, as is usually the case, resorting to a syntactic domain (such as the one of BDGs) has a number of negative consequences, some of which will be recalled in Section 3.4.

To identify a simpler, more natural solution, we first have to acknowledge that an element of our abstract domain should be a geometric shape, rather than (any) one of its graph representations. To stress this concept, such an element will be called a *Bounded Difference Shape* (BDS). A BDS corresponds to the equivalence class of all the BDGs representing it. The implementation of the abstract domain can freely choose between these possible representations, switching at will from one to the other, as long as the semantic operators are implemented as expected. Notice that, in such a context, the shortest-path closure operator is just a transparent implementation detail: on the abstract domain of BDSs it corresponds to the identity function.

The other step towards the solution of the divergence problem is the simple observation that a BDS is a convex polyhedron and the set of all BDSs is closed under the application of the standard widening on convex polyhedra. Thus, no divergence problem can be incurred when applying the standard widening to an increasing sequence of BDSs. As mentioned in Section 3.1, a crucial requirement in the specification of the standard widening is that the first argument polyhedron is described by a non-redundant system of constraints [33]. Thus it is not surprising that using closed BDGs has problems since it is very likely that they will encode redundant constraints. By contrast, we propose to select, in the equivalence class of BDGs representing the same geometric shape, a BDG which is maximal with respect to the graph lattice ordering '$\trianglelefteq$', since such a graph encodes no redundant constraints at all.

Definition 3 (Reduced graph.) A consistent graph G_1 is *reduced* if, for each consistent graph $G_2 \neq G_1$ such that $G_1 \trianglelefteq G_2$, we have $\mathrm{closure}(G_1) \neq \mathrm{closure}(G_2)$. A *reduction* for the consistent graph G is any reduced graph G_r such that $\mathrm{closure}(G) = \mathrm{closure}(G_\mathrm{r})$.

[2] This requirement was sometimes neglected in recent papers describing the standard widening on convex polyhedra; it was recently recalled and exemplified in [33]. Note that a similar requirement is implicitly present even in the specification of the widening on intervals.

Hence, a graph is reduced if it is maximal in the subset of graphs having the same shortest-path closure. In order to provide a correct and reasonably efficient implementation of the standard widening on the domain of BDSs, all we need is a reduction procedure mapping a BDG representation into (any) one of the equivalent reduced graphs. Such an algorithm was defined in [32] and called *shortest-path reduction*. Basically, it is an extension of the transitive reduction algorithm of [34] to the case of weighted directed graphs. Note that, since each equivalence class may have many maximal elements, shortest-path reduction is not a properly defined operator on the domain of BDGs. However, the shortest-path reduction algorithm of [32] provides a canonical form as soon as we fix a total order for the nodes in the graph.

In summary, the solution to the divergence problem for BDSs is to apply the operator specified in Definition 2 to a reduced BDG representation of the first argument of the widening. From the point of view of the user, this will be a transparent implementation detail: on the domain of BDSs, shortest-path reduction is the identity function, as was the case for shortest-path closure.

3.3 On the Precision of the Standard Widening

The standard widening on BDSs could result, if used with no precautions, in poorer precision with respect to its counterpart defined on the syntactic domain of BDGs. For increased precision, the specification of [6] prescribes two conditions that the abstract iteration sequence must satisfy:

1. the second argument of the widening should be represented by a closed BDG (note that, in this case, no divergence problem can arise);
2. the first BDG of the abstract iteration sequence $G_0 \trianglelefteq G_1 \trianglelefteq \ldots \trianglelefteq G_i \trianglelefteq \ldots$ should be closed too.

The effects of both improvements can be obtained also with the semantic domain of BDSs. As for the first one, this can be applied as is, leading to an implementation where the two arguments of the widening are represented by a reduced BDG and a closed BDG, respectively. The result of such a widening operator will depend on the specific reduced form computed for the first argument. The second precision improvement can be achieved by applying the well-known ‘widening up to’ technique defined in [35,36] or its variation called ‘staged widening with thresholds’ [37–39]: in practice, it is sufficient to add to the set of ‘up to’ thresholds all the constraints of the shortest-path closure of the first BDG G_0. Further precision improvements can be obtained by applying any delay strategy and/or the framework defined in [33].

3.4 Syntactic vs. Semantic Abstract Domains

In this section we have shown that when considering weakly-relational numeric abstractions, besides the syntactic domains of constraint systems, it is possible to define the semantic domains of the corresponding geometric shapes. To avoid misunderstandings, it is worth stressing that both kinds of abstract domain are well defined and may be safely adopted for the implementation of a static analysis application. Nonetheless, it can be argued that using a semantic abstract domain provides several advantages. Some of the advantages were already pointed out in [6, Section 5] where the domain of BDGs is compared to the domain of closed BDGs.[3] For instance, it is noted that the domain of closed BDGs allows for the specification of a nicer, injective meaning function; also, the least upper bound operator on BDGs is not the most precise approximation of the union of two geometric shapes. In summary, the discussion in [6, Section 5] makes clear that the solution to the divergence problem for the abstract iteration sequence was the main motivation for adopting a syntactic domain.

One disadvantage of syntactic abstract domains concerns the user-level interfaces of the corresponding software implementations. Namely, the user of a syntactic abstract domain (e.g., the developer of a specific static analysis application using this domain) has to be aware of many details that, in principle, should be hidden by the implementation. As an example, consider the shortest-path closure and reduction procedures for BDGs, which the user might rightfully see as semantics-preserving operations. As a matter of fact, for the syntactic domain of BDGs, these are not semantics-preserving: their application affects both the precision and the convergence of the abstract iteration. In such a situation, the documentation of the abstract domain software needs to include several warnings about the correct usage of these operators, so as to avoid possible pitfalls. In contrast, when adopting the semantic domain of BDSs, both the closure and reduction operators may be excluded from the public interface while the implementation can apply them where and when needed or appropriate. Such an approach is systematically pursued in the implementation of the *Parma Polyhedra Library* [40].

Another potential drawback of the adoption of a syntactic abstract domain can be found in the application of domain refinement operators. As an example, consider the application of the *finite powerset operator* [41] to the domains of BDGs and BDSs, so as to obtain two abstract domains that are able to represent finite disjunctions of the corresponding abstract elements. In both cases, by providing the widenings on BDGs and BDSs with appropriate finite convergence certificates [41], it will be possible to lift them to corresponding widenings on the powerset domains. However, when upgrading the syntactic domain, avoidable redundancies will be incurred, since different disjuncts inside a domain element may represent the same geometric shape; furthermore, these "duplicates" cannot be systematically removed, since by doing so we could change the value of the finite convergence certificate of the powerset element, possibly breaking the convergence guarantee of the lifted widening. As a consequence, both efficiency and precision are potentially degraded. In summary, the disadvantages of syntactic domains are amplified when applying domain refinements.

4 Rational Octagonal Graphs and Shapes

From a theoretical point of view, the observations made in the previous section are immediately applicable to any other weakly-relational numeric domain whose elements are convex polyhedra and is closed with respect to the application of the standard widening, therefore including the domains proposed in [8–11]. From a practical perspective, the success of such a construction depends on the availability of a reasonably efficient reduction procedure for the considered subclass of constraints, because algorithms for minimizing arbitrary linear inequality constraints are not efficient enough. In this section we provide such a reduction procedure for the *octagon* abstract domain [8].

[3] Similar observations, tailored to the case of octagons, are also in [8, Section VII].

We assume in the following that there is a fixed set $\mathcal{V} = \{v_0, \dots, v_{n-1}\}$ of n variables. The octagon abstract domain allows for the manipulation of *octagonal constraints* of the form $av_i + bv_j \leq d$, where $a, b \in \{-1, 0, +1\}$, $a \neq 0$, $v_i, v_j \in \mathcal{V}$, $v_i \neq v_j$ and $d \in \mathbb{Q}$ (the same class of constraints was considered in [15], where octagons were called *simple sections*). Octagonal constraints can be encoded using potential constraints by splitting each variable v_i into two forms: a positive form v_i^+, interpreted as $+v_i$; and a negative form v_i^-, interpreted as $-v_i$. Then any octagonal constraint $av_i + bv_j \leq d$ can be written as a potential constraint $v - v' \leq d_0$ where $v, v' \in \{v_i^+, v_i^-, v_j^+, v_j^-\}$ and $d_0 \in \mathbb{Q}$. Namely, an octagonal constraint such as $v_i + v_j \leq d$ can be translated into the potential constraint $v_i^+ - v_j^- \leq d$; alternatively, the same octagonal constraint can be translated into $v_j^+ - v_i^- \leq d$. Furthermore, unary (octagonal) constraints such as $v_i \leq d$ and $-v_i \leq d$ can be encoded as $v_i^+ - v_i^- \leq 2d$ and $v_i^- - v_i^+ \leq 2d$, respectively, so that the special variable $\mathbf{0}$ is no longer needed.

From now on, we assume that the set of nodes is $\mathcal{N}^\pm := \{0, \dots, 2n-1\}$. These will denote the positive and negative forms of the variables in $\mathcal{V}$: for all $i \in \mathcal{N}^\pm$, if $i = 2k$, then i represents the positive form v_k^+ and, if $i = 2k+1$, then i represents the negative form v_k^- of the variable v_k. To simplify the presentation, for each $i \in \mathcal{N}^\pm$, we let $\bar{\imath}$ denote $i+1$, if i is even, and $i-1$, if i is odd, so that, for all $i \in \mathcal{N}^\pm$, we also have $\bar{\imath} \in \mathcal{N}^\pm$ and $\bar{\bar{\imath}} = i$. Then we can rewrite a potential constraint $v - v' \leq d$ where $v \in \{v_k^+, v_k^-\}$ and $v' \in \{v_l^+, v_l^-\}$ as the potential constraint $i - j \leq d$ in $\mathcal{N}^\pm$ where, if $v = v_k^+$, $i = 2k$ and if $v = v_k^-$, $i = 2k+1$; similarly, if $v' = v_l^+$, $j = 2l$ and if $v' = v_l^-$, $j = 2l+1$.

It follows that any finite system of octagonal constraints on $\mathcal{V}$, translated into a set of potential constraints in $\mathcal{N}^\pm$ as above, can be encoded by a graph G in $\mathcal{N}^\pm$. In particular, any *satisfiable* system of octagonal constraints can be encoded by a *consistent* graph in $\mathcal{N}^\pm$. However, the converse does not hold since in any valuation ρ of an encoding of a set of octagonal constraints we must also have $\rho(i) = -\rho(\bar{\imath})$, so that the arcs (i, j) and $(\bar{\jmath}, \bar{\imath})$ should have the same weight. Therefore, to encode rational octagonal constraints, we restrict attention to consistent graphs over $\mathcal{N}^\pm$ where the arcs in all such pairs are *coherent*.

Definition 4 (Octagonal graph.) An *octagonal graph* in $\mathcal{N}^\pm$ is any consistent graph $G = (\mathcal{N}^\pm, w)$ satisfying the coherence assumption:

$$\forall i, j \in \mathcal{N}^\pm : w(i, j) = w(\bar{\jmath}, \bar{\imath}). \tag{4}$$

The set $\mathbb{O}$ of all octagonal graphs (with the usual addition of the bottom element, representing an unsatisfiable system of constraints) is a sub-lattice of $\mathbb{G}_\perp$, sharing the same least upper bound and greatest lower bound operators. Note that, at the implementation level, coherence can be automatically and efficiently enforced by letting arc (i, j) and arc $(\bar{\jmath}, \bar{\imath})$ share the same representation. This also implies that an octagonal constraint such as $v_i + v_j \leq d$ will always be translated into both $v_i^+ - v_j^- \leq d$ and $v_j^+ - v_i^- \leq d$.

The octagon abstract domain developed in [8] is thus a syntactic domain having octagonal graphs as elements. When dealing with octagonal graphs, observe that the coherence assumption links the positive and negative forms of variables. A closure by entailment procedure, besides classical transitivity as encoded by the explicit paths in the graph, should also consider the following *strengthening* inference rule:

$$\frac{i - \bar{\imath} \leq d_1 \qquad \bar{\jmath} - j \leq d_2}{2(i - j) \leq d_1 + d_2} \tag{5}$$

Thus shortest-path closure for consistent graphs does not provide a canonical form for octagonal graphs; to this end, *strong closure*, first defined in [8], is used instead.

Definition 5 (Strongly closed graph.) An octagonal graph $G = (\mathcal{N}^{\pm}, w)$ is *strongly closed* if it is closed and the following property holds:

$$\forall i, j \in \mathcal{N}^{\pm} : 2w(i, j) \leq w(i, \bar{\imath}) + w(\bar{\jmath}, j). \tag{6}$$

The *strong closure* of an octagonal graph G in $\mathcal{N}^{\pm}$ is

$$\text{S-closure}(G) := \bigsqcup \bigl\{\, G' \in \mathbb{O} \bigm| G' \trianglelefteq G \text{ and } G' \text{ is strongly closed} \,\bigr\}.$$

When trivially extended with a bottom element, strong closure is a kernel operator on the lattice of octagonal graphs.

4.1 A Strong Reduction Procedure for Octagonal Graphs

By repeating the reasoning motivating the bounded difference shapes, we define the semantic abstract domain of *octagonal shapes*, whose elements are equivalence classes of octagonal graphs representing the same geometric shape. Hence, strong closure maps an octagonal graph representation of a non-empty octagonal shape into the minimum element of the corresponding equivalence class. The dual procedure, mapping the octagonal graph into (any) one of the maximal elements in its equivalence class, is called *strong reduction*.

Definition 6 (Strongly reduced graph.) An octagonal graph G_1 is *strongly reduced* if, for each octagonal graph $G_2 \neq G_1$ such that $G_1 \trianglelefteq G_2$, $\text{S-closure}(G_1) \neq \text{S-closure}(G_2)$. A *strong reduction* for the octagonal graph G is any strongly reduced octagonal graph G_{R} such that $\text{S-closure}(G) = \text{S-closure}(G_{\mathrm{R}})$.

In the above definition, we only compare G_1 with other *octagonal* graphs $G_2 \in \mathbb{O}$, thereby disregarding those trivial redundancies that are due to the coherence assumption. This is not a real problem because, as discussed before, any reasonable implementation will automatically and efficiently filter away this kind of redundancies.

We now generalize the shortest-path reduction algorithm of [32] so as to obtain a strong reduction procedure for octagonal graphs. Clearly, the algorithm of [32] cannot be used without modifications, since it takes no account of the redundancies caused by the strengthening inference rule (5). Nonetheless, the high-level structure of the strong reduction procedure is the same as that defined in [32] for shortest-path reduction:

1. compute the closure by entailment of the constraint graph;
2. partition the nodes into equivalence classes based on equality constraints;
3. decompose the graph so as to separate those arcs that link different equivalence classes (encoding only inequalities) from the partition information (encoding the equivalence classes themselves, i.e., all the equalities);
4. reduce the subgraph that gives constraints on different equivalence classes;

5. reduce the partition information;
6. merge the results of steps 4 and 5 to obtain the reduced constraint graph.

We now describe each of the above steps, formally stating the correctness of the overall procedure.

Step 1 of the algorithm can be performed by applying any strong closure procedure, e.g., the one originally defined in [8] or the more efficient one given in Section 5.

Step 2 is also easily implemented by observing that, in a strongly closed octagonal graph, equality constraints correspond to zero-cycles having length two.

Definition 7 (Zero-equivalence.) Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph. The nodes $i, j \in \mathcal{N}^{\pm}$ are *zero-equivalent* in G, denoted $i \equiv_G j$, if and only if $w(i, j) = -w(j, i)$.

While step 6 carries over from BDGs to octagonal graphs, the formal definition of steps 3–5 of the reduction algorithm is more difficult for octagonal graphs than it was for BDGs, as it requires some understanding of the structure of the zero-equivalence classes. As a first observation, note that $i \equiv_G j$ if and only if $\bar{\imath} \equiv_G \bar{\jmath}$, so that we have the following lemma.

Lemma 1 *Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph and $\mathcal{E} \subseteq \mathcal{N}^{\pm}$ a zero-equivalence class for G. Then $\overline{\mathcal{E}} := \{\, \bar{\imath} \in \mathcal{N}^{\pm} \mid i \in \mathcal{E} \,\}$ is also a zero-equivalence class for G.*

Let G be a strongly closed octagonal graph and, for a zero-equivalence class $\mathcal{E}$ of G, let $\overline{\mathcal{E}}$ be defined as in Lemma 1. Then we say that $\mathcal{E}$ is *non-singular* if $\mathcal{E} \cap \overline{\mathcal{E}} = \varnothing$ and *singular* if $\mathcal{E} = \overline{\mathcal{E}}$.

Lemma 2 *Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph. Then there is at most one singular zero-equivalence class for G.*

If it exists, the singular zero-equivalence class encodes all the *unary* equality constraints.[4] In contrast, all the other (non-singular) zero-equivalence classes can only encode *binary* equality constraints.

We associate to each zero-equivalence class $\mathcal{E} \subseteq \mathcal{N}^{\pm}$ a *leader* $\ell_{\mathcal{E}} := \min \mathcal{E}$; the class having the leader in positive (resp., negative) form will be said to be a positive (resp., negative) zero-equivalence class. Note that, this means that the singular zero-equivalence class, if present, is always positive and, for non-singular zero-equivalence classes $\mathcal{E}$ and $\overline{\mathcal{E}}$, we have $\ell_{\overline{\mathcal{E}}} = \overline{\ell}_{\mathcal{E}}$.

We are now ready to provide a formal specification for step 3 of the strong reduction algorithm. As was the case in [32], the first subgraph resulting from the decomposition, relating nodes in different zero-equivalence classes, is obtained by only connecting the leaders; however, the leader of the singular zero-equivalence class should *not* be connected to the other leaders. The second subgraph only encodes those constraints relating nodes in the same zero-equivalence class.

Definition 8 (Non-singular leaders and zero-equivalence subgraphs.) Assume $G = (\mathcal{N}^{\pm}, w)$ is a strongly closed octagonal graph and $\mathcal{L} \subseteq \mathcal{N}^{\pm}$ is the set of leaders of the non-singular zero-equivalence classes for G. The *non-singular leaders' subgraph*

[4] When computing a reduced BDG, such a singular zero-equivalence class is always present: it is the zero-equivalence class containing the special variable **0**.

of G is the graph $L = (\mathcal{N}^{\pm}, w_{\mathrm{L}})$, where the weight function w_{L} is defined, for each $i, j \in \mathcal{N}^{\pm}$, by

$$w_{\mathrm{L}}(i,j) := \begin{cases} w(i,j), & \text{if } i = j \text{ or } \{i,j\} \subseteq \mathcal{L}; \\ +\infty, & \text{otherwise.} \end{cases}$$

The *zero-equivalence subgraph* of G is the graph $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$, where the weight function w_{E} is defined, for each $i, j \in \mathcal{N}^{\pm}$, by

$$w_{\mathrm{E}}(i,j) := \begin{cases} w(i,j), & \text{if } i \equiv_G j; \\ +\infty, & \text{otherwise.} \end{cases}$$

The following result states that the two subgraphs are still strongly closed and the non-singular leaders' subgraph encodes no equality constraints, therefore describing a fully dimensional octagonal shape.

Lemma 3 *Let L and E be the non-singular leaders' subgraph and the zero-equivalence subgraph of the strongly closed octagonal graph G, respectively. Then, L and E are strongly closed octagonal graphs and L is zero-cycle free.*

Step 4 of the strong reduction algorithm is implemented by checking, for each proper arc in the non-singular leaders' subgraph, whether it can be obtained from the other arcs by a single application of the constraint inference rules. Once again, note that we disregard redundancies caused by the coherence assumption.

Definition 9 (Strongly atomic arc and subgraph.) Let $G = (\mathcal{N}^{\pm}, w)$ be an octagonal graph. An arc (i, j) of G is *atomic* if it is proper and, for all $k \in \mathcal{N}^{\pm} \setminus \{i, j\}$, $w(i, j) < w(i, k) + w(k, j)$. The arc (i, j) is *strongly atomic* if it is atomic and either $i = \overline{\jmath}$ or $2w(i, j) < w(i, \overline{\imath}) + w(\overline{\jmath}, j)$.

The *strongly atomic subgraph* of G is the graph $A = (\mathcal{N}^{\pm}, w_{\mathrm{A}})$ where the weight function w_{A} is defined, for all $i, j \in \mathcal{N}^{\pm}$, by

$$w_{\mathrm{A}}(i,j) = \begin{cases} w(i,j), & \text{if } (i,j) \text{ is strongly atomic in } G; \\ +\infty, & \text{otherwise.} \end{cases}$$

The implementation of step 5 of the algorithm, i.e., the strong reduction of the zero-equivalence subgraph, is performed by reducing each zero-equivalence class in isolation. Once again, we exploit the total ordering defined on $\mathcal{N}^{\pm}$.

