Solving Boolean Equation Systems using Small Progress Measures

Jeroen Keiren
j.j.a.keiren@student.tue.nl

12th January 2009

Abstract

We investigate a parity game interpretation of boolean equation systems. Given this interpretation, we present a version of the algorithm for solving parity games based on small progress measures [3]. The algorithm is applied directly on the boolean equation system (BES), eliminating the intermediate transformation step of BES to parity game. The resulting algorithm runs in $O(ad(\mathcal{E})\Sigma_{X\in\mathcal{X}}|rhs(X)| \cdot (\frac{|\mathcal{X}|}{\lfloor ad(\mathcal{E})/2 \rfloor})^{\lfloor ad(\mathcal{E})/2 \rfloor})$ time, which is an improvement on the currently known bounds for solving boolean equation systems.

1 Introduction

An approach for verifying correctness of concurrent systems is model checking. A desired property is expressed as a logical formula and it is checked whether a model satisfies this formula. Various logics (CTL, LTL) are much used in this area. In this paper we will consider modal μ-calculus, which subsumes the other logics.

Generally, model checking procedures for modal μ-calculus can be split into two categories. Local and global procedures. Local procedures are used to show that a certain state in the system satisfies a requirement, whereas global procedures compute for all states whether they satisfy a requirement. In this paper we will restrict ourselves to the global approach. For the global model checking problem several approaches have been presented in the literature. For the full fragment of modal μ-calculus efficient algorithms are not known, and the problem is known to be in $NP \cap co-NP$ as well as in $UP \cap co-UP$. It is still expected that a polynomial algorithm can be found. Known methods for solving the model checking problem for modal μ-calculus include BDD based methods using iteration for fixpoint computation [1], translation to the problem of finding a winning strategy in a parity game [6] as well as translating the problem to finding solutions for a boolean equation system [4]. We combine the latter two approaches, and present a way to apply the algorithms known for parity games to boolean equation systems.

In Section 2 we introduce boolean equation systems and parity games. Section 3 recapitulates the algorithm for finding winning strategies of parity games by Jurdziński [3]. We present an interpretation of a BES as a parity game in Section 4. This interpretation is used to transform the algorithm such that it can be applied directly to boolean equation systems in Section 5.

2 Preliminaries

2.1 Boolean equation systems

Boolean equation systems (BESses) [4] are a class of equation system that can be employed to perform model checking of model μ-calculus formulae. It has been shown [4] that solving a BES is equivalent to model-checking. Boolean equation systems are also used for this purpose in mCRL2 [2], a language for specifying concurrent systems and protocols in an algebraic style.

Definition 2.1 [Boolean Equation System] We assume a set $\mathcal{X}$ of boolean variables, with typical elements X, X_1, X_2, ... and a type $\mathbb{B}$ with elements true, false representing the booleans. Furthermore we have fixed-point symbols μ for least fixed-point and ν for greatest fixed-point.

A boolean equation system is a system of fixpoint equations, inductively defined as follows:

- ϵ is the empty BES
- if $\mathcal{E}$ is a BES, then $(\sigma X = \phi)\mathcal{E}$ is also a BES, with $\sigma \in \{\mu, \nu\}$ a fixpoint symbol and ϕ a negation free formula over $\mathcal{X}$.

Negation free formulae ϕ are defined as follows:

$$\phi ::= \mathsf{false} \mid \mathsf{true} \mid X \mid \phi \wedge \phi \mid \phi \vee \phi$$

where $X \in \mathcal{X}$ is a proposition variable of type $\mathbb{B}$.

Finding a solution of a BES amounts to finding an assignment of true or false to each variable X_i such that all equations are satisfied. Furthermore if $\sigma_i = \mu$, then the assignment to X_i is as strong as possible, and if $\sigma_i = \nu$ it is as weak as possible.

We now define the solution of a BES formally. An environment $\eta{:}\mathcal{X} \to \{\mathsf{true}, \mathsf{false}\}$ is a function that assigns a boolean value to each variable $X \in \mathcal{X}$.

Definition 2.2 [Solution of a BES] The solution $[\![\_]\!]{:}BES \to (\mathcal{X} \to \{\mathsf{true}, \mathsf{false}\}) \to \mathcal{X} \to \{\mathsf{true}, \mathsf{false}\}$ of a BES $\mathcal{E}$ in an environment η is defined inductively as follows:

$$\begin{aligned} [\![\epsilon]\!]\eta &= \eta \\ [\![(\mu X = f)\mathcal{E}]\!]\eta &= [\![\mathcal{E}]\!]\eta[X := f([\![\mathcal{E}]\!]\eta[X := \mathsf{false}])] \\ [\![(\nu X = f)\mathcal{E}]\!]\eta &= [\![\mathcal{E}]\!]\eta[X := f([\![\mathcal{E}]\!]\eta[X := \mathsf{true}])] \end{aligned}$$

We also write $[\![\mathcal{E}]\!]\eta(X)$ to denote the solution of X for BES $\mathcal{E}$ in environment η.

In the rest of this paper we assume that the constants true and false do not occur in the right hand sides of the equations. We may do this as we can replace each occurrence of true by a reference to X_{true} and each occurrence of false by a reference to X_{false}. Where X_{true} and X_{false} are defined as follows:

$$\begin{aligned} \nu X_{\mathsf{true}} &= X_{\mathsf{true}} \\ \mu X_{\mathsf{false}} &= X_{\mathsf{false}} \end{aligned}$$

Assuming that $\mathsf{false} < \mathsf{true}$ we define an ordering on boolean equation systems as follows:

Definition 2.3 [Ordering $\leqslant$ on BES (Definition 3.15 in [4]] Given boolean equation systems $\mathcal{E} \equiv (\sigma_1 X_1 = f_1) \ldots (\sigma_n X_n = f_n)$ and $\mathcal{E}' \equiv (\sigma_1 X_1 = g_1) \ldots (\sigma_n X_n = g_n)$, then $\mathcal{E} \leqslant \mathcal{E}'$ iff $f_i \leqslant g_i$.

Definition 2.4 [Conjunctive/Disjunctive form] A BES $\mathcal{E}$ is in conjunctive/disjunctive form if every ϕ_i is of the form X_j, $\bigwedge_{k=0}^{n} X_{j_k}$ or $\bigvee_{k=0}^{n} X_{j_k}$, where $n \geqslant 1$.

