Algorithm 1 `br player 2 pruning`

$R \leftarrow \emptyset$
function BRP($\Gamma, \breve{v}, h, u_{\pi_1\pi_{\mathcal{A}}}$) $\triangleright$ $i \in \mathcal{P}$ has perfect recall
  if ($h \in Z$) **then**
    return $u_{\pi_1\pi_{\mathcal{A}}}(h)$
  $LB^*(I_2) \leftarrow -\infty$
  if ($h \in I | P(I) = 2$) **then**
    $I_2 \leftarrow I(h)$
    for $i = 1 : |A_o(h)|$ **do**
      $a \leftarrow A_o(h)_i$
      $flag \leftarrow 0$
      for $\bar{h} \in I_2$ **do**
        if ($LB^*(I_2) > \breve{v}(\bar{h}a)$) **then**
          $flag \leftarrow flag + 1$
      if ($flag = |I_2|$) **then**
        $R \leftarrow R \cup$ PRUNE(Γ, I_2, a)
        continue;
      if ($(I_2, a) \notin E$) **then**
        $LB'(I_2) \leftarrow 0$
        for $\bar{h} \in I_2$ **do**
          $LB'(I_2) \leftarrow LB'(I_2)$+BRP($\Gamma, \breve{v}, \bar{h}a, u_{\pi_1\pi_{\mathcal{A}}}$)
        if ($LB'(I_2) > LB^*(I_2)$) **then**
          $LB^*(I_2) \leftarrow LB'(I_2)$
        $E \leftarrow E \cup \{(I_2, a)\}$
  return R
function PRUNE($\Gamma, \bar{I}, a$)
  $I_{h,a} \leftarrow \{I(ha)\}_{h \in \bar{I}}$
  $\hat{H} \leftarrow \bigcup_{I \in I_{h,a}} \{h \in H_1 | \exists b, (I, b) \in X_{1,2}(h)\}$
  $\bar{R} \leftarrow \emptyset$
  for $I_1 \in \mathcal{I}_1$ **do**
    if ($\forall h \in I_1, h \in \hat{H}$) **then**
      $\bar{R} \leftarrow \bar{R} \cup \{I_1\}$
  return $\bar{R}$

Function PRUNE takes in input game Γ, the information set $\bar{I}$ and an action $a \in A(\bar{I})$; $I_{h,a}$ is the set of information sets that immediately follows from $\bar{I}$ given action a and $\hat{H}$ is the set of all nodes in a subtree having root node in some $I \in I_{h,a}$. The function returns set R that contains all the information set where we can fix an action. An information set can be part of R only if all its nodes belong to $\hat{H}$.
Function BRP takes in input game Γ, the vector containing an upper bound value $\breve{v}$ per information set, a node $h \in H$ and the vector of utilities $u_{\pi_1\pi_A}(z)$, $z \in Z$, that are marginalized with respect to the given behavioral strategies π_1 of player 1 and $\pi_{\mathcal{A}}$ of adversary player.
Function BRP returns R, the set of information sets which can be pruned. Pruning an information set practically means fixing a certain action. Indeed we know that the pruned information set will not be part of the best response, so we can fix a random action there because we will never take that path. Fixing an action simplify the combinatorial problem of choosing plans of player 1.
BRP is a recursive function that reads the marginalized utilities at leaves nodes and then propagate them up as lower bounds. The aim of the team of players $\{1, 2\}$ is maximizing the utility so
Algorithm 2 use two vectors of dimension $|\mathcal{I}_1|$:

- $\mathbf{a}$: the i-th element $\mathbf{a}_i$ is the action currently selected at the i-th information set of player 1. The action is expressed as an index (ID) from 1 to $|A(I_i)|$ following

