

Online appendix for the paper  
**CLP(H): *Constraint Logic Programming for Hedges***  
 published in Theory and Practice of Logic Programming

BESIK DUNDUA

*VIAM, Tbilisi State University, Georgia and LIACC, University of Porto, Portugal*  
 (e-mail: [bdundua@gmail.com](mailto:bdundua@gmail.com))

MÁRIO FLORIDO

*DCC-FC and LIACC, University of Porto, Portugal (e-mail: [amf@dcc.fc.up.pt](mailto:amf@dcc.fc.up.pt))*

TEMUR KUTSIA

*RISC, Johannes Kepler University Linz, Austria (e-mail: [kutsia@risc.jku.at](mailto:kutsia@risc.jku.at))*

MIRCEA MARIN

*West University of Timișoara, Romania (e-mail: [mmarin@info.uvt.ro](mailto:mmarin@info.uvt.ro))*

*submitted 28 September 2014; revised 07 January 2015; accepted 23 February 2015*

## Appendix A

### *Theorem 1*

If the constraint  $\mathcal{C}$  is solved, then  $\mathfrak{J} \models \exists \mathcal{C}$  holds for all intended structures  $\mathfrak{J}$ .

### *Proof*

Since  $\mathcal{C}$  is solved, each disjunct  $\mathcal{K}$  in it has a form  $v_1 \doteq e_1 \wedge \dots \wedge v_n \doteq e_n \wedge v'_1$  in  $\mathbf{R}_1 \wedge \dots \wedge v'_m$  in  $\mathbf{R}_m$  where  $m, n \geq 0$ ,  $v_i, v'_j \in \mathcal{V}$  and  $e_i$  is an expression corresponding to  $v_i$ . Moreover,  $v_1, \dots, v_n, v'_1, \dots, v'_m$  are distinct and  $\llbracket \mathbf{R}_j \rrbracket \neq \emptyset$  for all  $1 \leq j \leq m$ . Note that while  $v_i$ 's do not occur anywhere else in  $\mathcal{K}$ , it still might be the case that some  $v'_j$ ,  $1 \leq j \leq m$ , occurs in some  $e_k$ ,  $1 \leq k \leq n$ .

Let  $e'_j$  be an element of  $\llbracket \mathbf{R}_j \rrbracket$  for all  $1 \leq j \leq m$ . Assume that for each  $1 \leq i \leq n$ , the substitution  $\sigma'_i$  is a grounding substitution for  $e_i$  with the property that  $v'_j \sigma'_i = e'_j$  for all  $1 \leq j \leq m$ . Then  $\sigma = \{v_1 \mapsto e_1 \sigma'_1, \dots, v_n \mapsto e_n \sigma'_n, v'_1 \mapsto e'_1, \dots, v'_m \mapsto e'_m\}$  solves  $\mathcal{K}$ . Therefore,  $\mathfrak{J} \models \exists \mathcal{C}$  holds.  $\square$

### *Theorem 2 (Termination of solve)*

*solve* terminates on any quantifier-free constraint.

### *Proof*

We need to show that  $\text{NF}(\text{step})$  terminates for any quantifier-free constraint in DNF. We define a complexity measure  $cm(\mathcal{C})$  for such constraints, and show that  $cm(\mathcal{C}') < cm(\mathcal{C})$  holds whenever  $\mathcal{C}' = \text{step}(\mathcal{C})$ .

For a hedge  $H$  (resp., for a regular expression  $\mathbf{R}$ ), we denote by  $size(H)$  (resp., by  $size(\mathbf{R})$ ) its denotational length, e.g.,  $size(\epsilon) = 0$ ,  $size(\mathbf{eps}) = 1$ ,  $size(f(f(a)), \bar{x}) = 4$ , and  $size(f(f(a \cdot b^*))) = 6$ .

The complexity measure  $cm(\mathcal{K})$  of a conjunction of primitive constraints  $\mathcal{K}$  is the tuple  $\langle N_1, M_1, N_2, M_2, M_3 \rangle$  defined as follows ( $\{\!\!\{\}$  stands for a multiset):

- $N_1$  is the number of unsolved variables in  $\mathcal{K}$ .
- $M_1 := \{\!\!\{size(H) \mid H \text{ in } R \in \mathcal{K}, H \neq \epsilon\}\!\!\}$ .
- $N_2$  is the number of primitive constraints in the form  $\bar{x} \text{ in } R \text{ in } \mathcal{K}$ .
- $M_2 := \{\!\!\{size(R) \mid H \text{ in } R \in \mathcal{K}\}\!\!\}$ .
- $M_3 := \{\!\!\{size(t_1) + size(t_2) \mid t_1 \doteq t_2 \in \mathcal{K}\}\!\!\}$ .