That is, a BES is in conjunctive/disjunctive form if every right hand side is either a single variable, or it is a conjunction or a disjunction over propositional variables. Conjunctions and disjunctions may not appear mixed in a single right hand side. Note that every BES can be transformed into conjunctive/disjunctive form in polynomial time in a way that preserves the solution of variables occurring in both BESses.

Definition 2.5 [Conjunctive form] A BES $\mathcal{E}$ is in conjunctive form if every ϕ_i is of the form $\bigwedge_{k=0}^{n} X_{j_k}$, with $n \geqslant 0$.

That is, a BES in conjunctive form only contains conjuncts or single variables (or true or false) as right hand sides. It has been shown [4] that given a BES $\mathcal{E}$ and an environment η there is a BES $\mathcal{E}'$ in conjunctive form such that $\mathcal{E}' \leqslant \mathcal{E}$ and $[\![\mathcal{E}']\!]\eta = [\![\mathcal{E}]\!]\eta$.

In the rest of this paper we restrict ourselves to BESses of the following form:

$$\mathcal{E} = (\sigma_1 X_1 = \phi_1) \ldots (\sigma_n X_n = \phi_n)$$

for some $n \in \mathbb{N}$. Moreover $\mathcal{E}$ is in conjunctive/disjunctive form and true and false do not occur in $\mathcal{E}$.

A block in a BES is a set of consecutive equations of the BES with the same fixpoint operator.

Definition 2.6 [Block nesting depth] The block nesting depth ($bnd(\mathcal{E})$) of a BES $\mathcal{E}$ is the number of blocks of $\mathcal{E}$.

Intuitively, the *alternation depth* ($ad(\mathcal{E})$) of a BES $\mathcal{E}$ is the longest sequence of mutually dependent boolean equations in $\mathcal{E}$ with alternating fixpoint symbol. For a more formal definition we refer to [4].

For BES $\mathcal{E}$ its dependency graph $\mathcal{G}_\mathcal{E}$ consists of vertices and edges according to the following simple rules. If $\sigma X_i = f$ is an equation of $\mathcal{E}$, then X_i is a vertex of $\mathcal{G}_\mathcal{E}$, and for all $X_j \in f$, (X_i, X_j) are edges in $\mathcal{G}_\mathcal{E}$.

2.2 Parity games

A parity game is a graph game played by two players, *Even* and *Odd* on a game graph in which each vertex is assigned an integer priority. Player *Even* wins an infinite play if the lowest infinitely often occurring parity in a game is *Even*, otherwise player *Odd* wins the play. We use definitions similar to the ones given by Jurdziński [3]. Furthermore we use the generic *Player* to denote either *Even* or *Odd* in case definitions are defined analogously for both players.

A game graph is a directed graph $\mathcal{G} = (V, E, p{:}V \to \mathbb{N})$, in which V is a set of vertices, E is a total edge relation and p is a priority function, assigning an integer priority to each vertex.

Definition 2.7 [Parity Game] Given game graph $\mathcal{G} = (V, E, p{:}V \to \mathbb{N})$, and partition (V_{Even}, V_{Odd}) of V, $\Gamma = (V, E, p, (V_{Even}, V_{Odd}))$ is a parity game.

A parity game is played by the two players by placing a token on an initial vertex. Then moves are taken indefinitely according to the following simple rule: if the token is on a vertex $v \in V_{Player}$ then player *Player* moves the token along an outgoing edge of v. The result is an infinite path (also referred to as play) $\pi = \langle v_1, v_2, v_3, \dots \rangle$ in the game graph.

Let $Inf(\pi)$ denote the set of priorities occurring infinitely often in play π. Play π is winning for player *Even* if and only if $\min(Inf(\pi))$ is even, π is winning for player *Odd* otherwise.

For finding winning strategies it suffices to look at history free strategies. These are strategies that, independently of the path by which a vertex is reached, always the same successor is chosen. We define such a strategy for a player, fixing an outgoing edge for each vertex in the set corresponding to that player.

Definition 2.8 [Strategy] A function $\psi_{Player}{:}V_{Player} \to V$ is a strategy for player *Player* if $(v, \psi(v)) \in E$ for all $v \in V_{Player}$.

A play $\pi = \langle v_1, v_2, v_3, \dots \rangle$ is consistent with a strategy ψ_{Player} for player *Player* if and only if every vertex $u \in \pi$ is such that $u \in V_{Player}$ is immediately followed by $\psi(u)$

Definition 2.9 [Winning strategy] Strategy ψ_{Player} is a winning strategy for player *Player* from set $W \subseteq V$ if every play starting from a vertex in W, consistent with ψ_{Player} is winning for player *Player*.

Theorem 2.10 [Memoryless determinacy] For every parity game, there is a unique partition (W_{Even}, W_{Odd}) of V such that there is a winning strategy ψ_{Even} for player *Even* from his winning set W_{Even} and a winning strategy ψ_{Odd} for player *Odd* from her winning set W_{Odd}.

Definition 2.11 [i-cyle] We call a cycle in a parity game an i-cycle iff the lowest infinitely often occurring priority on the cycle is i.

We refer to an i-cycle with even i as an even cycle, similarly an i-cycle with odd i is referred to as an odd cycle.

3 Game parity progress measures

A technique for solving parity games, based on the notion of progress measures has been proposed by Jurdziński [3]. A parallel implementation of this algorithm has been presented by van de Pol and Weber [5]. In this section we will recapitulate the theoretic foundations, and shed more light on the proofs given in [3].

The algorithm attaches to each vertex a tuple with as length the maximal priority occurring in the parity game. Initially this is the tuple $\overline{0}$ with 0 in all positions. Furthermore, all even positions always remain 0, and odd positions i are limited to the number of vertices with priority i. On these tuples a lexicographic ordering is defined such that $(n_0, n_1, \dots n_k) \equiv_i (m_0, m_1, \dots m_l)$ if and only if $(n_0, n_1, \dots n_i) \equiv (m_0, m_1, \dots, m_i)$ with $\equiv \in \{<, \leqslant, =, \geqslant, >\}$. Note that a tuple suffixed with zeros preserves these relations.

Example 3.1 $(0,1,0,1) =_0 (0,2,0,1)$ is equivalent to $(0) = (0)$ and hence is true. $(0,1,0,1) <_1 (0,2,0,1)$ is equivalent to $(0,1) < (0,2)$ and hence is also true, whereas $(0,1,0,1) \geqslant_3 (0,2,0,1)$ is $(0,1,0,1) \geqslant (0,2,0,1)$ is false.