Algorithm 2 `update`

init $\mathbf{a} \leftarrow \mathbf{1}$
init $\mathbf{p} \leftarrow \mathbf{0}$
$\bar{\pi}_1 \leftarrow$ TOBEHAVIORAL($\mathbf{a}$)
function TOBEHAVIORAL($\mathbf{a}$)
    $\bar{\boldsymbol{\pi}}_{\mathbf{1}} \leftarrow \mathbf{0}_{|\mathcal{I}_1| \times |A(I_1)|}$
    for $x = 1 : |\mathcal{I}_1|, y = 1 : |A(I_1)|$ **do**
        if $\mathbf{a}_x = y$ **then**
            $\bar{\boldsymbol{\pi}}_{\mathbf{1}}(x, y) \leftarrow 1$
    return $\bar{\boldsymbol{\pi}}_{\mathbf{1}}$
function UPDATE(($\mathbf{a}$, $\mathbf{p}$, R))
    $\mathbf{p} \leftarrow$ UPDATEP($\mathbf{p}$, R)
    $\mathbf{a} \leftarrow$ UPDATEA($\mathbf{a}$, $\mathbf{p}$)
    return ($\mathbf{a}$, $\mathbf{p}$)
function UPDATEP($\mathbf{p}$, R)
    for $I_i \in \mathcal{I}_1$ **do**
        if $\mathbf{p}_i \neq 1 \wedge I_i \in R$ **then**
            $\mathbf{p}_i \leftarrow 1$
    return $\mathbf{p}$
function UPDATEA($\mathbf{a}$, $\mathbf{p}$)
    $B \leftarrow \mathcal{I}_1$
    for $i | \mathbf{p}_i = 1$ **do**
        $B \leftarrow B \setminus I_i$
    $\mathbf{a} \leftarrow$ NEXT(B)
    return $\mathbf{a}$

Algorithm 3 `Ordered Actions in` $A_o(I_i)$

function ORDER(Γ, I_i)
    $C \leftarrow I_i \times A(I_i)$
    $j \leftarrow 1$
    for $(h, a) \in I_i \times A(I_i)$ **do**
        if $(h, a) \in C$ **then**
            if $\breve{v}(ha) \geq \breve{v}(\bar{h}\bar{a}) \quad \forall(\bar{h}, \bar{a}) \in I_i \times A(I_i)$ **then**
                $A_o(I_i)_j \leftarrow a$
                $j \leftarrow j + 1$
                for $\bar{h} \in I_i$ **do**
                    $C \leftarrow C \setminus \{(\bar{h}, a)\}$

the order of $A_o(I_i)$ (e.g. $\mathbf{a}_2 = 1$ means that the second element of **a**, which is the action currently selected at the second information set of player one I_2, is the first element of the ordered set of action of I_2 that is $A_o(I_2)_1$). **a** is initialized as a vector of 1s that means that the first path selected is the one with the highest value according to information obtained from the perfect recall refinement.

- **p**: the i-th element $\mathbf{p}_i$ is equal to 1 iff the i-th information set of player 1 has been pruned. This information is useful in order to fix an action there.

Function TOBEHAVIORAL takes in input the vector **a** and create $\bar{\boldsymbol{\pi}}_{\mathbf{1}}$ a matrix $|\mathcal{I}_1| \times |A(I_1)|$. Note that the branching factor is the same for all the information sets of player 1 (i.e. $|A(I_1)| = |A(I_2)| = ... = |A(I_{|\mathcal{I}_1|})|$). The i-th row of the matrix is the pure behavioral strategy followed by player 1 at information set $I_i \in \mathcal{I}_1$. Behavioral strategy at information set I_i is a distribution of probabilities over actions $A(I_i)$; playing a pure strategy means setting $\bar{\boldsymbol{\pi}}_{\mathbf{1}}(i, j) = 1$ if the j-th action of $A_o(I_i)$ (whose ID is $\mathbf{a}(i)$) is chosen and all the other elements $\bar{\boldsymbol{\pi}}_{\mathbf{1}}(i, \bar{j}) = 0 \quad \forall \bar{j} \neq j$.