Given parity game $G = (V, V_0, V_1, E, \Omega)$ and pre-solved vertices $P_0 \subseteq V$ and $P_1 \subseteq V$ such that $P_0 \subseteq W_0 \subseteq V \backslash P_1$. The formula

$$S(G) = \nu Z_{d-1}.\mu Z_{d-2} \dots \nu Z_0.F_0(G, Z_{d-1}, \dots, Z_0)$$

$$F_0(G, Z_{d-1}, \dots, Z_0) = \{v \in V_0 \mid \exists_{w \in V}\ (v, w) \in E \land w \in Z_{\Omega(w)}\} \cup \{v \in V_1 \mid \forall_{w \in V}\ (v, w) \in E \implies w \in Z_{\Omega(w)}\}$$

solves W_0 for G.

In this section we prove that formula

$$S^P(G) = \nu Z_{d-1}.\mu Z_{d-2} \dots \nu Z_0.(F_0(G, Z_{d-1}, \dots, Z_0) \cap (V \backslash P_1) \cup P_0)$$

also solves W_0 for G. Note that the formula $F_0(G, Z_{d-1}, \dots, Z_0) \cap (V \backslash P_1) \cup P_0$ is still monotonic as shown in Lemma 0.1.

Lemma 0.1. *Given lattice $\langle 2^D, \subseteq \rangle$, monotonic function $f : 2^D \to 2^D$ and $A \subseteq D$. The functions $f^{\cup}(x) = f(x) \cup A$ and $f^{\cap}(x) = f(x) \cap A$ are also monotonic.*

Proof. Let $x, y \subseteq D$ and $x \subseteq y$ then $f(x) \subseteq f(y)$.

Let $e \in f(x) \cup A$. If $e \in f(x)$ then $e \in f(y)$ and $e \in f(y) \cup A$. If $e \in A$ then $e \in f(y) \cup A$. We find $f^{\cup}(x) \subseteq f^{\cup}(y)$.

Let $e \in f(x) \cap A$. We have $e \in f(x)$ and $e \in A$. Therefore $e \in f(y)$ and $e \in f(y) \cap A$. We find $f^{\cap}(x) \subseteq f^{\cap}(y)$. □

Fixed-point iteration index We introduce the notion of fixed-point *iteration index* to help with the proof.

Consider alternating fixed-point formula

$$\nu X_m.\mu X_{m-1} \dots .\nu X_0.f(X_m, X_{m-1}, \dots, X_0)$$

Using fixed-point iteration to solve this formula results in a number of intermediate values for the iteration variables $X_m, \dots X_0$. We define an iteration index that, intuitively, indicates where in the iteration process we are. For an alternating fixed-point formula with m fixed-point variables we define an iteration index $\zeta \subseteq \mathbb{N}^m$.

When applying iteration to formula $\nu X_j.f(X)$ we start with some value for X_j^0 and calculate $X_j^{i+1} = f(X_j^i)$. So we get a list of values for X_j, however when we have alternating fixed-point formula's we might iterate X_j multiple times but get different lists of values because the values for $X_m, \dots, X_{j-1}$ are different. We use the iteration index to distinguish between these different lists.

Iteration index $\zeta = (k_m, \dots, k_0)$ indicates where in the iteration process we are. We start at $\zeta = (0, 0, \dots, 0)$. We first iterate X_0, when we calculate X_0^1 we are at iteration index $\zeta = (0, 0, \dots, 1)$, when we calculate X_0^2 we are at iteration index $\zeta = (0, 0, \dots, 2)$ and so on. In general when we calculate a value for X_j^i then $k_j = i$ in ζ. This gives a natural order of indexes:

$$\begin{array}{c}
(0, \dots, 0, 0, 0) \\
(0, \dots, 0, 0, 1) \\
(0, \dots, 0, 0, 2) \\
\vdots \\
(0, \dots, 0, 1, 0) \\
(0, \dots, 0, 1, 1) \\
(0, \dots, 0, 1, 2) \\
\vdots
\end{array}$$

Formally we have $(k_{d-1}, \dots, k_0) < (j_{d-1}, \dots, j_0)$ if and only if for the largest $l \leq d-1$ such that $k_l \neq j_l$ we have $k_l < j_l$. We define $\{k_{d-1}, \dots, k_0\} - 1 = \{k_{d-1}, \dots, k_0 - 1\}$ and $\{k_{d-1}, \dots, k_0\} + 1 = \{k_{d-1}, \dots, k_0 + 1\}$ for convenience of notation.