The complexity measure  $cm(\mathcal{C})$  of a constraint  $\mathcal{C} = \mathcal{K}_1 \vee \dots \vee \mathcal{K}_n$  is defined as  $\{\!\!\{cm(\mathcal{K}_1), \dots, cm(\mathcal{K}_n)\}\!\!\}$ .

Measures are compared by the multiset extension of the lexicographic ordering on tuples. The components that are natural numbers ( $N_1$  and  $N_2$ ) are, of course, compared by the standard ordering on naturals. The multiset components  $M_1$ ,  $M_2$ , and  $M_3$  are compared by the multiset extension of the standard ordering on the naturals.

The strict part of the ordering on measures is obviously well-founded. The **Log** rules strictly reduces it. For the other rules, the table below shows which rule reduces which component of the measure. The symbols  $>$  and  $\geq$  indicate the strict and non-strict decrease, respectively. It implies the termination of the algorithm **solve**.

Rule	$N_1$	$M_1$	$N_2$	$M_2$	$M_3$
(M1), (M10), (E1)–(E7)	$>$				
(F5), (F7), (M2), (M3), (M8), (M11), (M12)	$\geq$	$>$			
(M9)	$\geq$	$\geq$	$>$		
(F6), (M4)–(M7)	$\geq$	$\geq$	$\geq$	$>$	
(D1), (D2), (F1)–(F4), (Del1)–(Del3)	$\geq$	$\geq$	$\geq$	$\geq$	$>$

□

*Lemma 1*

If  $\text{step}(\mathcal{C}) = \mathcal{D}$ , then  $\mathfrak{J} \models \forall (\mathcal{C} \leftrightarrow \exists_{var(\mathcal{C})} \mathcal{D})$  for all intended structures  $\mathfrak{J}$ .

*Proof*

By case distinction on the inference rules of the solver, selected by the strategy first in the application of **step**. We illustrate here two cases, when the selected rules are **(E3)** and **(M2)**. For the other rules the lemma can be shown similarly.

In **(E3)**,  $\mathcal{C}$  has a disjunct  $\mathcal{K} = (\bar{x}, H) \doteq T \wedge \mathcal{K}'$  with  $\bar{x} \notin var(T)$ , and  $\mathcal{D}$  is the result of replacing  $\mathcal{K}$  in  $\mathcal{C}$  with the disjunction  $\mathcal{C}' = \bigvee_{T=(T_1, T_2)} (\bar{x} \doteq T_1 \wedge H \vartheta \doteq T_2 \wedge \mathcal{K}' \vartheta)$  where  $\vartheta = \{\bar{x} \mapsto T_1\}$ . Therefore, it is sufficient to show that  $\mathfrak{J} \models \forall (\mathcal{K} \leftrightarrow \exists_{var(\mathcal{C})} \mathcal{C}')$ . Since  $var(\mathcal{C}') = var(\mathcal{K})$ , this amounts to showing that for all ground substitutions  $\sigma$  of  $var(\mathcal{K})$  we have  $\mathfrak{J} \models (\bar{x}\sigma, H\sigma) \doteq T\sigma \wedge \mathcal{K}'\sigma$  iff  $\mathfrak{J} \models (\bigvee_{T=(T_1, T_2)} (\bar{x} \doteq T_1 \wedge H\vartheta \doteq T_2 \wedge \mathcal{K}'\vartheta))\sigma$ .

- Assume  $\mathfrak{J} \models (\bar{x}\sigma, H\sigma) \doteq T\sigma \wedge \mathcal{K}'\sigma$ . We can split  $T\sigma$  into  $T_1\sigma$  and  $T_2\sigma$  such that  $\bar{x}\sigma = T_1\sigma$  and  $H\sigma = T_2\sigma$ . Now, we show  $v\vartheta\sigma = v\sigma$  for all  $v \in var(\bar{x}, H, T)$ . Indeed, if  $v \neq \bar{x}$ , the equality trivially holds. If  $v = \bar{x}$ , we have  $\bar{x}\vartheta\sigma = T_1\sigma = \bar{x}\sigma$ . Hence,  $\mathfrak{J} \models (\bigvee_{T=(T_1, T_2)} (\bar{x} \doteq T_1 \wedge H\vartheta \doteq T_2 \wedge \mathcal{K}'\vartheta))\sigma$ .