The strong reduction for a positive non-singular zero-equivalence class $\mathcal{E}$ follows that of [32]: it creates a single zero-cycle connecting all nodes in $\mathcal{E}$ following their total ordering, where the weights of the component arcs are as in the strong closure of the graph. By the coherence assumption, the nodes in the corresponding negative zero-equivalence class $\overline{\mathcal{E}}$ are automatically connected in the opposite order. Figure 1 shows the arcs in the strong reduction of both $\mathcal{E}$ and $\overline{\mathcal{E}}$, where $\mathcal{E} = \{z_0, \ldots, z_m\}$ is the positive class and where $z_0 < \cdots < z_m$. The strong reduction for a singular zero-equivalence class $\mathcal{E}$ is similar except that there is now a single zero-cycle connecting all the positive and negative nodes in $\mathcal{E}$. Figure 2 shows the strong reduction for the singular zero-equivalence class $\mathcal{E} = \{z_0, \overline{z}_0, \ldots, z_m, \overline{z}_m\}$, where $z_0 < \overline{z}_0 < \cdots < z_m < \overline{z}_m$. In both Figures 1 and 2, the dashed arcs are those that can be obtained from the non-dashed ones by application of the coherence assumption.

The following definition formalizes the above observations.

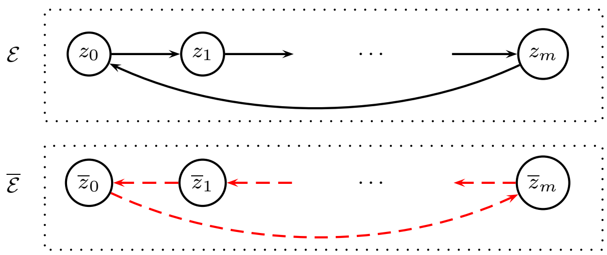

Fig. 1 Strong reduction for non-singular zero-equivalence classes

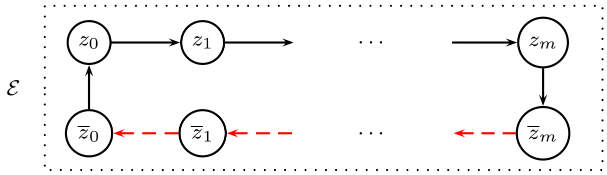

Fig. 2 Strong reduction for the singular zero-equivalence class

Definition 10 (Zero-equivalence reduction.) Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph and let w' be the weight function defined, for all $i, j \in \mathcal{N}^{\pm}$, as follows: if $i, j \in \mathcal{E}$ for some positive zero-equivalence class $\mathcal{E}$ of G and

- if $\mathcal{E} = \{z_0, \ldots, z_m\}$ is non-singular, assuming $z_0 < \cdots < z_m$,

$$w'(i,j) := \begin{cases} w(i,j), & \text{if } i = z_{h-1},\ j = z_h, \text{ for some } h = 1 \ldots, m; \\ w(i,j), & \text{if } i = z_m,\ j = z_0 \text{ and } m > 0; \\ +\infty, & \text{otherwise;} \end{cases}$$

- if $\mathcal{E} = \{z_0, \overline{z}_0, \ldots, z_m, \overline{z}_m\}$ is singular, assuming $z_0 < \overline{z}_0 < \cdots < z_m < \overline{z}_m$,

$$w'(i,j) := \begin{cases} w(i,j), & \text{if } i = z_{h-1},\ j = z_h, \text{ for some } h = 1 \ldots, m; \\ w(i,j), & \text{if } i = \overline{z}_0,\ j = z_0 \text{ or } i = z_m,\ j = \overline{z}_m; \\ +\infty, & \text{otherwise;} \end{cases}$$

and $w'(i,j) := +\infty$, otherwise. Then, the *zero-equivalence reduction* for G is the octagonal graph $Z = (\mathcal{N}^{\pm}, w_Z)$, where, for each $i, j \in \mathcal{N}^{\pm}$,

$$w_Z(i,j) := \min\bigl\{w'(i,j), w'(\overline{\jmath}, \overline{\imath})\bigr\}.$$

The final step 6 of the strong reduction algorithm is implemented by computing the greatest lower bound $A \sqcap Z$, where A is the strongly atomic subgraph of L and Z is the zero-equivalent reduction of E, as obtained at steps 4 and 5 of the algorithm.

Theorem 1 *Given an octagonal graph G, the strong reduction algorithm computes a strong reduction for G.*

If n is the cardinality of the original set $\mathcal{V}$ of variables, then steps 1 and 4 of the algorithm have worst-case complexity in $O(n^3)$, while all the others steps are in $O(n^2)$. Thus, the overall procedure has cubic complexity. As was the case for the reduction procedure of [32], once the ordering of variables is fixed, the strong reduction algorithm returns a canonical form for octagonal graphs.

4.2 A Semantic Widening for Octagonal Shapes

A correct implementation of the standard widening on octagonal shapes is obtained by computing any strong reduction of the octagonal graph representing the first argument. As in the case of BDSs, for maximum precision the strongly closed representation for the second argument should be computed. Even better, by adopting the following variant, we obtain a "truly semantic" widening for the domain of octagonal shapes.

Definition 11 (Widening octagonal shapes.) Let $S_1, S_2 \in \wp(\mathbb{R}^n)$, where $\varnothing \neq S_1 \subseteq S_2$, be two octagonal shapes represented by the strongly reduced octagonal graph G_1 and the strongly closed octagonal graph G_2, respectively. Let also $S \in \wp(\mathbb{R}^n)$ be the octagonal shape represented by the octagonal graph $G_1 \nabla G_2$. Let $\dim(T)$ denote the affine dimension of shape T. Then we define

$$S_1 \nabla S_2 := \begin{cases} S_2, & \text{if } \dim(S_1) < \dim(S_2); \\ S, & \text{otherwise.} \end{cases}$$

By refraining from applying the graph-based widening when the affine dimension of the geometric shapes is increasing, the operator becomes independent from the specific strongly reduced form computed, i.e., from the total ordering defined on the nodes of the graphs. Also note that the test $\dim(S_1) < \dim(S_2)$ can be efficiently decided by checking whether the nodes of the two octagonal graphs are partitioned into different collections of zero-equivalence classes.

Theorem 2 *The operator '∇' of* Definition 11 *is a proper widening on the domain of octagonal shapes. Let '∇_{s}' be the standard widening on the domain of convex polyhedra, as defined in* [19]*. Then, for all octagonal shapes $S_1, S_2 \in \mathbb{R}^n$ such that $\varnothing \neq S_1 \subseteq S_2$, we have $S_1 \nabla S_2 \subseteq S_1 \nabla_{\mathrm{s}} S_2$.*

Observe that, in the typical static analysis computation pattern, the result of a widening computed for one iteration will be used as the first input argument for the widening step in the subsequent iteration. By Definition 11, such an input argument will already be either strongly closed or strongly reduced (because the graph widening operator of Definition 2 preserves strong reduction); hence, the full computational cost of strong reduction will be incurred less frequently than expected.

A semantic widening for BDSs enjoying a result similar to Theorem 2 can be obtained by replacing, in Definition 11, the strongly reduced and strongly closed octagonal graph representations by the reduced and closed BDG representations, respectively.

5 An Improved Strong Closure Procedure for Octagonal Graphs

The strong closure algorithm defined in [8, 18], whose pseudo-code is reported in Figure 3, performs n local propagation steps, where n is the dimension of the vector space.

```
function Q_consistent(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    for $i := 0$ to $2n-1$ do
        if $w[i,i] < 0$ return false;
        else $w[i,i] := 0$;
    return true;
end function

procedure strengthening(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$)
    for $i := 0$ to $2n-1$ do
        for $j := 0$ to $2n-1$ do
            $w[i,j] := \min\Bigl(w[i,j], \bigl(w[i,\bar{\imath}] + w[\bar{\jmath},j]\bigr)/2\Bigr)$;
end procedure

function strong_closure(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    for $k := 0$ to $2n-2$ step 2 do
        { Modified Floyd-Warshall: $n$ steps }
        for $i := 0$ to $2n-1$ do
            for $j := 0$ to $2n-1$ do
                $w[i,j] := \min\bigl(w[i,j], w[i,k] + w[k,j],$
                                    $w[i,\bar{k}] + w[\bar{k},j],$
                                    $w[i,k] + w[k,\bar{k}] + w[\bar{k},j],$
                                    $w[i,\bar{k}] + w[\bar{k},k] + w[k,j]\bigr)$;
        { Strengthening: $n$ steps }
        strengthening($w$);
    return Q_consistent($w$);
end function
```

Fig. 3 The strong closure algorithm proposed in [8, 18] for rational octagonal graphs

In each of these steps, a rather involved variant of the constraint propagation for transitivity as found in the Floyd-Warshall algorithm is followed by another constraint propagation for strengthening, corresponding to the new inference rule (5).

A naive implementation of the algorithm in Figure 3 would perform $48n^3$ coefficient operations (the number of operations is obtained by summing the number of coefficient additions and the number of coefficient comparisons implementing the computation of the minima, whose realization typically requires a coefficient subtraction; we disregard simpler operations, such as zero initialization, division by the constant 2 and comparison with the constant 0, which are also less frequent). Several optimizations can be applied: for instance, some invariant computations can be moved out of the body of the inner loop; more importantly, the number of iterations can be lowered by resorting to an appropriate data structure [18, Section 4.5.1] recording the weight function w, here denoted to be similar to a two-dimensional array, so that it automatically implements the coherence assumption for octagonal graphs (i.e., property (4) of Definition 4). In particular, the finely tuned implementation that can be found in [42], performs $20n^3 + 24n^2$ coefficient operations. Nonetheless, a significant efficiency improvement,

```
function shortest_path_closure(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    { Classical Floyd-Warshall }
    for $k := 0$ to $2n-1$ do
        for $i := 0$ to $2n-1$ do
            for $j := 0$ to $2n-1$ do
                $w[i,j] := \min\bigl(w[i,j], w[i,k] + w[k,j]\bigr)$;
    return Q_consistent($w$);
end function

function strong_closure(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    if not shortest_path_closure($w$) return false;
    strengthening($w$);
    return true;
end function
```

Fig. 4 An improved strong closure algorithm for rational octagonal graphs

which is probably beyond any hand crafted and/or compiler assisted optimization of the algorithm in Figure 3, can be obtained thanks to the following theorem.

Theorem 3 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed octagonal graph. Consider the graph $G_{\mathrm{S}} = (\mathcal{N}^{\pm}, w_{\mathrm{S}})$, where w_{S} is defined, for each $i, j \in \mathcal{N}^{\pm}$, by*

$$w_{\mathrm{S}}(i,j) := \min\biggl\{ w(i,j), \frac{w(i,\bar{\imath})}{2} + \frac{w(\bar{\jmath},j)}{2} \biggr\}.$$

Then $G_{\mathrm{S}} = \text{S-closure}(G)$.

Intuitively, the theorem states that strong closure can be obtained by application of *any* shortest-path closure algorithm followed by a *single* strengthening step, leading to a much simpler implementation.

Figure 4 shows an improved strong closure algorithm based on Theorem 3 and on the classical Floyd-Warshall shortest-path closure algorithm. Here too the pseudo-code relies on a data structure that automatically implements the coherence assumption. The naive implementation of this new algorithm leads to the computation of $16n^3 + 8n^2$ coefficient operations: while it is still in the $\mathrm{O}(n^3)$ complexity class, the saving with respect to the highly optimized algorithm in [42] is always above 20%.

If the case of sparse graphs is considered, then Theorem 3 allows for a further improvement to the complexity bound: to this end, it is enough to modify the code in Figure 4 so as to compute the shortest-path closure using Johnson's algorithm [26]: the worst case complexity of such an implementation will be $\mathrm{O}(n^2 \log n + mn)$, which significantly improves upon the $\mathrm{O}(n^3)$ worst case complexity of the Floyd-Warshall algorithm when, e.g., $m \in \Theta(n)$. However, as observed elsewhere [18,43], some of the targeted applications (e.g., static analysis) typically require the computation of graphs that are dense, so that the Floyd-Warshall algorithm is often a better choice from a practical perspective.

Observe that Theorem 3 also allows for enhancements in the *incremental* variant of the strong closure algorithm, proposed in [18]. Once again, the improved algorithm can be obtained by *any* incremental version of shortest-path closure (e.g., the incremental version of the Floyd-Warshall algorithm) followed by a single strengthening step.

6 Integer Octagonal Graphs, Tight Closure and Tight Reduction

We now consider the case of integer octagonal constraints, i.e., octagonal constraints where the bounds are all integral and the variables are only allowed to take integral values. These can be encoded by suitably restricting the codomain of the weight function of octagonal graphs.

Definition 12 (Integer octagonal graph.) An *integer octagonal graph* is an octagonal graph $G = (\mathcal{N}^{\pm}, w)$ having an integral weight function:

$$\forall i, j \in \mathcal{N}^{\pm} : w(i, j) \in \mathbb{Z} \cup \{+\infty\}.$$

As an integer octagonal graph is also a rational octagonal graph, the constraint system that it encodes will be satisfiable when interpreted to take values in $\mathbb{Q}$. However, when interpreted to take values in $\mathbb{Z}$, this system may be unsatisfiable since the arcs encoding unary constraints can have an odd weight; we say that an octagonal graph is $\mathbb{Z}$-*consistent* if its encoded integer constraint system is satisfiable. For the same reason, the strong closure of an integer octagonal graph does not provide a canonical form for the integer constraint system that it encodes and we need to consider the following *tightening* inference rule:

$$\frac{i - \bar{\imath} \leq d}{i - \bar{\imath} \leq 2\lfloor d/2 \rfloor}. \tag{7}$$

Definition 13 (Tightly closed graph.) An octagonal graph $G = (\mathcal{N}^{\pm}, w)$ is *tightly closed* if it is a strongly closed integer octagonal graph and the following property holds:

$$\forall i \in \mathcal{N}^{\pm} : w(i, \bar{\imath}) \text{ is even.} \tag{8}$$

The *tight closure* of an octagonal graph G in $\mathcal{N}^{\pm}$ is

$$\text{T-closure}(G) := \bigsqcup \bigl\{\, G' \in \mathbb{O} \bigm| G' \trianglelefteq G \text{ and } G' \text{ is tightly closed} \,\bigr\}.$$

It follows from Definition 13 that any tightly closed integer octagonal graph will satisfy property (8), thereby encoding a satisfiable integer constraint system, and therefore will be $\mathbb{Z}$-consistent. Moreover, since the encoding of any satisfiable integer constraint system will result in a $\mathbb{Z}$-consistent integer octagonal graph G that satisfies property (8), its tight closure T-closure(G) will also be $\mathbb{Z}$-consistent. This means that, if G is *not* $\mathbb{Z}$-consistent, then T-closure$(G) = \bigsqcup \varnothing = \bot$; that is, the tight closure operator computes either a tightly closed graph or the bottom element. Therefore, tight closure is a kernel operator on the lattice of octagonal graphs, as was the case for strong closure.

An incremental closure procedure for obtaining the tight closure of an octagonal graph was defined in [13] and improved in [44]. The algorithm, which is also presented and discussed in [18, Section 4.3.5], maintains the tight closure of a system of octagonal constraints by performing at most $\mathrm{O}(n^2)$ operations each time a new constraint is added: thus, for m constraints, the worst case complexity is $\mathrm{O}(mn^2)$. In particular, for the case of a dense system of octagonal constraints where $m \in \mathrm{O}(n^2)$, the worst case complexity is $\mathrm{O}(n^4)$.

```
procedure tightening(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$)
    for $i := 0$ to $2n-1$ do
        $w[i,\bar{\imath}] := 2 \cdot \mathrm{floor}\bigl(w[i,\bar{\imath}]/2\bigr)$;
end procedure

procedure tight_closure_if_consistent(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$)
    shortest_path_closure($w$);
    tightening($w$);
    strengthening($w$);
end procedure
```

Fig. 5 A $\mathrm{O}(n^3)$ tight closure algorithm for $\mathbb{Z}$-consistent integer octagonal graphs

The following theorem shows that a more efficient tight closure algorithm can be obtained by a simple modification to the improved strong closure algorithm based on Theorem 3. Basically, the tightening inference rule (7) must be applied to ensure that property (8) holds before applying the strengthening inference rule (5).

Theorem 4 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed integer octagonal graph. Consider the graph $G_{\mathrm{T}} = (\mathcal{N}^{\pm}, w_{\mathrm{T}})$, where w_{T} is defined, for each $i, j \in \mathcal{N}^{\pm}$, by*

$$w_{\mathrm{T}}(i,j) := \min\biggl\{ w(i,j), \Bigl\lfloor \frac{w(i,\bar{\imath})}{2} \Bigr\rfloor + \Bigl\lfloor \frac{w(\bar{\jmath},j)}{2} \Bigr\rfloor \biggr\}. \tag{9}$$

Then, if G_{T} is an octagonal graph, $G_{\mathrm{T}} = \text{T-closure}(G)$.

Figure 5 shows the pseudo-code for a $\mathrm{O}(n^3)$ tight closure algorithm based on Theorem 4 and on the classical Floyd-Warshall shortest-path closure algorithm. As in the rational case, if the graph is sparse, a better complexity bound can be obtained by using Johnson's algorithm [26] for shortest-path closure: the worst case complexity will again be $\mathrm{O}(n^2 \log n + mn)$, which improves upon the $\mathrm{O}(mn^2)$ worst case complexity of [13, 44] when $m \in \Omega(\log n)$. In contrast, the same worst case complexity bound of [13,44] is obtained when considering the incremental variant of the tight closure algorithm.

It follows from the statement of Theorem 4 that an implementation based on it also needs to check the consistency of G_{T}. In principle, one could apply again a shortest-path closure procedure so as to check whether G_{T} contains some negative weight cycles. However, a more efficient solution can be obtained by exploiting the following result.

Theorem 5 *Let G and G_{T} be as defined in* Theorem 4. *Consider $G_{\mathrm{t}} = (\mathcal{N}^{\pm}, w_{\mathrm{t}})$ where, for each $i, j \in \mathcal{N}^{\pm}$,*

$$w_{\mathrm{t}}(i,j) := \begin{cases} 2\lfloor w(i,j)/2 \rfloor, & \text{if } j = \bar{\imath}; \\ w(i,j), & \text{otherwise}; \end{cases} \tag{10}$$

and suppose that, for all $i \in \mathcal{N}^{\pm}$, $w_{\mathrm{t}}(i,\bar{\imath}) + w_{\mathrm{t}}(\bar{\imath},i) \geq 0$. Then G_{T} is an octagonal graph.

The combination of the results stated in Theorems 4 and 5, together with the well known result for rational consistency, leads to an $\mathrm{O}(n^3)$ tight closure algorithm, such as that given by the pseudo-code in Figure 6, that computes the tight closure of any (possibly inconsistent) coherent integer-weighted graph returning the Boolean value 'true' if and only if the input graph is $\mathbb{Z}$-consistent.

```
function Z_consistent(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    for $i := 0$ to $2n-2$ step 2 do
        if $w[i,\bar{\imath}] + w[\bar{\imath},i] < 0$ return false;
    return true;
end function

function tight_closure(var $w\,[0\,..\,2n-1]\,[0\,..\,2n-1]$) : bool
    if not shortest_path_closure($w$) return false;
    tightening($w$);
    if not Z_consistent($w$) return false;
    strengthening($w$);
    return true;
end function
```

Fig. 6 A $\mathrm{O}(n^3)$ tight closure algorithm for integer coherent graphs

6.1 A Tight Reduction Procedure for Integer Octagonal Graphs

A definition of tight reduction of a $\mathbb{Z}$-consistent integer octagonal graph could be easily introduced by a blind adaptation of (strong) reduction. Such an approach, however, would result in a rather unnatural tight reduction procedure.

Example 1 Let $\mathcal{C} = \{v_0 \leq 0, -v_0 \leq 0\}$ be a system of integer octagonal constraints on $\mathcal{V} = \{v_0\}$. Then, $\mathcal{N}^{\pm} = \{v_0^+, v_0^-\}$ and we encode $\mathcal{C}$ by the integer octagonal graph $G = (\mathcal{N}^{\pm}, w)$, where $w(i,j) = 0$ for all $i, j \in \mathcal{N}^{\pm}$. Clearly, $G = \text{T-closure}(G)$. Let $G_1 = (\mathcal{N}^{\pm}, w_1)$ be the subgraph of G having only arcs (v_0^+, v_0^-) and (v_0^-, v_0^+): it is easy to observe that $\text{T-closure}(G_1) = \text{T-closure}(G)$ and no proper subgraph of G_1 satisfies this property. However, the maximum integer octagonal graph equivalent to G is $G_2 = (\mathcal{N}^{\pm}, w_2)$, where the two arcs mentioned above have weights $w_2(v_0^+, v_0^-) = 1$ and $w_2(v_0^-, v_0^+) = -1$. Note that G_2 represents the (non-redundant, but unnecessarily involved) integer octagonal constraints $\mathcal{C}_2 = \{2v_0 \leq 1, -2v_0 \leq -1\}$; in particular, G_2 is not a subgraph of the tight closure of G. □

To avoid cases such as Example 1, we require that a tight reduction for G is found among the subgraphs of $\text{T-closure}(G)$.

Definition 14 (Tightly reduced graph.) A $\mathbb{Z}$-consistent integer octagonal graph G_1 is *tightly reduced* if G_1 is a subgraph of $\text{T-closure}(G_1)$ and, for each integer octagonal graph G_2 such that G_2 is a proper subgraph of G_1, $\text{T-closure}(G_1) \neq \text{T-closure}(G_2)$. A *tight reduction* for the $\mathbb{Z}$-consistent integer octagonal graph G is any tightly reduced graph G_{R} such that $\text{T-closure}(G) = \text{T-closure}(G_{\mathrm{R}})$.

