# 1 Recursive algorithm for MPGs

**Definition 1.1.** *A multidimensional parity game (MPG) is a parity game defined over a set of configuration $\mathfrak{C}$ and a set of origin vertices $\mathfrak{V}$. Every vertex is represented by a pair of a configuration and an origin vertex. We have MPG being a tuple $G = (V, V_0, V_1, E, \Omega)$ such that:*

- $V \subseteq \mathfrak{C} \times \mathfrak{V}$,
- $V_0 \uplus V_1 = V$,
- $E \subseteq V \times V$ *such that* $((c, v), (c', v')) \in E \implies c = c'$ *and*
- $\Omega : \mathfrak{V} \to \mathbb{N}$.

**Definition 1.2.** *For a set $X \subseteq \mathfrak{C} \times \mathfrak{V}$ we define $con(X) \subseteq \mathfrak{C}$ to be the set of configurations that occur in $X$, formally: $con(X) = \{c \mid (c, v) \in X\}$.*

**Definition 1.3.** *For a set $X \subseteq \mathfrak{C} \times \mathfrak{V}$ we define $cX = \{v \in \mathfrak{V} \mid (c, v) \in X\}$ and $Xv = \{c \in \mathfrak{C} \mid (c, v) \in X\}$.*

**Definition 1.4.** *For a set $E \subseteq (\mathfrak{C} \times \mathfrak{V}) \times (\mathfrak{C} \times \mathfrak{V})$ we define $cE = \{(v, w) \mid ((c, v), (c, w)) \in E\}$.*

**Definition 1.5.** *An MPG $G = (V, V_0, V_1, E, \Omega)$ can be played for a configuration $c \in \mathfrak{C}$ which is playing PG $(cV, cV_0, cV_1, cE, \Omega)$, we denote this as $cG$.*

**Definition 1.6.** *$\alpha$-MAttr* $: MPG \to \mathcal{P}(V) \to \mathcal{P}(V)$

$$\begin{aligned}\alpha\text{-}MAttr(G, U) = \mu A.U \cup &\{(c, v) \in V_\alpha \mid \exists (c, v') \in V : (c, v') \in A \wedge ((c, v), (c, v')) \in E\} \\ \cup &\{(c, v) \in V_{\overline{\alpha}} \mid \forall (c, v') \in V : ((c, v), (c, v')) \in E \implies (c, v') \in A\}\end{aligned}$$

**Definition 1.7.** $\backslash^M : MPG \to \mathcal{P}(V) \to MPG$
$(V, V_0, V_1, E, \Omega) \backslash^M CV = (V', V_0', V_1', E', \Omega)$ *such that:*
$V' = V \backslash CV$
$E' = E \cap (V' \times V')$
$V_0' = V_0 \cap V'$
$V_1' = V_1 \cap V'$

**Definition 1.8.** *[1] A set $U \subseteq V$ is an $\alpha$-trap for PG $G$ iff:*

$$\begin{aligned}&\forall v \in U : \\ &\quad v \in V_\alpha \implies \forall (v, w) \in E : w \in U \\ &\quad \wedge \\ &\quad v \in V_{\overline{\alpha}} \implies \exists (v, w) \in E : w \in U\end{aligned}$$

*If the token is in $\alpha$-trap $U$ then player $\overline{\alpha}$ can play a strategy such that the token always remains in $U$.*

**Definition 1.9.** *A set $U \subseteq V$ is a $\alpha$-MTrap for MPG $G$ iff:*

$$\begin{aligned}&\forall (c, v) \in U : \\ &\quad v \in V_\alpha \implies \forall ((c, v), (c, w)) \in E : (c, w) \in U \\ &\quad \wedge \\ &\quad v \in V_{\overline{\alpha}} \implies \exists ((c, v), (c, w)) \in E : (c, w) \in U\end{aligned}$$

