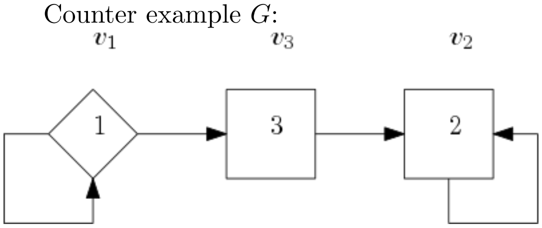
Algorithm 1 RecursivePG($PG\ G = (V, V_0, V_1, E, \Omega)$)

$m \leftarrow \min\{\Omega(v) \mid v \in V\}$
$h \leftarrow \max\{\Omega(v) \mid v \in V\}$
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
$U \leftarrow \{v \in V \mid \Omega(v) = h\}$
$A \leftarrow \alpha\text{-}Attr(G, U)$
$(W_0', W_1') \leftarrow$ RecursivePG$(G \backslash A)$
if $W_{\overline{\alpha}}' = \emptyset$ **then**
    $W_\alpha \leftarrow A \cup W_\alpha'$
    $W_{\overline{\alpha}} \leftarrow \emptyset$
else
    $B \leftarrow \overline{\alpha}\text{-}Attr(G, W_{\overline{\alpha}}')$
    $(W_0'', W_1'') \leftarrow$ RecursivePG$(G \backslash B)$
    $W_\alpha \leftarrow W_\alpha''$
    $W_{\overline{\alpha}} \leftarrow W_{\overline{\alpha}}'' \cup B$
end if
return (W_0, W_1)

Conjecture 0.1. *For any* RecursivePG(G') *that is invoked during* RecursivePG(G) *it holds that any vertex* $v \in W_{\overline{\alpha}}'$ *is won by player* $\overline{\alpha}$ *in game* G.

Counter example G:



All vertices are won by player 0.
RecursivePG(G):
$h = 3, \alpha = 1$
$A = \{v_3\}$
> RecursivePG$(G \backslash A)$:
> $\hat{h} = 2, \hat{\alpha} = 0$
> $\hat{A} = \{v_2\}$
> > RecursivePG$(G \backslash A \backslash \hat{A})$
>
> $\hat{W}_0' = \emptyset$
> $\hat{W}_1' = \{v_1\} = \hat{W}_{\overline{\hat{\alpha}}}'$
> **Vertex v_1 is in $\hat{W}_{\overline{\hat{\alpha}}}'$ however in G the vertex is won by player $\hat{\alpha}$.**
> $\hat{B} = \{v_1\}$
> > RecursivePG$(G \backslash A \backslash \hat{B})$
>
> $\hat{W}_0'' = \{v_2\}, \hat{W}_1'' = \emptyset$

$W_0' = W_{\overline{\alpha}}' = \{v_2\}$
$W_1' = W_\alpha' = \{v_1\}$
$B = V$
$W_0 = V$