Unsurprisingly, the strong reduction algorithm of Section 4.1, in isolation, will not compute a tight reduction for a $\mathbb{Z}$-consistent integer octagonal graph, since it disregards redundancies caused by the tightening inference rule (8).

Example 2 Let $\mathcal{C} = \{v_0 - v_1 \leq 0, v_0 + v_1 \leq 3, v_0 \leq 1\}$ be an integer octagonal constraint system on $\mathcal{V} = \{v_0, v_1\}$. By applying the strong reduction algorithm of Section 4.1, we would conclude that no constraint in $\mathcal{C}$ is redundant. However, $v_0 \leq 1$ is indeed redundant, as it can be obtained by transitivity (3) followed by tightening (7). □

In the example above, the constraint $v_0 \leq 1$ is encoded by arc (v_0^+, v_0^-) having weight 2: this arc is strongly atomic, hence it is not identified as redundant by the strong reduction procedure. However, if the arc is untightened, we would obtain an arc having weight 3, hence *not* (strongly) atomic, i.e., redundant.

Definition 15 (Graph untightening.) Let $G = (\mathcal{N}^\pm, w)$ be a $\mathbb{Z}$-consistent integer octagonal graph. Arc $(i, \bar{\imath})$ is *tightly redundant* in G if $w(i, \bar{\imath})$ is even and there exists $k \in \mathcal{N}^\pm \setminus \{i, \bar{\imath}\}$ such that $w(i,k) + w(k, \bar{\imath}) \leq w(i, \bar{\imath}) + 1$. The *untightening* of G is the graph $U = (\mathcal{N}^\pm, w_{\mathrm{U}})$, where the weight function w_{U} is defined, for all $i, j \in \mathcal{N}^\pm$, by

$$w_{\mathrm{U}}(i,j) = \begin{cases} +\infty, & \text{if } j = \bar{\imath} \text{ and } (i, \bar{\imath}) \text{ is tightly redundant in } G; \\ w(i,j), & \text{otherwise.} \end{cases}$$

A correct and efficient tight reduction procedure is obtained by a simple adaptation of the strong reduction procedure of Section 4.1: in step 1 of the algorithm, we compute tight closure (rather than strong closure); steps 2–5 are unchanged and yield graphs A and Z (resulting from steps 4 and 5, respectively); in step 6, the final result is computed as $U \sqcap Z$, where U is the untightening of A.

Theorem 6 *Given a $\mathbb{Z}$-consistent integer octagonal graph G, the tight reduction algorithm computes a tight reduction for G.*

The tight reduction algorithm has cubic complexity and, when the variable ordering is fixed, computes a canonical form for integer octagonal graphs.

To the best of our knowledge, this is the first time that a reduction procedure for the class of integer octagonal constraints is proposed and proved correct. Previous approaches were based either on more general classes of constraints (thereby incurring into avoidable inefficiencies), or on the rational relaxation of the constraints (thereby failing to identify all redundancies).

7 A Note on Floating-Point Computations

The formal definitions and theoretical results concerning weighted directed graphs assume that the data type adopted for the weights allows for their exact representation and propagation, therefore avoiding any possible problems with overflow, underflow and rounding errors. In actual implementations, by trading some efficiency, such an assumption can be easily fulfilled by adopting unbounded precision data types (integer or rational, depending on the underlying problem), as provided by several software libraries such as GMP, the GNU Multiple Precision arithmetic library.

Catering for inexact data types needs some care since, in particular, this requires that the notion of a correct approximation is made explicit. As an example, consider the case of shortest-path closure for a weighted graph whose variables are meant to take rational values. The computation of the weight of a path corresponds to several applications of the constraint inference rule for transitivity: approximating such a weight from above leads to a correct (but possibly imprecise) inference, whereas approximating it from below is unsafe. Therefore, an implementation willing to correctly approximate the rational shortest-path closure using unbounded precision integers will just need to correctly approximate the input graph (rounding upwards all the rational weights) and then apply any standard closure algorithm.

Example 3 Consider the graph $G = (\mathcal{N}, w)$ having $k + 1$ nodes $\mathcal{N} = \{0, \dots, k\}$ connected to form a simple path by $k \geq 1$ arcs having the same rational weight:

$$\forall i \in \{1, \dots, k\} : w(i - 1, i) = 1/k.$$

Clearly, the one and only (hence shortest) path $\pi = 0 \cdots k$ from node 0 to node k has rational weight $w(\pi) = 1$, which happens to be an integer. By approximating the rational weight function w using an integer weight function $w_{\mathbb{Z}}$ and then applying closure, we will compute $w_{\mathbb{Z}}(\pi) = \sum_{i=1}^{k} w_{\mathbb{Z}}(i - 1, i) = \sum_{i=1}^{k} \lceil 1/k \rceil = k$. □

Even though a correct approximation has occurred with respect to the ideal rational computation, the implementation based on unbounded integers can be said to be precise in that no further approximations are introduced besides those incurred during the initial integer encoding of the rational weights. In few words, the representation of input weights is approximate, but their propagation is accurate. In particular, the approximated algorithm still implements a lower closure operator on the domain of weighted graphs.

Things are more difficult if we consider a data type that causes an inexact (though correct) propagation of the weights. This is the case, for instance, when adopting bounded precision floating point coefficients. Correctness can be achieved, almost as easily as before, by applying the appropriate rounding mode not only when approximating the rational weights of the input graph, but also when computing operations such as floating point addition, which is an inexact operation. When reasoning on precision, however, an important issue surfaces: the basic definition of weight of a path as given in Section 2, assumes that addition is an associative operation, but such a property fails to hold for floating point data types. In order to compute the minimum sum of the arc weights in a path, we would need to identify an optimal order of application for additions, probably resulting in algorithms that are too slow.

As an alternative approach, we could incrementally modify the weighted graph by repeated applications of the transitive inference rule (3), until either a fixpoint is reached or an inconsistency is detected. This will result in the definition of a lower closure operator on the lattice $\mathbb{G}$, below denoted as 'fpw', that correctly approximates the ideal one. However, the computed weights might be larger than the minimal ones, as shown in the following example.

Example 4 Let '$\uparrow$' denote the upward approximation of a rational number using a floating point number and '$\boxplus$' denote floating point addition with upward rounding. Consider the graph $G = (\mathcal{N}, w)$ where the nodes in $\mathcal{N} = \{a, b, c, d\}$ are connected by three arcs having (rational and floating point) weights $w(a, b) = \uparrow(1/3)$, $w(b, c) = \uparrow(1/2) = 1/2$ and $w(c, d) = -\uparrow(1/3)$. Clearly, the exact rational weight of path $\pi = abcd$ is $w(abcd) = 1/2$. In principle, the same rational weight could also be computed using floating points, as follows:

$$w(b, c) \boxplus \bigl[w(a, b) \boxplus w(c, d)\bigr] = 1/2.$$

However, the result is obtained by a reassociation of the sequence of additions that corresponds to none of the subpaths of π; hence, it can not be computed by incremental applications of the transitivity inference rule, which would instead result in the following approximation:

$$\begin{aligned}
\mathrm{fpw}(abcd) &= \mathrm{fpw}(ab) \boxplus \mathrm{fpw}(bcd) \\
&= \mathrm{fpw}(ab) \boxplus \bigl[\mathrm{fpw}(bc) \boxplus \mathrm{fpw}(cd)\bigr] \\
&= w(a,b) \boxplus \bigl[w(b,c) \boxplus w(c,d)\bigr] \\
&= {\uparrow}(1/3) \boxplus \Bigl[1/2 \boxplus \bigl(-{\uparrow}(1/3)\bigr)\Bigr] > 1/2.
\end{aligned}$$

□

It is tempting to adopt 'fpw' as the natural definition of *floating point weight* of a path, since by construction it enjoys an optimal substructure property, so that its implementation can be based on dynamic programming techniques. However, the following example shows that the classical Floyd-Warshall algorithm for rational/integer shortest-path closure is not enough to compute 'fpw'.

Example 5 Consider the graph $G = (\mathcal{N}, w)$ where the nodes in $\mathcal{N} = \{a, b, c, d\}$ are connected by three arcs having floating point weights $w(a,b) = 1/2$, $w(b,c) = {\uparrow}(1/3)$ and $w(c,d) = -{\uparrow}(1/3)$. Let w_{fw} be the weight function computed by the classical Floyd-Warshall algorithm, instantiated to use the given floating point data type with upward rounding operations. Clearly, the shortest path from node a to node d has (rational and floating point) weight $w(abcd) = \mathrm{fpw}(abcd) = 1/2$. However, the algorithm will instead compute

$$\begin{aligned}
w_{\mathrm{fw}}(a,d) &= w_{\mathrm{fw}}(abc) \boxplus w_{\mathrm{fw}}(cd) \\
&= \bigl[w_{\mathrm{fw}}(ab) \boxplus w_{\mathrm{fw}}(bc)\bigr] \boxplus w_{\mathrm{fw}}(cd) \\
&= \bigl[w(a,b) \boxplus w(b,c)\bigr] \boxplus w(c,d) \\
&= \bigl[1/2 \boxplus {\uparrow}(1/3)\bigr] \boxplus \bigl(-{\uparrow}(1/3)\bigr) > 1/2,
\end{aligned}$$

even though $w_{\mathrm{fw}}(bcd) = \mathrm{fpw}(bcd) = 0$. This happens because the algorithm, by assuming associativity, does not attempt to improve the computed path weight using the other association $\mathrm{fpw}(ab) \boxplus \mathrm{fpw}(bcd)$. □

In principle, the computation of 'fpw' can be carried out by repeatedly applying the Floyd-Warshall algorithm until reaching a fixpoint: we conjecture that such an approach would achieve an $\mathrm{O}(n^4)$ worst case complexity bound.

It is worth stressing that the observations and examples made above, illustrating the issues caused by inexact floating point addition, can be easily adapted so as to show corresponding issues due to overflows. Such a case is even more general, in that it applies not only to floating point data types, but also to bounded precision integer data types.

In summary, the implementation of a truly semantic abstract domain is not possible using only bounded precision and inexact computations. Nonetheless, by quotienting the syntactic domain of graph representations according to the approximate closure/reduction operators, we obtain a more abstract (but still syntactic) domain where most, although not all, of the redundancies have been removed. As a consequence, the negative side to the adoption of a syntactic abstract domain will be greatly mitigated.

8 Related Work

The shortest-path reduction algorithm of [32] was also considered in [18]. Here the reduction procedure is used, as originally proposed in [32], as a tool for the computation of *hollow* representations for BDGs, i.e., representations that are appropriate to the case of sparse graphs, so as to obtain memory space savings. This thesis does not appear to identify the positive interaction between reduction and widening and, as a consequence, the thesis conjectures that the computation of hollow representations could compromise the convergence of the abstract iteration sequence (see [18, Section 3.8.2]). When adopting our semantic widenings, the sparse representation can be used even when storing abstract elements at widening points. An adaptation of the reduction algorithm for the case of octagonal graphs is defined in [18, Section 4.5.2] although this differs from the one proposed in Section 4.1. It turns out that the algorithm of [18, Section 4.5.2] may not obtain a strongly reduced graph in the sense of Definition 6: the adapted hollow representation for octagonal graphs can still encode some redundant constraints, as it does not take into proper account the peculiarities of the singular zero-equivalence class. No reduction algorithm is proposed in [18] for the integer octagonal case.

In 2001, Seater and Wonnacott observed that the Omega test [45] could be used to decide the satisfiability of a (non trivially redundant) system of UTVPI integer constraints achieving an $\mathrm{O}(n^3)$ worst-case complexity bound [46]. In 2005, Lahiri and Musuvathi proposed an $\mathrm{O}(mn)$ algorithm for the same satisfiability problem [47], where m is the number of constraints. They also sketched, without formal definitions and proofs, an $\mathrm{O}(n^3)$ tight closure algorithm. Still in 2005, Miné proposed a modification of the strong (i.e., non-tight) closure algorithm for *rational* octagonal constraints and argued that this would provide a good and efficient approximation of tight closure [18, 48]. In the same year we showed that the algorithm for computing the strong closure of rational octagonal constraints as described in [8, 18, 48] could be simplified with a consequential improvement in its efficiency [20, 21].

Some of the issues related to the adoption of an inexact data type for the representation of graph weights have also been discussed in [18, Sections 7.3.2 and 7.5.3].

9 Conclusions

We have address the problem of the construction of full-fledged, efficient and provably correct weakly-relational numeric abstract domains.

By considering the semantic abstract domains of geometric shapes, instead of the more concrete abstract domains of their syntactic representations in terms of constraint networks, we have shown how proper widening operators can be derived for several weakly-relational numeric abstractions, including the domain of bounded difference shapes and (rational and integer) octagonal shapes.

For what concerns the efficient representation of octagonal shapes by means of octagonal graphs, we have specified and proved correct strong and tight reduction procedures, as well as more efficient strong and tight closure procedures.

The theoretical results concerning weighted directed graphs hold when the data type adopted for the representation of weights allows for exact computations. If a floating-point data type is considered, then most of these results will be broken due to rounding errors, so that the implementation of a truly semantic abstract domain will not be possible. Nonetheless, the (approximate) reduction operators allow for the removal of most of the syntactic redundancies.

The algorithms presented in this paper have been experimentally validated and are available as part of the Parma Polyhedra Library [40], which is free software distributed under the GNU General Public License. (`http://www.cs.unipr.it/ppl/`).

Acknowledgements We would like to thank Katy Dobson and Elena Mazzi for useful discussions on topics related to this work.

References

1. Cousot, P., Cousot, R.: Static determination of dynamic properties of programs. In: B. Robinet (ed.) Proceedings of the Second International Symposium on Programming, pp. 106–130. Dunod, Paris, France, Paris, France (1976).
2. Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, pp. 84–96. ACM Press, Tucson, Arizona (1978).
3. Bagnara, R.: Data-flow analysis for constraint logic-based languages. Ph.D. thesis, Dipartimento di Informatica, Università di Pisa, Pisa, Italy (1997). Printed as Report TD-1/97
4. Bagnara, R., Giacobazzi, R., Levi, G.: Static analysis of CLP programs over numeric domains. In: M. Billaud, P. Castéran, M. Corsini, K. Musumbu, A. Rauzy (eds.) Actes "Workshop on Static Analysis '92", *Bigre*, vol. 81–82, pp. 43–50. Atelier Irisa, IRISA Campus de Beaulieu, Bordeaux (1992). Extended abstract
5. Bagnara, R., Giacobazzi, R., Levi, G.: An application of constraint propagation to dataflow analysis. In: Proceedings of "The Ninth Conference on Artificial Intelligence for Applications", pp. 270–276. IEEE Computer Society Press, Los Alamitos, CA, Orlando, Florida (1993)
6. Miné, A.: A new numerical abstract domain based on difference-bound matrices. In: O. Danvy, A. Filinski (eds.) Proceedings of the 2nd Symposium on Programs as Data Objects (PADO 2001), *Lecture Notes in Computer Science*, vol. 2053, pp. 155–172. Springer-Verlag, Berlin, Aarhus, Denmark (2001)
7. Shaham, R., Kolodner, E.K., Sagiv, S.: Automatic removal of array memory leaks in Java. In: D.A. Watt (ed.) Proceedings of the 9th International Conference on Compiler Construction (CC 2000), *Lecture Notes in Computer Science*, vol. 1781, pp. 50–66. Springer-Verlag, Berlin, Berlin, Germany (2000)
8. Miné, A.: The octagon abstract domain. In: Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01), pp. 310–319. IEEE Computer Society Press, Stuttgart, Germany (2001)
9. Simon, A., King, A., Howe, J.M.: Two variables per linear inequality as an abstract domain. In: M. Leuschel (ed.) Logic Based Program Synthesis and Tranformation, 12th International Workshop, *Lecture Notes in Computer Science*, vol. 2664, pp. 71–89. Springer-Verlag, Berlin, Madrid, Spain (2002)
10. Clarisó, R., Cortadella, J.: The octahedron abstract domain. In: R. Giacobazzi (ed.) Static Analysis: Proceedings of the 11th International Symposium, *Lecture Notes in Computer Science*, vol. 3148, pp. 312–327. Springer-Verlag, Berlin, Verona, Italy (2004)
11. Sankaranarayanan, S., Sipma, H.B., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: R. Cousot (ed.) Verification, Model Checking and Abstract Interpretation: Proceedings of the 6th International Conference (VMCAI 2005), *Lecture Notes in Computer Science*, vol. 3385, pp. 25–41. Springer-Verlag, Berlin, Paris, France (2005)
12. Miné, A.: A few graph-based relational numerical abstract domains. In: M.V. Hermenegildo, G. Puebla (eds.) Static Analysis: Proceedings of the 9th International Symposium, *Lecture Notes in Computer Science*, vol. 2477, pp. 117–132. Springer-Verlag, Berlin, Madrid, Spain (2002)
13. Jaffar, J., Maher, M.J., Stuckey, P.J., Yap, R.H.C.: Beyond finite domains. In: A. Borning (ed.) Principles and Practice of Constraint Programming: Proceedings of the Second International Workshop, *Lecture Notes in Computer Science*, vol. 874, pp. 86–94. Springer-Verlag, Berlin, Rosario, Orcas Island, Washington, USA (1994)

14. Lagarias, J.C.: The computational complexity of simultaneous Diophantine approximation problems. SIAM Journal on Computing **14**(1), 196–209 (1985)
15. Balasundaram, V., Kennedy, K.: A technique for summarizing data access and its use in parallelism enhancing transformations. In: B. Knobe (ed.) Proceedings of the ACM SIGPLAN'89 Conference on Programming Language Design and Implementation (PLDI), *ACM SIGPLAN Notices*, vol. 24(7), pp. 41–53. ACM Press, Portland, Oregon, USA (1989)
16. Ball, T., Cook, B., Lahiri, S.K., Zhang, L.: Zapato: Automatic theorem proving for predicate abstraction refinement. In: R. Alur, D. Peled (eds.) Computer Aided Verification: Proceedings of the 16th International Conference, *Lecture Notes in Computer Science*, vol. 3114, pp. 457–461. Springer-Verlag, Berlin, Boston, MA, USA (2004)
17. Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X.: The Astrée analyzer. In: M. Sagiv (ed.) Programming Languages and Systems, Proceedings of the 14th European Symposium on Programming, *Lecture Notes in Computer Science*, vol. 3444, pp. 21–30. Springer-Verlag, Berlin, Edinburgh, UK (2005)
18. Miné, A.: Weakly relational numerical abstract domains. Ph.D. thesis, École Polytechnique, Paris, France (2005)
19. Halbwachs, N.: Détermination automatique de relations linéaires vérifiées par les variables d'un programme. Thèse de 3$^{\text{ème}}$ cycle d'informatique, Université scientifique et médicale de Grenoble, Grenoble, France (1979)
20. Bagnara, R., Hill, P.M., Mazzi, E., Zaffanella, E.: Widening operators for weakly-relational numeric abstractions. In: C. Hankin, I. Siveroni (eds.) Static Analysis: Proceedings of the 12th International Symposium, *Lecture Notes in Computer Science*, vol. 3672, pp. 3–18. Springer-Verlag, Berlin, London, UK (2005)
21. Bagnara, R., Hill, P.M., Mazzi, E., Zaffanella, E.: Widening operators for weakly-relational numeric abstractions. Quaderno 399, Dipartimento di Matematica, Università di Parma, Italy (2005). Available at `http://www.cs.unipr.it/Publications/`
22. Bagnara, R., Hill, P.M., Zaffanella, E.: An improved tight closure algorithm for integer octagonal constraints. In: F. Logozzo, D. Peled, L. Zuck (eds.) Verification, Model Checking and Abstract Interpretation: Proceedings of the 9th International Conference (VMCAI 2008), *Lecture Notes in Computer Science*, vol. 4905, pp. 8–21. Springer-Verlag, Berlin, San Francisco, USA (2008).
23. Birkhoff, G.: Lattice Theory, *Colloquium Publications*, vol. XXV, third edn. American Mathematical Society, Providence, Rhode Island, USA (1967)
24. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages, pp. 238–252. ACM Press, New York (1977)
25. Cousot, P., Cousot, R.: Systematic design of program analysis frameworks. In: Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages, pp. 269–282. ACM Press, New York (1979).
26. Cormen, T.H., Leiserson, T.E., Rivest, R.L.: Introduction to Algorithms. The MIT Press, Cambridge, MA (1990)
27. Pratt, V.R.: Two easy theories whose combination is hard (1977). Memo sent to Nelson and Oppen concerning a preprint of their paper [49]
28. Bellman, R.: Dynamic Programming. Princeton University Press (1957)
29. Allen, J.F., Kautz, H.A.: A model of naive temporal reasoning. In: J.R. Hobbs, R. Moore (eds.) Formal Theories of the Commonsense World, pp. 251–268. Ablex, Norwood, NJ (1985)
30. Davis, E.: Constraint propagation with interval labels. Artificial Intelligence **32**(3), 281–331 (1987)
31. Dill, D.L.: Timing assumptions and verification of finite-state concurrent systems. In: J. Sifakis (ed.) Proceedings of the International Workshop on Automatic Verification Methods for Finite State Systems, *Lecture Notes in Computer Science*, vol. 407, pp. 197–212. Springer-Verlag, Berlin, Grenoble, France (1989)
32. Larsen, K., Larsson, F., Pettersson, P., Yi, W.: Efficient verification of real-time systems: Compact data structure and state-space reduction. In: Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS'97), pp. 14–24. IEEE Computer Society Press, San Francisco, CA (1997)
33. Bagnara, R., Hill, P.M., Ricci, E., Zaffanella, E.: Precise widening operators for convex polyhedra. Science of Computer Programming **58**(1–2), 28–56 (2005).