**Lemma 1.1.** *If $(c, v) \in \alpha{-}MAttr(G, U)$ then there exists a $(c, w) \in U$.*

**Lemma 1.2.** *Given MPG $G = (V, V_0, V_1, E, \Omega)$. It holds that $U \subseteq V$ is an $\alpha$-MTrap iff for all $c \in con(V)$ the set $cU$ is an $\alpha$-trap in the game $cG$.*

**Lemma 1.3.** *The set $U = V \backslash \alpha{-}MAttr(G, X)$ is an $\alpha$-MTrap in $G$.*

**Lemma 1.4.** *If $V \backslash U$ is an $\alpha$-MTrap then $G \backslash U$ is an MPG.*

**Lemma 1.5.** *Let $X \subseteq V$ be an $\alpha$-MTrap in $G$. Then $\overline{\alpha}{-}MAttr(G, X)$ is also an $\alpha$-MTrap in $G$.*

**Lemma 1.6.** *Given MPG $G = (V, V_0, V_1, E, \Omega)$ and $X \subseteq V$. If $V \backslash X$ is an $\alpha$-MTrap in $G$ and $X' \subseteq V \backslash X$ is an $\alpha$-MTrap in $G \backslash^M X$ then $X'$ is an $\alpha$-MTrap in $G$.*

**Algorithm 1** RecursiveMPG($MPG\ G = (V, V_0, V_1, E, \Omega)$)

1: $m \leftarrow \min\{\Omega(v) \mid (c, v) \in V\}$
2: $h \leftarrow \max\{\Omega(v) \mid (c, v) \in V\}$
3: **if** $h = m$ or $V = \emptyset$ **then**
4:     **if** $h$ is even or $V = \emptyset$ **then**
5:         **return** $(V, \emptyset)$
6:     **else**
7:         **return** $(\emptyset, V)$
8:     **end if**
9: **end if**
10: $\alpha \leftarrow 0$ if $h$ is even and 1 otherwise
11: $U \leftarrow \{(c, v) \in V \mid \Omega(v) = h\}$
12: $A \leftarrow \alpha\text{-}MAttr(G, U)$
13: $(W'_0, W'_1) \leftarrow Recursive(G \setminus^M A)$
14: **if** $W'_{\overline{\alpha}} = \emptyset$ **then**
15:     $W_\alpha \leftarrow A \cup W'_\alpha$
16:     $W_{\overline{\alpha}} \leftarrow \emptyset$
17: **else**
18:     $B \leftarrow \overline{\alpha}\text{-}MAttr(G, W'_{\overline{\alpha}})$
19:     $(W''_0, W''_1) \leftarrow Recursive(G \setminus^M B)$
20:     $W_\alpha \leftarrow W''_\alpha$
21:     $W_{\overline{\alpha}} \leftarrow W''_{\overline{\alpha}} \cup B$
22: **end if**
23: **return** $(W_0, W_1)$

**Theorem 1.7.** *Given MPG $G = (V, V_0, V_1, E, \Omega)$ it holds that $(c, v) \in W_\alpha$ resulting* RecursiveMPG($G$) *iff player $\alpha$ has a winning memoryless strategy in $cG$.*

*Proof.* Proof by induction, similar to [1].

**Induction hypothesis (IH):**

For $(W_0, W_1) =$ RecursiveMPG($G = (V, V_0, V_1, E, \Omega)$) we have

1. $W_0 \uplus W_1 = V$,
2. for any $\alpha \in \{0, 1\}$ it holds that $W_\alpha$ is an $\overline{\alpha}$-MTrap in $G$ and
3. for every $c \in con(W_\alpha)$ there is a strategy $\sigma^c_\alpha$ such that $v \in cW_\alpha$ is winning for player $\alpha$ in game $cG$.

We will refer to the parts of the IH as IH1, IH2 and IH3.

**Base** $\max\{\Omega(v) \mid (c, v) \in V\} = \min\{\Omega(v) \mid (c, v) \in V\}$:

There is only one priority, so any infinite play for any configuration is won by the player with the parity of this one priority. So the entire graph is won by one player (proving IH1), it is a $\alpha$-MTrap for any $\alpha \in \{0, 1\}$ and the winner of the graph is not affected by the strategies (proving IH2 and IH3). In line 1-9 of the algorithm this is implemented, so in this case the IH holds.