Definition 3.2 A function $\varrho{:}V \to \mathbb{N}^d$, with d the maximal priority in the game, is a parity progress measure for parity graph $\mathcal{G} = (V, E, p{:}V \to \mathbb{N})$ if for all $(v, w) \in E$ we have

$$\begin{cases} \varrho(v) \geqslant_{p(v)} \varrho(w) & \text{if } p(v) \text{ is even} \\ \varrho(v) >_{p(v)} \varrho(w) & \text{if } p(v) \text{ is odd} \end{cases}$$

Consider game graph $\mathcal{G} = (V, E, p{:}V \to \mathbb{N})$. For every $i \in \mathbb{N}$ we denote with $V_i \subseteq V$ the set of vertices in $\mathcal{G}$ with priority i. Furthermore we let $n_i = |V_i|$, the number of vertices with priority i. We define the finite subset $M_{\mathcal{G}}$ of $\mathbb{N}^d$, such that it is the finite set of d-tuples with zeros on even positions (counting from 0), and non-negative integers bounded by n_i on odd positions i as follows:

$$M_{\mathcal{G}} = \begin{cases} [1] \times [n_1 + 1] \times [1] \times [n_3 + 1] \times \cdots \times [1] \times [n_{d-1} + 1] & \text{if } d \text{ is even} \\ [1] \times [n_1 + 1] \times [1] \times [n_3 + 1] \times \cdots \times [1] \times [n_{d-2} + 1] \times [1] & \text{if } d \text{ is odd} \end{cases}$$

In the sequel we use notation $\mathcal{G} \restriction V$ to denote the graph $\mathcal{G}$ from which all vertices not in V have been removed, along with the dangling edges that result.

Theorem 3.3 [Small parity progress measure] There is a parity progress measure $\varrho{:}V \to M_{\mathcal{G}}$ for parity graph $\mathcal{G}$ if and only if all cycles in $\mathcal{G}$ are even.

Proof $\Rightarrow$) It is straightforward to see that if there is a parity progress measure for $\mathcal{G}$, then all cycles are even. For a proof see [3].

$\Leftarrow$) We prove that if all cycles in a parity graph $\mathcal{G}$ are even, then there is a parity progress measure $\varrho{:}V \to M_{\mathcal{G}}$ for $\mathcal{G}$ by induction on the number of vertices in $\mathcal{G}$. We additionally show that if $p(v)$ is odd, then $\varrho(v) >_{p(v)} \overline{0}$. This proof is mostly similar to the one by Jurdziński (Theorem 5 in [3]). The proof as given there omits some low-level details that have proven to be essential for a thorough understanding, which is why we repeat the proof here and fill in those details. The proof is constructive, and in itself provides a recursive algorithm for computing a progress measure. An example of this is included in Appendix A.

If $\mathcal{G}$ has only one vertex v, the theorem holds trivially, as there are no edges. We satisfy our additional claim by assigning $\varrho(v) = \overline{0}$ if $p(v)$ is even, or assigning $\varrho(v)$ the tuple with 1 on position $p(v)$ and 0 on all other positions in case $p(v)$ is odd.

If $V_0 \cup V_1 = \emptyset$ we reduce all priorities by a multiple of two, hence we assume that $V_0 \cup V_1 \neq \emptyset$. Note that if the priorities have been reduced, we need to shift the computed progress measure to the right by the same multiple of two to get a progress measure for the original problem.

Suppose that $V_0 \neq \emptyset$. By induction hypothesis a parity progress measure $\varrho{:}(V \setminus V_0) \to M_{\mathcal{G}}$ exists for subgraph $\mathcal{G} \restriction (V \setminus V_0)$. Now consider ϱ in which we set $\varrho(v) = \overline{0}$ for all $v \in V_0$. This is a progress measure for $\mathcal{G}$, as for all $v \in V_0$ and for all $(v, w) \in E$ it holds that $\varrho(v) \geqslant_0 \varrho(w)$, as all even positions in the progress measures are zero.

Suppose that $V_0 = \emptyset$ and $V_1 \neq \emptyset$. There is a partition (W_1, W_2) of V such that $W_1 \neq \emptyset$ and $W_2 \neq \emptyset$, and there is no edge from W_1 to W_2 in $\mathcal{G}$. This partition can be constructed as follows. Pick an arbitrary $u \in V_1$. We define $U \subseteq V$ to be the set of vertices in $\mathcal{G}$ reachable from u in at least one step. If $U = \emptyset$ we choose partition $(W_1, W_2) = (\{u\}, V \setminus \{u\})$. If $U \neq \emptyset$ we choose partition $(W_1, W_2) = (U, V \setminus U)$, note that $W_2 = V \setminus U \neq \emptyset$, because if u would be reachable from itself in at least one step, this means that u is on a cycle; as $V_0 = \emptyset$, 1 (the priority of u) would be the lowest priority on the cycle, thus the cycle is odd, leading to a contradiction.

Consider subgraphs $\mathcal{G}_1 = \mathcal{G} \restriction W_1$ and $\mathcal{G}_2 = \mathcal{G} \restriction W_2$ of $\mathcal{G}$. There are parity progress measures $\varrho_1{:}W_1 \to M_{\mathcal{G}_1}$ ($\varrho_2{:}W_2 \to M_{\mathcal{G}_2}$) for $\mathcal{G}_1$ ($\mathcal{G}_2$) by induction hypothesis. Let $n_i^1 = |W_1 \cap V_i|$ for $i \in \mathbb{N}$. Now function $\varrho{:}V \to M_{\mathcal{G}}$ is a parity progress measure for $\mathcal{G}$, where ϱ is defined as follows:

$$\varrho(v) = \begin{cases} \varrho_1(v) & \text{if } v \in W_1 \\ \varrho_2(v) + (0, n_1^1, 0, n_3^1, \dots) & \text{if } v \in W_2 \end{cases}$$

Because there are no edges from W_1 to W_2, it is straightforward to see that this is a progress measure for vertices in W_1. There may be edges from vertices in W_2 to W_1. Let (v, w) be such an edge. We need to verify that $\varrho(v) >_{p(v)} \varrho(w)$ if $p(v)$ is odd, and $\varrho(v) \geqslant_{p(v)}$ if $p(v)$ is even. Observe that $\varrho_1(w)$ at position i can be at most n_i^1, hence $(0, n_1^1, 0, n_3^1, \dots)$ is the maximal progress measure that can be obtained in ϱ_1. Furthermore from our additional claim it follows that if $p(v)$ is odd $\varrho_2(v) >_{p(v)} \overline{0}$ and if $p(v)$ is even $\varrho_2(v) \geqslant_{p(v)} \overline{0}$. From this we conclude that ϱ is indeed a parity progress measure for $\mathcal{G}$. □