34. Aho, A.V., Garey, M.R., Ullman, J.D.: The transitive reduction of a directed graph. SIAM Journal on Computing **1**(2), 131–137 (1972)
35. Halbwachs, N., Proy, Y.E., Raymond, P.: Verification of linear hybrid systems by means of convex approximations. In: B. Le Charlier (ed.) Static Analysis: Proceedings of the 1st International Symposium, *Lecture Notes in Computer Science*, vol. 864, pp. 223–237. Springer-Verlag, Berlin, Namur, Belgium (1994).
36. Halbwachs, N., Proy, Y.E., Roumanoff, P.: Verification of real-time systems using linear relation analysis. Formal Methods in System Design **11**(2), 157–185 (1997)
37. Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X.: Design and implementation of a special-purpose static program analyzer for safety-critical real-time embedded software. In: T.Æ. Mogensen, D.A. Schmidt, I.H. Sudborough (eds.) The Essence of Computation, Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones [on occasion of his 60th birthday], *Lecture Notes in Computer Science*, vol. 2566, pp. 85–108. Springer-Verlag, Berlin (2002)
38. Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X.: A static analyzer for large safety-critical software. In: Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI'03), pp. 196–207. ACM Press, San Diego, California, USA (2003)
39. Miné, A.: Relational abstract domains for the detection of floating-point run-time errors. In: D. Schmidt (ed.) Programming Languages and Systems: Proceedings of the 13th European Symposium on Programming, *Lecture Notes in Computer Science*, vol. 2986, pp. 3–17. Springer-Verlag, Berlin, Barcelona, Spain (2004)
40. Bagnara, R., Hill, P.M., Zaffanella, E.: The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. Science of Computer Programming **72**(1–2), 3–21 (2008).
41. Bagnara, R., Hill, P.M., Zaffanella, E.: Widening operators for powerset domains. Software Tools for Technology Transfer **8**(4/5), 449–466 (2006). In the printed version of this article, all the figures have been improperly printed (rendering them useless). See [50].
42. Miné, A.: The Octagon Abstract Domain Library. Semantics and Abstract Interpretation Computer Science Lab., École Normale Supérieure, Paris, France, release 0.9.6 edn. (2002). Available at `http://www.di.ens.fr/~mine/oct/`
43. Venet, A., Brat, G.: Precise and efficient static array bound checking for large embedded C programs. In: Proceedings of the ACM SIGPLAN 2004 Conference on Programming Language Design and Implementation (PLDI'04), pp. 231–242. ACM Press, Washington, DC, USA (2004)
44. Harvey, W., Stuckey, P.J.: A unit two variable per inequality integer constraint solver for constraint logic programming. In: M. Patel (ed.) ACSC'97: Proceedings of the 20th Australasian Computer Science Conference, vol. 19, pp. 102–111. Australian Computer Science Communications (1997)
45. Pugh, W.: The Omega test: A fast and practical integer programming algorithm for dependence analysis. In: Supercomputing 91 (1991)
46. Seater, R., Wonnacott, D.: Polynomial time array dataflow analysis. In: H.G. Dietz (ed.) Languages and Compilers for Parallel Computing, 14th International Workshop, *Lecture Notes in Computer Science*, vol. 2624, pp. 411–426. Springer-Verlag, Berlin, Cumberland Falls, KY, USA (2001)
47. Lahiri, S.K., Musuvathi, M.: An efficient decision procedure for UTVPI constraints. In: B. Gramlich (ed.) Frontiers of Combining Systems: Proceedings of the 5th International Workshop, FroCoS 2005, *Lecture Notes in Artificial Intelligence*, vol. 3717, pp. 168–183. Springer-Verlag, Berlin, Vienna, Austria (2005)
48. Miné, A.: The octagon abstract domain. Higher-Order and Symbolic Computation **19**(1), 31–100 (2006)
49. Nelson, G., Oppen, D.C.: Fast decision algorithms based on Union and Find. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77), pp. 114–119. IEEE Computer Society Press, Providence, RI, USA (1977). The journal version of this paper is [51]
50. Bagnara, R., Hill, P.M., Zaffanella, E.: Widening operators for powerset domains. Software Tools for Technology Transfer **9**(3/4), 413–414 (2007). Erratum to [41] containing all the figures properly printed.
51. Nelson, G., Oppen, D.C.: Fast decision procedures based on congruence closure. Journal of the ACM **27**(2), 356–364 (1980). An earlier version of this paper is [49]

A Proofs

In the following we provide some further notation and terminology, as well as the formal proofs for all the results stated in the previous sections. Most of the proofs are based on auxiliary lemmas, some of which are well-known results of graph theory.

A.1 Further Notation and Simple Results on Weighted Graphs

For weighted directed graphs $G_1, G_2 \in \mathbb{G}$, we will write $G_1 \lhd G_2$ when $G_1 \unlhd G_2$ and $G_1 \neq G_2$.

Let $G = (\mathcal{N}^{\pm}, w)$ be an octagonal graph. If $\pi = j_0 \cdots j_p$ is a path in G, then $\overline{\pi}$ denotes the path $\overline{\jmath}_p \cdots \overline{\jmath}_0$. Note that, by the coherence assumption (4) of Definition 4, $w(\pi) = w(\overline{\pi})$. A path π in G is said to be (strongly) atomic if all the arcs in π are (strongly) atomic in G.

Definition 16 (Weakly closed graph.) An octagonal graph $G = (\mathcal{N}^{\pm}, w)$ is *weakly closed* if it satisfies rule (1) in Definition 1 and

$$\forall i, j, k \in \mathcal{N}^{\pm} : w(i,k) + w(k,j) \geq \min\Big\{ w(i,j), \big(w(i,\overline{\imath}) + w(\overline{\jmath}, j)\big)/2 \Big\}. \tag{11}$$

The following three lemmas (namely, Lemmas 4, 5 and 6) recall results for weighted graphs that are well-known and/or rather simple. The reader that, for the sake of completeness, is interested in their proofs is referred to the corresponding lemmas in [21] (Lemmas 4, 5 and 7, respectively).

Lemma 4 *Let $G = (\mathcal{N}, w)$ be a consistent graph. Let also $\pi = i \cdots j$ be a path in G. Then there exists a simple path $\pi' = i \cdots j$ such that $w(\pi') \leq w(\pi)$. Moreover each arc in π' is also an arc in π.*

Lemma 5 *Let $G = (\mathcal{N}, w)$ be a closed graph. Then, for any path $\pi = i \cdots j$ in G, it holds that $w(i,j) \leq w(\pi)$.*

Lemma 6 *Let $G = (\mathcal{N}, w)$ be a closed and zero-cycle free graph. Let also $\pi = i \cdots j$ be a proper path in G such that $w(i,j) = w(\pi)$. Then π is a simple path.*

Lemma 7 *Let $G = (\mathcal{N}, w)$ be a consistent graph and $G_{\mathrm{c}} = (\mathcal{N}, w_{\mathrm{c}})$ be the closure of G. Let also (z_1, z_2) be an arc in G_{c}. Then there exists a simple path $\pi = z_1 \cdots z_2$ in G such that $w_{\mathrm{c}}(z_1, z_2) = w(\pi)$.*

Proof For each $i, j \in \mathcal{N}$, let $\mathrm{sp}(i,j) := \min\big\{ w(\pi) \bigm| \pi = i \cdots j \text{ is a path in } G \big\}$, so that the graph $G_{\mathrm{sp}} = (\mathcal{N}, \mathrm{sp})$ directly encodes the shortest-path weights of G. (sp is well defined, because G is consistent and $\min \varnothing = +\infty$.) For each arc (i,j) of G_{sp}, let also $\pi_{ij} = i \cdots j$ be a corresponding simple shortest-path in G such that $w(\pi_{ij}) = \mathrm{sp}(i,j)$.

Note that G_{sp} satisfies property (1) of Definition 1, since the 0-length path i is a shortest-path from node i to itself, so that $w(\pi_{ii}) = \mathrm{sp}(i,i) = 0$. To see that G_{sp} also satisfies property (2) of Definition 1, by contraposition, we assume that there exist $i, j, k \in \mathcal{N}$ such that $\mathrm{sp}(i,j) > \mathrm{sp}(i,k) + \mathrm{sp}(k,j)$. Hence, $w(\pi_{ij}) > w(\pi_{ik}) + w(\pi_{kj})$, i.e., $w(\pi_{ij}) > w(\pi_{ik} :: \pi_{kj})$, contradicting the assumption that π_{ij} is a shortest-path in G. Hence, G_{sp} is closed and, since by construction $G_{\mathrm{sp}} \unlhd G$, by Definition 1 we obtain $G_{\mathrm{sp}} \unlhd G_{\mathrm{c}}$.

Now let (i,j) be any arc in G_{sp}, so that π_{ij} is a shortest-path in G. Since $G_{\mathrm{c}} \unlhd G$, π_{ij} is also a path in G_{c} and $w_{\mathrm{c}}(\pi_{ij}) \leq w(\pi_{ij}) = \mathrm{sp}(i,j)$. By Lemma 5, $w_{\mathrm{c}}(i,j) \leq w_{\mathrm{c}}(\pi_{ij})$. Hence, $w_{\mathrm{c}}(i,j) \leq \mathrm{sp}(i,j)$, so that $G_{\mathrm{c}} \unlhd G_{\mathrm{sp}}$.

As $G_{\mathrm{c}} = G_{\mathrm{sp}}$, the thesis follows by definition of G_{sp}, taking $\pi = \pi_{z_1 z_2}$. □

A.2 Proofs of the Results Stated in Sections 5 and 6

We anticipate here the proofs of the results stated in Sections 5 and 6 because the proofs of theorems stated in Section 4.1 use these results.

Lemma 8 *Let $G = (\mathcal{N}^{\pm}, w)$ be a weakly closed octagonal graph. Then, for all $i, j \in \mathcal{N}^{\pm}$,*

$$w(i, \bar{\imath}) \leq 2w(i, j) + w(j, \bar{\jmath}).$$

Proof Let $i, j \in \mathcal{N}^{\pm}$. Rule (11) of Definition 16, applied to the triple of nodes $i, \bar{\imath}, j$ gives

$$w(i, \bar{\imath}) \leq w(i, j) + w(j, \bar{\imath}).$$

Moreover, with the triple of nodes $j, \bar{\imath}, \bar{\jmath}$, the same rule (11) implies that one of the following two inequalities holds:[5]

$$w(j, \bar{\imath}) \leq w(j, \bar{\jmath}) + w(\bar{\jmath}, \bar{\imath}), \qquad 2w(j, \bar{\imath}) \leq w(j, \bar{\jmath}) + w(i, \bar{\imath}).$$

If the first holds, then

$$\begin{aligned} w(i, \bar{\imath}) &\leq w(i, j) + w(j, \bar{\imath}) \\ &\leq w(i, j) + w(j, \bar{\jmath}) + w(\bar{\jmath}, \bar{\imath}) \\ &= 2w(i, j) + w(j, \bar{\jmath}), \end{aligned}$$

where the last step follows from the coherence assumption (4) of Definition 4. On the other hand, if the second inequality holds, then

$$\begin{aligned} 2w(i, \bar{\imath}) &\leq 2w(i, j) + 2w(j, \bar{\imath}) \\ &\leq 2w(i, j) + w(j, \bar{\jmath}) + w(i, \bar{\imath}). \end{aligned}$$

and hence again we have

$$w(i, \bar{\imath}) \leq 2w(i, j) + w(j, \bar{\jmath}).$$

□

Lemma 9 *Let $G = (\mathcal{N}^{\pm}, w)$ be a weakly closed octagonal graph and $i, j \in \mathcal{N}^{\pm}$ be such that*

$$2w(i, j) > w(i, \bar{\imath}) + w(\bar{\jmath}, j). \tag{12}$$

Let $G_{\mathrm{s}} = (\mathcal{N}^{\pm}, w_{\mathrm{s}})$ where, for each $h_1, h_2 \in \mathcal{N}^{\pm}$,

$$w_{\mathrm{s}}(h_1, h_2) := \begin{cases} \bigl(w(i, \bar{\imath}) + w(\bar{\jmath}, j)\bigr)/2, & \text{if } (h_1, h_2) \in \bigl\{(i, j), (\bar{\jmath}, \bar{\imath})\bigr\}; \\ w(h_1, h_2), & \text{otherwise.} \end{cases} \tag{13}$$

Then G_{s} is a weakly closed octagonal graph.

Proof By Definition 4, the graph G_{s} is octagonal. By construction, $G_{\mathrm{s}} \trianglelefteq G$. As G is weakly closed, rule (1) of Definition 1 holds so that $w(i, i) = 0$. As G is consistent, it has no negative weight cycles and hence, $2w(i, i) \leq w(i, \bar{\imath}) + w(\bar{\imath}, i)$. Thus, by hypothesis (12), we have $i \neq j$ and G_{s} satisfies rule (1) of Definition 1.

Observe that, as G is weakly closed, rule (11) of Definition 16 holds. Also, since $G_{\mathrm{s}} \trianglelefteq G$,

$$\forall h_1, h_2 \in \mathcal{N}^{\pm} : w_{\mathrm{s}}(h_1, h_2) \leq w(h_1, h_2). \tag{14}$$

Suppose $z_1, z_2, z \in \mathcal{N}^{\pm}$. Then, by Definition 16, to prove that G_{s} is weakly closed, we need to show that

$$w_{\mathrm{s}}(z_1, z) + w_{\mathrm{s}}(z, z_2) \geq \min\Bigl\{w_{\mathrm{s}}(z_1, z_2), \bigl(w_{\mathrm{s}}(z_1, \bar{z}_1) + w_{\mathrm{s}}(\bar{z}_2, z_2)\bigr)/2\Bigr\}. \tag{15}$$

If $(z_1, z), (z, z_2) \notin \bigl\{(i, j), (\bar{\jmath}, \bar{\imath})\bigr\}$, then, by (13), we have $w_{\mathrm{s}}(z_1, z) = w(z_1, z)$ and $w_{\mathrm{s}}(z, z_2) = w(z, z_2)$ and it follows from rule (11) and property (14) that (15) holds.

[5] Note that rule (11) has the form $A \geq \min\{B, C\}$; from this we can conclude that $A \geq B$ (when B is the minimum) or $B \geq C$ (when C is the minimum).

The remaining cases where $(z_1, z) \in \bigl\{(i,j), (\bar{\jmath}, \bar{\imath})\bigr\}$ or $(z, z_2) \in \bigl\{(i,j), (\bar{\jmath}, \bar{\imath})\bigr\}$ are all similar so, to complete the proof, we just consider the case when $(z_1, z) = (i,j)$. By hypothesis (13), we have

$$2w_{\mathrm{s}}(i,j) + 2w_{\mathrm{s}}(j,z_2) = w(i,\bar{\imath}) + w(\bar{\jmath},j) + 2w(j,z_2);$$

hence, by the coherence assumption (4) of Definition 4,

$$2w_{\mathrm{s}}(i,j) + 2w_{\mathrm{s}}(j,z_2) = w(i,\bar{\imath}) + w(\bar{\jmath},j) + 2w(\bar{z}_2,\bar{\jmath});$$

hence, by Lemma 8 (instantiated using nodes $\bar{z}_2$ and $\bar{\jmath}$),

$$2w_{\mathrm{s}}(i,j) + 2w_{\mathrm{s}}(j,z_2) \leq w(i,\bar{\imath}) + w(\bar{z}_2,z_2);$$

hence, by hypothesis (13),

$$2w_{\mathrm{s}}(i,j) + 2w_{\mathrm{s}}(j,z_2) \leq w_{\mathrm{s}}(i,\bar{\imath}) + w_{\mathrm{s}}(\bar{z}_2,z_2).$$

Therefore (15) holds. □

Lemma 10 *Let $G = (\mathcal{N}^{\pm}, w)$ be weakly closed. Consider the graph $G_{\mathrm{S}} = (\mathcal{N}^{\pm}, w_{\mathrm{S}})$ where*

$$\forall z_1, z_2 \in \mathcal{N}^{\pm} : w_{\mathrm{S}}(z_1,z_2) := \min\Bigl\{w(z_1,z_2), \bigl(w(z_1,\bar{z}_1) + w(\bar{z}_2,z_2)\bigr)/2\Bigr\}. \qquad (16)$$

Then G_{S} is strongly closed.

Proof We first prove that G_{S} is weakly closed, by induction on the number of pairs $z_1, z_2 \in \mathcal{N}^{\pm}$ such that $2w(z_1,z_2) > w(z_1,\bar{z}_1) + w(\bar{z}_2,z_2)$. In the base case, when no such pair exists, by (16) we obtain $G_{\mathrm{S}} = G$, so that G_{S} is also weakly closed.

In the inductive case, there exist $i, j \in \mathcal{N}^{\pm}$ such that $2w(i,j) > w(i,\bar{\imath}) + w(\bar{\jmath},j)$; so that $i \neq \bar{\jmath}$. Since G and nodes i, j satisfy the hypotheses in Lemma 9, we can define $G' = (\mathcal{N}^{\pm}, w')$ equal to the octagonal graph $G_{\mathrm{s}} = (\mathcal{N}^{\pm}, w_{\mathrm{s}})$ defined in Lemma 9; it follows that G' is a weakly closed octagonal graph and

$$w'(i,j) = \min\Bigl\{w(i,j), \bigl(w(i,\bar{\imath}) + w(\bar{\jmath},j)\bigr)/2\Bigr\}. \qquad (17)$$

Let $G'_{\mathrm{S}} = (\mathcal{N}^{\pm}, w'_{\mathrm{S}})$ where

$$\forall z_1, z_2 \in \mathcal{N}^{\pm} : w'_{\mathrm{S}}(z_1,z_2) := \min\Bigl\{w'(z_1,z_2), \bigl(w'(z_1,\bar{z}_1) + w'(\bar{z}_2,z_2)\bigr)/2\Bigr\}.$$

Let $z_1, z_2 \in \mathcal{N}^{\pm}$ where $(z_1, z_2) \notin \bigl\{(i,j), (\bar{\jmath},\bar{\imath})\bigr\}$; then $w'(z_1,z_2) = w(z_1,z_2)$; also, as $i \neq \bar{\jmath}$, $w'(z_1,\bar{z}_1) = w(z_1,\bar{z}_1)$ and $w'(z_2,\bar{z}_2) = w(z_2,\bar{z}_2)$; thus we have $w'(z_1,z_2) \leq \bigl(w'(z_1,\bar{z}_1) + w'(\bar{z}_2,z_2)\bigr)/2$ if and only if $w(z_1,z_2) \leq \bigl(w(z_1,\bar{z}_1) + w(\bar{z}_2,z_2)\bigr)/2$. Therefore, the induction hypothesis can be applied to G'_{S}, so that G'_{S} must be weakly closed.

By property (17), $G_{\mathrm{S}} = G'_{\mathrm{S}}$ so that G_{S} is weakly closed. By hypothesis (16), $w_{\mathrm{S}}(z_1,\bar{z}_1) = w(z_1,\bar{z}_1)$ and $w_{\mathrm{S}}(\bar{z}_2,z_2) = w(\bar{z}_2,z_2)$, so that, again by hypothesis (16), we have

$$\forall z_1, z_2 \in \mathcal{N}^{\pm} : w_{\mathrm{S}}(z_1,z_2) = \min\Bigl\{w_{\mathrm{S}}(z_1,z_2), \bigl(w_{\mathrm{S}}(z_1,\bar{z}_1) + w_{\mathrm{S}}(\bar{z}_2,z_2)\bigr)/2\Bigr\};$$

hence, by Definition 16, property (2) of Definition 1 holds, so that G_{S} is closed; and, also, property (6) of Definition 5 holds, so that G_{S} is strongly closed. □

Proof (of Theorem 3 on page 17) Let $G^{\mathrm{s}} = (\mathcal{N}^{\pm}, w^{\mathrm{s}})$ be such that $G^{\mathrm{s}} = \text{S-closure}(G)$. By property (6) of Definition 5, for each $i, j \in \mathcal{N}^{\pm}$, $w^{\mathrm{s}}(i,j) \leq w_{\mathrm{S}}(i,j)$ so that $G^{\mathrm{s}} \trianglelefteq G_{\mathrm{S}}$. Moreover, by construction, $G_{\mathrm{S}} \trianglelefteq G$ so that we obtain $G^{\mathrm{s}} \trianglelefteq G_{\mathrm{S}} \trianglelefteq G$. Since G is closed, by Definition 16, it is also weakly closed. Hence, by Lemma 10, G_{S} is strongly closed. Hence, $G^{\mathrm{s}} = G_{\mathrm{S}}$. □

The proof of Theorem 4 relies on an auxiliary lemma.

Lemma 11 *Let $G = (\mathcal{N}^{\pm}, w)$ be a weakly closed octagonal graph. Let $G_{\mathrm{t}} = (\mathcal{N}^{\pm}, w_{\mathrm{t}})$ where, for each $h_1, h_2 \in \mathcal{N}^{\pm}$,*

$$w_{\mathrm{t}}(h_1, h_2) := \begin{cases} w(h_1, h_2) - 1, & \text{if } h_1 = \overline{h}_2 \text{ and } w(h_1, h_2) \text{ is odd;} \\ w(h_1, h_2), & \text{otherwise.} \end{cases} \tag{18}$$

Then, if G_{t} is consistent, G_{t} is a weakly closed octagonal graph.

