Algorithm 1: greedy-explanations(I_0, T_p)

$I_{end} \leftarrow$ propagate($I_0 \wedge T_p$);
Seq $\leftarrow$ empty sequence;
$I \leftarrow I_0$;
while $I \neq I_{end}$ **do**
    $(E, S, N) \leftarrow$min-explanation(I, T_p);
    $nested \leftarrow$ nested-explanation(E, S, N, T_p);
    if $nested \neq \emptyset$ **then**
        append $nested$ to Seq;
    else
        append (E, S, N) to Seq;
    end
    $I \leftarrow I \cup N$;
end

Algorithm 2: nested-explanation(E, S, N, T_p)

$nested\_explanations \leftarrow \{\}$;
$step\_cost \leftarrow f(E, S, N)$;
for $n_i \in N$ **do**
    expensive $\leftarrow false$;
    nested_seq $\leftarrow \{\}$;
    $I' \leftarrow E \wedge \neg\{n_i\}$;
    while $consistent(I') \wedge \neg expensive$ **do**
        $(E', S', N') \leftarrow$min-explanation(I', S);
        append (E', S', N') to nested_seq;
        $I' \leftarrow I' \cup N'$;
        if $f(E', S', N') \leq step\_cost$ **then**
            expensive $\leftarrow true$;
        end
    end
    if $\neg expensive$ **then**
        append $nested\_seq$ to $nested\_explanations$;
    end
end
return $nested\_explanations$

Algorithm 3: candidate-explanations(I, C)

```
input  : A partial interpretation I and a set of constraints C
Candidates ← {};
J ← propagate(I ∧ C);
for a ∈ J \ I do
    // Minimal expl.  of each new fact:
    X ← MUS(¬a ∧ I ∧ C) ;
    E ← I ∩ X;                                  // facts used
    S ← C ∩ X;                                  // constraints used
    A ← propagate(E ∧ S);                       // all implied facts
    add (E, S, A) to Candidates
end
return Candidates
```

Algorithm 4: min-explanation(I, C)

```
input  : A partial interpretation I and a set of constraints C
best ← none;
for S ⊆ C ordered ascending by g(S) do
    if best ≠ none ∧ g(S) > f(best) then
        break;
    end
    cand ← best explanation from candidate-explanations(I, S);
    if best = none ∨ f(best) > f(cand) then
        best ← cand;
    end
end
return best
```

Algorithm 5: greedy-explanations(I_0, T)

$I_{end} \leftarrow$ propagate($I_0 \wedge T$);
Seq $\leftarrow$ empty sequence;
$I \leftarrow I_0$;
while $I \neq I_{end}$ **do**
    $(E, S, N) \leftarrow$min-explanation(I, T);
    $nested \leftarrow$ nested-explanations(E, S, N);
    append $((E, S, N), nested)$ to Seq;
    $I \leftarrow I \cup N$;
end

Algorithm 6: nested-explanations(E, S, N)

input : A partial interpretation I and a set of constraints C
$nested\_explanations \leftarrow \{\}$;
for $n_i \in N$ **do**
    store $\leftarrow true$;
    nested_seq $\leftarrow \{\}$;
    $I' \leftarrow E \wedge \neg\{n_i\}$;
    while $consistent(I')$ **do**
        $(E', S', N') \leftarrow$min-explanation(I', S);
        if $f(E', S', N') \geq f(E, S, N)$ **then**
            store $\leftarrow false$; **break**;
        end
        append (E', S', N') to nested_seq;
        $I' \leftarrow I' \cup N'$;
    end
    if $store$ **then**
        append $nested\_seq$ to $nested\_explanations$;
    end
end
return $nested\_explanations$