We write X_j^ζ to indicate the value of variable X_j at moment ζ of the iteration process. Variable X_j doesn't change values when a variable X_l with $j > l$ changes values, there we have for indexes $\zeta = (k_m, \ldots, k_j, k_{j-1}, \ldots, k_0)$ and $\zeta' = (k_m, \ldots, k_j, k'_{j-1}, \ldots, k'_0)$ that $X_j^\zeta = X_j^{\zeta'}$.

We can use the fixed-point iteration definition to define the values for X_j^ζ. Let $\zeta = (k_m, \ldots, k_0)$, we have:

$$X_0^{\zeta+1} = f(X_m^\zeta, X_{m-1}^\zeta, \ldots, X_0^\zeta)$$

and for any even $0 < j \leq m$

$$X_j^{(\ldots,k_j+1,\ldots)} = \mu X_{j-1} \cdots = \bigcup_{i \geq 0} X^{(\ldots,k_j,i,\ldots)}$$

and for any odd $0 < j \leq m$

$$X_j^{(\ldots,k_j+1,\ldots)} = \nu X_{j-1} \cdots = \bigcap_{i \geq 0} X^{(\ldots,k_j,i,\ldots)}$$

Γ-game We define Γ, which transforms a parity game, to help with the proof.

Given parity game $G = (V, V_0, V_1, E, \Omega)$ with winning set W_0 such that $P_0 \subseteq W_0 \subseteq V \backslash P_1$. We define $\Gamma(G, P_0, P_1) = (V', V'_0, V'_1, E', \Omega')$ such that

$$\begin{aligned}
V' &= (V \backslash P_0 \backslash P_1) \cup \{s_0, s_1\} \\
V'_0 &= (V_0 \cap V') \cup \{s_1\} \\
V'_1 &= (V_1 \cap V') \cup \{s_0\} \\
E' &= (E \cap (V' \times V')) \cup \{(v, s_\alpha) \mid (v, w) \in E \wedge w \in P_\alpha\} \\
\Omega'(v) &= \begin{cases} 0 & \text{if } v \in \{s_0, s_1\} \\ \Omega(v) & \text{otherwise} \end{cases}
\end{aligned}$$

Parity game $\Gamma(G, P_0, P_1)$ contains vertices s_0 and s_1 such that they have no outgoing edges and s_α is owned by player $s_{\overline{\alpha}}$. Clearly if the token ends in s_α then player α wins. Vertices that had edges to a vertex in P_α now have an edge to s_α.

Note that this parity is not total, as shown in [Monadic second-order logic on tree-like structures by Igor Walukiewicz] the formula $S(G)$ also solves non-total games.

Lemma 0.2. *Given parity game $G = (V, V_0, V_1, E, \Omega)$ with winning set W_0 such that $P_0 \subseteq W_0 \subseteq V \backslash P_1$ and parity game $G' = \Gamma(G, P_0, P_1)$ with winning set Q_0 we have $W_0 \cap (V \backslash P_0 \backslash P_1) = Q_0 \cap (V \backslash P_0 \backslash P_1)$.*

Proof. Let vertex $v \in V \backslash P_0 \backslash P_1$. Assume v is won by player α in G using strategy $\sigma_\alpha : V_\alpha \to V$. We construct a strategy $\sigma'_\alpha : V'_\alpha \to V'$ for game G' as follows:

$$\sigma'_\alpha(w) = \begin{cases} s_\beta & \text{if } \sigma_\alpha(w) \in P_\beta \text{ for some } \beta \in \{0, 1\} \\ \sigma_\alpha(w) & \text{otherwise} \end{cases}$$

This strategy maps the vertices to the same successors except when a vertex is mapped to a vertex in P_β, in which case σ'_α maps the vertex to s_β.

Let π' be a valid path in G', starting from v and conforming to σ'_α. Since vertices s_0 and s_1 don't have any successors we distinguish three cases for π':

- Assume π' ends in $s_{\overline{\alpha}}$. Let $\pi' = (x_0 \ldots x_m s_{\overline{\alpha}})$ with $v = x_0$. Because s_0 and s_1 don't have successors we find $x_i \in V \backslash P_0 \backslash P_1$. Moreover for every $x_i x_{i+1}$ we have $(x_i, x_{i+1}) \in E'$, any such edge is also in E because the edges between vertices in $V \backslash P_0 \backslash P_1$ were left intact when creating G'. Finally we find that $(x_m, y) \in E$ with $y \in P_{\overline{\alpha}}$. There must exist a valid path $\pi = (x_0 \ldots x_m y \ldots)$ in game G. Moreover this path conforms to σ_α because σ'_α and σ_α map to the same vertices for all $x_0 \ldots x_{m-1}$ and x_m maps to a vertex in $P_{\overline{\alpha}}$. Player $\overline{\alpha}$ has a winning strategy from y so we conclude that π is won by $\overline{\alpha}$ in game G. Because π exists and conforms to σ_α we find that σ_α is not winning for α from v in G. This is a contradiction so we conclude that π' never ends in $s_{\overline{\alpha}}$.