**Base** $V = \emptyset$:

An empty set is trivially an $\alpha$-MTrap so returning $(\emptyset, \emptyset)$ satisfies the IH. This is implemented in line 1-5 in the algorithm.

**Step:**

Let $\alpha$ be 0 if the highest priority in the graph in even and 1 otherwise. (line 10)

Using line 11 and 12 and lemma 1.3 we get that $V \setminus A$ is an $\alpha$-MTrap in G.

Using lemma 1.4 we find that $G \setminus^M A$ is an MPG. Since $U$ is non-empty $A$ is non-empty and therefore $G \setminus^M A$ is smaller than $G$. Therefore we can apply the IH on it and we find $W'_0$ and $W'_1$. Let the associated strategies be $w^c_0$ and $w^c_1$.

We distinguish two cases:

- $W_{\overline{\alpha}} = \emptyset$:

  We claim that sets $W_\alpha = W'_\alpha \cup A$ and $W_{\overline{\alpha}} = \emptyset$ satisfy the IH. Clearly $W_\alpha = V$, so the winning sets are the entire graph and the empty set which are both an $\alpha$-MTrap and an $\overline{\alpha}$-MTrap trivially (proving IH1 and IH2).

  To prove IH3 we will consider game $cG$ for any $c \in con(W_\alpha) = con(V)$. By showing that player $\alpha$ has a winning strategy from every $cV$ we prove IH3.

  Consider play $\pi$ in game $cG$ and strategy $\sigma^c_\alpha$ that plays towards $cU$ when the token is in $cA \setminus cU$, plays $w^c_\alpha$ when the token is in $cV \setminus cA$ and plays an arbitrary edge when the token is in $cU$.

Since $cA$ is an attractor the token will always reach $cU$ when in $cA$ and $\sigma^c_\alpha$ is played. So the token can only escape $cA$ through $cU$. Consider the token being in $cV\backslash cA$, if player $\overline{\alpha}$ plays to stay in $cV\backslash cA$ then player $\alpha$ wins, since strategy $w^c_\alpha$ is winning for every vertex in $cV\backslash cA$ if the token doesn't escape. If the token is played towards $cA$ by player $\overline{\alpha}$ then the play can eventually return to $cV\backslash cA$ in which case $cU$ is visited or the play can remain inside $cA$ in which case $cU$ is visited infinitely often. So a play can stay in $cV\backslash cA$ in which case player $\alpha$ wins, can play towards and stay in $cA$ in which case player $\alpha$ wins or alternate between the two in which case $cU$ is infinitely often visited and player $\alpha$ wins.

This is implemented in line 14-16 of the algorithm.

- $W_{\overline{\alpha}} \neq \emptyset$:

  Recall that $V\backslash A$ is an $\alpha$-MTrap in G, by IH we know that $W'_{\overline{\alpha}}$ is an $\alpha$-MTrap in $G\backslash^M A$, therefore (using lemma 1.6) $W'_{\overline{\alpha}}$ is an $\alpha$-MTrap in $G$.

  Let $B = \overline{\alpha}{-}MAttr(G, W'_{\overline{\alpha}})$ (line 18). , using lemma 1.5 we know that $B$ is also an $\alpha$-MTrap.

  By lemma 1.4 we find that $G\backslash^M B$ is an MPG. Since $B$ is non-empty the game $G\backslash^M B$ is smaller than the game $G$, therefore we can apply the IH and we find $W''_0$ and $W''_1$. Let the associated strategies be $q^c_0$ and $q^c_1$. Finally let $W_\alpha = W''_\alpha$ and $W_{\overline{\alpha}} = W''_{\overline{\alpha}} \cup B$ (lines 18-21).

  Since $W''_\alpha \uplus W''_{\overline{\alpha}} = V\backslash B$ by IH we have $W_\alpha \uplus W_{\overline{\alpha}} = V$ (proving IH1).

