

1 Explanation Generation

Algorithm 1: CSP-Explain(\mathcal{T} , f [, \mathcal{I}_0])

```

input           :  $\mathcal{T}$  set of constraints
input           :  $f$  a consistent objective function
optional input:  $\mathcal{I}_0$  a partial interpretation
output          : Explanation sequence
1 begin
2    $\mathcal{I}_{end} \leftarrow \text{propagate}(\mathcal{I}_0, \mathcal{T})$  // Goal state
3    $\mathcal{I} \leftarrow \mathcal{I}_0$  // Initial partial interpretation
4    $Seq \leftarrow \emptyset$  // explanation sequence
5   while  $\mathcal{I} \neq \mathcal{I}_{end}$  do
6     for  $i \in \mathcal{I}_{end} \setminus \mathcal{I}$  do
7        $X_i \leftarrow \text{OMUS}(\{\neg i\} \wedge \mathcal{I} \wedge \mathcal{S})$ 
8        $E_i \leftarrow \mathcal{I} \cap X_i$  // Facts used
9        $S_i \leftarrow \mathcal{T} \cap X_i$  // Constraint used
10       $\mathcal{N}_i \leftarrow \text{propagate}(E_i \wedge S_i)$  // Newly derived facts
11    end
12     $(E_{best}, S_{best}, N_{best}) \leftarrow (E_i, S_i, N_i)$  with lowest  $f(E_i, S_i, N_i)$ 
13    append  $(E_{best}, S_{best}, N_{best})$  to  $Seq$ 
14     $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_{best}\}$ 
15  end
16 end

```

Algorithm 2: CSP-Explain-Incremental(\mathcal{T} , f [, \mathcal{I}_0])

```
1  $\mathcal{I}_{end} \leftarrow \text{propagate}(\mathcal{I}_0, \mathcal{T})$ 
2  $\mathcal{I} \leftarrow \mathcal{I}_0$ 
3  $\mathcal{W} \leftarrow \emptyset$  // Collection of MSSes generated during OMUS
4  $Seq \leftarrow \emptyset$ 
5 while  $\mathcal{I} \neq \mathcal{I}_{end}$  do
6   for  $i \in \mathcal{I}_{end} \setminus \mathcal{I}$  do
7      $X_i, \mathcal{MSS} \leftarrow \text{OMUS-DELAYED}(\{\neg i\} \wedge \mathcal{I} \wedge \mathcal{S}, \mathcal{M})$ 
8      $E_i \leftarrow \mathcal{I} \cap X_i$ 
9      $S_i \leftarrow \mathcal{T} \cap X_i$ 
10     $\mathcal{N}_i \leftarrow \text{propagate}(E_i \wedge S_i)$ 
11     $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{MSS}$ 
12  end
13   $(E_{best}, S_{best}, N_{best}) \leftarrow (E_i, S_i, N_i)$  with lowest  $f(E_i, S_i, N_i)$ 
14  append  $(E_{best}, S_{best}, N_{best})$  to  $Seq$ 
15   $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_{best}\}$ 
16 end
```

Algorithm 3: OMUS-Delayed($\mathcal{F}, \mathcal{M}, f_{cost}$)

```
// F = unsatisfiable CNF formula; M = Collection of MSSes;  $f_{cost}$  = cost function
1  $\mathcal{K} \leftarrow \emptyset$ 
  // grow mss from input mss
2 foreach  $MSS \in \mathcal{M}$  do
3    $MSS' \leftarrow \text{Grow}(\mathcal{F} \cap MSS)$ 
4    $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{F} \setminus MSS'\}$ 
5 end
6  $\text{mode} \leftarrow \text{mode\_greedy}$ 
7 while true do
8   while true do
9     switch  $\text{nonOptLevel}$  do
10      case  $\text{mode\_incr}$ 
11         $hs \leftarrow \text{FindIncrementalHittingSet}(\mathcal{K}, \mathcal{C}, hs)$ 
12      case  $\text{mode\_greedy}$ 
13         $hs \leftarrow \text{FindGreedyHittingSet}(\mathcal{K})$ 
14      end
15       $(\text{sat?}, \kappa) \leftarrow \text{SatSolver}(hs)$ 
16      if  $\text{not sat?}$  then
17        switch  $\text{nonOptLevel}$  do
18          case  $\text{mode\_incr}$ 
19             $\text{mode} \leftarrow \text{mode\_greedy}$ 
20          case  $\text{mode\_greedy}$ 
21             $\text{mode} \leftarrow \text{mode\_opt}$ 
22            break
23          end
24        else
25           $MSS \leftarrow \text{Grow}(hs)$ 
26           $\mathcal{M} \leftarrow \mathcal{M} \cup \{MSS\}$ 
27           $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{F} \setminus MSS\}$ 
28           $\text{mode} \leftarrow \text{mode\_incr}$ 
29        end
30       $hs \leftarrow \text{OptimalHittingSet}(\mathcal{K}, f_{cost})$ 
31       $(\text{sat?}, \kappa) \leftarrow \text{SatSolver}(hs)$ 
32      if  $\text{not sat?}$  then
33        return  $hs, \mathcal{M}$ 
34      end
35       $MSS \leftarrow \text{Grow}(hs)$ 
36       $\mathcal{M} \leftarrow \mathcal{M} \cup \{MSS\}$ 
37       $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{F} \setminus MSS\}$ 
38       $\text{mode} \leftarrow \text{mode\_incr}$ 
39 end
```

Algorithm 4: OMUS-Delayed($\mathcal{F}, \mathcal{M}, f_{cost}$)

```
// F = unsatisfiable CNF formula; M = Collection of MSSes;  $f_{cost}$  = cost function
1  $\mathcal{K} \leftarrow \emptyset$ 
  // grow mss from input mss
2 foreach  $MSS \in \mathcal{M}$  do
3    $MSS' \leftarrow \text{Grow}(\mathcal{F} \cap MSS)$ 
4    $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{F} \setminus MSS'\}$ 
5 end
6  $mode \leftarrow mode\_greedy$ 
7 while true do
8   switch nonOptLevel do
9     case mode\_incr
10       $MCS \leftarrow \{\mathcal{F} \setminus MSS\}$ 
11       $hs \leftarrow \text{FindIncrementalHittingSet}(\mathcal{K}, MCS, hs)$ 
12     case mode\_greedy
13       $hs \leftarrow \text{FindGreedyHittingSet}(\mathcal{K})$ 
14     case mode\_opt
15       $hs \leftarrow \text{OptimalHittingSet}(\mathcal{K}, \mathcal{W})$ 
16   end
17    $(sat?, \kappa) \leftarrow \text{SatSolver}(hs)$ 
18   if not sat? then
19     switch nonOptLevel do
20       case mode\_incr
21         $mode \leftarrow mode\_greedy$ 
22       case mode\_greedy
23         $mode \leftarrow mode\_opt$ 
24       break
25       case mode\_opt
26        return  $hs, \mathcal{M}$ 
27     end
28   else
29      $MSS \leftarrow \text{Grow}(hs)$ 
30      $\mathcal{M} \leftarrow \mathcal{M} \cup \{MSS\}$ 
31      $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{F} \setminus MSS\}$ 
32      $mode \leftarrow mode\_incr$ 
33 end
```