The parity progress measure is still too restrictive to be used for computing a winning strategy, hence we add a largest element $\top$ to $M_{\mathcal{G}}$, such that $M_{\mathcal{G}}^\top = M_{\mathcal{G}} \cup \{\top\}$. Given $\varrho{:}V \to M_{\mathcal{G}}^\top$ and $(v, w) \in E$ then $\mathit{Prog}(\varrho, v, w)$ is the least $m \in M_{\mathcal{G}}^\top$ such that

$$\begin{cases} m \geqslant_{p(v)} \varrho(w) & \text{if } p(v) \text{ is even} \\ m >_{p(v)} \varrho(w), \text{ or } m = \varrho(w) = \top & \text{if } p(v) \text{ is odd} \end{cases}$$

A function $\varrho{:}V \to M_{\mathcal{G}}^\top$ is a game parity progress measure if and only if for all $v \in V$:

- if $v \in V_{\mathit{Even}}$ then $\exists_{(v,w)\in E} \varrho(v) \geqslant_{p(v)} \mathit{Prog}(\varrho, v, w)$
- if $v \in V_{\mathit{Odd}}$ then $\forall_{(v,w)\in E} \varrho(v) \geqslant_{p(v)} \mathit{Prog}(\varrho, v, w)$

We define strategy $\psi_{\mathit{Even}}{:}V_{\mathit{Even}} \to V$ for player *Even* such that for all v $\psi_{\mathit{Even}}(v) = u$, with $\varrho(u) = \min\{\varrho(w) \mid v \to w\}$. In words, ψ_{Even} is the successor u of v which minimizes $\varrho(u)$. The winning set $||\varrho||$ is the set $\{v \mid v \in V \text{ and } \varrho(v) \neq \top\}$.

It was proven by Jurdziński [3] that strategy ψ_{Even} computed from game parity progress measure ϱ is a winning strategy for player *Even* from $||\varrho||$. Furthermore it was shown that there is a game parity progress measure $\varrho{:}V \to M_{\mathcal{G}}^\top$ such that $||\varrho||$ is the winning set of player *Even*.

3.1 Algorithm

In this section we repeat the algorithm for solving parity games based on small progress measures as presented by Jurdziński [3]. We define an ordering $\sqsubseteq$ on the set of functions $V \to M_{\mathcal{G}}^\top$. Given functions $\mu, \varrho{:}V \to M_{\mathcal{G}}^\top$, $\mu \sqsubseteq \varrho$ if and only iff $\mu(v) \leqslant \varrho(v)$ for all $v \in V$. As we are dealing with finite graphs, $V \to M_{\mathcal{G}}^\top$ is finite. Furthermore there are greatest and least elements, hence $\sqsubseteq$ defines a complete lattice. We use $\sqsubset$ if $\mu \sqsubseteq \varrho$ and $\mu \neq \varrho$.

The algorithm uses a family of $\mathit{Lift}(\_, v)$ operators on $V \to M_{\mathcal{G}}^\top$ for all $v \in V$. $\mathit{Lift}(\varrho, v)$, for $v \in V$, is defined as follows:

$$\mathit{Lift}(\varrho, v)(u) = \begin{cases} \varrho(u) & \text{if } u \neq v \\ \min_{(v,w)\in E} \mathit{Prog}(\varrho, v, w) & \text{if } u = v \in V_{\mathit{Even}} \\ \max_{(v,w)\in E} \mathit{Prog}(\varrho, v, w) & \text{if } u = v \in V_{\mathit{Odd}} \end{cases}$$

Observe that for every $v \in V$, $Lift(\_, v)$ is $\sqsubseteq$-monotonous, i.e. for $\varrho_1 \sqsubseteq \varrho_2$, $Lift(\varrho_1, v) \sqsubseteq (\varrho_2, v)$. Furthermore, a game parity progress measure can be computed by determining the simultaneous fixed point of all $Lift(\_, v)$ operators. This leads to the following algorithm in [3]:

Algorithm 3.4 [ProgressMeasureLifting]
μ := $\lambda v \in V.\overline{0}$
`while` $\mu \sqsubset Lift(\mu, v)$ `for some` $v \in V$ `do` $\mu := Lift(\mu, v)$

Given parity game $\Gamma = (V, E, p{:}V \to \mathbb{N}, (V_{Even}, V_{Odd}))$, and maximal priority d in the game. `ProgressMeasureLifting` computes winning sets for both players and a winning strategy for player *Even* from his winning set in $O(d|E| \cdot (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$ time and $O(d|V|)$ space.

4 Equivalence of Boolean Equation Systems and parity games

Given a boolean equation system $\mathcal{E}$ in conjunctive/disjunctive form, we can find a parity game $\Gamma_{\mathcal{E}}$ such that $\mathcal{E}(X)$ is true for some variable X if and only if player *Even* has a winning strategy on $\Gamma_{\mathcal{E}}$ from vertex X.

We translate a BES to a parity game as follows. As game graph we take the dependency graph of the BES. Furthermore vertices get assigned priority according to the block nesting depth in the BES. The first block gets assigned priority 0 if the fixpoint symbol of the block is ν, and 1 if the fixpoint symbol of the block is μ. Each next block is assigned the priority of the preceding block, incremented by 1. Observe that it is invariant under this translation that vertices corresponding to μ-equations get odd priority, and vertices for ν-equations get even priority. We also assign all X_i such that $\sigma X_i = \bigwedge_{k=0}^{n} X_{j_k}$ to V_{Odd} and the other X_i to V_{Even}.

Analogously, we translate a parity game to a BES as follows. For each vertex $v \in V$, with edges $(v, w_1), \ldots (v, w_{n_v})$, we create the following equation. If v in V_{Odd}, we create $\sigma X_v = \bigwedge_{i=0}^{n_v} X_i$, for vertices v in V_{Even}, equation $\sigma X_v = \bigvee_{i=0}^{n_v} X_i$ is introduced. In both bases $\sigma = \mu$ if $p(v)$ is odd, and $\sigma = \nu$ otherwise.