Proof Suppose that the graph G_{t} is consistent. By construction, G_{t} is octagonal and $G_{\mathrm{t}} \trianglelefteq G$. As G is weakly closed, rule (1) of Definition 1 holds so that, for all $h \in \mathcal{N}^{\pm}$, $w(h, h) = 0$ and, by hypothesis (18), $w_{\mathrm{t}}(h, h) = w(h, h)$; hence, G_{t} satisfies rule (1) of Definition 1 too. Observe that, as G is weakly closed, rule (11) of Definition 16 holds.

Suppose $z_1, z_2, z \in \mathcal{N}^{\pm}$. Then, by Definition 16, to prove that G_{t} is weakly closed, we need to show that

$$w_{\mathrm{t}}(z_1, z) + w_{\mathrm{t}}(z, z_2) \geq \min\Big\{ w_{\mathrm{t}}(z_1, z_2), \big(w_{\mathrm{t}}(z_1, \overline{z}_1) + w_{\mathrm{t}}(\overline{z}_2, z_2)\big)/2 \Big\}. \tag{19}$$

If $z_1 = z_2$, then (19) is equivalent to $w_{\mathrm{t}}(z_1, z) + w_{\mathrm{t}}(z, z_1) \geq 0$, which holds due to the hypothesis that G_{t} is consistent. If $w_{\mathrm{t}}(z_1, z) = w(z_1, z)$ and $w_{\mathrm{t}}(z, z_2) = w(z, z_2)$ then, as $G_{\mathrm{t}} \trianglelefteq G$, it follows from rule (11) that (19) holds.

Suppose now that $z_1 \neq z_2$ and $w_{\mathrm{t}}(z_1, z) \neq w(z_1, z)$; then $w_{\mathrm{t}}(z_1, z_2) = w(z_1, z_2)$ and, by hypothesis (18), $z = \overline{z}_1$. By application of Lemma 8 to graph G and nodes $\overline{z}_2$ and z_1, we obtain $w(\overline{z}_2, z_2) \leq 2w(\overline{z}_2, z_1) + w(z_1, \overline{z}_1)$ so that, by coherence,. $w(\overline{z}_2, z_2) \leq 2w(\overline{z}_1, z_2) + w(z_1, \overline{z}_1)$. Note however that, if $w(\overline{z}_2, z_2) = 2w(\overline{z}_1, z_2) + w(z_1, \overline{z}_1)$, then $w(\overline{z}_2, z_2)$ will be odd if and only if $w(z_1, \overline{z}_1)$ is odd so that, by hypothesis (18), $w_{\mathrm{t}}(\overline{z}_2, z_2) = 2w_{\mathrm{t}}(\overline{z}_1, z_2) + w_{\mathrm{t}}(z_1, \overline{z}_1)$; on the other hand, if $w(\overline{z}_2, z_2) < 2w(\overline{z}_1, z_2) + w(z_1, \overline{z}_1)$, then $w(\overline{z}_2, z_2) \leq 2w(\overline{z}_1, z_2) + w(z_1, \overline{z}_1) - 1$ so that, by hypothesis (18), $w_{\mathrm{t}}(\overline{z}_2, z_2) \leq 2w_{\mathrm{t}}(\overline{z}_1, z_2) + w_{\mathrm{t}}(z_1, \overline{z}_1)$. Therefore, in both cases, $2w_{\mathrm{t}}(z_1, \overline{z}_1) + 2w_{\mathrm{t}}(\overline{z}_1, z_2) \geq w_{\mathrm{t}}(z_1, \overline{z}_1) + w_{\mathrm{t}}(\overline{z}_2, z_2)$ so that (19) holds. □

Proof (of Theorem 4 on page 19) Let $G^{\mathrm{T}} := \text{T-closure}(G)$. By definition of G_{T}, $G_{\mathrm{T}} \trianglelefteq G$ so that $\text{T-closure}(G_{\mathrm{T}}) \trianglelefteq G^{\mathrm{T}}$. As G_{T} is an octagonal graph, G_{T} is consistent, and hence $G^{\mathrm{T}} \neq \bot$; let $G^{\mathrm{T}} = (\mathcal{N}^{\pm}, w^{\mathrm{T}})$. Let $G_{\mathrm{t}} = (\mathcal{N}^{\pm}, w_{\mathrm{t}})$ be as defined in Lemma 11 so that G_{t} is weakly closed. Then, for all $i, j \in \mathcal{N}^{\pm}$,

$$w_{\mathrm{T}}(i, j) = \min\Big\{ w_{\mathrm{t}}(i, j), \frac{w_{\mathrm{t}}(i, \bar{\imath})}{2} + \frac{w_{\mathrm{t}}(\bar{\jmath}, j)}{2} \Big\}.$$

Therefore, by Lemma 10, G_{T} is strongly closed so that, by Definition 13, $G_{\mathrm{T}} \trianglelefteq G^{\mathrm{T}}$.

By Definitions 1, 5 and 13, for all $i, j \in \mathcal{N}^{\pm}$, both $w^{\mathrm{T}}(i, j) \leq w(i, j)$ and

$$w^{\mathrm{T}}(i, j) \leq \Big\lfloor \frac{w(i, \bar{\imath})}{2} \Big\rfloor + \Big\lfloor \frac{w(\bar{\jmath}, j)}{2} \Big\rfloor$$

so that, by definition of G_{T}, we have $G^{\mathrm{T}} \trianglelefteq G_{\mathrm{T}}$. □

Theorem 5 is a corollary of the following result proved in [47, Lemma 4].

Lemma 12 *Let $G = (\mathcal{N}^{\pm}, w)$ be an integer octagonal graph with no negative weight cycles and $G_{\mathrm{t}} = (\mathcal{N}^{\pm}, w_{\mathrm{t}})$, where w_{t} satisfies* (10)*, have a negative weight cycle. Then there exists $i, \bar{\imath} \in \mathcal{N}^{\pm}$ and a cycle $\pi = (i \cdot \pi_1 \cdot \bar{\imath}) :: (\bar{\imath} \cdot \pi_2 \cdot i)$ in G such that $w(\pi) = 0$ and the weight of the shortest path in G from i to $\bar{\imath}$ is odd.*

Proof (of Theorem 5 on page 19) The proof is by contradiction; suppose G_{T} is not an octagonal graph; then by Definitions 1, 5 and 13, G_{T} is inconsistent. We show that G_{t} is also inconsistent. Again, we assume to the contrary that G_{t} is consistent and derive a contradiction. Let $i, j \in \mathcal{N}^{\pm}$. By (10), we have $w_{\mathrm{t}}(i, j) \leq w(i, j)$ and $w_{\mathrm{t}}(i, \bar{\imath})/2 + w_{\mathrm{t}}(\bar{\jmath}, j)/2 = k_{ij}$, where $k_{ij} := \lfloor w(i, \bar{\imath})/2 \rfloor + \lfloor w(\bar{\jmath}, j)/2 \rfloor$. Letting $\text{S-closure}(G_{\mathrm{t}}) = (\mathcal{N}^{\pm}, w_{\mathrm{S}})$, we have, by Definition 5, $w_{\mathrm{S}}(i, j) \leq w_{\mathrm{t}}(i, j)$ and $w_{\mathrm{S}}(i, j) \leq w_{\mathrm{t}}(i, \bar{\imath})/2 + w_{\mathrm{t}}(\bar{\jmath}, j)/2$. Thus $w_{\mathrm{S}}(i, j) \leq \min\{w(i, j), k_{ij}\}$. As this holds for all $i, j \in \mathcal{N}^{\pm}$, by (9), $\text{S-closure}(G_{\mathrm{t}}) \trianglelefteq G_{\mathrm{T}}$, contradicting the assumption that G_{t} was consistent. Hence G_{t} is inconsistent and therefore contains a negative weight cycle.

By Lemma 12, there exists $i, \bar{\imath} \in \mathcal{N}^{\pm}$ and a cycle $\pi = (i \cdot \pi_1 \cdot \bar{\imath}) :: (\bar{\imath} \cdot \pi_2 \cdot i)$ in G such that $w(\pi) = 0$ and the weight of the shortest path in G from i to $\bar{\imath}$ is odd. As G is closed, $w(i, \bar{\imath}) \leq w(i \cdot \pi_1 \cdot \bar{\imath})$ and $w(\bar{\imath}, i) \leq w(\bar{\imath} \cdot \pi_2 \cdot i)$. Thus $w(i, \bar{\imath}) + w(\bar{\imath}, i) \leq w(\pi) = 0$. Moreover, $(i\bar{\imath})$ is a path and hence the shortest path from i to $\bar{\imath}$ so that $w(i\bar{\imath})$ is odd; hence, by (10), $w(i, \bar{\imath}) = w_{\mathrm{t}}(i, \bar{\imath}) + 1$ and $w(\bar{\imath}, i) \geq w_{\mathrm{t}}(\bar{\imath}, i)$. Therefore $w_{\mathrm{t}}(i, \bar{\imath}) + w_{\mathrm{t}}(\bar{\imath}, i) < 0$. □

A.3 Proofs of the Results Stated in Section 4.1

Proof (of Lemma 1 on page 12) As $\mathcal{E}$ is a zero-equivalence class, it is not empty and thus, by construction, $\overline{\mathcal{E}}$ is not empty.

Suppose $i \in \overline{\mathcal{E}}$ and $j \in \mathcal{N}^{\pm}$; then by construction, $\bar{\imath} \in \mathcal{E}$. As $\mathcal{E}$ is a zero-equivalence class, by Definition 7, $\bar{\jmath} \in \mathcal{E}$ if and only if $w(\bar{\imath}, \bar{\jmath}) = -w(\bar{\jmath}, \bar{\imath})$; by the coherence rule (4), $w(\bar{\jmath}, \bar{\imath}) = -w(\bar{\imath}, \bar{\jmath})$ if and only if $w(i, j) = -w(j, i)$; hence $w(i, j) = -w(j, i)$ if and only if $j \in \overline{\mathcal{E}}$. Hence, by Definition 7, $\overline{\mathcal{E}}$ is a zero-equivalence class. □

Lemma 13 *Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph and $\mathcal{E} \subseteq \mathcal{N}^{\pm}$ be a zero-equivalence class for G. Then, for each $i, j \in \mathcal{E}$ and $k \in \mathcal{N}^{\pm}$, we have $w(i, k) = w(i, j) + w(j, k)$ and $w(k, i) = w(k, j) + w(j, i)$.*

Proof As G is closed, by property (2) of Definition 1, we have $w(i, k) \leq w(i, j) + w(j, k)$ and $w(k, i) \leq w(k, j) + w(j, i)$. Hence, we only have to prove that $w(i, k) \geq w(i, j) + w(j, k)$ and $w(k, i) \geq w(k, j) + w(j, i)$. This is equivalent to proving that:

$$\begin{aligned} w(j, k) &\leq -w(i, j) + w(i, k), \\ w(k, j) &\leq -w(j, i) + w(k, i). \end{aligned}$$

Since $i, j \in \mathcal{E}$, we have $i \equiv_G j$ so that, by Definition 7, $w(i, j) = -w(j, i)$. Hence,

$$\begin{aligned} w(j, k) &\leq w(j, i) + w(i, k), \\ w(k, j) &\leq w(i, j) + w(k, i), \end{aligned}$$

which are true, again by property (2) of Definition 1. □

Lemma 14 *Let $G = (\mathcal{N}^{\pm}, w)$ be a strongly closed octagonal graph with a singular zero-equivalence class $\mathcal{E}$. Then, for each $i \in \mathcal{E}$ and $j \in \mathcal{N}^{\pm}$,*

$$2w(i, j) = w(i, \bar{\imath}) + w(\bar{\jmath}, j).$$

Proof Let $i \in \mathcal{E}$ and $j \in \mathcal{N}^{\pm}$. As G is a strongly closed octagonal graph, by property (6) of Definition 5, we have $2w(i, j) \leq w(i, \bar{\imath}) + w(\bar{\jmath}, j)$. Therefore, it remains to be proved that $2w(i, j) \geq w(i, \bar{\imath}) + w(\bar{\jmath}, j)$.

By hypothesis, $i \in \mathcal{E}$ so that, as $\mathcal{E} = \overline{\mathcal{E}}$, we also have $\bar{\imath} \in \mathcal{E}$. Therefore, by Lemma 13, $w(\bar{\imath}, j) = w(\bar{\imath}, i) + w(i, j)$. Since the graph G is octagonal, by the coherence assumption (4) of Definition 4, this can be rewritten as $w(\bar{\jmath}, i) = w(\bar{\imath}, i) + w(i, j)$. Thus, by applying property (2) of Definition 1, we obtain:

$$\begin{aligned} w(\bar{\jmath}, j) &\leq w(\bar{\jmath}, i) + w(i, j) \\ &= w(\bar{\imath}, i) + w(i, j) + w(i, j) \\ &= w(\bar{\imath}, i) + 2w(i, j). \end{aligned}$$

Therefore, $2w(i, j) \geq -w(\bar{\imath}, i) + w(\bar{\jmath}, j)$. As we observed that $i \equiv_G \bar{\imath}$ then, by Definition 7, $w(i, \bar{\imath}) = -w(\bar{\imath}, i)$, so that we obtain $2w(i, j) \geq w(i, \bar{\imath}) + w(\bar{\jmath}, j)$. □

Proof (of Lemma 2 on page 12) Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be singular zero-equivalence classes for the strongly closed octagonal graph $G = (\mathcal{N}^{\pm}, w)$. Let $i \in \mathcal{E}_1$ and $j \in \mathcal{E}_2$. Then, as $\mathcal{E}_1 = \overline{\mathcal{E}}_1$ and $\mathcal{E}_2 = \overline{\mathcal{E}}_2$, we have $i, \bar{\imath} \in \mathcal{E}_1$ and $j, \bar{\jmath} \in \mathcal{E}_2$. Hence, by Definition 7, $w(i, \bar{\imath}) = -w(\bar{\imath}, i)$ and $w(\bar{\jmath}, j) = -w(j, \bar{\jmath})$. Moreover, by Lemma 14,

$$\begin{aligned} 2w(j, i) &= w(j, \bar{\jmath}) + w(\bar{\imath}, i), \\ 2w(i, j) &= w(i, \bar{\imath}) + w(\bar{\jmath}, j). \end{aligned}$$

Thus we obtain

$$\begin{aligned} 2w(i, j) &= w(i, \bar{\imath}) + w(\bar{\jmath}, j) \\ &= -w(\bar{\imath}, i) - w(j, \bar{\jmath}) \\ &= -2w(j, i). \end{aligned}$$

Hence, $w(i, j) = -w(j, i)$ so that, by Definition 7, $i \equiv_G j$. Thus, as i, j were arbitrary nodes in $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively, we have the result $\mathcal{E}_1 = \mathcal{E}_2$. □

Proof (of Lemma 3 on page 13) Let $G = (\mathcal{N}^{\pm}, w)$, $L = (\mathcal{N}^{\pm}, w_{\mathrm{L}})$ and $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$; let also $\mathcal{L}$ be the set of leaders of the non-singular zero-equivalence classes for G. By Definition 8, L and E are subgraphs of G.

We first prove that the graphs L and E are octagonal; by Definition 4, this means that we have to show that they are consistent and satisfy (4). Observe that since, by hypothesis, the graph G is octagonal, we can assume that G is consistent and satisfies the coherence assumption (4).

To show that L (resp., E) is consistent, consider any cyclic path π in L (resp., in E). Then, as L and E are subgraphs of G, π is also a cyclic path in G and $w_{\mathrm{L}}(\pi) = w(\pi)$ (resp., $w_{\mathrm{E}}(\pi) = w(\pi)$). As G is consistent, we have $w(\pi) \geq 0$ so that $w_{\mathrm{L}}(\pi) \geq 0$ (resp., $w_{\mathrm{E}}(\pi) \geq 0$). Therefore L and E are both consistent.

To show that (4) for L holds, let $i, j \in \mathcal{N}^{\pm}$. If $i = j$, then $\bar{\imath} = \bar{\jmath}$ and, as G is a closed graph, $w(i,i) = w(\bar{\imath},\bar{\imath}) = 0$. By Definition 8, $w_{\mathrm{L}}(i,i) = w(i,i)$ and $w_{\mathrm{L}}(\bar{\imath},\bar{\imath}) = w(\bar{\imath},\bar{\imath})$ so that $w_{\mathrm{L}}(i,i) = w_{\mathrm{L}}(\bar{\imath},\bar{\imath}) = 0$. Suppose next that $i \neq j$. If $i \notin \mathcal{L}$ or $j \notin \mathcal{L}$ then, by Lemma 1 and the definition of the total ordering on $\mathcal{N}^{\pm}$, $\bar{\imath} \notin \mathcal{L}$ or $\bar{\jmath} \notin \mathcal{L}$; thus, by Definition 8, $w_{\mathrm{L}}(i,j) = w_{\mathrm{L}}(\bar{\jmath},\bar{\imath}) = +\infty$. Suppose now that $i, j \in \mathcal{L}$. Then, we also have $\bar{\imath}, \bar{\jmath} \in \mathcal{L}$ so that, again, by Definition 8, $w_{\mathrm{L}}(i,j) = w(i,j)$ and $w_{\mathrm{L}}(\bar{\jmath},\bar{\imath}) = w(\bar{\jmath},\bar{\imath})$. Since G satisfies (4), $w(i,j) = w(\bar{\jmath},\bar{\imath})$ so that $w_{\mathrm{L}}(i,j) = w_{\mathrm{L}}(\bar{\jmath},\bar{\imath})$. Hence, the graph L satisfies (4).

To show that (4) for E holds, let $i, j \in \mathcal{N}^{\pm}$. If $i = j$, then $\bar{\imath} = \bar{\jmath}$ and, as G is a closed graph, $w(i,i) = w(\bar{\imath},\bar{\imath}) = 0$. Since $i \equiv_G i$ and $\bar{\imath} \equiv_G \bar{\imath}$, by Definition 8, $w_{\mathrm{E}}(i,i) = w(i,i)$ and $w_{\mathrm{E}}(\bar{\imath},\bar{\imath}) = w(\bar{\imath},\bar{\imath})$ so that $w_{\mathrm{E}}(i,i) = w_{\mathrm{E}}(\bar{\imath},\bar{\imath}) = 0$. Suppose next that $i \neq j$. By Lemma 1, $i \equiv_G j$ if and only if $\bar{\imath} \equiv_G \bar{\jmath}$. Thus, by Definition 8, if $i \not\equiv_G j$ we have $w_{\mathrm{E}}(i,j) = w_{\mathrm{E}}(\bar{\jmath},\bar{\imath}) = +\infty$ and, if $i \equiv_G j$, we have $w_{\mathrm{E}}(i,j) = w(i,j)$ and $w_{\mathrm{E}}(\bar{\jmath},\bar{\imath}) = w(\bar{\jmath},\bar{\imath})$. In the latter case, as G satisfies (4), $w(i,j) = w(\bar{\jmath},\bar{\imath})$ so that $w_{\mathrm{E}}(i,j) = w_{\mathrm{E}}(\bar{\jmath},\bar{\imath})$. Hence, E is a graph that satisfies (4).

Secondly, using the hypothesis that G is strongly closed, we prove that L and E are strongly closed. Namely, we show that given that the weight function w satisfies properties (1) and (2) of Definition 1 and property (6) of Definition 5, we show that the weight functions w_{L} and w_{E} also satisfy properties (1), (2) and (6).

Consider property (1). Let $i \in \mathcal{N}^{\pm}$; as w satisfies (1), $w(i,i) = 0$, so that, by Definition 8, we also have $w_{\mathrm{L}}(i,i) = 0$ and $w_{\mathrm{E}}(i,i) = 0$.

Consider property (2). Let $i, j, k \in \mathcal{N}^{\pm}$; then we have to show that:

$$w_{\mathrm{L}}(i,j) \leq w_{\mathrm{L}}(i,k) + w_{\mathrm{L}}(k,j), \tag{20}$$

$$w_{\mathrm{E}}(i,j) \leq w_{\mathrm{E}}(i,k) + w_{\mathrm{E}}(k,j). \tag{21}$$

When $k \in \{i,j\}$, (20) and (21) trivially hold. Suppose now that $k \notin \{i,j\}$. Consider first the graph L. If $\{i,j,k\} \nsubseteq \mathcal{L}$, then, by Definition 8, $w_{\mathrm{L}}(i,k) = +\infty$ or $w_{\mathrm{L}}(k,j) = +\infty$, so that (20) holds. If, instead, $i, j, k \in \mathcal{L}$, then, by Definition 8, $w_{\mathrm{L}}(i,j) = w(i,j)$, $w_{\mathrm{L}}(i,k) = w(i,k)$ and $w_{\mathrm{L}}(k,j) = w(k,j)$. Hence, as w satisfies (2), (20) holds. Consider now the graph E. If $i \not\equiv_G k$ or $j \not\equiv_G k$ then, by Definition 8, $w_{\mathrm{E}}(i,k) = +\infty$ or $w_{\mathrm{E}}(k,j) = +\infty$, so that (21) holds. If, instead, $i \equiv_G k$ and $j \equiv_G k$ so that $i \equiv_G j$; then, by Definition 8, $w_{\mathrm{E}}(i,j) = w(i,j)$, $w_{\mathrm{E}}(i,k) = w(i,k)$ and $w_{\mathrm{E}}(k,j) = w(k,j)$. Hence, as w satisfies (2), (21) holds.

