Solving Parity Games by a Reduction to SAT

Martin Lange
Institut für Informatik, University of Munich, Germany

Abstract. This paper presents a reduction from the problem of solving parity games to the satisfiability problem for formulas of propositional logic in conjunctive normal form. It uses Jurdziński’s characterisation of winning strategies via progress measures. The reduction is motivated by the apparent success that using SAT solvers has had in symbolic verification. The paper reports on a prototype implementation of the reduction and presents some runtime results.

1 Introduction

Solving a parity game is an intrinsic and interesting problem in theoretical computer science. It is equivalent to the model checking problem for the modal \( \mu \)-calculus [7], and is closely related to the problem of solving other games like mean pay-off or stochastic games.

It is also one of the few inhabitants of the complexity class \( \text{NP} \cap \text{co-NP} \) [8] – even \( \text{UP} \cap \text{co-UP} \) [10] – and many people believe that it is in fact in \( \text{PTIME} \). Although no-one has been able to show inclusion in \( \text{PTIME} \) so far, people have invented many algorithms for solving parity games.

Recursive methods like Zielonka’s algorithm [21], etc. solve a game with at most \( p \) different priorities by referring several times to games with strictly less than \( p \) many different priorities. Consequently, their running time is exponential in the number of priorities in the game.

Strategy improvement as done by Jurdziński and Vöge’s algorithm [12] based on Puri’s [17] – and similar to Hoffman and Karp’s [9] as well as Ludwig’s [13] algorithms for stochastic games – uses the fact that strategies can be partially ordered with a winning strategy being maximal w.r.t. this order. This performs very well in practice but it is not known whether a polynomial number of iteration steps always suffices to find a winning strategy.

A randomised and also subexponential algorithm is due to Björklund, Peterson, Sandberg, and Vorobyov [16].

Every model checker for the full \( \mu \)-calculus is in principal also an algorithm for solving parity games. Several of the former have emerged beginning with tableau-like methods [4, 20], automata-theoretic ones [7], equation solvers [1] and symbolic model checking procedures [2].

The algorithm with the currently best asymptotic complexity is Jurdziński’s small progress measures procedure [11]. It is exponential in the number of odd priorities occurring in the game, i.e. in the half of the maximal priority. A similar asymptotic bound is achieved by Seidl’s fixpoint iteration [19].
Opposing common undergraduate syllabi, polynomial time is not a synonym for efficiency. The famous SAT problem is NP-complete [5] and, hence, theoretically does not admit efficient algorithms. However, there are many SAT solvers that are astonishingly efficient in practice, for example CHAFF [15]. Such solvers are used successfully for example in bounded model checking [3].

Inspired by this we present a different approach to solving parity games: a reduction to SAT. Theoretically this is not too exciting since it is clear that such a reduction must exists. Furthermore, SAT is believed to be harder than PARITY. Again, clever heuristics implemented in nowadays SAT solvers can make up for this and result in an algorithm that is efficient in practice. Furthermore, there are fragments of SAT that can be solved in polynomial time. Hence, our reduction opens up a new possibility for showing inclusion of PARITY in PTIME.

The reduction is based on a comment by Emerson where he explains inclusion of the model checking problem for the modal \( \mu \)-calculus in NP. He essentially writes “Guess a rank for each \( \mu \)-subformula at each state in a transition system. Show that the lexicographic order on the tuples through the transition system is well-founded.” [6].

Note that an NP-algorithm is not the same as a reduction to SAT. What we want is a formula of propositional logic that is satisfiable iff the existential player has a winning strategy starting in a certain node of the parity game. We cannot let the SAT solver “guess” a strategy and verify that it is winning by a graph-theoretical method. The resulting propositional formula has to express already that the strategy is winning.

Following the idea about ranks consequently with the aim of a local characterisation of winning strategies we define the notion of a \( \mu \)-annotation — effectively and unintentionally re-inventing Jurdziński progress measures [11]. We stick to the term \( \mu \)-annotation because in this setting they are not dynamically updated and, hence, do not measure any progress. We hereby attribute the theory of \( \mu \)-annotations to Jurdziński explicitly, but keep the corresponding results and proofs in this paper in order to put the reduction to SAT on a solid theoretical foundation.

The rest of the paper is organised as follows. Sections 2 and 3 recall definitions. Section 4 contains the aforementioned theory regarding \( \mu \)-annotations, resp. progress measures. Sections 5 and 6 present two slightly different translations from PARITY to SAT based on the existence of such annotations. Section 7 presents experimental results of one of these translations, and Section 8 discusses further work.

2 Parity Games

A parity game is a tuple \( G = (V, E, v_0, p) \) where \( (V, E) \) is a finite, directed graph and \( V \) is partitioned into two sets \( V_\exists \) and \( V_\forall \), \( v_0 \in V \) is the starting node, and \( p : V \to \mathbb{N} \) is a priority function. \( G \) is assumed to be total, i.e. for every \( v \in V \) there is a \( v' \in V \) with \( (v, v') \in E \).
A play of $G$ is a maximal path $\pi = v_0v_1v_2 \ldots$ through $G$ starting in $v_0$. It is constructed in the following way. Given a node $v \in V_G$, player $x$ chooses a $v' \in V$ with $(v, v') \in E$ and the construction of the play continues with $v'$.