- Assume  $\mathfrak{J} \models (\bigvee_{T=(T_1, T_2)} (\bar{x} \doteq T_1 \wedge H \vartheta \doteq T_2 \wedge \mathcal{K}' \vartheta)) \sigma$ . Then there exists the split  $T = (T_1, T_2)$  such that  $\mathfrak{J} \models (\bar{x} \sigma \doteq T_1 \sigma \wedge H \vartheta \sigma \doteq T_2 \sigma \wedge \mathcal{K}' \vartheta \sigma)$ . Again, we can show  $v \vartheta \sigma = v \sigma$  for all  $v \in \text{var}(\bar{x}, H, T)$ . Hence,  $\mathfrak{J} \models (\bar{x} \sigma, H \sigma) = T \sigma \wedge \mathcal{K}' \sigma$ . It finishes the proof for (E3).

Now, let the selected rule be (M2). In this case  $\mathcal{C}$  has a disjunct  $\mathcal{K} = (t, H)$  in  $\mathbf{R} \wedge \mathcal{K}'$  with  $H \neq \epsilon$  and  $\mathbf{R} \neq \mathbf{eps}$ . Then  $\mathcal{D}$  is the result of replacing  $\mathcal{K}$  in  $\mathcal{C}$  with  $\mathcal{C}' = \bigvee_{(f(\mathbf{R}_1), \mathbf{R}_2) \in \text{lf}(\mathbf{R})} (t \text{ in } f(\mathbf{R}_1) \wedge H \text{ in } \mathbf{R}_2 \wedge \mathcal{K}')$ . Therefore, to show  $\mathfrak{J} \models \forall(\mathcal{C} \leftrightarrow \exists_{\text{var}(\mathcal{C})} \mathcal{D})$ , it is enough to show that  $\mathfrak{J} \models \forall(\mathcal{K} \leftrightarrow \exists_{\text{var}(\mathcal{C})} \mathcal{C}')$ . Since  $\text{var}(\mathcal{C}') = \text{var}(\mathcal{K})$ , this amounts to showing that for all ground substitutions  $\sigma$  of  $\text{var}(\mathcal{K})$  we have  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\mathbf{R} \wedge \mathcal{K}' \sigma$  iff  $\mathfrak{J} \models (\bigvee_{(f(\mathbf{R}_1), \mathbf{R}_2) \in \text{lf}(\mathbf{R})} (t \text{ in } f(\mathbf{R}_1) \wedge H \text{ in } \mathbf{R}_2 \wedge \mathcal{K}')) \sigma$ . Recall that the linear form  $\text{lf}(\mathbf{R})$  of a regular expression  $\mathbf{R}$  has the property:

$$\llbracket \mathbf{R} \rrbracket \setminus \{\epsilon\} = \bigcup_{(f(\mathbf{R}_1), \mathbf{R}_2) \in \text{lf}(\mathbf{R})} \llbracket f(\mathbf{R}_1) \cdot \mathbf{R}_2 \rrbracket, \quad (\text{LF})$$

- Assume  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\mathbf{R} \wedge \mathcal{K}' \sigma$ . By the property (LF) and by the definitions of intended structure and entailment, we get that  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\mathbf{R} \wedge \mathcal{K}' \sigma$  implies  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\text{lf}(\mathbf{R}) \wedge \mathcal{K}' \sigma$ . Hence, we can conclude  $\mathfrak{J} \models (\bigvee_{(f(\mathbf{R}_1), \mathbf{R}_2) \in \text{lf}(\mathbf{R})} (t \sigma \text{ in } f(\mathbf{R}_1) \wedge H \sigma \text{ in } \mathbf{R}_2 \wedge \mathcal{K}' \sigma))$ .
- Assume  $\mathfrak{J} \models (\bigvee_{(f(\mathbf{R}_1), \mathbf{R}_2) \in \text{lf}(\mathbf{R})} (t \sigma \text{ in } f(\mathbf{R}_1) \wedge H \sigma \text{ in } \mathbf{R}_2 \wedge \mathcal{K}' \sigma))$ . Then we have  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\text{lf}(\mathbf{R}) \wedge \mathcal{K}' \sigma$  which, by (LF), implies  $\mathfrak{J} \models (t \sigma, H \sigma)$  in  $\mathbf{R} \wedge \mathcal{K}' \sigma$ .

□

### Theorem 3

If  $\text{solve}(\mathcal{C}) = \mathcal{D}$ , then  $\mathfrak{J} \models \forall(\mathcal{C} \leftrightarrow \exists_{\text{var}(\mathcal{C})} \mathcal{D})$  for all intended structures  $\mathfrak{J}$ , and  $\mathcal{D}$  is either partially solved or the false constraint.