Consider property (6). Let $i, j \in \mathcal{N}^{\pm}$; then we have to show that:

$$2w_{\mathrm{L}}(i,j) \leq w_{\mathrm{L}}(i,\bar{\imath}) + w_{\mathrm{L}}(\bar{\jmath},j), \tag{22}$$

$$2w_{\mathrm{E}}(i,j) \leq w_{\mathrm{E}}(i,\bar{\imath}) + w_{\mathrm{E}}(\bar{\jmath},j). \tag{23}$$

When $i = j$, it follows from the consistency and property (1), proved in previous paragraphs for L and E that (22) and (23) hold. Suppose now that $i \neq j$. Consider first the graph L. If $\{i,j\} \nsubseteq \mathcal{L}$, then, by Definition 8, $w_{\mathrm{L}}(i,\bar{\imath}) = +\infty$ or $w_{\mathrm{L}}(\bar{\jmath},j) = +\infty$ so that (22) holds. If, instead, $i, j \in \mathcal{L}$; then by Lemma 1 and the definition of the total ordering on $\mathcal{N}^{\pm}$, $\bar{\imath}, \bar{\jmath} \in \mathcal{L}$ so that $w_{\mathrm{L}}(i,j) = w(i,j)$, $w_{\mathrm{L}}(i,\bar{\imath}) = w(i,\bar{\imath})$ and $w_{\mathrm{L}}(\bar{\jmath},j) = w(\bar{\jmath},j)$. Hence, as w satisfies (4), (22) holds. Consider now the graph E. If $i \not\equiv_G \bar{\imath}$ or $j \not\equiv_G \bar{\jmath}$, then, by Definition 8, $w_{\mathrm{E}}(i,\bar{\imath}) = +\infty$ or $w_{\mathrm{E}}(\bar{\jmath},j) = +\infty$, so that (23) holds. Suppose, instead that both $i \equiv_G \bar{\imath}$ and $j \equiv_G \bar{\jmath}$; then, by Definition 8, $w_{\mathrm{E}}(i,j) = w(i,j)$, $w_{\mathrm{E}}(i,\bar{\imath}) = w(i,\bar{\imath})$ and $w_{\mathrm{E}}(\bar{\jmath},j) = w(\bar{\jmath},j)$. Hence, as w satisfies (4), (23) holds.

Finally, we prove that L is a zero-cycle free graph. Let $\pi = j_0 j_1 \cdots j_p$ be a proper cycle in L; thus $j_0 = j_p$ and $j_0 \neq j_1$. As $w_{\mathrm{L}}(j_0,j_1) < +\infty$, by Definition 8, we have $j_0, j_1 \in \mathcal{L}$, so that $j_0 \not\equiv_G j_1$. Hence, by Definition 7, $w_{\mathrm{L}}(j_0,j_1) + w_{\mathrm{L}}(j_1,j_0) > 0$. We have already shown that L is closed; hence, by Lemma 5, $w_{\mathrm{L}}(j_1 \cdots j_p) \geq w_{\mathrm{L}}(j_1,j_p) = w_{\mathrm{L}}(j_1,j_0)$ so that $w_{\mathrm{L}}(\pi) \geq w_{\mathrm{L}}(j_0,j_1) + w_{\mathrm{L}}(j_1,j_0) > 0$. □

Lemma 15 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed and zero-cycle free octagonal graph. Suppose that (i, j) is an arc in G. Then there exists a simple and atomic path $\pi = i \cdots j$ in G such that $w(i, j) = w(\pi)$.*

Proof If $i = j$, let $\pi = i$ so that π is a simple and atomic path such that $w(\pi) = 0$. As G is closed, $w(i, j) = 0$ so that $w(i, j) = w(\pi)$ as required.

Suppose now that $i \neq j$. Let $\pi = j_0 \cdots j_p$ be a path of length p in G where $i = j_0$, $j = j_p$ and $w(i, j) = w(\pi)$ (for instance we can take $p = 1$). Suppose that $j_{a-1} = j_a$, for some $a \in \{1, \dots, p\}$. Then, as G is closed, $w(j_{a-1}, j_a) = 0$ and, since $i \neq j$, we can drop the non-proper arc (j_{a-1}, j_a) from path π and obtain another path π' from i to j such that $w(i, j) = w(\pi')$. Therefore, we can assume that π is a proper path. By Lemma 6, the path is also simple. Thus, as $\mathcal{N}$ is finite, we can assume that p is maximal such that $w(i, j) = w(\pi)$ and π is a simple path in G.

Suppose, for some $q = 1, \dots, p$, that the arc (j_{q-1}, j_q) in π is not atomic in G. Then, by property (2) of Definition 1 and Definition 9, there must exist $k \in \mathcal{N} \setminus \{j_{q-1}, j_q\}$ such that $w(j_{q-1}, j_q) = w(j_{q-1}, k) + w(k, j_q)$. Thus, letting $\pi_1 = j_0 \cdots j_{q-1}$ and $\pi_2 = j_q \cdots j_p$, we obtain

$$\begin{aligned}
w(j_0, j_p) &= w\bigl(\pi_1 :: (j_{q-1} j_q) :: \pi_2\bigr) \\
&= w(\pi_1) + w(j_{q-1}, j_q) + w(\pi_2) \\
&= w(\pi_1) + w(j_{q-1}, k) + w(k, j_q) + w(\pi_2) \\
&= w\bigl(\pi_1 :: (j_{q-1} k j_q) :: \pi_2\bigr).
\end{aligned}$$

By Lemma 6, the path $\pi' = \pi_1 :: (j_{q-1} k j_q) :: \pi_2$, which has length $p + 1$, is simple too, contradicting the maximality assumption for path π. Thus (j_{q-1}, j_q) is atomic in G. As this holds for any arc in π, the path π is atomic. □

Lemma 16 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed and zero-cycle free octagonal graph. Suppose that, for some $i \in \mathcal{N}^{\pm}$, there is a proper path $\pi = j_0 \cdots j_p$, where $i = j_0$, $\bar{\imath} = j_p$ and $w(i, \bar{\imath}) = w(\pi)$. Then there is at most one arc in the path π that is atomic but not strongly atomic in G.*

Proof The proof is by contraposition. Suppose that, for some $q, r \in \{1, \dots, p\}$ such that $q < r$, the arcs (j_{q-1}, j_q) and (j_{r-1}, j_r) are atomic but not strongly atomic in G. Then, by Definition 9,

$$\begin{aligned}
2w(j_{q-1}, j_q) &\geq w(j_{q-1}, \bar{\jmath}_{q-1}) + w(\bar{\jmath}_q, j_q), \\
2w(j_{r-1}, j_r) &\geq w(j_{r-1}, \bar{\jmath}_{r-1}) + w(\bar{\jmath}_r, j_r).
\end{aligned}$$

Let $\pi_1 = j_0 \cdots j_{q-1}$, $\pi_2 = j_q \cdots j_{r-1}$ and $\pi_3 = j_r \cdots j_p$, so that $\pi = \pi_1 :: (j_{q-1} j_q) :: \pi_2 :: (j_{r-1} j_r) :: \pi_3$. Since $w(i, \bar{\imath}) = w(\pi)$, we obtain

$$\begin{aligned}
2w(i, \bar{\imath}) &= 2w\bigl(\pi_1 :: (j_{q-1} j_q) :: \pi_2 :: (j_{r-1} j_r) :: \pi_3\bigr) \\
&= 2w(\pi_1) + 2w(j_{q-1}, j_q) + 2w(\pi_2) + 2w(j_{r-1}, j_r) + 2w(\pi_3).
\end{aligned}$$

As G is an octagonal graph, this can be rewritten

$$\begin{aligned}
2w(i, \bar{\imath}) &\geq w(\pi_1) + w(j_{q-1}, \bar{\jmath}_{q-1}) + w(\overline{\pi}_1) \\
&\qquad + w(\bar{\jmath}_q, j_q) + w(\pi_2) + w(j_{r-1}, \bar{\jmath}_{r-1}) + w(\overline{\pi}_2) \\
&\qquad\quad + w(\overline{\pi}_3) + w(\bar{\jmath}_r, j_r) + w(\pi_3) \\
&= w\bigl(\pi_1 :: (j_{q-1} \bar{\jmath}_{q-1}) :: \overline{\pi}_1\bigr) \\
&\qquad + w\bigl((\bar{\jmath}_q j_q) :: \pi_2 :: (j_{r-1} \bar{\jmath}_{r-1}) :: \overline{\pi}_2\bigr) \\
&\qquad\quad + w\bigl(\overline{\pi}_3 :: (\bar{\jmath}_r j_r) :: \pi_3\bigr).
\end{aligned}$$

Note that the path $\pi_2' = (\bar{\jmath}_q j_q) :: \pi_2 :: (j_{r-1} \bar{\jmath}_{r-1}) :: \overline{\pi}_2$ is a proper cycle from node $\bar{\jmath}_q$ to itself. Moreover, both paths $\pi_1' = \pi_1 :: (j_{q-1} \bar{\jmath}_{q-1}) :: \overline{\pi}_1$ and $\pi_3' = \overline{\pi}_3 :: (\bar{\jmath}_r j_r) :: \pi_3$ go from node i to node $\bar{\imath}$ so that, by Lemma 5, we have $w(i, \bar{\imath}) \leq w(\pi_1')$ and $w(i, \bar{\imath}) \leq w(\pi_3')$. As a consequence,

$$2w(i, \bar{\imath}) \geq w(i, \bar{\imath}) + w(\pi_2') + w(i, \bar{\imath}).$$

Therefore we obtain $w(\pi_2') \leq 0$, contradicting the hypothesis that the graph G is consistent and zero-cycle free. It follows that $q = r$. □

Lemma 17 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed and zero-cycle free octagonal graph. Suppose that $(i, \bar{\imath})$ is an arc in G. Then there exists a strongly atomic path $\pi = j_0 \cdots j_p$ in G, where $i = j_0$, $\bar{\imath} = j_p$ and $w(i, \bar{\imath}) = w(\pi)$.*

Proof By Lemma 15, there exists a simple and atomic path π in G from i to $\bar{\imath}$ and $w(i, \bar{\imath}) = w(\pi)$. Since the path is simple, we can take $p = \|\pi\|$ to be maximal in the set of paths having these properties. Suppose that π is not strongly atomic in G. Then, by Lemma 16, there must be exactly one arc (j_{q-1}, j_q) in π that is atomic but not not strongly atomic in G. Let $h = j_{q-1}$ and $k = j_q$. Thus, $\pi = \pi_h :: (hk) :: \pi_k$, where the subpaths $\pi_h = j_0 \cdots j_{q-1}$ and $\pi_k = j_q \cdots j_p$ are strongly atomic in G. As the graph G is octagonal, the paths $\overline{\pi}_h$ and $\overline{\pi}_k$ are also strongly atomic and satisfy $w(\pi_h) = w(\overline{\pi}_h)$ and $w(\pi_k) = w(\overline{\pi}_k)$.

Since the arc (h, k) is atomic but not strongly atomic, by Definition 9,

$$2w(h, k) \geq w(h, \overline{h}) + w(\overline{k}, k).$$

Therefore, as G is octagonal,

$$\begin{aligned}
2w(i, \bar{\imath}) &= 2w(\pi_h) + 2w(h, k) + 2w(\pi_k) \\
&\geq 2w(\pi_h) + w(h, \overline{h}) + w(\overline{k}, k) + 2w(\pi_k) \\
&= w(\pi_h) + w(h, \overline{h}) + w(\overline{\pi}_h) \\
&\qquad + w(\overline{\pi}_k) + w(\overline{k}, k) + w(\pi_k) \\
&= w\bigl(\pi_h :: (h\overline{h}) :: \overline{\pi}_h\bigr) + w\bigl(\overline{\pi}_k :: (\overline{k}k) :: \pi_k\bigr).
\end{aligned}$$

Note that both paths $\pi' = \pi_h :: (h\overline{h}) :: \overline{\pi}_h$ and $\pi'' = \overline{\pi}_k :: (\overline{k}k) :: \pi_k$ go from node i to node $\bar{\imath}$. Thus, by Lemma 5, we have $w(i, \bar{\imath}) \leq w(\pi')$ and $w(i, \bar{\imath}) \leq w(\pi'')$. As a consequence, we obtain

$$w(i, \bar{\imath}) = w(\pi') = w(\pi'').$$

As $h \neq \overline{h}$ and $k \neq \overline{k}$ and as π_h, $\overline{\pi}_h$ and π_k, $\overline{\pi}_k$ are proper paths, both π' and π'' are proper paths too; and hence, by Lemma 6, they are both simple paths. Moreover, by the maximality assumption for p, we have $\|\pi'\| = \|\pi''\| = p$. Consider now just one of these paths: π'.

To conclude the proof, we will show that $(h, \overline{h})$ is a strongly atomic arc in G, so that π' is a strongly atomic path in G. Suppose instead that $(h, \overline{h})$ is not strongly atomic in G. Then, by Definition 9, it is not atomic, so that there exists $\ell \in \mathcal{N}^{\pm} \setminus \{h, \overline{h}\}$ such that $w(h, \overline{h}) = w(h, \ell) + w(\ell, \overline{h})$. Consider the path $\pi''' = \pi_h :: (h\ell\overline{h}) :: \overline{\pi}_h$. Then π''' is a proper path in G such that $w(i, \bar{\imath}) = w(\pi''')$ and $\|\pi'''\| > p$; contradicting the assumption that p was maximal. It follows that the arc $(h, \overline{h})$ is strongly atomic in G. □

Lemma 18 *Let $G = (\mathcal{N}^{\pm}, w)$ be a closed and zero-cycle free octagonal graph. Suppose that (i, j) is an arc in G. Then one of the following properties holds:*

1. *there exists a strongly atomic path $\pi = i \cdots j$ in G where $w(i, j) = w(\pi)$;*
2. *$i \neq \bar{\jmath}$, $2w(i, j) \geq w(i, \bar{\imath}) + w(\bar{\jmath}, j)$ and there exist strongly atomic paths $\pi_i = i \cdots \bar{\imath}$, $\pi_j = \bar{\jmath} \cdots j$ in G where $w(i, \bar{\imath}) = w(\pi_i)$ and $w(\bar{\jmath}, j) = w(\pi_j)$.*

Proof By Lemma 15, there exists a simple and atomic path $\pi = i \cdots j$ in G where $w(i, j) = w(\pi)$. If $i = j$, then $\|\pi\| = 0$, so that the path π is strongly atomic and condition 1 holds. Suppose now that $i \neq j$.

If the path is strongly atomic in G, then condition 1 holds. Therefore to complete the proof we assume that π is not strongly atomic in G and show that condition 2 holds.

By Definition 9, there exists at least one arc (h, k) in π that is atomic but not strongly atomic in G. As π is a simple path, $h \neq k$ so that, again by Definition 9,

$$2w(h, k) = w(h, \overline{h}) + w(\overline{k}, k).$$

As the graph is octagonal, we have $w(i, h) = w(\overline{h}, \bar{\imath})$ and $w(k, j) = w(\bar{\jmath}, \overline{k})$. Moreover, by Lemma 5,

$$\begin{aligned}
w(i, \bar{\imath}) &\leq w(i, h) + w(h, \overline{h}) + w(\overline{h}, \bar{\imath}), \\
w(\bar{\jmath}, j) &\leq w(\bar{\jmath}, \overline{k}) + w(\overline{k}, k) + w(k, j).
\end{aligned}$$

Thus, as $w(i,j) = w(\pi)$, we have

$$\begin{aligned}
2w(i,j) &= 2w(i,h) + 2w(h,k) + 2w(k,j) \\
&= 2w(i,h) + w(h,\overline{h}) + w(\overline{k},k) + 2w(k,j) \\
&= w(i,h) + w(h,\overline{h}) + w(\overline{h},\overline{\imath}) \\
&\qquad + w(\overline{\jmath},\overline{k}) + w(\overline{k},k) + w(k,j) \\
&\geq w(i,\overline{\imath}) + w(\overline{\jmath},j).
\end{aligned}$$

By Lemma 17, there exist strongly atomic paths $\pi_i = i \cdots \overline{\imath}$ and $\pi_j = \overline{\jmath} \cdots j$ in G where $w(i,\overline{\imath}) = w(\pi_i)$ and $w(\overline{\jmath},j) = w(\pi_j)$; therefore, condition 2 holds. □

Lemma 19 *Let G be a strongly closed and zero-cycle free octagonal graph and let A be its strongly atomic subgraph. Then A is an octagonal subgraph of G and* S-closure$(A) = G$.

Proof It follows from Definition 9 that A is a subgraph of G and, for all $i,j \in \mathcal{N}^{\pm}$, (i,j) is an arc in A if and only if $(\overline{\jmath},\overline{\imath})$ is an arc in A. Therefore, as G is octagonal, A is also octagonal.

Since $G \trianglelefteq A$ and the strong closure operator is both monotonic and idempotent, we obtain that $G \trianglelefteq$ S-closure(A). Therefore to prove S-closure$(A) = G$, we just need to show that S-closure$(A) \trianglelefteq G$. Letting $G = (\mathcal{N}^{\pm}, w)$, S-closure$(A) = (\mathcal{N}^{\pm}, w_{\mathrm{S}})$ and (i,j) be any arc of G, we will show that $w_{\mathrm{S}}(i,j) \leq w(i,j)$. Since G is closed and zero-cycle free, either one of two cases of Lemma 18 holds.

If case 1 of Lemma 18 holds, then there exists a strongly atomic path $\pi = i \cdots j$ in G where $w(i,j) = w(\pi)$. By Definition 9, π is also a path in A having the same weight $w(\pi)$. Since strong closure is a reductive operator, $w_{\mathrm{S}}(\pi) \leq w(\pi)$. Moreover, by Lemma 5, $w_{\mathrm{S}}(i,j) \leq w_{\mathrm{S}}(\pi)$ and hence $w_{\mathrm{S}}(i,j) \leq w(i,j)$.

If case 2 of Lemma 18 holds, then $2w(i,j) \geq w(i,\overline{\imath}) + w(\overline{\jmath},j)$ and there exist strongly atomic paths $\pi_i = i \cdots \overline{\imath}$, $\pi_j = \overline{\jmath} \cdots j$ in G where $w(i,\overline{\imath}) = w(\pi_i)$ and $w(\overline{\jmath},j) = w(\pi_j)$. By property (6) of Definition 5, we also have $2w(i,j) \leq w(i,\overline{\imath}) + w(\overline{\jmath},j)$, so that we obtain

$$2w(i,j) = w(i,\overline{\imath}) + w(\overline{\jmath},j).$$

By Definition 9, π_i and π_j are also paths in A having the same weights $w(\pi_i)$ and $w(\pi_j)$. Since strong closure is a reductive operator, $w_{\mathrm{S}}(\pi_i) \leq w(\pi_i)$ and $w_{\mathrm{S}}(\pi_j) \leq w(\pi_j)$. By Lemma 5, $w_{\mathrm{S}}(i,\overline{\imath}) \leq w_{\mathrm{S}}(\pi_i)$ and $w_{\mathrm{S}}(\overline{\jmath},j) \leq w_{\mathrm{S}}(\pi_j)$; hence we have $w_{\mathrm{S}}(i,\overline{\imath}) \leq w(i,\overline{\imath})$ and $w_{\mathrm{S}}(\overline{\jmath},j) \leq w(\overline{\jmath},j)$. Thus, $w_{\mathrm{S}}(i,\overline{\imath}) + w_{\mathrm{S}}(\overline{\jmath},j) \leq 2w(i,j)$ and, by property (6) of Definition 5, $w_{\mathrm{S}}(i,j) \leq w(i,j)$.