  $V\backslash B$ is an $\overline{\alpha}$-MTrap in G by lemma 1.3.

  $W''_\alpha$ is an $\overline{\alpha}$-MTrap in $G\backslash^M B$ by IH.

  $W_\alpha = W''_\alpha$ is an $\overline{\alpha}$-MTrap in $G$ because it is an $\overline{\alpha}$-MTrap in $G\backslash^M B$ and $V\backslash B$ is an $\overline{\alpha}$-MTrap in G (using lemma 1.6). (proving IH2 for $\alpha$)

  $W''_{\overline{\alpha}}$ is an $\alpha$-MTrap in $G\backslash^M B$ by IH.

  So $\overline{\alpha}$ has a strategy such that the token can not go from $W''_{\overline{\alpha}}$ to $W''_\alpha$ directly.

  Player $\overline{\alpha}$ can make sure $W''_{\overline{\alpha}}$ can only be left by going to $B$. In $B$ player $\overline{\alpha}$ has a strategy to stay in $B$ so player $\overline{\alpha}$ can force the token to stay inside $W_{\overline{\alpha}} = W''_{\overline{\alpha}} \cup B$, hence it is an $\alpha$-MTrap (proving IH2 for $\overline{\alpha}$).

  What is left to show is that for game $cG$ for any $c \in con(W_\beta)$ player $\beta$ has a winning strategy for $cW_\beta$ with $\beta \in \{0,1\}$.

  Consider game $cG$, let the token be on vertex $v \in cV$. We distinguish three cases:

  - If $v \in cW'_{\overline{\alpha}}$. We know that $cW'_{\overline{\alpha}}$ is an $\alpha$-trap in $cG$ because $V\backslash A$ is also an $\alpha$-trap and player $\overline{\alpha}$ has a winning strategy for every $v$.
  - If $v \in cB\backslash cW'_{\overline{\alpha}}$. Player $\overline{\alpha}$ has a strategy such that the token eventually ends up in $cW'_{\overline{\alpha}}$ and therefore player $\overline{\alpha}$ wins.
  - If $v \in cW''_{\overline{\alpha}}$. As shown above player $\overline{\alpha}$ has a strategy such that the token remains in $cW''_{\overline{\alpha}}$ or goes to $cB$. In the latter case player $\overline{\alpha}$ wins as shown above. In the first case player $\overline{\alpha}$ also wins by playing strategy $q^c_{\overline{\alpha}}$.
  - if $v \in cW''_\alpha$. Since $cW''_\alpha$ is an $\overline{\alpha}$-trap in $cG$, player $\alpha$ can play strategy $q^c_\alpha$ such that the token remains in $cW''_\alpha$ and player $\alpha$ wins.

  This proves IH3.

$\square$

**Theorem 1.8.** *Given VPG $G = (V, V_0, V_1, E, \Omega, \mathfrak{C}, \theta)$ with winning sets $Q^c_0$ and $Q^c_1$. It holds that $v \in Q^c_\alpha$ iff $(c, v) \in W_\alpha$ resulting from* RecursiveMPG$(V', V'_0, V'_1, E', \Omega)$ *where:*

- $V' = \mathfrak{C} \times V$,
- $V'_0 = \mathfrak{C} \times V_0$,
- $V'_1 = \mathfrak{C} \times V_1$,
- $E' = \{((c, v), (c, w)) \mid (v, v') \in E \wedge c \in \theta(v, w)\}$

# 2 Recursive algorithm for RPGs

A relaxed VPG (RPG) is a VPG that is not necessarily total.