Given a play $\pi = v_0v_1 \ldots$ let $\inf \pi := \{ v \in V |$ there are infinitely many $i \in \mathbb{N}$ s.t. $v = v_i \}$. Player $\exists$ wins the play $\pi = v_0v_1 \ldots$ if $\min \{ p(v) | v \in \inf \pi \}$ is even. If it is odd then player $\forall$ wins the play $\pi$.

A strategy for player $x$ is a function $\sigma : V^*V \rightarrow V$ that tells player $x$ which choice to make depending on the current construction of a play. A strategy is called positional if for all $\alpha, \beta \in V^*$ and all $v \in V_G$ we have $\sigma(\alpha v) = \sigma(\beta v)$. Hence, the choices made according to a positional strategy only depend on the last node visited. In such a case we will rather use $\sigma : V \rightarrow V$.

A play $\pi = v_0v_1 \ldots$ is called conforming to (a positional) strategy $\sigma$ for player $x$ if for all $i \in \mathbb{N}$ if $v_i \in V_G$ then $v_{i+1} = \sigma(v_0 \ldots v_i)$. A strategy $\sigma$ for player $x$ is called a winning strategy if every play conforming to $\sigma$ is won by player $x$.

Given a parity game $G$ and a strategy $\sigma$ for player $\exists$ we write $G|_\sigma$ for the parity game that is induced by $\sigma$ on $G$. Formally, $G|_\sigma = (V, E \cap (V \times V) \cup \{(v, \sigma(v)) | v \in V_G\}, v_0, p)$. The problem of solving a parity game $G = (V, E, v_0, p)$ is to determine whether or not player $\exists$ has a winning strategy for $G$. Let PARITY := $\{ G |$ player $\exists$ has a winning strategy for $G \}$.

(a) Player $\exists$ has a winning strategy for $G$ iff player $\forall$ does not have a winning strategy for $G$.
(b) A player has a winning strategy for $G$ iff she has a positional winning strategy for $G$.

**Theorem 2.** [8] The problem of solving a parity game is in $NP \cap co-NP$.

Given a parity game $G$, the finite unraveling $R(G)$ is informally obtained by unfolding the graph $G$ to a tree with back-edges only to nodes of minimal priority on the thus created loop. Formally, it is defined as $R(G) = (V', E', v_0, p')$ with

$$
V' := \{ v_0 \ldots v_n \in V^+ | \forall i = 1, \ldots, n : (v_{i-1}, v_i) \in E \text{ and } \forall v \in V \text{ there are at most two } i, j \text{ s.t. } v_i = v_j = v \}$$

$$
V'_3 := V' \cap V^*V_3$$

$$
V'_0 := V' \cap V^*V_0$$

$$
E' := \{ (v_0 \ldots v_n, v_0 \ldots v_n v_{n+1}) \in V' \times V' | (v_n, v_{n+1}) \in E \} \cup \{ (v_0 \ldots v_k \ldots v_n, v_0 \ldots v_k) \in V' \times V' | (v_n, v_k) \in E \text{ and } \forall i = k, \ldots, n : p(v_k) \leq p(v_i) \}$$

$$
p'(v_0 \ldots v_n) := p(v_n)$$

For two nodes $v, v' \in V'$ we write $v \sim v'$ iff there is a $w \in V$ s.t. $v = v_0v_1 \ldots w$ and $v' = v_0v'_1 \ldots w$ for some $v_1, v'_1, \ldots$. Note that $\sim$ is an equivalence relation preserving priorities.
Lemma 1. Player $\exists$ wins the parity game $G$ iff she wins the parity game $R(G)$.

Proof. Suppose she has a winning strategy $\sigma$ for $G$. It immediately carries over to a strategy $\sigma'$ on $R(G)$ via $\sigma'(v_0 \ldots v_n) = v_0 \ldots v_n \sigma(v_0 \ldots v_n)$. It is not hard to see that $\sigma'$ is a winning strategy. The converse direction follows from Theorem 1: if she does not have a winning strategy then player $\forall$ has a winning strategy which carries over to a strategy on $R(G)$ in the same way. $\square$

3 Propositional Logic and SAT

Given a set of propositional variables $V = \{X, Y, \ldots\}$, formulas of propositional logic are built in the following way.

$$\Phi ::= X \mid \Phi \lor \Phi \mid \Phi \land \Phi \mid \neg \Phi$$

We use the usual abbreviations: $\varphi \rightarrow \psi := \neg \varphi \lor \psi$ and $\top ::= X \lor \neg X$ for some $X \in V$.

A variable assignment is a mapping $\zeta : V \rightarrow \{0, 1\}$. A formula is called satisfiable iff there is an assignment to the variables in it that makes the formula true under the usual interpretations of the boolean connectives. Let $\text{SAT} := \{\Phi \mid \Phi$ is satisfiable $\}$.