We can therefore conclude that S-closure$(A) \trianglelefteq G$. □

Lemma 20 *Let G be a strongly closed octagonal graph and Z the zero-equivalence reduction for G. Then Z is an octagonal subgraph of G.*

Proof Let $G = (\mathcal{N}^{\pm}, w)$ and $Z = (\mathcal{N}^{\pm}, w_{\mathrm{Z}})$. Let the weight function w' be as defined in Definition 10; then $(\mathcal{N}^{\pm}, w')$ is a subgraph of G. To show that Z is a subgraph of G, consider any $i,j \in \mathcal{N}^{\pm}$ such that $w_{\mathrm{Z}}(i,j) < +\infty$. By Definition 10, $w_{\mathrm{Z}}(i,j) = \min\bigl\{w'(i,j), w'(\overline{\jmath},\overline{\imath})\bigr\}$ and hence, $w_{\mathrm{Z}}(i,j) = w(i,j)$ or $w_{\mathrm{Z}}(i,j) = w(\overline{\jmath},\overline{\imath})$; since G is an octagonal graph, $w_{\mathrm{Z}}(i,j) = w(i,j)$. Therefore, Z is a subgraph of G, which implies $G \trianglelefteq Z$ and, as G is consistent, Z is a consistent graph too. Moreover, for all $i,j \in \mathcal{N}^{\pm}$, we have

$$w_{\mathrm{Z}}(i,j) = w_{\mathrm{Z}}(\overline{\jmath},\overline{\imath}) = \min\bigl\{w'(i,j), w'(\overline{\jmath},\overline{\imath})\bigr\}.$$

Thus, w_{Z} also satisfies the coherence assumption (4) of Definition 4. Therefore, Z is octagonal and hence, an octagonal subgraph of G. □

Lemma 21 *Let $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$ be the zero-equivalence subgraph of a strongly closed octagonal graph G. Let also $i,j \in \mathcal{N}^{\pm}$ and $\pi = i \cdots j$ be a path in E. Then $w_{\mathrm{E}}(\pi) = w_{\mathrm{E}}(i,j)$.*

Proof The proof is by induction on $p = \|\pi\|$. If $p = 0$, then $i = j$, $\pi = i$ and $w_{\mathrm{E}}(\pi) = 0$; thus, as (i,i) is an arc in E and G is closed, the result $w_{\mathrm{E}}(i,i) = 0$ holds by property (1) of Definition 1. For the inductive case, when $p > 0$, let $\pi = i \cdot \pi'$ be a path in E, where the subpath

$\pi' = k \cdots j$ is such that $\|\pi'\| = p - 1$. Thus, by the inductive hypothesis, $w_{\mathrm{E}}(\pi') = w_{\mathrm{E}}(k, j)$ and we obtain

$$\begin{aligned} w_{\mathrm{E}}(\pi) &= w_{\mathrm{E}}(i, k) + w_{\mathrm{E}}(\pi') \\ &= w_{\mathrm{E}}(i, k) + w_{\mathrm{E}}(k, j) \\ &= w_{\mathrm{E}}(i, j), \end{aligned}$$

where the last step holds by Lemma 13, as i, k are nodes in E. □

Lemma 22 *Let $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$ be the zero-equivalence subgraph of a strongly closed octagonal graph and $Z = (\mathcal{N}^{\pm}, w_{\mathrm{Z}})$ the zero-equivalence reduction for E. Let also $i, j \in \mathcal{N}^{\pm}$ be such that $i \equiv_E j$. Then there exists a path $\pi = i \cdots j$ in Z such that $w_{\mathrm{Z}}(\pi) = w_{\mathrm{E}}(i, j)$.*

Proof Let w' and w_{Z} be as specified in Definition 10; thus, $(\mathcal{N}^{\pm}, w')$ is a subgraph of the octagonal graph Z. We first show that, for any zero-equivalence class $\mathcal{E}$ for E, if $\mathcal{E}$ contains more than one node, then there is a cycle $\pi_{\mathcal{E}}$ in Z that contains all the nodes in $\mathcal{E}$.

Suppose first that the zero-equivalence class $\mathcal{E} = \{z_0, \ldots, z_m\}$, where $m > 0$, is non-singular. If $\mathcal{E}$ is positive and $z_0 < \cdots < z_m$, then, by Definition 10, there is a cycle $\pi_{\mathcal{E}} = z_0 \cdots z_m z_0$ in $(\mathcal{N}^{\pm}, w')$. Hence, $\pi_{\mathcal{E}}$ is also a cycle in Z. If $\mathcal{E}$ is a negative zero-equivalence class, then $\overline{\mathcal{E}} = \{\overline{z}_0, \ldots, \overline{z}_m\}$ is positive and $\overline{z}_0 < \cdots < \overline{z}_m$ so that, by the previous argument, there is a cycle $\pi_{\overline{\mathcal{E}}} = \overline{z}_0 \cdots \overline{z}_m \overline{z}_0$ in Z connecting the nodes of $\overline{\mathcal{E}}$. By Lemma 20, Z is octagonal, so that there is also a cycle $\pi_{\mathcal{E}} = \overline{\pi_{\overline{\mathcal{E}}}} = z_0 z_m \cdots z_0$ in Z that connects the nodes of $\mathcal{E}$.

Suppose next that the zero-equivalence class $\mathcal{E} = \{z_0, \overline{z}_0, \ldots, z_m, \overline{z}_m\}$ is singular and that $z_0 < \overline{z}_0 < \cdots z_m < \overline{z}_m$. Then, by Definition 10, there is a path $\pi^+ = z_0 \cdots z_m$ in $(\mathcal{N}^{\pm}, w')$ and, hence, in Z; Since $\mathcal{E} = \overline{\mathcal{E}}$ and Z is octagonal, there is a path $\pi^- = \overline{z}_m \cdots \overline{z}_0$ in Z. Also, by Definition 10, $(\mathcal{N}^{\pm}, w')$ contains the arcs $(\overline{z}_0, z_0)$ and $(z_m, \overline{z}_m)$. Therefore, $\pi_{\mathcal{E}} = \pi^+ :: (z_m \overline{z}_m) :: \pi^- :: (\overline{z}_0 z_0)$ is a cycle in Z that contains all the nodes in $\mathcal{E}$.

As $i \equiv_E j$, by Definition 7, for some zero-equivalence class $\mathcal{E}$ in E, $i, j \in \mathcal{E}$. Note that, since $i \neq j$, if $\mathcal{E} = \{z_0, \ldots, z_m\}$ is non-singular, then it must be $m > 0$. Thus i and j are nodes in the cycle $\pi_{\mathcal{E}}$ as defined above; thus, there exists a path $\pi = i \cdots j$ in Z. As Z is a subgraph of E, $w_{\mathrm{Z}}(\pi) = w_{\mathrm{E}}(\pi)$. By Lemma 21, $w_{\mathrm{E}}(\pi) = w_{\mathrm{E}}(i, j)$ and thus, $w_{\mathrm{Z}}(\pi) = w_{\mathrm{E}}(i, j)$. □

Lemma 23 *Let $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$ be the zero-equivalence subgraph of a strongly closed octagonal graph. Let also $Z = (\mathcal{N}^{\pm}, w_{\mathrm{Z}})$ be the zero-equivalence reduction for E. Then* S-closure$(Z) = E$.

Proof Let S-closure$(Z) = Z^{\mathrm{S}} = (\mathcal{N}^{\pm}, w_{\mathrm{Z}}^{\mathrm{S}})$. By Lemma 3, E is strongly closed octagonal graph. By Lemma 20, Z is an octagonal subgraph of E. Hence, $E \trianglelefteq Z$ and $E \trianglelefteq Z^{\mathrm{S}}$. Consider any $i, j \in \mathcal{N}^{\pm}$. Then $w_{\mathrm{Z}}^{\mathrm{S}}(i, j) \geq w_{\mathrm{E}}(i, j)$; it remains to show that $w_{\mathrm{E}}(i, j) \geq w_{\mathrm{Z}}^{\mathrm{S}}(i, j)$.

If $w_{\mathrm{E}}(i, j) = +\infty$. then, as $w_{\mathrm{Z}}^{\mathrm{S}}(i, j) \geq w_{\mathrm{E}}(i, j)$, $w_{\mathrm{Z}}^{\mathrm{S}}(i, j) = +\infty$. Suppose now that $w_{\mathrm{E}}(i, j) < +\infty$ so that, by Definition 8, $i \equiv_E j$. By Lemma 22, there exists a path $\pi = i \cdots j$ in Z such that $w_{\mathrm{E}}(i, j) = w_{\mathrm{Z}}(\pi)$; as $Z^{\mathrm{S}} \trianglelefteq Z$, we also have $w_{\mathrm{Z}}(\pi) \geq w_{\mathrm{Z}}^{\mathrm{S}}(\pi)$; and, by Lemma 5, $w_{\mathrm{Z}}^{\mathrm{S}}(\pi) \geq w_{\mathrm{Z}}^{\mathrm{S}}(i, j)$; therefore $w_{\mathrm{E}}(i, j) \geq w_{\mathrm{Z}}^{\mathrm{S}}(i, j)$. □

Proof (of Theorem 1 on page 14) Let $G = (\mathcal{N}^{\pm}, w)$ be the strong closure of the input octagonal graph, computed at step 1 of the strong reduction procedure; let $L = (\mathcal{N}^{\pm}, w_{\mathrm{L}})$ be the non-singular leaders' subgraph of G; $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$ the zero-equivalence subgraph of G; $A = (\mathcal{N}^{\pm}, w_{\mathrm{A}})$ the strongly atomic subgraph of L; $Z = (\mathcal{N}^{\pm}, w_{\mathrm{Z}})$ the zero-equivalence reduction of E; $G_{\mathrm{R}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}) = A \sqcap Z$; and $G_{\mathrm{R}}^{\mathrm{S}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}^{\mathrm{S}}) =$ S-closure(G_{R}). Then we need to show that G_{R} is a strongly reduced octagonal graph and $G_{\mathrm{R}}^{\mathrm{S}} = G$.

By Definition 8 and Lemma 3, both L and E are octagonal subgraphs of G. By Lemma 19, A is an octagonal subgraph of L and, by Lemma 20, Z is an octagonal subgraph of E; hence A and Z are octagonal subgraphs of G. Therefore, $G_{\mathrm{R}} = A \sqcap Z$ is an octagonal subgraph of G.

Next we show that $G_{\mathrm{R}}^{\mathrm{S}} = G$. By Lemma 3, L is strongly closed and zero-cycle free; by Lemma 19, S-closure$(A) = L$. By Lemma 23, S-closure$(Z) = E$. Thus, we have

$$\begin{aligned} G_{\mathrm{R}}^{\mathrm{S}} &= \text{S-closure}\bigl(\text{S-closure}(A \sqcap Z)\bigr) \\ &= \text{S-closure}\bigl(\text{S-closure}(A) \sqcap \text{S-closure}(Z)\bigr) \\ &= \text{S-closure}(L \sqcap E). \end{aligned}$$

As observed before, L and E are subgraphs of G and hence $G \trianglelefteq G_{\mathrm{R}}^{\mathrm{S}}$. It remains to show that $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq G$. Let (i,j) be an arc in G and suppose that $i \in \mathcal{E}_i$ and $j \in \mathcal{E}_j$, where $\mathcal{E}_i$ and $\mathcal{E}_j$ are zero-equivalence classes for G. Then, to prove that $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq G$, we just need to show that

$$w_{\mathrm{R}}^{\mathrm{S}}(i,j) \leq w(i,j). \tag{24}$$

To do this, without loss of generality, we need to consider three cases:

1. $\mathcal{E}_i$ and $\mathcal{E}_j$ are singular zero-equivalence classes in G;
2. $\mathcal{E}_i$ is singular and $\mathcal{E}_j$ is a non-singular zero-equivalence class in G;
3. $\mathcal{E}_i$ and $\mathcal{E}_j$ are non-singular zero-equivalence classes in G.

Case 1. By Lemma 2, $\mathcal{E}_i = \mathcal{E}_j$ so that $w(i,j) = w_{\mathrm{E}}(i,j)$. Therefore, as $w_{\mathrm{R}}^{\mathrm{S}}(i,j) \leq w_{\mathrm{E}}(i,j)$, Equation (24) holds.

Case 2. Let $\ell_j \in \mathcal{N}^{\pm}$ be the leader of $\mathcal{E}_j$ so that $\overline{\ell}_j$ is the leader of the non-singular zero-equivalence class $\overline{\mathcal{E}}_j$. By Definition 8,

$$\begin{aligned} w(i,\bar{\imath}) &= w_{\mathrm{E}}(i,\bar{\imath}), & w(\bar{\jmath},\overline{\ell}_j) &= w_{\mathrm{E}}(\bar{\jmath},\overline{\ell}_j), \\ w(\ell_j,j) &= w_{\mathrm{E}}(\ell_j,j), & w(\overline{\ell}_j,\ell_j) &= w_{\mathrm{L}}(\overline{\ell}_j,\ell_j). \end{aligned} \tag{25}$$

By Lemma 13,

$$\begin{aligned} w(\bar{\jmath},j) &= w(\bar{\jmath},\overline{\ell}_j) + w(\overline{\ell}_j,j) \\ &= w(\bar{\jmath},\overline{\ell}_j) + w(\overline{\ell}_j,\ell_j) + w(\ell_j,j) \end{aligned}$$

so that, by Lemma 14 and Equation (25),

$$\begin{aligned} 2w(i,j) &= w(i,\bar{\imath}) + w(\bar{\jmath},j) \\ &= w(i,\bar{\imath}) + w(\bar{\jmath},\overline{\ell}_j) + w(\overline{\ell}_j,\ell_j) + w(\ell_j,j) \\ &= w_{\mathrm{E}}(i,\bar{\imath}) + w_{\mathrm{E}}(\bar{\jmath},\overline{\ell}_j) + w_{\mathrm{L}}(\overline{\ell}_j,\ell_j) + w_{\mathrm{E}}(\ell_j,j). \end{aligned}$$

Moreover, by Lemma 5,

$$\begin{aligned} 2w_{\mathrm{R}}^{\mathrm{S}}(i,j) &\leq w_{\mathrm{R}}^{\mathrm{S}}(i,\bar{\imath}) + w_{\mathrm{R}}^{\mathrm{S}}(\bar{\jmath},\overline{\ell}_j) + w_{\mathrm{R}}^{\mathrm{S}}(\overline{\ell}_j,\ell_j) + w_{\mathrm{R}}^{\mathrm{S}}(\ell_j,j) \\ &\leq w_{\mathrm{E}}(i,\bar{\imath}) + w_{\mathrm{E}}(\bar{\jmath},\overline{\ell}_j) + w_{\mathrm{L}}(\overline{\ell}_j,\ell_j) + w_{\mathrm{E}}(\ell_j,j) \end{aligned}$$

since $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq L$ and $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq E$. Therefore Equation (24) holds.

Case 3. By Definition 8,

$$w(i,\ell_i) = w_{\mathrm{E}}(i,\ell_i), \quad w(\ell_j,j) = w_{\mathrm{E}}(\ell_j,j), \quad w(\ell_i,\ell_j) = w_{\mathrm{L}}(\ell_i,\ell_j).$$

Therefore, by Lemma 13,

$$\begin{aligned} w(i,j) &= w(i,\ell_i) + w(\ell_i,j) \\ &= w(i,\ell_i) + w(\ell_i,\ell_j) + w(\ell_j,j) \\ &= w_{\mathrm{E}}(i,\ell_i) + w_{\mathrm{L}}(\ell_i,\ell_j) + w_{\mathrm{E}}(\ell_j,j). \end{aligned}$$

Moreover, by Lemma 5,

$$\begin{aligned} w_{\mathrm{R}}^{\mathrm{S}}(i,j) &\leq w_{\mathrm{R}}^{\mathrm{S}}(i,\ell_i) + w_{\mathrm{R}}^{\mathrm{S}}(\ell_i,\ell_j) + w_{\mathrm{R}}^{\mathrm{S}}(\ell_j,j) \\ &\leq w_{\mathrm{E}}(i,\ell_i) + w_{\mathrm{L}}(\ell_i,\ell_j) + w_{\mathrm{E}}(\ell_j,j), \end{aligned}$$

since $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq L$ and $G_{\mathrm{R}}^{\mathrm{S}} \trianglelefteq E$. Therefore Equation (24) holds and hence $G_{\mathrm{R}}^{\mathrm{S}} = G$.

Finally, to show that G_{R} is strongly reduced, as specified in Definition 6, we assume there exists a strongly reduced octagonal graph $G_2 = (\mathcal{N}^{\pm}, w_2)$ such that $G_{\mathrm{R}} \lhd G_2$ and $G = \text{S-closure}(G_2)$ and derive a contradiction. Let $G_2^{\mathrm{c}} = (\mathcal{N}^{\pm}, w_2^{\mathrm{c}}) = \mathrm{closure}(G_2)$, $G_{\mathrm{R}}^{\mathrm{c}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}^{\mathrm{c}}) = \mathrm{closure}(G_{\mathrm{R}})$ and $G_2^{\mathrm{S}} = (\mathcal{N}^{\pm}, w_2^{\mathrm{S}}) = \text{S-closure}(G_2)$ (so that $w = w_{\mathrm{R}}^{\mathrm{S}} = w_2^{\mathrm{S}}$).

As $G_{\mathrm{R}} \lhd G_2$, there exists $i, j \in \mathcal{N}^{\pm}$ such that $w_{\mathrm{R}}(i,j) < w_2(i,j)$; as G_2^{c} and G are closed, $w_2^{\mathrm{c}}(i,i) = w(i,i) = 0$ so that we can assume that $i \neq j$. Moreover $w_{\mathrm{R}}(i,j) < +\infty$ so that, as G_{R} is a subgraph of G, $w_{\mathrm{R}}(i,j) = w(i,j)$; as closure is both idempotent and reductive, $G = \mathrm{closure}(G) \unlhd G_{\mathrm{R}}^{\mathrm{c}} \unlhd G_{\mathrm{R}}$; hence we have

$$w(i,j) = w_{\mathrm{R}}^{\mathrm{c}}(i,j) = w_{\mathrm{R}}(i,j).$$

Since $G_{\mathrm{R}} \lhd G_2$ we have $G_{\mathrm{R}}^{\mathrm{c}} \unlhd G_2^{\mathrm{c}}$ so that $w(i,j) \leq w_2^{\mathrm{c}}(i,j)$.

Suppose first that $w(i,j) < w_2^{\mathrm{c}}(i,j)$. By Theorem 3, since $\text{S-closure}(G_2^{\mathrm{c}}) = G$, we have $2w(i,j) = w_2^{\mathrm{c}}(i,\bar{\imath}) + w_2^{\mathrm{c}}(\bar{\jmath},j)$. Hence, $2w(i,j) = w(i,\bar{\imath}) + w(\bar{\jmath},j)$, so that $i \neq \bar{\jmath}$, $(i,\bar{\imath})$ and $(\bar{\jmath},j)$ are arcs in G and, by Definition 9, arc (i,j) is not strongly atomic in G. Hence, since (i,j) is an arc in G_{R}, it must be $i \equiv_G j$, so that i, j are in the same equivalence class $\mathcal{E}$ and (i,j) is an arc in Z; thus, as $i \neq j$, either i or j is not the leader of $\mathcal{E}$; without loss of generality, we assume that i is not the leader. Since $w = w_2^{\mathrm{S}} = w_{\mathrm{R}}^{\mathrm{S}}$, we have $w(i,\bar{\imath}) = w_2^{\mathrm{S}}(i,\bar{\imath}) = w_{\mathrm{R}}^{\mathrm{S}}(i,\bar{\imath})$. By Theorem 3, $w_2^{\mathrm{S}}(i,\bar{\imath}) = w_2^{\mathrm{c}}(i,\bar{\imath})$ and $w_{\mathrm{R}}^{\mathrm{S}}(i,\bar{\imath}) = w_{\mathrm{R}}^{\mathrm{c}}(i,\bar{\imath})$; hence $w(i,\bar{\imath}) = w_2^{\mathrm{c}}(i,\bar{\imath}) = w_{\mathrm{R}}^{\mathrm{c}}(i,\bar{\imath})$. By Lemma 7, there exists $\pi_i = i \cdots \bar{\imath}$ in G_2 such that

$$w_2(\pi_i) = w_2^{\mathrm{c}}(i,\bar{\imath}) = w(i,\bar{\imath}) \leq w_{\mathrm{R}}(\pi_i),$$

where the last inequality follows from Lemma 5 and by closure being a reductive operator. Note that as $G_{\mathrm{R}} \lhd G_2$, π_i must also be a path in G_{R}. By Definition 10 and the assumption that i is not a leader of $\mathcal{E}$, if, for some $h \in \mathcal{N}^{\pm}$, (i,h) is an arc in G_{R}, then $h = j$; hence $\pi_i = (ij) \cdots \bar{\imath}$; also, as $G_{\mathrm{R}} \lhd G_2$, for all $h_1, h_2 \in \mathcal{N}^{\pm}$, $w_2(h_1,h_2) \geq w_{\mathrm{R}}(h_1,h_2)$; hence $w_2(\pi_i) > w_{\mathrm{R}}(\pi_i)$ which is a contradiction.

Suppose now that $w(i,j) = w_2^{\mathrm{c}}(i,j)$ so that $w_2^{\mathrm{c}}(i,j) < w_2(i,j)$. By Lemma 7, there exists $\pi = i \cdots j$ in G_2 such that

$$w_2(\pi) = w_2^{\mathrm{c}}(i,j) = w(i,j) < w_2(i,j).$$

As $w_2(\pi) < w_2(i,j)$, there exists $k \in \mathcal{N}^{\pm}$ such that $\pi = (ik) :: \pi'$ for some path $\pi' = k \cdots j$ in G_2. Since (i,j) is an arc in G_{R}, by Definitions 9 and 10, $w(i,j) < w(i,k) + w(k,j)$. As G is closed, by Lemma 5, $w(k,j) \leq w(\pi')$ and hence $w(i,j) < w(i,k) + w(\pi') = w(\pi)$; and therefore $w_2(\pi) < w(\pi)$, contradicting $G \lhd G_2$. □

A.4 Proofs of the Results Stated in Section 4.2

The standard widening operator '∇_{s}' for topologically closed convex polyhedra defined in the PhD thesis of N. Halbwachs [19, Définition 5.3.3, p. 57] is slightly different from the specification originally proposed in [2], in that the former does not depend on the particular constraint systems chosen for representing the arguments of the widening. Nonetheless, the following result, which is taken from [33], states that the two definitions happen to be equivalent when applied to polyhedra S_1 and S_2 such that $S_1 \subseteq S_2$ and $\dim(S_1) = \dim(S_2)$.