$\alpha\text{-}RAttr\colon RPG \to \mathcal{P}(\mathfrak{C} \times V) \to \mathcal{P}(\mathfrak{C} \times V)$

$$\begin{aligned}\alpha\text{-}RAttr(G, U) = \mu A.U &\\ &\cup \{(c, v) \in \mathfrak{C} \times V_\alpha \mid \exists v' \in V : (c, v') \in A \wedge (v, v') \in E \wedge c \in \theta(v, v')\} \\ &\cup \{(c, v) \in \mathfrak{C} \times V_{\overline{\alpha}} \mid \forall v' \in V : (v, v') \in E \wedge c \in \theta(v, v') \implies (c, v') \in A\}\end{aligned}$$

$\setminus^R : RPG \to \mathcal{P}(\mathfrak{C} \times V) \to RPG$
$(V, V_0, V_1, E, \Omega, \mathfrak{C}, \theta) \setminus^R CV = (V', V_0', V_1', E', \Omega, \mathfrak{C}, \theta')$ such that:
$\theta'(u, v) = \theta(u, v) \setminus \bigcup\{c \mid (c, w) \in CV \wedge (u = w \vee v = w)\}$
$E' = \{e \in E \mid \theta'(e) \neq \emptyset\}$
$V' = \{v \in V \mid \exists_w (v, w) \in E' \vee (w, v) \in E'\}$
$V_0' = V_0 \cap V'$
$V_1' = V_1 \cap V'$

**Algorithm 2** RECURSIVERPG($RPG\ G = (V, V_0, V_1, E, \Omega, \mathfrak{C}, \theta)$, $X \subseteq \mathfrak{C} \times V$

$m \leftarrow \min\{\Omega(v) \mid v \in V\}$
$h \leftarrow \max\{\Omega(v) \mid v \in V\}$
**if** $h = m$ or $V = \emptyset$ **then**
  **if** $h$ is even or $V = \emptyset$ **then**
    **return** $(V, \emptyset)$
  **else**
    **return** $(\emptyset, V)$
  **end if**
**end if**
$\alpha \leftarrow 0$ if $h$ is even and 1 otherwise
$U \leftarrow (\mathfrak{C} \times \{v \in V \mid \Omega(v) = h\}) \setminus X$
$A \leftarrow \alpha\text{-}RAttr(G, U)$
$(W_0', W_1') \leftarrow Recursive(G \setminus^R A, A \cup X)$
**if** $W_{\overline{\alpha}}' = \emptyset$ **then**
  $W_\alpha \leftarrow A \cup W_\alpha'$
  $W_{\overline{\alpha}} \leftarrow \emptyset$
**else**
  $B \leftarrow \overline{\alpha}\text{-}RAttr(G, W_{\overline{\alpha}}')$
  $(W_0, W_1) \leftarrow Recursive(G \setminus^R B, B \cup X)$
  $W_{\overline{\alpha}} \leftarrow W_{\overline{\alpha}} \cup B$
**end if**
**return** $(W_0, W_1)$

**Definition 2.1.** *RPG* $R = (V, V_0, V_1, E, \Omega, \mathfrak{C}, \theta)$ *with exclusion set* $X \subseteq \mathfrak{C} \times V$ *and MPG* $M = (V', V_0', V_1', E', \Omega')$ *are* $X$*-equivalent iff:*

- $V' = (\mathfrak{C} \times V) \setminus X$,
- $V_0' = (\mathfrak{C} \times V_0) \setminus X$,
- $V_1' = (\mathfrak{C} \times V_1) \setminus X$,
- $((c, v), (c, w)) \in E'$ *iff* $(v, w) \in E$ *and* $c \in \theta(v, w)$,
- $\Omega = \Omega'$

**Lemma 2.1.** *If RPG $R$ with exclusion set $X$ and MPG $M$ are $X$-equivalent then* $\textsc{RecursiveRPG}(R, X) = \textsc{RecursiveMPG}(M)$.

**Theorem 2.2.** *Given VPG $G = (V, V_0, V_1, E, \Omega, \mathfrak{C}, \theta)$ with winning sets $Q_0^c$ and $Q_1^c$. It holds that $v \in Q_\alpha^c$ iff $(c, v) \in W_\alpha$ resulting from* $\textsc{RecursiveRPG}(G, \emptyset)$.

# References


[1] W. Zielonka, "Infinite games on finitely coloured graphs with applications to automata on infinite trees," *Theoretical Computer Science*, vol. 200, no. 1, pp. 135 – 183, 1998.