A formula is said to be in conjunctive normal form, if it is of the form $\bigwedge_{i \in I} \bigvee_{j \in J} l_{i,j}$ where for all $i, j; l_{i,j}$ is a literal, i.e. a possibly negated variable. Let $\text{SAT-CNF} := \text{SAT} \cap \{\Phi \mid \Phi$ is in conjunctive normal form $\}$.

Theorem 3. [5] SAT and SAT-CNF are both NP-complete under polynomial time reductions.

4 Characterising Winning Strategies Locally

Given a parity game $G = (V, E, v_0, p)$, let $p_0 \geq \max \{p(v) \mid v \in V \}$ be an odd upper bound on the priorities occurring in $G$. A $\mu$-annotation for $G$ is a tuple $\pi = (a_1, a_3, \ldots, a_{p_0}) \in \mathbb{N}^{\frac{p_0+1}{2}}$.

Given two $\mu$-annotations $\pi = (a_1, \ldots, a_{p_0})$ and $\bar{\pi} = (b_1, \ldots, b_{p_0})$ for $G$ and a $p \leq p_0$, we define

$\pi \leq_p \bar{\pi} \iff \begin{cases} a_i \leq b_i & \text{for all } i = 1, \ldots, p-1 \text{ if } p \text{ is even} \\ a_p < b_p \text{ and } a_i \leq b_i & \text{for all } i = 1, \ldots, p-2 \text{ o.w.} \end{cases}$

We write $\pi^{(p)}$ for some odd $p \in \mathbb{N}$ to denote the $p$-component of $\pi$.

A $\mu$-annotation for $G$ is an $\eta$ that assigns to each $v \in V$ an annotation in the above sense. It is called successful, iff for all $v \in V$:

- if $v \in V_\forall$ then for all $w \in vE$: $\eta(w) \leq_{p(w)} \eta(v)$,
- if $v \in V_\exists$ then there is a $w \in vE$: $\eta(w) \leq_{p(w)} \eta(v).$
Lemma 2. Let $G$ be a parity game, $\eta$ be a $\mu$-annotation for $G$, and $\pi = v_0 v_1 \ldots$ be a play of $G$. If for all $i \in \mathbb{N}$: $\eta(v_{i+1}) \preceq_{p(v_{i+1})} \eta(v_i)$ then the minimal priority occurring infinitely often in $\pi$ is even.

Proof. Suppose that the minimal priority $p$ that occurs infinitely often in $\pi$ is odd. Then there are infinitely many nodes $v_i, v_{i+1}, \ldots$ on $\pi$, s.t. $\eta(v_i)(p) > \eta(v_{i+1})(p) \ldots$ since eventually there is no lower even priority anymore that would allow $\eta(v_i)(p) < \eta(v_{i+1})(p)$ for some $j \in \mathbb{N}$. But then we cannot have $\eta(v_{i+1}) \preceq_{p(v_{i+1})} \eta(v_i)$ for all $i \in \mathbb{N}$, because the natural numbers are Noetherian.

Lemma 3. Let $G$ be a parity game and $\sigma$ be a strategy for a player $\exists$. An $\eta$ is a successful $\mu$-annotation for $G$ iff it is a successful $\mu$-annotation for $G|_\sigma$.

Proof. The “only if” part is trivial since $G|_\sigma$ is a substructure of $G$. For the “if” part note that $G$ and $G|_\sigma$ have the same set of nodes. Hence, a successful $\mu$-annotation $\eta$ for $G|_\sigma$ is also a $\mu$-annotation for $G$. It is easy to see that it is also successful. Let $G = (V_3, V_\ell, E, v_0, p)$ and $G|_\sigma = (V_3, V_\ell, E', v_0, p)$. Take any node $v \in V$. If $v \in V_3$ then $vE = vE'$, hence success immediately carries over to $G$ on these nodes. If $v \in V_3$ then $vE' = \{w\}$ for some $w \in V$ with $\eta(w) \preceq_{p(w)} \eta(v)$. Now $vE \supseteq \{w\}$, i.e. there is a $w$ with $\eta(w) \preceq_{p(w)} \eta(v)$.

Lemma 4. Let $G$ be a parity game. There is a successful $\mu$-annotation for $G$ iff there is a successful $\mu$-annotation for $\mathcal{R}(G)$.

Proof. Let $V$ be the node set of $G$ and $V'$ be the node set of $\mathcal{R}(G)$. A successful $\mu$-annotation $\eta$ for $G$ immediately carries over to a $\mu$-annotation $\eta'$ on $\mathcal{R}(G)$ via $\eta'(v_0 \ldots v_n) = \eta(v_n)$. It is not hard to see that $\eta'$ is also successful.

Now suppose there is a successful $\mu$-annotation $\eta'$ for $\mathcal{R}(G)$. In a first step we construct a successful $\mu$-annotation $\eta_{\text{min}}$ with minimal values. This can be done by successively decreasing annotation values whilst preserving success in each step. For loops in $\mathcal{R}(G)$ this has to be done simultaneously for all nodes on this loop. Next, observe that an annotation at a node $v$ only depends on the annotations at nodes reachable from $v$. Furthermore, for two nodes $v, v' \in V'$ the paths emerging from $v$ and $v'$ projected onto their last component are the same. Hence, we have $\eta_{\text{min}}(v) = \eta_{\text{min}}(v')$ if $v \sim v'$. Finally, this yields a well-defined $\mu$-annotation for $G$ via $\eta(w) = \eta_{\text{min}}(v')$ for some $v' = v_0 \ldots w$. Again, it is not hard to see that $\eta$ is successful.