- Assume π' ends in s_α. In this case player α wins the path.

- Assume π' never visits s_α or $s_{\overline{\alpha}}$. Assume the path is on by player $\overline{\alpha}$, as we argued above we find that this path is also valid in game G, conforms to σ_α and is winning for $\overline{\alpha}$. Therefore σ_α is not winning for player α from v, this is a contradiction so we conclude that player α wins the path π'.

We find that π' is always won by player α in game G'. We conclude that any vertex $v \in V\backslash P_0\backslash P_1$ has the same winner in game G as in game G'. □

Proof

Theorem 0.3. *Given parity game $G = (V, V_0, V_1, E, \Omega)$ with winning set W_0 such that $P_0 \subseteq W_0 \subseteq V\backslash P_1$. The formula*

$$S^P(G) = \nu Z_{d-1}.\mu Z_{d-2} \ldots \nu Z_0.(G, F_0(Z_{d-1}, \ldots, Z_0) \cap (V\backslash P_1) \cup P_0)$$

correctly solves W_0 for G.

Proof. Let $G' = \Gamma(G, P_0, P_1)$. We consider $S(G')$, which calculates the winning set for player 0 for game G'. Formula $F_0(G', Z_{d-1}, \ldots, Z_0)$ will always include s_0 and never include s_1 regardless of the values for $Z_{d-1} \ldots Z_0$. Clearly any $\nu Z_i \ldots$ or $\mu Z_i \ldots$ contains s_0 and doesn't contain s_1. So we can calculate $S(G')$ using fixed-point iteration starting greatest fixed-point variables at $V'\backslash\{s_1\}$ and least fixed-point variables at $\{s_0\}$.

We can also calculate $S^P(G)$ using fixed-point iteration starting at P_0 and $V\backslash P_1$ because clearly any $\nu Z_i \ldots$ or $\mu Z_i \ldots$ contains all vertices from P_0 and none from P_1.

We will go through the iteration of formula's $S^P(G)$ and $S(G')$ using iteration index ζ to indicate where in the iteration we are. We write Z_i to denote variables in $S(G')$ and Y_i to denote variables in $S^P(G)$.

Trivially, for any ζ and i we have $P_0 \subseteq Y_i^\zeta \subseteq V\backslash P_1$ and $\{s_0\} \subseteq Z_i^\zeta \subseteq V\backslash\{s_1\}$

We define operator $\simeq: V \times V' \to \mathbb{B}$ such that for $Y \subseteq V$ and $Z \subseteq V'$ we have $Y \simeq Z$ if and only if:

$$Y\backslash P_0\backslash P_1 = Z\backslash\{s_0, s_1\}$$

We will prove that for any $\zeta = (k_{d-1}, \ldots, k_0)$ we have $Y_i^\zeta \simeq Z_i^\zeta$ for every $i \in [0, d-1]$.

Proof by induction on ζ.

Base $\zeta = (0, 0, \ldots, 0)$: we have for least fixed-point variables Z_i^ζ and Y_i^ζ the values $\{s_0\}$ and P_0, clearly $Z_i^\zeta \simeq Y_i^\zeta$.

For greatest fixed-point variables Z_j^ζ and Y_j^ζ we have $Z_j^\zeta\backslash\{s_0, s_1\} = V\backslash P_1\backslash P_0$. So we find $Z_j^\zeta \simeq Y_j^\zeta$.

Step: Consider $\zeta = (k_{d-1}, \ldots, k_0)$. Let $0 \leq j \leq d-1$. If $k_j = 0$ then $Z_j^\zeta = Z_j^{(0,0,\ldots,0)}$ and $Y_j^\zeta = Y_j^{(0,0,\ldots,0)}$, furthermore $Z_j^{(0,0,\ldots,0)} \simeq Y_j^{(0,0,\ldots,0)}$ so we find $Z_j^\zeta \simeq Y_j^\zeta$. If $k_j > 0$ then we distinguish three cases for j to show that $Z_j^\zeta \simeq Y_j^\zeta$:

- Case: $j = 0$. We have the following equations:

$$Y_0^\zeta = F_0(G, \tilde{Z}_{d-1}^{\zeta-1}, \ldots, Y_0^{\zeta-1}) \cap (V\backslash P_1) \cup P_0$$

  and

$$Z_0^\zeta = F_0(G', Z_{d-1}^{\zeta-1}, \ldots, Z_0^{\zeta-1})$$

  Consider vertex $v \in V\backslash P_0\backslash P_1$. We distinguish two cases:

  - Assume $v \in V_0$.
    If $v \in Y_0^\zeta$ then v must have an edge in game G to w such that $w \in Y_{\Omega(w)}^{\zeta-1}$. We find $w \notin P_1$ because vertices from P_1 are never in the iteration variable. If $w \in P_0$ then it follows from the way we created G' that in G' there exists an edge from v to s_0 and since s_0 is always in the iteration variable we find $v \in Z_0^\zeta$. If $w \notin P_0$ then because $Y_{\Omega(w)}^{\zeta-1} \simeq Z_{\Omega(w)}^{\zeta-1}$ we find $w \in Z_{\Omega(w)}^{\zeta-1}$ and therefore $v \in Z_0^\zeta$.

If $v \in Z_0^\zeta$ then v must have an edge in game G' to w such that $w \in Z_{\Omega(w)}^{\zeta-1}$. We find $w \neq s_1$ because w is never in the iteration variable. If $w = s_0$ then it follows from the way we created G' that in G there exists an edge from v to a vertex in P_0 and since any vertex in P_0 is always in the iteration variable we find $v \in Y_0^\zeta$. If $w \neq s_0$ then because $Y_{\Omega(w)}^{\zeta-1} \simeq Z_{\Omega(w)}^{\zeta-1}$ we find $w \in Y_{\Omega(w)}^{\zeta-1}$ and therefore $v \in Y_0^\zeta$.

– Assume $v \in V_1$.

If $v \in Y_0^\zeta$ then for any successor w of v in game G it holds that $w \in Y_{\Omega(w)}^{\zeta-1}$. Consider successor x of v in game G'. We distinguish three cases:

  * $x = s_0$: In this case $x \in Z_{\Omega(x)}^{\zeta-1}$ because s_0 is always in the iteration variables.
  * $x = s_1$: Because of the way G' is constructed we find vertex v must have a successor w in P_1, however we found $w \in Y_{\Omega(w)}^{\zeta-1}$. This is a contradiction because vertices in P_1 are never in the iteration variables. So this case can not happen.
  * $x \notin \{s_0, s_1\}$: We have $x \in V' \backslash \{s_0, s_1\}$ and therefore x is also a successor of v in game G. We find $x \in Y_{\Omega(x)}^{\zeta-1}$ and because $Y_{\Omega(x)}^{\zeta-1} \simeq Z_{\Omega(x)}^{\zeta-1}$ we have $x \in Z_{\Omega(x)}^{\zeta-1}$.

We always find $x \in Z_{\Omega(x)}^{\zeta-1}$, therefore $v \in Z_0^\zeta$.

If $v \in Z_0^\zeta$ then for any successor w of v in game G' it holds that $w \in Z_{\Omega(w)}^{\zeta-1}$. Consider successor x of v in game G. We distinguish three cases:

  * $x \in P_0$: In this case $x \in Y_{\Omega(x)}^{\zeta-1}$ because vertices in P_0 are always in the iteration variables.
  * $x \in P_1$: Because of the way G' is constructed we find vertex v must have successor s_1 in game G', however we found that for any successor w of v in game G' we have $w \in Z_{\Omega(w)}^{\zeta-1}$. This is a contradiction because s_1 is never in the iteration variable. So this case can not happen.
  * $x \in V \backslash P_0 \backslash P_1$: We find that x is also a successor of v in game G'. We find $x \in Z_{\Omega(w)}^{\zeta-1}$ and because $Y_{\Omega(x)}^{\zeta-1} \simeq Z_{\Omega(x)}^{\zeta}$ we have $x \in Y_{\Omega(x)}^{\zeta}$.

We always find $x \in Y_{\Omega(x)}^{\zeta-1}$, therefore $v \in Y_0^\zeta$.

- Case: $j > 0$ being even. We have

$$Z_j^\zeta = \mu Z_{j-1} \cdots = \bigcup_{i \geq 0} Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$$

and

$$Y_j^\zeta = \mu Y_{j-1} \cdots = \bigcup_{i \geq 0} Y_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$$

Let $v \in V \backslash P_0 \backslash P_1$.

