

Definition 1 (Coarsest Perfect Recall Refinement) *The coarsest perfect recall refinement Γ' of the imperfect recall game $\Gamma = \langle \mathcal{P}, A, H, Z, P, \pi_c, u, \mathcal{I} \rangle$ is a tuple $\langle \mathcal{P}, A', H, Z, P, \pi_c, u, \mathcal{I}' \rangle$ where $\forall i \in \mathcal{P}, \forall I_i \in \mathcal{I}_i, H(I_i)$ defines the information set partition \mathcal{I}' . A' is a modification of A , which guarantees that $\forall I \in \mathcal{I}', \forall h_k, h_l \in I, A'(h_k) = A'(h_l)$, while for all distinct $I^k, I^l \in \mathcal{I}', \forall a^k \in A(I^k), \forall a^l \in A(I^l), a^k \neq a^l$.*

Algorithm 1 Coarsest Perfect Recall Refinement

```

1: function CPRR( $\Gamma, i$ )  $\triangleright i \in \mathcal{P}$  has imperfect recall
2:    $O = \{I \in \mathcal{I}_i \mid \exists h \in I, X_i(h) = \emptyset\}$ 
3:    $\mathcal{I}_i^* \leftarrow \mathcal{I}_i \setminus O$ 
4:    $\mathcal{L}_i \leftarrow \emptyset$ 
5:    $L \leftarrow \emptyset$ 
6:   for  $I \in \mathcal{I}_i^*$  do
7:      $L \leftarrow I$ 
8:     for  $x \in I$  do
9:       if  $x \in L$  then
10:         $I_x = \text{CHECK}(\Gamma, I, x)$ 
11:         $L = L \setminus \{I_x\}$ 
12:         $\mathcal{L}_i = \mathcal{L}_i \cup I_x$ 
13:    $\tilde{\mathcal{I}}_i = O \cup \mathcal{L}_i$ 
14:   return  $\tilde{\mathcal{I}}_i$ 
15: function CHECK( $\Gamma, I, x$ )
16:    $I_x = \{x\}$ 
17:    $P = \{I_i \in \mathcal{I}_i \mid I_i \in X_i(x)\}$ 
18:   for  $\bar{x} \in I \setminus \{x\}$  do
19:      $\bar{P} = \{p \in P \mid \text{seq}_i\{p \rightarrow x\} \neq \emptyset \wedge \text{seq}_i\{p \rightarrow \bar{x}\} \neq \emptyset\}$ 
20:     for  $\bar{p} \in \bar{P}$  do
21:       if  $\text{seq}_i\{\bar{p} \rightarrow x\} = \text{seq}_i\{\bar{p} \rightarrow \bar{x}\}$  then
22:          $I_x = I_x \cup \{\bar{x}\}$ 
23:   return  $I_x$ 

```

The algorithm takes in input a generic extensive-form imperfect recall game and returns its coarsest perfect recall refinement by splitting some information sets of the original game. Consider an information set I with n nodes: if $n = 1$, I will be in the final perfect recall partition $\tilde{\mathcal{I}}_i$ because we cannot further split it. If $n > 1$ the algorithm select each couple (x, \bar{x}) of nodes in I and checks all the common fathers \bar{p} : if the sequence of nodes and actions that bring from \bar{p} to x (i.e. $\text{seq}_i\{\bar{p} \rightarrow x\}$) and the sequence from \bar{p} to \bar{x} (i.e. $\text{seq}_i\{\bar{p} \rightarrow \bar{x}\}$) are the same, we can keep these two nodes in the same information set of the perfect recall partition. The fact that the sequences are identical means that each possible loss of memory of player i in the path from \bar{p} to x and \bar{x} is not due to his own previous action.

In this algorithm we don't have to follow a specific order when we visit the information sets $I \in \mathcal{I}_i^*$. An information set is splitted checking all the superior information sets. Each of these information sets is a potential cause of division so if we check all of them as we do in our algorithm, it is impossible to pass over a possible split. Moreover the split of inferior information sets does not modify the superior ones.

The perfect recall refinement given by this algorithm is the coarsest: in the refinement we assemble the biggest number of nodes possible such that the information set they create is perfect recall.