1 Naive Computation of Inconsistency Explanations in MCS

Similar as for the computation of diagnoses, we use dlvhex to compute all inconsistency explanations. Using a saturation technique, this is easily achieved. As dlhex uses the FLP-reduct semantics, we can allow negation-as-failure in the check-part of our saturation encoding, so we may reuse &*con_out_i* from the diagnosis computation.

On the other hand, however, we have to ensure that the external atom also holds for the saturated guess which is tricky as, input beliefs of a context also depend on bridge rule applicability which is ignored in the saturated case. Therefore we inform the external atom whether saturation is the case; we do this by adding the saturation atom "spoil" as input to &*con_out_i*. Technically, to avoid having to redo every external-atom, we use a meta-atom that will internally use &*con_out_i*.

The full encoding for an MCS M with bridge rules br_M, contexts $con(M)$, and OUT_i, resp., IN_i being the output-, resp. input-projected beliefs of context $i \in con(M)$: is as follows:

```
% guess an explanation candidate, ∀r ∈ br_M:
e1(r) v ne1(r).
e2(r) v ne2(r).
:- e1(r), ne1(r).
:- e2(r), ne2(r).

% guess a subset of the candidate explanation
% for which holds that E1 ⊆ r1 ⊆ br_M
% and r2 ⊆ br_M \ E2:
r1(R) :- e1(R).
r1(R) v nr1(R) :- ne1(R).
r2(R) v nr2(R) :- ne2(R).

% guess a belief state, every a ∈ OUT_i
% is either present or absent:
pres_i(a) v abs_i(a).

% derive applicable rule heads, ∀r ∈ br_M,
% of form r = (i : b) ← (i_1 : b_1),...,(i_{k−1} : b_{k−1}), not(i_k : b_k),...,not(i_m : b_m).
% with i_j ∈ con(M), and b_j ∈ OUT_j
in_i(b) :- r1(r), pres_i_1(b_1),..., pres_i_{k−1}(b_{k−1}), abs_k(b_k),..., abs_i_m(b_m).

% derive heads of unconditional rules, ∀r ∈ br_M with
% r = (i : b) ← ...
in_i(b) :- r2(r).

% inform external atom about which beliefs to use for output projection,
```

```
% i.e. OUT_i, so ∀o ∈ OUT_i add
out_i(o).

% spoil if the guessed belief state is no equilibrium
% of the guessed MCS, ∀i ∈ con(M) where the mcs-master file
% specifies a
% #context(i,"external_atom","param").
spoil :- not
    &saturation_meta_context["external_atom",spoil,i,pres_i,in_i,out_i,"param"].

% #context(2,"dlv_asp_context_acc","./medExample/kb2.dlv").
% therefore becomes
% spoil :- not &saturation_meta_atom["dlv_asp_context_acc",spoil,
%           2,pres_2,in_2,out_2,"./medExample/kb2.dlv"].

% spoil if our guesses are wrong by themselves, ∀r ∈ br_M, a ∈ OUT_i:
spoil :- r1(r), nr1(r).
spoil :- r2(r), nr2(r).
spoil :- pres_i(a), abs_i(a).

% saturate the guesses, i.e., subsets, belief states, and derived rule heads
% ∀r ∈ br_M, a ∈ OUT_i, b ∈ IN_i:
r1(r) :- spoil.
r2(r) :- spoil.
nr1(r) :- spoil.
nr2(r) :- spoil.
abs_i(a) :- spoil.
pres_i(a) :- spoil.
in_i(b) :- spoil.

% ensure saturation
:- not spoil.
```