Theorem 4.1 [Equivalence of BES and parity game](Theorem 8.7 in [4]) Player *Even* has a winning strategy for $\mathcal{G}_{\mathcal{E}}$ from vertex X_i if and only if $(\llbracket \mathcal{E} \rrbracket \eta)(X_i) = \mathsf{true}$.

Proof $\Leftarrow$) Assume $(\llbracket \mathcal{E} \rrbracket \eta)(X_i) = \mathsf{true}$. There exists a BES $\mathcal{E}'$ in conjunctive form, where $\mathcal{E}' \leqslant \mathcal{E}$ and $\llbracket \mathcal{E}' \rrbracket \eta = \llbracket \mathcal{E} \rrbracket \eta$, such that all conjunctions in $\mathcal{E}$ are contained in $\mathcal{E}'$, and from each disjunction in $\mathcal{E}$ only one variable is contained in $\mathcal{E}'$. A winning strategy for player *Even* on $\mathcal{G}_{\mathcal{E}}$ is obtained by choosing in each vertex in $V_{\mathcal{E},Even}$ this successor which is also contained in $\mathcal{G}_{\mathcal{E}'}$.

We show by contraposition that player *Even* has a winning strategy for $\mathcal{G}_{\mathcal{E}'}$. Assume that player *Even* does not have a winning strategy for $\mathcal{G}_{\mathcal{E}'}$ from vertex X_i. Hence there is an odd cycle reachable from X_i in $\mathcal{G}_{\mathcal{E}'}$. We want to show that $(\llbracket \mathcal{E} \rrbracket \eta)(X_i) = \mathsf{false}$. There must be a least vertex X_j, reachable from X_i, with odd priority on such a cycle. This corresponds to an equation $\mu X_j = f_j$.

Let us assume $f_j \neq \mathsf{false}$. Assume the cycle has the form $v_0, v_1, \ldots, v_n$ with $v_0 = v_n = X_j$. This gives a substitution sequence as follows: substitute the equation corresponding to v_1 into

f_j, leading to $\mu X_j = f_j^1$. Do this for all vertices $v_1, \ldots v_{n-1}$, leading in $(n-1)$ substitutions to $\mu X_j = f_j^{n-1}$. As f_j^{n-1} can only consist of a conjunction or a single variable (as $\mathcal{E}'$ is in conjunctive form), in both cases this reduces to $\mu X_j = \mathsf{false}$ as X_j must occur as a conjunct in f_j^{n-1}. Therefore also $([\![\mathcal{E}']\!]\eta)(X_j) = \mathsf{false}$.

There is a sequence from X_i to the first occurrence of X_j, which can be traversed backwards applying substitution steps for constants we get that $([\![\mathcal{E}']\!]\eta)(X_i) = [\![\mathcal{E}]\!]\eta = \mathsf{false}$.

For the case where $f_j = \mathsf{false}$ only the substitution argument needs to be applied to derive the same contradiction.

$\Rightarrow$) Dual to the previous case. □

5 BES parity progress measures

Using the equivalence given in the previous section, we provide a definition of the algorithm from Section 3 directly on boolean equation systems.

Define function *rhs* giving the variables occurring in the right hand side of a boolean equation $\sigma_i X_i = f_i$ as follows:

$$rhs(X_i) = \{X_j \mid X_j \in f_i\}$$

Instead of translating all boolean variables to vertices, we keep the original set of variables $\mathcal{X}$. Additionally we use the sets computed using *rhs* instead of explicitly adding edges. Depending on the fixpoint symbol of the first block, the block nesting depth *bnd* of each equation gives the priority of and equation. If the fixpoint symbol of the first block is μ, then the block nesting depth is incremented by one.

Now that we have obtains suitable notation we can define the theory underlying progress measures from Section 3 in terms of BESses. This theory immediately gives us the implementation of the algorithm in terms of boolean equation systems. We omit the proofs in this section, as they follow immediately from the correspondence between BES and parity game, and the corresponding proof in Section 3.

The algorithm we present attaches to each equation a tuple with as length the maximal priority occurring in the BES. Initially this is the tuple $\overline{0}$. Furthermore all even positions always remain 0, and odd positions i are limited to the number of equations in the block with priority i. We use the same lexicographic ordering as in Section 3.

Definition 5.1 A function $\varrho_{\mathcal{E}}{:}\mathcal{X} \to \mathbb{N}^d$, with $d = bnd(\mathcal{E})$, is a parity progress measure for BES $\mathcal{E}$ if for all $X \in \mathcal{X}$ and $X_i \in rhs(X)$ we have

$$\begin{cases} \varrho_{\mathcal{E}}(X) \geqslant_{p(X)} \varrho(X_i) & \text{if } p(X) \text{ is even} \\ \varrho_{\mathcal{E}}(X) >_{p(X)} \varrho(X_i) & \text{if } p(X) \text{ is odd} \end{cases}$$

Consider a BES $\mathcal{E}$. For every $i \in \mathbb{N}$ we denote with $\mathcal{X}_i$ the set of equations in $\mathcal{E}$ with priority i. We let $n_i = |\mathcal{X}_i|$, the number of equations with priority i. We define finite subset $M_{\mathcal{E}}$ of $\mathbb{N}^d$, such that it is the finite subset of d-tuples with zeros on even positions, and non-negative integers bounded by n_i on odd positions i as follows:

$$M_{\mathcal{E}} = \begin{cases} [1] \times [n_1 + 1] \times [1] \times [n_3 + 1] \times \cdots \times [1] \times [n_{d-1} + 1] & \text{if } d \text{ is even} \\ [1] \times [n_1 + 1] \times [1] \times [n_3 + 1] \times \cdots \times [1] \times [n_{d-2} + 1] \times [1] & \text{if } d \text{ is odd} \end{cases}$$

Theorem 5.2 There is a parity progress measure for BES $\mathcal{E}$ if and only if all cycles in the dependency graph of $\mathcal{E}$ are even.