Theorem 4. Player $\exists$ wins the parity game $G$ iff there is a successful $\mu$-annotation for $G$.

Proof. ($\Rightarrow$) Suppose there is a successful $\mu$-annotation $\eta$ for $G = (V, E, v_0, p)$. It immediately yields a positional strategy $\sigma$ for player $\exists$ defined by $\sigma(v) = w$ only if $\eta(w) \preceq_{p(w)} \eta(v)$. By assumption, for every $v \in V_3$ there is at least one such $w$. If there is more than one satisfying the inequality then $\sigma(v)$ can simply be defined as any of them. It remains to be seen that $\sigma$ is a winning strategy.
Let player $\forall$ play against $\sigma$ with any other strategy $\sigma'$. The result is a play $\pi = v_0v_1\ldots$. Since $\eta$ is a successful $\mu$-annotation we have for all $i \in \mathbb{N}: \eta(v_{i+1}) \leq p(v_{i+1}) \eta(v_i)$. Lemma 2 then shows that player $\exists$ wins the play. Since she does so for any play $\pi$ which is conforming to $\sigma$, she wins the game $G$.

$(\Rightarrow)$ Suppose player $\exists$ has a winning strategy $\sigma$ for $G$. According to Theorem 1 we can assume $\sigma$ to be positional. We will use $\sigma$ to construct a $\mu$-annotation $\eta$ for $\mathcal{R}(G|_{\sigma})$. Let $p_{\text{max}}$ be the maximal odd priority used in $G$. At the beginning set $\eta(v_0) = (2 \cdot |V|, \ldots, 2 \cdot |V|)$. Suppose a node $v$ on level $n$ of the quasi-tree $\mathcal{R}(G)$ has already been assigned an annotation $\eta(v) = (a_1, \ldots, a_{p_{\text{max}}})$. Then for every son $w$ of $v$ that is not a predecessor of $v$ set

$$\eta(w) := \begin{cases} (a_1, \ldots, a_{p(w)-2}, a_{p(w)} - 1, 2 \cdot |V|, \ldots, 2 \cdot |V|) & \text{if } p(w) \text{ is odd,} \\ (a_1, \ldots, a_{p(w)-1}, 2 \cdot |V|, \ldots, 2 \cdot |V|) & \text{otherwise.} \end{cases}$$

Note that the depth of the quasi-tree $\mathcal{R}(G)$ is at most $2 \cdot |V|$ which is why the starting value for $\eta$ at $v_0$ suffices.

It remains to be seen that $\eta$ is successful. Clearly, if $w$ is a son but not a predecessor of $v$ in $\mathcal{R}(G|_{\sigma})$ then $\eta(w) \leq_{p(w)} \eta(v)$. Now suppose there is an edge from $v$ to $w$ in $\mathcal{R}(G|_{\sigma})$ s.t. $w$ is also a predecessor of $v$. According to Lemma 1, player $\exists$ also wins the parity game $\mathcal{R}(G|_{\sigma})$. Hence, the minimal priority occurring on the path $w_0\ldots w_n$, with $w = w_0$ and $v = w_n$, is even. Let $p_0$ be this priority.

By the construction above we have $\eta(w_i)^{(p)} = \eta(w_j)^{(p)}$ for all $i, j = 0, \ldots, n$ and all $p \leq p_0$, in particular $i = 0$ and $j = n$. Furthermore, $\mathcal{R}(G|_{\sigma})$ is constructed s.t. $p(w) = p_0 \leq p(w_i)$ for all $i = 0, \ldots, n$. Thus, we also have $\eta(w) \leq_{p(w)} \eta(v)$.

Applying Lemmas 3 and 4 shows that there also is a successful $\mu$-annotation for $G$ which concludes the proof. 

\begin{corollary}
Let $G$ be a parity game with node set $V$. There is a successful $\mu$-annotation for $G$ iff there is a successful $\mu$-annotation $\eta$ for $G$ s.t. for all $v \in V$: if $\eta(v) = (a_1, \ldots, a_p)$ then for all $i = 1, \ldots, p: 0 \leq a_i < |V|$. 

\begin{proof}
Given a successful $\mu$-annotation for $G$ we can apply Theorem 4 twice in order to first obtain a positional winning strategy $\sigma$ for player $\exists$ and then a successful $\mu$-annotation $\eta'$ for $\mathcal{R}(G|_{\sigma})$ with node set $V'$. We have $\eta(v)^{\mu} \leq 2 \cdot |V|$ for all $v \in V'$. The proof of Lemma 4 reduces this to a minimal successful $\mu$-annotation $\eta$ for $\mathcal{R}(G)$ which is also a successful $\mu$-annotation for $G$. But then we must have $\eta(v)^{\mu} < |V|$ because the length of a maximal descending chain of natural numbers in any component of the annotations is bounded by the maximal number of transitions on a loop in $G|_{\sigma}$ which is $|V|$. Finally, the proof of Lemma 3 shows that a successful $\mu$-annotation for $G|_{\sigma}$ is also a successful $\mu$-annotation for $G$.
\end{proof}
\end{corollary}