Proposition 1 *Let $S_1, S_2 \subseteq \mathbb{R}^n$ be two topologically closed convex polyhedra such that $\varnothing \neq S_1 \subseteq S_2$ and $\dim(S_1) = \dim(S_2)$. Let also $\mathcal{C}_1$ be a finite system of non-strict linear inequalities describing S_1 and suppose that $\mathcal{C}_1$ contains no redundant constraint. Then the result of the standard widening $S_1 \nabla_{\mathrm{s}} S_2$ is described by the constraint system*

$$\mathcal{C}_{\mathrm{s}} := \{\, \beta \in \mathcal{C}_1 \mid \text{all the points in } S_2 \text{ satisfy } \beta \,\}.$$

Proof See the proof in [33, Proposition 6]. □

Proof (of Theorem 2 on page 15) Let $S_1, S_2 \in \wp(\mathbb{R}^n)$, where $\varnothing \neq S_1 \subseteq S_2$, be two octagonal shapes represented by the strongly reduced octagonal graph G_1 and the strongly closed octagonal graph G_2, respectively. Let also $G = G_1 \nabla G_2 = (\mathcal{N}^{\pm}, w)$ and S the octagonal shape represented by G. Let $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}$ be the systems of octagonal constraints encoded by G_1, G_2 and G, respectively. Note that, in such a construction, each pair of coherent arcs generates a single octagonal constraint. Since the octagonal graph G_1 is strongly reduced, the corresponding constraint system $\mathcal{C}_1$ contains no redundant constraints.

We first assume $\dim(S_1) = \dim(S_2)$ so that, by Definition 11, $S_1 \nabla S_2 = S$. We will show that $S = S_1 \nabla_{\mathrm{s}} S_2$. Let $\mathcal{C}_{\mathrm{s}}$ be as defined in Proposition 1. Then it follows from Proposition 1 that to prove $S = S_1 \nabla_{\mathrm{s}} S_2$, we just need to show that $\mathcal{C} = \mathcal{C}_{\mathrm{s}}$. To prove $\mathcal{C} \subseteq \mathcal{C}_{\mathrm{s}}$, suppose that $\beta = (v_i - v_j \leq d_1) \in \mathcal{C}$, so that $w(i,j) = d_1 < +\infty$. Then, by Definition 2, $w_1(i,j) = d_1$ and $w_2(i,j) = d_2 \leq d_1$. Thus, there exists $\gamma = (v_i - v_j \leq d_2) \in \mathcal{C}_2$. Since all the points of S_2 satisfy γ, they also satisfy β and hence, $\beta \in \mathcal{C}_{\mathrm{s}}$. To prove the other inclusion $\mathcal{C}_{\mathrm{s}} \subseteq \mathcal{C}$, suppose that $\beta = (v_i - v_j \leq d_1) \in \mathcal{C}_{\mathrm{s}}$ so that, since $\mathcal{C}_{\mathrm{s}} \subseteq \mathcal{C}_1$ and $\mathcal{C}_1$ contains no redundancies, we have $w_1(i,j) = d_1$. By definition of $\mathcal{C}_{\mathrm{s}}$, all the points of S_2 satisfy β. Since the octagonal graph G_2 is strongly closed, there exists $\gamma = (v_i - v_j \leq d_2) \in \mathcal{C}_2$ such that $d_1 \geq d_2$. Hence, $w_1(i,j) \geq w_2(i,j)$ and, by Definition 2, we obtain $\beta \in \mathcal{C}$.

Suppose now that $\dim(S_1) \neq \dim(S_2)$. By hypothesis, $S_1 \subseteq S_2$ so that $\dim(S_1) < \dim(S_2)$. Then, by Definition 11, $S_1 \nabla S_2 = S_2$. Since the standard widening '∇_{s}' is an upper bound operator, we obtain $S_1 \nabla S_2 \subseteq S_1 \nabla_{\mathrm{s}} S_2$.

Thus, in both cases the operator '∇' computes an upper bound of its arguments which is at least as precise as the upper bound computed by the standard widening '∇_{s}'. To complete the proof, we only have to show that '∇' is a proper widening operator, i.e., it enforces the convergence of any abstract iteration sequence. This property is easily shown to hold by the observation, made above, that the operator of Definition 11 can behave differently from the standard widening only when there is a strict increase in the affine dimension of the arguments. Since such an increase can only happen a finite number of times, the operator '∇' inherits the convergence guarantee of '∇_{s}'. □

A.5 Proof of Theorem 6 on page 21

The following two lemmas are simple adaptations of Lemma 3 and Lemma 19 to the case of tightly closed graphs.

Lemma 24 *Let L and E be the non-singular leaders' subgraph and the zero-equivalence subgraph of the tightly closed graph G, respectively. Then, L and E are tightly closed graphs and L is zero-cycle free.*

Proof Since G is tightly closed, it is also strongly closed. Hence, by Lemma 3, the graphs L and E are strongly closed octagonal graphs and E is zero-cycle free. Since they are subgraphs of G, which is tightly closed, L and E are integer graphs and satisfy property (8) of Definition 13, so that they are tightly closed too. □

Lemma 25 *Let G be a tightly closed and zero-cycle free graph and $A = (\mathcal{N}^{\pm}, w_{\mathrm{A}})$ be its tightly atomic subgraph; let also $U = (\mathcal{N}^{\pm}, w_{\mathrm{U}})$ be the untightening of A. Then U is an integer octagonal subgraph of G and* T-closure$(U) = G$.

Proof Since G is tightly closed, it is also strongly closed so that, by Lemma 19, A is an octagonal subgraph of G and S-closure$(A) = G$; as G is tightly closed, we obtain T-closure$(A) = G$.

By Definition 15, U is an octagonal subgraph of A; hence, U is an integer octagonal subgraph of G. Since $G \trianglelefteq U$, by monotonicity and idempotency, $G \trianglelefteq$ T-closure(U). Therefore, to prove T-closure$(U) = G$, we just need to show that T-closure$(U) \trianglelefteq G$.

Let $A = (\mathcal{N}^{\pm}, w_{\mathrm{A}})$, $U = (\mathcal{N}^{\pm}, w_{\mathrm{U}})$ and $U_{\mathrm{T}} = (\mathcal{N}^{\pm}, w_{\mathrm{T}}) =$ T-closure(U); we prove that $U_{\mathrm{T}} \trianglelefteq A$. Let $i, j \in \mathcal{N}^{\pm}$ be any pair of nodes. If $w_{\mathrm{U}}(i,j) \leq w_{\mathrm{A}}(i,j)$ then, since tight closure is a reductive operator, $w_{\mathrm{T}}(i,j) \leq w_{\mathrm{A}}(i,j)$. On the other hand, by Definition 15, if $w_{\mathrm{U}}(i,j) > w_{\mathrm{A}}(i,j)$, then we have that $j = \bar{\imath}$, $w_{\mathrm{A}}(i,\bar{\imath})$ is even and there exists $k \in \mathcal{N}^{\pm} \setminus \{i, \bar{\imath}\}$ such that $w_{\mathrm{A}}(i,k) + w_{\mathrm{A}}(k,\bar{\imath}) \leq w_{\mathrm{A}}(i,\bar{\imath}) + 1$. Note that, by Definition 15 and reductivity of tight closure, we have $w_{\mathrm{T}}(i,k) \leq w_{\mathrm{U}}(i,k) = w_{\mathrm{A}}(i,k)$ and $w_{\mathrm{T}}(k,\bar{\imath}) \leq w_{\mathrm{U}}(k,\bar{\imath}) = w_{\mathrm{A}}(k,\bar{\imath})$. Since U_{T} is tightly closed, by property (2) of Definition 1, we obtain

$$\begin{aligned} w_{\mathrm{T}}(i,\bar{\imath}) &\leq w_{\mathrm{T}}(i,k) + w_{\mathrm{T}}(k,\bar{\imath}) \\ &\leq w_{\mathrm{A}}(i,k) + w_{\mathrm{A}}(k,\bar{\imath}) \\ &\leq w_{\mathrm{A}}(i,\bar{\imath}) + 1. \end{aligned}$$

Note that $w_{\mathrm{A}}(i,\bar{\imath}) + 1$ is an odd integer and, by property (8) of Definition 13, $w_{\mathrm{T}}(i,\bar{\imath})$ is even, so that we obtain $w_{\mathrm{T}}(i,\bar{\imath}) \leq w_{\mathrm{A}}(i,\bar{\imath})$.

Hence, $U_{\mathrm{T}} \trianglelefteq A$ and, since the tight closure operator is monotonic and idempotent, we obtain $U_{\mathrm{T}} =$ T-closure$(U_{\mathrm{T}}) \trianglelefteq$ T-closure$(A) = G$, completing the proof. □

The following proof of Theorem 6 is a simple adaptation of the proof of Theorem 1, exploiting the two lemmas above.

Proof (of Theorem 6 on page 21) Let $G = (\mathcal{N}^{\pm}, w)$ be the tight closure of the input integer octagonal graph, computed at step 1 of the tight reduction procedure; let $L = (\mathcal{N}^{\pm}, w_{\mathrm{L}})$ be the non-singular leaders' subgraph of G; $E = (\mathcal{N}^{\pm}, w_{\mathrm{E}})$ the zero-equivalence subgraph of G; $A = (\mathcal{N}^{\pm}, w_{\mathrm{A}})$ the strongly atomic subgraph of L; $U = (\mathcal{N}^{\pm}, w_{\mathrm{U}})$ the untightening of A; $Z = (\mathcal{N}^{\pm}, w_{\mathrm{Z}})$ the zero-equivalence reduction of E; $G_{\mathrm{R}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}) = U \sqcap Z$; and $G_{\mathrm{R}}^{\mathrm{T}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}^{\mathrm{T}}) = \text{T-closure}(G_{\mathrm{R}})$. Then we need to show that G_{R} is a tightly reduced octagonal graph and $G_{\mathrm{R}}^{\mathrm{T}} = G$.

By Definition 8 and Lemma 24, both L and E are octagonal subgraphs of G. By Lemma 25, U is an octagonal subgraph of L and, by Lemma 20, Z is an octagonal subgraph of E; hence U and Z are octagonal subgraphs of G. Hence, $G_{\mathrm{R}} = U \sqcap Z$ is an octagonal subgraph of G.

Next we show $G_{\mathrm{R}}^{\mathrm{T}} = G$. By Lemma 24, L and E are tightly closed and L is zero-cycle free; by Lemma 25, $\text{T-closure}(U) = L$; by Lemma 23, $\text{S-closure}(Z) = E$, hence $\text{T-closure}(Z) = E$. Thus, we have

$$\begin{aligned}
G_{\mathrm{R}}^{\mathrm{T}} &= \text{T-closure}\bigl(\text{T-closure}(U \sqcap Z)\bigr) \\
&= \text{T-closure}\bigl(\text{T-closure}(U) \sqcap \text{T-closure}(Z)\bigr) \\
&= \text{T-closure}(L \sqcap E).
\end{aligned}$$

As observed before, L and E are subgraphs of G and hence $G \trianglelefteq G_{\mathrm{R}}^{\mathrm{T}}$. It remains to show that $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq G$. Let (i, j) be an arc in G and suppose that $i \in \mathcal{E}_i$ and $j \in \mathcal{E}_j$, where $\mathcal{E}_i$ and $\mathcal{E}_j$ are zero-equivalence classes for G. Then, to prove that $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq G$, we just need to show that

$$w_{\mathrm{R}}^{\mathrm{T}}(i, j) \leq w(i, j). \tag{26}$$

To do this, without loss of generality, we need to consider three cases:

1. $\mathcal{E}_i$ and $\mathcal{E}_j$ are singular zero-equivalence classes in G;
2. $\mathcal{E}_i$ is singular and $\mathcal{E}_j$ is a non-singular zero-equivalence class in G;
3. $\mathcal{E}_i$ and $\mathcal{E}_j$ are non-singular zero-equivalence classes in G.

Case 1. By Lemma 2, $\mathcal{E}_i = \mathcal{E}_j$ so that $w(i, j) = w_{\mathrm{E}}(i, j)$. Therefore, as $w_{\mathrm{R}}^{\mathrm{T}}(i, j) \leq w_{\mathrm{E}}(i, j)$, Equation (26) holds.

Case 2. Let $\ell_j \in \mathcal{N}^{\pm}$ be the leader of $\mathcal{E}_j$ so that $\overline{\ell}_j$ is the leader of the non-singular zero-equivalence class $\overline{\mathcal{E}}_j$. By Definition 8,

$$\begin{aligned}
w(i, \overline{\imath}) &= w_{\mathrm{E}}(i, \overline{\imath}), & w(\overline{\jmath}, \overline{\ell}_j) &= w_{\mathrm{E}}(\overline{\jmath}, \overline{\ell}_j), \\
w(\ell_j, j) &= w_{\mathrm{E}}(\ell_j, j), & w(\overline{\ell}_j, \ell_j) &= w_{\mathrm{L}}(\overline{\ell}_j, \ell_j).
\end{aligned} \tag{27}$$

By Lemma 13,

$$\begin{aligned}
w(\overline{\jmath}, j) &= w(\overline{\jmath}, \overline{\ell}_j) + w(\overline{\ell}_j, j) \\
&= w(\overline{\jmath}, \overline{\ell}_j) + w(\overline{\ell}_j, \ell_j) + w(\ell_j, j)
\end{aligned}$$

so that, by Lemma 14 and Equation (27),

$$\begin{aligned}
2w(i, j) &= w(i, \overline{\imath}) + w(\overline{\jmath}, j) \\
&= w(i, \overline{\imath}) + w(\overline{\jmath}, \overline{\ell}_j) + w(\overline{\ell}_j, \ell_j) + w(\ell_j, j) \\
&= w_{\mathrm{E}}(i, \overline{\imath}) + w_{\mathrm{E}}(\overline{\jmath}, \overline{\ell}_j) + w_{\mathrm{L}}(\overline{\ell}_j, \ell_j) + w_{\mathrm{E}}(\ell_j, j).
\end{aligned}$$

Moreover, by Lemma 5,

$$\begin{aligned}
2w_{\mathrm{R}}^{\mathrm{T}}(i, j) &\leq w_{\mathrm{R}}^{\mathrm{T}}(i, \overline{\imath}) + w_{\mathrm{R}}^{\mathrm{T}}(\overline{\jmath}, \overline{\ell}_j) + w_{\mathrm{R}}^{\mathrm{T}}(\overline{\ell}_j, \ell_j) + w_{\mathrm{R}}^{\mathrm{T}}(\ell_j, j) \\
&\leq w_{\mathrm{E}}(i, \overline{\imath}) + w_{\mathrm{E}}(\overline{\jmath}, \overline{\ell}_j) + w_{\mathrm{L}}(\overline{\ell}_j, \ell_j) + w_{\mathrm{E}}(\ell_j, j)
\end{aligned}$$

since $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq L$ and $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq E$. Therefore Equation (26) holds.

Case 3. By Definition 8,

$$w(i, \ell_i) = w_{\mathrm{E}}(i, \ell_i), \quad w(\ell_j, j) = w_{\mathrm{E}}(\ell_j, j), \quad w(\ell_i, \ell_j) = w_{\mathrm{L}}(\ell_i, \ell_j).$$

Therefore, by Lemma 13,

$$\begin{aligned} w(i, j) &= w(i, \ell_i) + w(\ell_i, j) \\ &= w(i, \ell_i) + w(\ell_i, \ell_j) + w(\ell_j, j) \\ &= w_{\mathrm{E}}(i, \ell_i) + w_{\mathrm{L}}(\ell_i, \ell_j) + w_{\mathrm{E}}(\ell_j, j). \end{aligned}$$

Moreover, by Lemma 5,

$$\begin{aligned} w_{\mathrm{R}}^{\mathrm{T}}(i, j) &\leq w_{\mathrm{R}}^{\mathrm{T}}(i, \ell_i) + w_{\mathrm{R}}^{\mathrm{T}}(\ell_i, \ell_j) + w_{\mathrm{R}}^{\mathrm{T}}(\ell_j, j) \\ &\leq w_{\mathrm{E}}(i, \ell_i) + w_{\mathrm{L}}(\ell_i, \ell_j) + w_{\mathrm{E}}(\ell_j, j), \end{aligned}$$

since $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq L$ and $G_{\mathrm{R}}^{\mathrm{T}} \trianglelefteq E$. Therefore Equation (26) holds and hence $G_{\mathrm{R}}^{\mathrm{T}} = G$.

Finally, to show that G_{R} is tightly reduced as specified in Definition 14, having already shown that G_{R} is a subgraph of T-closure$(G_{\mathrm{R}}) = G$, we assume there exists a proper octagonal subgraph $G_2 = (\mathcal{N}^{\pm}, w_2)$ of G_{R} such that $G = \text{T-closure}(G_2)$ and derive a contradiction.

Let $G_2^{\mathrm{c}} = (\mathcal{N}^{\pm}, w_2^{\mathrm{c}}) = \mathrm{closure}(G_2)$, $G_{\mathrm{R}}^{\mathrm{c}} = (\mathcal{N}^{\pm}, w_{\mathrm{R}}^{\mathrm{c}}) = \mathrm{closure}(G_{\mathrm{R}})$ and $G_2^{\mathrm{T}} = (\mathcal{N}^{\pm}, w_2^{\mathrm{T}}) =$ T-closure(G_2) (so that $w = w_{\mathrm{R}}^{\mathrm{T}} = w_2^{\mathrm{T}}$). As G_2 is a proper octagonal subgraph of G_{R}, there exists $i, j \in \mathcal{N}^{\pm}$ such that $w_{\mathrm{R}}(i, j) < w_2(i, j) = +\infty$; as G_2^{c} and G are closed, $w_2^{\mathrm{c}}(i, i) = w(i, i) = 0$ so that we can assume that $i \neq j$. Moreover $w_{\mathrm{R}}(i, j) < +\infty$ so that, as G_{R} is a subgraph of G, $w_{\mathrm{R}}(i, j) = w(i, j)$; as closure is both idempotent and reductive, $G = \mathrm{closure}(G) \trianglelefteq G_{\mathrm{R}}^{\mathrm{c}} \trianglelefteq G_{\mathrm{R}}$; hence we have

$$w(i, j) = w_{\mathrm{R}}^{\mathrm{c}}(i, j) = w_{\mathrm{R}}(i, j).$$

Since $G_{\mathrm{R}} \triangleleft G_2$ we have $G_{\mathrm{R}}^{\mathrm{c}} \trianglelefteq G_2^{\mathrm{c}}$ so that $w(i, j) \leq w_2^{\mathrm{c}}(i, j)$.

Suppose first that $w(i, j) < w_2^{\mathrm{c}}(i, j)$. By Theorem 4, since T-closure$(G_2^{\mathrm{c}}) = G$, we have

$$w(i, j) = \left\lfloor \frac{w_2^{\mathrm{c}}(i, \bar\imath)}{2} \right\rfloor + \left\lfloor \frac{w_2^{\mathrm{c}}(\bar\jmath, j)}{2} \right\rfloor.$$

Hence, $w(i, j) = \left\lfloor \frac{w(i, \bar\imath)}{2} \right\rfloor + \left\lfloor \frac{w(\bar\jmath, j)}{2} \right\rfloor$. By property (8) of Definition 13, both $w(i, \bar\imath)$ and $w(\bar\jmath, j)$ are even integers, so that $2w(i, j) = w(i, \bar\imath) + w(\bar\jmath, j)$. Therefore, $i \neq \bar\jmath$ and the two arcs $(i, \bar\imath)$ and $(\bar\jmath, j)$ are in G; thus, by Definition 9, arc (i, j) is not strongly atomic in G, so that (i, j) is not an arc in L; hence, it is neither an arc in U. Since (i, j) is a proper arc in $G_{\mathrm{R}} = U \sqcap Z$, it must be an arc in Z, so that i and j are in the same equivalence class $\mathcal{E}$ (i.e., $i \equiv_G j$). As $i \neq j$, either i or j is not the leader of $\mathcal{E}$; without loss of generality, we assume that i is not the leader. By Definition 10, in G_{R} the only arc exiting from node i is (i, j); since $w_2(i, j) = +\infty$ and G_2 is a subgraph of G_{R}, there is no arc in G_2 exiting from i, so that i and j cannot belong to the same 0-weight loop; hence $i \not\equiv_{G_2} j$ and $G \neq G_2^{\mathrm{T}}$, which is a contradiction.

Suppose now that $w(i, j) = w_2^{\mathrm{c}}(i, j)$ so that $w_2^{\mathrm{c}}(i, j) < w_2(i, j)$. By Lemma 7, there exists $\pi = i \cdots j$ in G_2 such that

$$w_2(\pi) = w_2^{\mathrm{c}}(i, j) = w(i, j) < w_2(i, j).$$

As $w_2(\pi) < w_2(i, j)$, there exists $k \in \mathcal{N}^{\pm}$ such that $\pi = (ik) :: \pi'$ for some path $\pi' = k \cdots j$ in G_2. Since (i, j) is an arc in G_{R}, by Definitions 9 and 10, $w(i, j) < w(i, k) + w(k, j)$. As G is closed, by Lemma 5, $w(k, j) \leq w(\pi')$ and hence $w(i, j) < w(i, k) + w(\pi') = w(\pi)$; and therefore $w_2(\pi) < w(\pi)$, contradicting $G \triangleleft G_2$. □