As before we extend the parity progress measure by adding largest element $\top$ to $M_{\mathcal{E}}$, such that $M_{\mathcal{E}}^{\top} = M_{\mathcal{E}} \cup \{\top\}$. Given $\varrho_{\mathcal{E}}{:}\mathcal{X} \to M_{\mathcal{E}}^{\top}$, $X \in \mathcal{X}$ and $X_i \in \mathit{rhs}(X)$, then $\mathit{Prog}(\varrho_{\mathcal{E}}, X, X_i)$ is the least $m \in M_{\mathcal{E}}^{\top}$ such that

$$\begin{cases} m \geqslant_{p(X)} \varrho_{\mathcal{E}}(X_i) & \text{if } p(X) \text{ is even} \\ m >_{p(X)} \varrho_{\mathcal{E}}(X_i), \text{ or } m = \varrho_{\mathcal{E}}(X_i) = \top & \text{if } p(X) \text{ is odd} \end{cases}$$

A function $\varrho_{\mathcal{E}}{:}\mathcal{X} \to M_{\mathcal{E}}^{\top}$ is a BES parity progress measure if and only if for all $X \in \mathcal{X}$:

- if $X \in X_{\mathit{Even}}$ then $\exists_{Y \in \mathit{rhs}(X)} \varrho_{\mathcal{E}}(X) \geqslant_{p(X)} \mathit{Prog}(\varrho_{\mathcal{E}}, X, Y)$
- if $X \in X_{\mathit{Even}}$ then $\forall_{Y \in \mathit{rhs}(X)} \varrho_{\mathcal{E}}(X) \geqslant_{p(X)} \mathit{Prog}(\varrho_{\mathcal{E}}, X, Y)$

We now define strategy $\psi_{\mathit{Even}}{:}\mathcal{X}_{\mathit{Even}} \to \mathcal{X}$ for player *Even* such that for all X $\psi_{\mathit{Even}}(X) = Y$, with $\varrho(Y) = \min\{\varrho(Z) \mid Z \in \mathit{rhs}(X)\}$. In other words, $\psi_{\mathit{Even}}(X)$ is the variable Y in the right hand side of X which minimizes $\varrho(Y)$. The set of variables $||\varrho_{\mathcal{E}}||$ having solution `true` is the set $\{X | X \in \mathcal{X} \text{ and } \varrho_{\mathcal{E}}(X) \neq \top\}$.

Corollary 5.3 Strategy ψ_{Even} computed from BES parity progress measure $\varrho_{\mathcal{E}}$ is a winning strategy for player *Even* from $||\varrho_{\mathcal{E}}||$.

Corollary 5.4 There is a BES parity progress measure $\varrho_{\mathcal{E}}{:}\mathcal{X} \to M_{\mathcal{E}}^{\top}$ such that $||\varrho_{\mathcal{E}}||$ is the set of variables that are `true`.

5.1 Algorithm

We have seen the theoretic foundations needed for the parity lifting algorithm. We define an ordering $\sqsubseteq$ on the set of functions $\mathcal{X} \to M_{\mathcal{E}}^{\top}$. Given functions $\mu_{\mathcal{E}}, \varrho_{\mathcal{E}}{:}\mathcal{X} \to M_{\mathcal{E}}^{\top}$, $\mu_{\mathcal{E}} \sqsubseteq \varrho_{\mathcal{E}}$ if and only if $\mu_{\mathcal{E}}(X) \leqslant \varrho_{\mathcal{E}}(X)$ for all $X \in \mathcal{X}$. Also we are dealing with finite boolean equation systems, hence $\mathcal{X} \to M_{\mathcal{E}}^{\top}$ is finite. Additionally there are greatest and least elements, hence $\sqsubseteq$ defines a complete lattice.

The algorithm uses a family of $\mathit{Lift}(\_, X)$ operators on $\mathcal{X} \to M_{\mathcal{E}}^{\top}$ for all $X \in \mathcal{X}$. $\mathit{Lift}(\varrho_{\mathcal{E}}, X)$, for $X \in \mathcal{X}$ is defined as follows:

$$\mathit{Lift}(\varrho_{\mathcal{E}}, X)(Y) = \begin{cases} \varrho_{\mathcal{E}}(Y) & \text{if } X \neq Y \\ \min_{Y \in \mathit{rhs}(X)} \mathit{Prog}(\varrho_{\mathcal{E}}, X, Y) & \text{if } X = Y \in \mathcal{X}_{\mathit{Even}} \\ \max_{Y \in \mathit{rhs}(X)} \mathit{Prog}(\varrho_{\mathcal{E}}, X, Y) & \text{if } X = Y \in \mathcal{X}_{\mathit{Odd}} \end{cases}$$

As before, for every $X \in \mathcal{X}$, $\mathit{Lift}(\_, X)$ is $\sqsubseteq$-monotonous. A BES parity progress measure can be computed by determining the simultaneous fixed point of all $\mathit{Lift}(\_, X)$ operators. Hence, we get the following algorithm:

Algorithm 5.5 [BESProgressMeasureLifting]
$\mu_{\mathcal{E}}$ `:=` $\lambda X \in \mathcal{X}.\overline{0}$
`while` $\mu_{\mathcal{E}} \sqsubset \mathit{Lift}(\mu_{\mathcal{E}}, X)$ `for some` $X \in \mathcal{X}$ `do` $\mu_{\mathcal{E}} := \mathit{Lift}(\mu_{\mathcal{E}}, X)$

Given the algorithm for parity games and the similarity between BES and parity games we find that `ProgressMeasureLifting` computes the set of variables that is `true` in $O(d\Sigma_{X\in\mathcal{X}}|rhs(X)|\cdot(\frac{|\mathcal{X}|}{\lfloor d/2\rfloor}^{\lfloor d/2\rfloor})$ time, given that d is the maximal block nesting depth.

We can further improve this result by using the alternation depth instead of the block nesting depth in the computation of the priorities. When we do this we derive an algorithm that runs in $O(ad(\mathcal{E})\Sigma_{X\in\mathcal{X}}|rhs(X)|\cdot(\frac{|\mathcal{X}|}{\lfloor ad(\mathcal{E})/2\rfloor})^{\lfloor ad(\mathcal{E})/2\rfloor})$

6 Conclusions

In this paper we have seen a known algorithm from the literature for finding a winning strategy in a parity game. We have used a translation from boolean equation systems to parity games that was also known from the literature. Given this translation we have obtained a way to interpret a BES as parity game. This gives rise to an implementation of the parity game algorithm directly on boolean equation systems, thus constituting a new algorithm for solving boolean equation systems.

In the literature various algorithms are known for the full fragment of boolean equation systems. Approximation based algorithms are known that run in $O(|\mathcal{E}|^{ad(\mathcal{E})})$ as well as $O(ad(\mathcal{E})^2|\mathcal{E}|^{\lfloor ad(\mathcal{E})/2\rfloor+1})$. Gauss elimination and tableaux based methods run in $O(2^{|\mathcal{E}|})$. Hence we see that the algorithm we have presented improves the best known bound for solving the general class of boolean equation systems.