5 Using Unary Encodings of Annotation Values

Theorems 4 and 1 yield a reduction from PARITY to SAT. Given a parity game $G = (V, E, v_0, p)$, let $p_{\text{max}}$ be the maximal priority used in $G$. We build a propositional formula $\Phi_{G}^1$, s.t. a satisfying variable assignment for $\Phi_{G}^1$ encodes a successful $\mu$-annotation for $G$. $\Phi_{G}^1$ contains variables
- $S_v$ for every $v \in V$. They are used to mark nodes that can be visited in a play that is conforming to $\sigma$.
- $T_{v,w}$ for every $v \in V$ and every $w \in vE$. They are used to mark edges that player $\exists$ moves along in a play conforming to $\sigma$.
- $Y_{p,a}^v$ for every $v \in V$, every odd priority $p$ used in $G$ and every $a \in \{0, \ldots, |V| - 1\}$. Intuitively, $Y_{p,a}^v$ asserts that $\eta(v)^{(p)} = a$.

Part $\Psi$ of $\Phi_G$ expresses that the variables $S_v$ and $T_{v,w}$ represent a positional strategy for player $\exists$. Furthermore, $\Phi_G^1$ asserts that this strategy is a winning strategy, i.e., the corresponding annotation values form a successful annotation for $G$. Finally, $\Psi'$ says that every $p$-component of every node $v$ must have at least one value.

$$\Phi_G^1 := \Psi \land \Psi' \land \bigwedge_{v \in V} \bigwedge_{w \in vE} T_{v,w} \rightarrow A(v, w)$$

$$A(v, w) := \bigwedge_{p < p(w)} \bigwedge_{p \text{ odd}} E(v, w, p) \land \begin{cases} L(v, w) & \text{if } p(w) \text{ is odd} \\
1 & \text{o.w.} \end{cases}$$

$$E(v, w, p) := \bigwedge_{a=0}^{\lfloor |V|/2 \rfloor} \left( Y_{p,a}^w \rightarrow \bigvee_{b \leq a} Y_{p,b}^w \right)$$

$$L(v, w) := \bigwedge_{a=0}^{\lfloor |V|/2 \rfloor} \left( Y_{p(w), a}^w \rightarrow \bigvee_{b < a} Y_{p(w), b}^w \right)$$

$$\Psi := S_{v_0} \land \bigwedge_{v \in V} \bigwedge_{w \in vE} \left( S_v \rightarrow T_{v,w} \right) \land \bigwedge_{v \in V} \left( S_v \rightarrow \bigvee_{w \in vE} T_{v,w} \right) \land \bigwedge_{(v,w) \in E} T_{v,w} \rightarrow S_w$$

$$\Psi' := \bigwedge_{v \in V} \bigwedge_{p \leq p_{\text{max}}} \bigwedge_{\lfloor |V|/2 \rfloor} Y_{p,a}^w$$

Formula $A(v, w)$ asserts that $\eta(w) \leq p(w) \eta(v)$ for an annotation $\eta$ given as an assignment to the variables $Y_{p,a}^v$. Formula $E(v, w, p)$ says that $\eta(w)^{(p)} \leq \eta(v)^{(p)}$, and formula $L(v, w)$ says that $\eta(w)^{(p(w))} < \eta(v)^{(p(w))}$.

**Theorem 5.** Player $\exists$ has a winning strategy for the parity game $G$ iff $\Phi_G^1$ is satisfiable.

**Proof.** Suppose player $\exists$ wins the parity game $G = (V, E, v_0, p)$. According to Theorem 1, she has a positional winning strategy $\sigma$. This immediately yields
a valuation for the variables \( S_v \) and \( T_{v,w} \) fulfilling the subformula \( \Psi \). Furthermore, according to Theorem 4, there is a successful \( \mu \)-annotation \( \eta \) for \( G \). This immediately yields an assignment for the variables \( Y_{v,p}^{w} \) defined by \( Y_{v,p}^{w} = 1 \) iff \( \eta(v)^{(p)} = a \). It is not hard to see that the remaining conjuncts are all fulfilled by this evaluation.

Now suppose that \( \Phi_1^G \) is satisfiable. Then, the satisfying variable assignment defines a positional strategy \( \sigma \) for player \( \exists \). We define an annotation \( \eta \) for \( G \) by \( \eta(v)^{(p)} = a \) iff \( a = \min \{ b \mid Y_{v,p}^{w} = 1 \} \). Again, it is not hard to see that \( \eta \) is successful. Note that \( \Phi_G \) asserts that whenever \( (v,w) \in E \) and both \( v \) and \( w \) belong to the strategy, then \( \eta(w) \leq_{p(w)} \eta(v) \). This implies success of \( \eta \). □

Most SAT solvers expect their input to be in conjunctive normal form, and their performance depends on the input size as well as the number of variables occurring in it.