If $v \in Z_j^\zeta$ then there exists some i such that $v \in Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$. Since $\{k_{d-1},\ldots,k_j-1,i,\ldots\} < \zeta$ we apply induction to find $Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}} \simeq Y_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$. Because $v \in V \backslash P_0 \backslash P_1$ we find $v \in Y_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$ and therefore $v \in Y_j^\zeta$.

If $v \in Y_j^\zeta$ then we apply symmetrical reasoning to find $v \in Z_j^\zeta$.

- Case: $j > 0$ being odd. We have

$$Z_j^\zeta = \nu Z_{j-1} \cdots = \bigcap_{i \geq 0} Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$$

and

$$Y_j^\zeta = \nu Y_{j-1} \cdots = \bigcap_{i \geq 0} Y_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$$

Let $v \in V \backslash P_0 \backslash P_1$.

If $v \in Z_j^\zeta$ then for all $i \geq 0$ we have $v \in Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$. Assume $v \notin Y_j^\zeta$, there must exist an $l \geq 0$ such that $v \notin Y_j^{\{k_{d-1},\ldots,k_j-1,l,\ldots\}}$. Since $\{k_{d-1},\ldots,k_j-1,l,\ldots\} < \zeta$ we apply induction to find $Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,l,\ldots\}} \simeq Y_{j-1}^{\{k_{d-1},\ldots,k_j-1,l,\ldots\}}$. Because $v \in V \backslash P_0 \backslash P_1$ we find $v \notin Z_{j-1}^{\{k_{d-1},\ldots,k_j-1,i,\ldots\}}$ which is a contradiction so we have $v \in Y_j^\zeta$.

If $v \in Y_j^\zeta$ then we apply symmetrical reasoning to find $v \in Z_j^\zeta$.

This proves that for any ζ we have $Y_i^\zeta \simeq Z_i^\zeta$ for every $i \in [0, d-1]$.

We have shown that when starting the iteration of $S(G')$ and $S^P(G)$ at specific values then we get a similar results for vertices in $V \backslash P_0 \backslash P_1$. We choose these values such that they solve the formula's correctly, so we conclude that $S(G') \backslash \{s_0, s_1\} = S^P(G) \backslash P_0 \backslash P_1$. Lemma 0.2 shows that $S(G')$ correctly vertices in $V \backslash P_0 \backslash P_1$ for game G. So $S^P(G)$ also correctly solves vertices $V \backslash P_0 \backslash P_1$ for game G.

Moreover any vertex in P_0 is in $S^P(G)$, which is correct because P_0 vertices are winning for player 0. Any vertex in P_1 is not in $S^P(G)$, which is correct because P_1 vertices are winning for player 1. We conclude that all vertices are correctly solved. ☐

This gives the following algorithm where we start iteration at P_0 and $V \backslash P_1$. Moreover we ignore vertices in P_0 or P_1 in the diamond and box calculation, finally we always add vertices in P_0 to the results of the diamond and box operator.

Algorithm 1 Fixed-point iteration with P_0 and P_1

function FPITER($G = (V, V_0, V_1, E, \Omega), P_0, P_1$)
    for $i \leftarrow d-1, \ldots, 0$ **do**
        INIT(i)
    end for
    repeat
        $Z_0' \leftarrow Z_0$
        $Z_0 \leftarrow P_0 \cup$ DIAMOND() $\cup$ BOX()
        $i \leftarrow 0$
        while $Z_i = Z_i' \wedge i < d-1$ **do**
            $i \leftarrow i+1$
            $Z_i' \leftarrow Z_i$
            $Z_i \leftarrow Z_{i-1}$
            INIT($i-1$)
        end while
    until $i = d-1 \wedge Z_{d-1} = Z_{d-1}'$
    return $(Z_{d-1}, V \backslash Z_{d-1})$
end function

function INIT(i)
    $Z_i \leftarrow P_0$ if i is odd, $V \backslash P_1$ otherwise
end function

function DIAMOND
    return $\{v \in V_0 \backslash P_0 \backslash P_1 \mid \exists_{w \in V} (v, w) \in E \wedge w \in Z_{\Omega(w)}\}$
end function

function BOX
    return $\{v \in V_1 \backslash P_0 \backslash P_1 \mid \forall_{w \in V} (v, w) \in E \implies w \in Z_{\Omega(w)}\}$
end function

Note: In the paper [The Fixpoint-Iteration Algorithm for Parity Games], that describes an algorithm to solve $S(G)$, total parity games are used. However the argumentation only relies on the fact that every vertex has a unique owner and priority. So the algorithm can also solve non total games.