As it was observed by Jurdziński, the lifting strategy is important for practical performance of the algorithm. Hence an interesting open question is whether we can find a lifting strategy that works well on boolean equation systems in practice. Now that we have a means of translating parity game algorithms to BES, it could be investigated how well different parity game algorithms perform on boolean equation systems. Additionally, it might be useful in practice to combine various strategies for solving boolean equation systems. Also an extension of the algorithm to on the fly solving is worth investigating. Last but not least, the question whether the algorithm we have given can be generalized to BESses with arbitrary right hand sides (eliminating the preprocessing step) deserves an answer.

References

[1] E.A. Emerson and C.L Lei. Efficient model checking in fragments of the propositional mu-calculus. In *Proceedings of LICS 1986*, pages 267–278. IEEE Computer Society, 1986.

[2] J.F. Groote, A. Mathijssen, M.v. Weerdenburg, and Y. Usenko. The formal specification language mcrl2. In *Proceedings of Methods for Modelling Software Systems*, volume 06351 of *Dagstuhl Seminar Proceedings*, 2007.

[3] M. Jurdziński. Small progress measures for solving parity games. In H. Reichel and S. Tison, editors, *Proceedings of STACS 2000*, pages 290–301. Springer, 2000.

[4] A. Mader. *Verification of Modal Properties Using Boolean Equation Systems*. PhD thesis, Technische Universität München, 1997.

[5] J.v.d. Pol and M. Weber. A multi-core solver for parity games. In I. Cerna and G. Lüttgen, editors, *Proceedings of PDMC 2008*, volume 220, pages 19–34. Elsevier Science Publishers B. V., 2008.

[6] C. Stirling. Model checking and other games. Notes for Mathfit Workshop on Finite Model Theory, University of Wales Swansea, 1996.

A Constructive small progress measures

In Section 3 we have given a proof for the existence of a parity progress measure for a graph $\mathcal{G}$ if and only if all cycles in $\mathcal{G}$ are even. In order to get a better intuition for this result we provide an example in this section.

A.1 Computing a progress measure

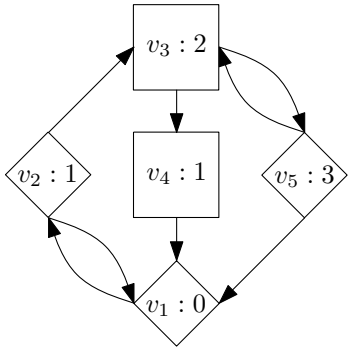

Figure 1: Parity game Γ

Consider parity game $\Gamma = (V, E, p{:}V \to \mathbb{N}, (V_{Even}, V_{Odd}))$ as shown in Figure 1. Note that by convention we draw vertices in V_{Even} as squares and vertices in V_{Odd} as diamonds. We observe that all cycles are even and $V_0 \cup V_1 \neq \emptyset$. We compute a parity progress measure $\varrho{:}V \to M_{\mathcal{G}}$ according to the construction of the proof of Theorem 3.3.

In the first step we find that $V_0 = \{v_1\}$, $V_1 = \{v_2, v_4\}$. By the induction hypothesis there is a progress measure $\varrho{:}(V \setminus V_0) \to M_{\mathcal{G}}$ for the subgraph $\mathcal{G}^1 = \mathcal{G} \restriction (V \setminus V_0)$. Setting $\varrho(v_1) = (0, 0)$ is then a progress measure for $\mathcal{G}$.

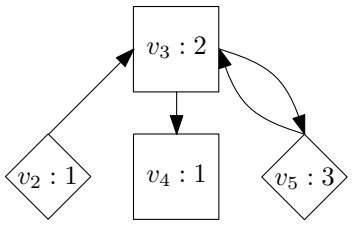

Figure 2: Parity game Γ^1

Let us compute $\varrho{:}(V \setminus V_0) \to M_{\mathcal{G}}$ for the subgraph shown in Figure 2. Denote this graph with $\mathcal{G}^1$. We observe that $V^{1,0} = \emptyset$, $V^{1,1} = \{v_2, v_4\}$. Hence we compute partition $W^{1,1}$, $W^{1,2}$. Pick $v_2 \in V^{1,1}$, then $U^1 = \{v_3, v_4, v_5\} \subseteq V^1$, and we choose $W^{1,1} = U^1$ and $W^{1,2} = V^1 \setminus U^1$. Let $\mathcal{G}^{1,1} = \mathcal{G}^1 \restriction W^1$, $\mathcal{G}^{1,2} = \mathcal{G}^1 \restriction W^2$. By induction hypothesis there are parity progress measures $\varrho^{1,1}{:}W^{1,1} \to M_{\mathcal{G}^{1,1}}$ for $\mathcal{G}^{1,1}$ and $\varrho^{1,2}{:}W^{1,2} \to M_{\mathcal{G}^{1,2}}$ for $\mathcal{G}^{1,2}$. Note that we remember that $n_1^{1,1} = 1$, $n_3^{1,1} = 1$

Figure 3: Parity game $\Gamma^{1,2}$

As subgraph $\mathcal{G}^{1,2}$ as shown in Figure 3 consists of a single vertex (v_2) with odd priority (1), we choose $\varrho^{1,2}{:}W^{1,2} \to M_{\mathcal{G}^{1,2}}$ such that $\varrho^{1,2}(v_2) = (0, 1)$.

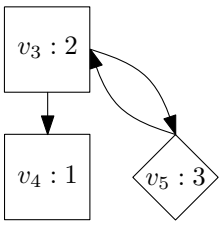

Figure 4: Parity game $\Gamma^{1,1}$

Compute $\varrho^{1,1}{:}W^{1,1} \to M_{\mathcal{G}^{1,1}}$ for the subgraph from Figure 4. Observe that $V_0^{1,1} = \emptyset$, $V_1^{1,1} = \{v_4\}$. We compute partition $W^{1,1,1}, W^{1,1,2}$. Pick $v_4 \in V^{1,1}$. There are no states reachable by a non-trivial path starting in v_4, hence $U^{1,1} = \emptyset$. We find $W^{1,1,1} = \{v_4\}$, $W^{1,1,2} = \{v_3, v_5\}$. Let $G^{1,1,1} = G^{1,1} \restriction W^{1,1,1}$, $G^{1,1,2} = G^{1,1} \restriction W^{1,1,2}$. By induction hypothesis there are parity progress measures $\varrho^{1,1,1}{:}W^{1,1,1} \to M_{\mathcal{G}^{1,1,1}}$ for $\mathcal{G}^{1,1,1}$ and $\varrho^{1,1,2}{:}W^{1,1,2} \to M_{\mathcal{G}^{1,1,2}}$ for $\mathcal{G}^{1,1,2}$. Note that we remember that $n_1^{1,1,1} = 1$, $n_3^{1,1,1} = 0$.