**Proposition 1.** Given a parity game \( G \) with nodes \( V \), edges \( E \) and maximal priority \( p_{\text{max}} \), there is a \( \Phi' \) in conjunctive normal form that is equivalent to \( \Phi_1^G \) w.r.t. satisfiability s.t. \( |\Phi'| = O(|V|^2 \cdot (|E| + \lceil \frac{E}{2} \rceil) + |V| \cdot |E| \cdot \lceil \frac{E_{max}}{2} \rceil) \) and the number of variables used in \( \Phi' \) is \(|V| + |E| + |V|^2 \cdot \lceil \frac{E_{max}}{2} \rceil\).

**Proof.** In order to transform \( \Phi_1^G \) to conjunctive normal form, one only needs the equivalence \( \varphi \rightarrow (\psi_1 \land \psi_2) \equiv (\varphi \rightarrow \psi_1) \land (\varphi \rightarrow \psi_2) \). Note that this results in a linear blow-up only since the formulas on the left-hand side of the implication are of constant size. The result of this transformation applied to \( \Phi_1^G \) is presented in Appendix A. □

### 6 Using Binary Encodings of Annotation Values

We present a second reduction from \( \text{PARITY} \) to \( \text{SAT} \) that proceeds along the same lines but encodes the annotation values binarily. This yields asymptotically smaller formulas and uses less variables. The disadvantage is their transformation into conjunctive normal form which is more complex and requires the introduction of additional variables. However, those variables are merely macros, hence, their values in an assignment are completely determined by the values of the variables used in the first place.

Given a parity game \( G \) with nodes \( V \), edges \( E \) and maximal priority \( p_{\text{max}} \), let \( n := |V| \), \( m := \lceil \log n \rceil - 1 \). In addition to the variables \( S_v \) and \( T_{v,w} \) for every \( v \in V, w \in vE \) as introduced in Section 5, formula \( \Phi_2^G \) uses variables \( X_{p,i}^{v} \) for every \( v \in V \), every odd priority \( p \leq p_{\text{max}} \), and every \( i = 0, \ldots, m \). Intuitively, \( X_{p,i}^{v} \) represents the \( i \)-th bit of \( \eta(v)^{(p)} \) for an annotation \( \eta \) of \( G \).

\[
\Phi_2^G := \Psi \land \bigwedge_{v \in V} \bigwedge_{w \in vE} T_{v,w} \rightarrow A(v, w)
\]

\[
A(v, w) := \bigwedge_{p < p(w)} E_m(v, w, p) \land \begin{cases} L^m(v, w) & \text{if } p(w) \text{ is odd} \\ 1 & \text{o.w.} \end{cases}
\]
where $\Psi$ is as defined in $\Phi^1_G$.

Formula $A(v, w)$ is the same as used in $\Phi^1_G$. Formula $E^i(v, w, p)$ says that the $i$-th bit of $\eta(w)^{(p)}$ is less or equal than that of $\eta(v)^{(p)}$, and if they are equal then the same holds recursively for the next lower bit. Formula $L^i(v, w)$ does the same for the $p(w)$ components of $\eta(w)$ and $\eta(v)$ but requires them to differ at one bit at least.

The following is proved in just the same way as Theorem 5 with the values encoded binarily instead of unarily. Note that for fixed $v$ and $p$– any assignment to the variables $X^v_{p,i}$ defines a value between 0 and $|V| - 1$. Thus, it is not necessary to explicitly require one to exist as it is done by $\Psi'$ in $\Phi^3_G$.

**Theorem 6.** Player $\exists$ has a winning strategy for the parity game $G$ iff $\Phi^2_G$ is satisfiable.

**Proposition 2.** For every parity game $G$ with nodes $V$, edges $E$ and maximal priority $p_{\max}$, there is a $\Phi'$ in conjunctive normal form that is equivalent to $\Phi^2_G$ w.r.t. satisfiability s.t. $|\Phi'| = O(|E| \cdot \frac{p_{\max}}{2} \cdot \log |V|)$ and the number of variables used in $\Phi'$ is

$$|V| \cdot (1 + \lceil \frac{p_{\max}}{2} \rceil \cdot \lceil \log |V| \rceil) + |E| \cdot (1 + \lceil \log |V| \rceil) \cdot (4 \cdot \lceil \frac{p_{\max}}{2} \rceil + 2)$$

**Proof.** $\Phi^3_G$ can be brought into conjunctive normal form using limited expansion: a subformula $\psi$ that violates the conjunctive normal form is replaced by a new variable $\psi$ and top-level clauses are added that express $Z_\psi \iff \psi$. This is done recursively on $\psi$'s subformulas for as long as they are not literals. $\square$

### 7 Experimental Results

We report on a prototype implementation of the translation from PARITY to SAT in Section 6. The programming language used for this implementation is the lazy functional language Haskell using the Glasgow Haskell Compiler. The tests were carried out on a machine with two Intel® Xeon™ 2.4 GHz processors and 4GB of RAM. The second processor remained unused. We do not report on the implementation using unary encodings, since its performance was worse than the binary one throughout all the tests.

The program takes parameters $n$, $p$ and $o$, and creates a random graph with $n$ nodes. The probability that $v \in V_\exists$ is 50% for each $v \in V$. Moreover, each $v$ is
assigned a priority that is chosen randomly from an even distribution between 0 and \( p \) inclusively. Finally, every \( v \) has exactly outdegree \( a \). We have restricted the test runs to \( a = 2 \) in order to reduce dimensions for presenting the results. Note that every parity game can easily be transformed into an equivalent game with constant outdegree 2. The random graphs are directly translated into propositional formulas in conjunctive normal form and fed to the SAT solver zChaff.

Two sets of benchmarks were carried out on such random graphs of successively increasing size \( n \): one with \( p = 1 \) which corresponds to model checking formulas of the modal \( \mu \)-calculus with at most one alternation; the other with \( p = \lceil \sqrt{n} \rceil \) which seems to be a reasonable choice for a principally unbounded but feasible number of priorities. Each input was processed 5 times. The minimal and maximal running times that occurred in this test series are shown in Figure 1. A value of “†” means that zChaff was aborted manually after running for more than 6h. For \( n > 1600 \) and \( p_{\text{max}} = \lceil \sqrt{n} \rceil \) there is an increasing number of instances that do not terminate within 6h. For \( p_{\text{max}} = 1 \), the series shows a steady growth with more diverging running times. For example, checking games with \( n = 20000 \) takes approx. 1 min for the reduction and 30 sec to 5 min for the solving.

\[
\begin{array}{|c|cc|cc|}
\hline
n & \multicolumn{2}{c|}{p_{\text{max}} = 1} & \multicolumn{2}{c|}{p_{\text{max}} = \lceil \sqrt{n} \rceil} \\
    & \text{reduction} & \text{solving} & \text{reduction} & \text{solving} \\
\hline
100 & 0.05s & 0.03s - 0.05s & 0.3s & 0.1s - 0.2s \\
200 & 0.1s & 0.06s - 0.1s & 1s - 1.2s & 0.7s - 11m54s \\
300 & 0.2s & 0.2s & 2.2s - 2.6s & 1.1s - 2.1s \\
400 & 0.3s & 0.2s - 0.3s & 4s - 4.2s & 1.4s - 4.4s \\
500 & 0.4s & 0.2s - 0.3s & 5.5s - 5.9s & 3.3s - 37.3s \\
600 & 0.6s & 0.4s & 8.3s - 9.2s & 4.8s - 1m4s \\
700 & 0.7s & 0.3s - 0.5s & 10.9s - 12.1s & 3.8s - 1m25s \\
800 & 0.8s & 0.6s & 14s - 15.5s & 6s - 1m44s \\
900 & 0.9s & 0.6s - 0.7s & 17s - 20s & 9.2s - 2h6m \\
1000 & 1s & 0.7s - 0.8s & 20s - 23s & 6.3s - 4m19s \\
1100 & 1.2s & 0.9s - 1s & 29s - 34s & 9.8s - 1m11s \\
1200 & 1.4s & 1s & 40s & 16s - 6m40s \\
1300 & 1.5s & 1.1s - 1.5s & 45s & 18s - † \\
1400 & 1.6s & 1.2s - 1.3s & 52s & 20s - 19m48s \\
1500 & 1.7s & 1.3s & 58s - 1m6s & 16s - 2h28m \\
1600 & 1.8s & 1.4s - 1.8s & 1m12s & 16s - † \\
\hline
\end{array}
\]

8 Conclusions

We have shown how the apparent power of recent SAT solvers can be used to solve parity games. The experimental results look promising: most parity games
are solved very quickly, but there are a few examples of formulas that could not be solved in reasonable time.

What remains to be done is to

- extend the translation in order to solve parity games globally, i.e., compute the entire winning region of one of the players rather than decide for a designated node only which winning region it belongs to;
- speed up the implementation and make it more memory-efficient by re-implementing it in a different language, for example C++;
- compare the optimised procedure with implementations of other algorithms for parity games, for instance omega [18], as well as model checkers for the modal $\mu$-calculus like SMV [14], etc;
- check how this algorithms performs on families of parity games that are known to cause exponential behaviour of other algorithms;
- check whether the formulas created by one of the translations above can be solved in polynomial time.

References

The following formula in conjunctive normal form is equivalent to $\Phi^1_G$ w.r.t. satisfiability:

\[
\phi' := S_{v_0} \land \bigwedge_{v \in V_0} \bigwedge_{w \in vE} (\neg S_v \lor T_{v,w}) \land \bigwedge_{v \in V_0} \bigwedge_{w \in vE} (\neg S_v \lor \bigvee \bigwedge_{p \leq p_{\max} \land \text{odd}} T_{v,w}) \\
\land \bigwedge_{v \in V} \bigg( \bigwedge_{p \leq p_{\max} \land \text{odd}} \bigwedge_{a=0}^{[V]-1} Y^v_{p,a} \land \bigwedge_{w \in vE} \bigwedge_{p(w) \text{ odd}}^{[V]-1} \neg T_{v,w} \lor \neg Y^v_{p(w),a} \lor \bigvee_{b \leq a} Y^w_{p(w),b} \\
\land \bigwedge_{w \in vE} \left( (\neg T_{v,w} \lor S_w) \land \bigwedge_{p \leq p(w)}^{[V]-1} \bigwedge_{a=0}^{[V]-1} \neg T_{v,w} \lor \neg Y^v_{p,a} \lor \bigvee_{b \leq a} Y^w_{p(b),b} \right) \bigg)
\]