Figure 5: Parity game $\Gamma^{1,1,1}$

Subgraph $G^{1,1,1}$ as shown in Figure 5 consists of a single vertex (v_4) with odd priority (1), hence we choose $\varrho^{1,1,1}{:}W^{1,1,1} \to M_{\mathcal{G}^{1,1,1}}$ such that $\varrho^{1,1,1}(v_4) = (0, 1)$.

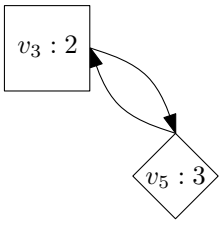

Figure 6: Parity game $\Gamma^{1,1,2}$

Compute $\varrho^{1,1,2}{:}W^{1,1,2} \to M_{\mathcal{G}^{1,1,2}}$. Observe that $V_0^{1,1,2} = V_1^{1,1,2} = \emptyset$, hence we reduce all priorities by 2, leading to the game in Figure 7.

Now compute $\varrho'^{1,1,2}{:}W'^{1,1,2} \to M_{\mathcal{G}'^{1,1,2}}$. Observe that $V_0'^{1,1,2} = \{v_3\} \neq \emptyset$, $V_1'^{1,1,2} = \{v_5\}$. By induction hypothesis there is a progress measure $\varrho'^{1,1,2}(V'^{1,1,2} \setminus V_0'^{1,1,2}) \to M_{\mathcal{G}'^{1,1,2}}$ for the subgraph $G'^{1,1,2,1} = G'^{1,1,2} \restriction (V'^{1,1,2} \setminus V_0'^{1,1,2})$. Setting $\varrho'^{1,1,2}(v_3) = (0, 0)$ gives a progress measure for $\mathcal{G}'^{1,1,2}$

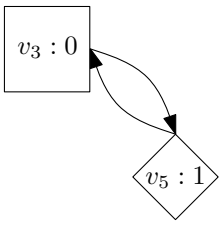

Figure 7: Parity game $\Gamma'^{1,1,2}$

Figure 8: Parity game $\Gamma'^{1,1,2,1}$

Subgraph $\mathcal{G}'^{1,1,2,1}$ as shown in Figure 8 consists of a single vertex (v_5) with odd priority (1), hence we choose $\varrho'^{1,1,2,1}{:}W'^{1,1,2} \to M_{\mathcal{G}'^{1,1,2,1}}$ such that $\varrho'^{1,1,2,1}(v_5) = (0,1)$.

Combining these last two results we obtain $\varrho'^{1,1,2}{:}W'^{1,1,2} \to M_{\mathcal{G}'^{1,1,2}}$ with $\varrho'^{1,1,2}(v_3) = (0,0)$ and $\varrho'^{1,1,2}(v_5) = (0,1)$. However, remember that we had decreased the priorities by 2. We compensate for this by right-shifting all progress measures by 2. This leads to $\varrho^{1,1,2}{:}W^{1,1,2} \to M_{\mathcal{G}^{1,1,2}}$ with $\varrho^{1,1,2}(v_3) = (0,0,0,0)$ and $\varrho^{1,1,2}(v_5) = (0,0,0,1)$.

We continue combining these results until we find a parity progress measure. We combine $\varrho^{1,1,1}$ and $\varrho^{1,1,2}$ into $\varrho^{1,1}$. We know that $\varrho^{1,1} = \varrho^{1,1,1} \cup (\varrho^{1,1,2} + (0, n_1^{1,1,1}, 0, n_3^{1,1,1})) = \varrho^{1,1,1} \cup (\varrho^{1,1,2} + (0,1,0,0))$. Hence $\varrho^{1,1}(v_3) = (0,1,0,0)$, $\varrho^{1,1}(v_4) = (0,1,0,0)$ and $\varrho^{1,1}(v_5) = (0,1,0,1)$.

To this we add $\varrho^{1,1}$ to obtain ϱ^1. We know that $\varrho^1 = \varrho^{1,1} \cup (\varrho^{1,2} + (0, n_1^{1,1}, 0, n_3^{1,1})) = \varrho^{1,1} \cup (\varrho^{1,2} + (0,1,0,1))$. As such we find $\varrho(v_3) = (0,1,0,0)$, $\varrho(v_4) = (0,1,0,0)$, $\varrho(v_5) = (0,1,0,1)$ and $\varrho(v_2) = (0,2,0,1)$. To this we still need to add that $\varrho(v_1) = (0,0,0,0)$. Now ϱ is a parity progress measure for $\mathcal{G}$.

A.2 Verifying progress measure

Now that we have computed a progress measure, let us check if it adheres to Definition 3.2. We check the inequalities along all edges and we find that ϱ is indeed a progress measure. An overview of the relevant inequalities is shown in Table 1.

Abstract	**Concrete**	**Truth**
$\varrho(v_1) \geqslant_0 \varrho(v_2)$	$(0,0,0,0) \geqslant_0 (0,2,0,1)$	$\top$
$\varrho(v_2) >_1 \varrho(v_1)$	$(0,2,0,1) >_1 (0,0,0,0)$	$\top$
$\varrho(v_2) >_1 \varrho(v_3)$	$(0,2,0,1) >_1 (0,1,0,0)$	$\top$
$\varrho(v_3) \geqslant_2 \varrho(v_4)$	$(0,1,0,0) \geqslant_2 (0,1,0,0)$	$\top$
$\varrho(v_4) >_1 \varrho(v_1)$	$(0,1,0,0) >_1 (0,0,0,0)$	$\top$
$\varrho(v_3) \geqslant_2 \varrho(v_5)$	$(0,1,0,0) \geqslant_2 (0,1,0,1)$	$\top$
$\varrho(v_5) >_3 \varrho(v_3)$	$(0,1,0,1) >_3 (0,1,0,0)$	$\top$
$\varrho(v_5) >_3 \varrho(v_1)$	$(0,1,0,1) >_3 (0,0,0,0)$	$\top$

Table 1: Inequalities for progress measure