A Transformation of $\Phi^1_G$ to CNF
B Transformation of $\Phi^2_G$ to CNF

In addition to the variables $S_v$, $T_{v,w}$ and $X_{p,i}^v$ used in $\Phi^2_G$ we need variables

- $E_{p,i}^{v,w}$ to abbreviate $E^i(v, w, p)$,
- $L_{p,i}^{v,w}$ to abbreviate $L^i(v, w)$,
- $J_{p,i}^{v,w}$ to abbreviate $\neg X_{p,i}^{v,w} \lor X_{p,i}^{v,w}$,
- $K_{p,i}^{v,w}$ to abbreviate $\neg K_{p,i}^{v,w} \lor E_{p,i}^{v,w}$,
- $H_{v,w}^i$ to abbreviate $\neg H_{v,w}^i \lor L_{i}^{v,w}$

Then we can write $\Phi^2_G$ in conjunctive normal form as

$$\Phi := S_{v_0} \land \bigwedge_{v \in V} \bigwedge_{w \in vE} \left( \neg S_v \lor T_{v,w} \right) \land \bigwedge_{v \in V_2} \left( \neg S_v \lor \bigvee_{w \in vE} T_{v,w} \right)$$

$$\land \bigwedge_{v \in V} \left( \bigwedge_{w \in vE} \left( \neg T_{v,w} \lor L_{p,0}^{v,w} \right) \land D_L(v, w) \land D_H(v, w) \land D_I(v, w) \land D_K(v, w, p) \land D_T(v, w) \land D_E(v, w) \land D_J(v, w, p) \right) \right)$$

where

$$D_E(v, w, p) :=$$

$$\left( \neg E_{p,0}^{v,w} \lor \neg X_{p,0}^{w,v} \lor X_{p,0}^{w,v} \right) \land \left( E_{p,0}^{v,w} \lor X_{p,0}^{w,v} \right) \land \left( E_{p,0}^{v,w} \lor \neg X_{p,0}^{w,v} \right) \land$$

$$\bigwedge_{i=1}^m \left( E_{p,i}^{v,w} \lor \neg J_{p,i}^{v,w} \lor J_{p,i}^{v,w} \right) \land \left( \neg E_{p,i}^{v,w} \lor I_{p,i}^{v,w} \right) \land \left( \neg E_{p,i}^{v,w} \lor J_{p,i}^{v,w} \right)$$

$$D_L(v, w) :=$$

$$\left( L_0^{v,w} \lor X_{p(w),0}^{v,w} \lor \neg X_{p(w),0}^{v,w} \right) \land$$

$$\left( \neg L_0^{v,w} \lor \neg X_{p(w),0}^{v,w} \right) \land \left( \neg L_0^{v,w} \lor X_{p(w),0}^{v,w} \right) \land$$

$$\bigwedge_{i=1}^m \left( L_i^{v,w} \lor \neg I_{p(i),i}^{v,w} \lor \neg H_{i}^{v,w} \right) \land \left( \neg L_i^{v,w} \lor I_{p(i),i}^{v,w} \right) \land \left( \neg L_i^{v,w} \lor H_{i}^{v,w} \right)$$
\[ D_I(v, w, p) := \]
\[
\bigwedge_{i=1}^{m} \left( \neg I_{p, i}^{v, w} \lor \neg X_{p, i}^{v, w} \lor X_{p, i}^{v, w} \right) \land \left( I_{p, i}^{v, w} \lor X_{p, i}^{v, w} \right) \land \left( I_{p, i}^{v, w} \lor \neg X_{p, i}^{v, w} \right)
\]

\[ D_K(v, w, p) := \]
\[
\bigwedge_{i=1}^{m} \left( \neg K_{p, i}^{v, w} \lor X_{p, i}^{v, w} \lor \neg X_{p, i}^{v, w} \right) \land \left( K_{p, i}^{v, w} \lor \neg X_{p, i}^{v, w} \right) \land \left( K_{p, i}^{v, w} \lor X_{p, i}^{v, w} \right)
\]

\[ D_J(v, w, p) := \]
\[
\bigwedge_{i=1}^{m} \left( \neg J_{p, i}^{v, w} \lor E_{p, i-1}^{v, w} \lor \neg K_{p, i}^{v, w} \right) \land \left( J_{p, i}^{v, w} \lor \neg E_{p, i-1}^{v, w} \right) \land \left( J_{p, i}^{v, w} \lor K_{p, i}^{v, w} \right)
\]

\[ D_H(v, w) := \]
\[
\bigwedge_{i=1}^{m} \left( \neg H_{i}^{v, w} \lor \neg K_{p(w), i}^{v, w} \lor L_{i-1}^{v, w} \right) \land \left( H_{i}^{v, w} \lor K_{p(w), i}^{v, w} \right) \land \left( H_{i}^{v, w} \lor \neg L_{i-1}^{v, w} \right)
\]