### Proof

We assume without loss of generality that  $\mathcal{C}$  is in DNF.  $\mathfrak{J} \models \forall(\mathcal{C} \leftrightarrow \exists_{\text{var}(\mathcal{C})} \mathcal{D})$  follows from Lemma 1 and the following property: If  $\mathfrak{J} \models \forall(\mathcal{C}_1 \leftrightarrow \exists_{\text{var}(\mathcal{C}_1)} \mathcal{C}_2)$  and  $\mathfrak{J} \models \forall(\mathcal{C}_2 \leftrightarrow \exists_{\text{var}(\mathcal{C}_2)} \mathcal{C}_3)$ , then  $\mathfrak{J} \models \forall(\mathcal{C}_1 \leftrightarrow \exists_{\text{var}(\mathcal{C}_1)} \mathcal{C}_3)$ . The property itself relies on the fact that  $\mathfrak{J} \models \forall(\exists_{\text{var}(\mathcal{C}_1)} \exists_{\text{var}(\mathcal{C}_2)} \mathcal{C}_3 \leftrightarrow \exists_{\text{var}(\mathcal{C}_1)} \mathcal{C}_3)$ , which holds because all variables introduced by the rules of the solver in  $\mathcal{C}_3$  are fresh not only for  $\mathcal{C}_2$ , but also for  $\mathcal{C}_1$ .

As for the partially solved constraint, by the definition of  $\text{solve}$  and Theorem 2,  $\mathcal{D}$  is in a normal form. Assume by contradiction that it is not partially solved. By inspection of the solver rules, based on the definition of partially solved constraints, we can see that there is a rule that applies to  $\mathcal{D}$ . But this contradicts the fact that  $\mathcal{D}$  is in a normal form. Hence,  $\mathcal{D}$  is partially solved. □

### Lemma 2

Let  $v \doteq e$  be an equation, where  $v$  is a variable and  $e$  is the corresponding expression such that  $v$  does not occur in  $e$ . Let  $\mathcal{K}_1$  and  $\mathcal{K}_2$  be two arbitrary (possibly empty) conjunctions of extended literals such that the conjunction  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$  is well-moded. Let  $\theta = \{v \mapsto e\}$  be a substitution. Then  $\mathcal{K}_1 \wedge \mathcal{K}_2 \theta \wedge v \doteq e$  is also well-moded.

*Proof*

The point in this lemma is that it does not matter how  $\mathcal{K}_1$  and  $\mathcal{K}_2$  are chosen. We consider two cases. First, when  $v \doteq e$  is the leftmost literal containing  $v$  in a well-moded sequence corresponding to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$  and, second, when this is not the case.

*Case 1.* Let  $\tilde{E}_1, v \doteq e, \tilde{E}_2$  be a well-moded sequence corresponding to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ , such that  $\tilde{E}_1$  does not contain  $v$ . Note that there is no assumption (apart from what guarantees well-modedness of  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ ) on the appearance of literals in  $\tilde{E}_1$  and  $\tilde{E}_2$ : They may contain literals from  $\mathcal{K}_1$  only, from  $\mathcal{K}_2$  only, or from both  $\mathcal{K}_1$  and  $\mathcal{K}_2$ .

Well-modedness of  $\tilde{E}_1, v \doteq e, \tilde{E}_2$  requires the variables of  $e$  to appear in  $\tilde{E}_1$ . Consider the sequence  $\tilde{E}_1, v \doteq e, \tilde{E}_2[\theta]$ , where the notation  $\tilde{E}[\theta]$  stands for such an instance of  $\tilde{E}$  in which  $\theta$  affects only literals from  $\mathcal{K}_2$ . Then  $\tilde{E}_1, v \doteq e$  is well-moded and it can be safely extended by  $\tilde{E}_2[\theta]$  without violating well-modedness, because the variables in  $v \doteq e$  still precede (in the well-moded sequence) the literals from  $\tilde{E}_2[\theta]$ , and the relative order of the other variables (in the well-moded sequence) does not change. Hence,  $\tilde{E}_1, v \doteq e, \tilde{E}_2[\theta]$  is a well-moded sequence that corresponds to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ .

*Case 2.* Let  $\tilde{E}_1, L, \tilde{E}_2, v \doteq e, \tilde{E}_3$  be a well-moded sequence corresponding to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ , where  $L$  is the leftmost literal that contains  $v$  in an output position. Again, we make no assumption on literal appearances in the subsequences of the sequence. Then  $\tilde{E}_1, L, v \doteq e, \tilde{E}_2, \tilde{E}_3$  is also a well-moded sequence (corresponding to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ ), because  $v$  still appears in an output position in  $L$  left to  $v \doteq e$ , the variables in  $e$  still precede literals from  $\tilde{E}_3$ , and the relative order of the other variables does not change. For literals in  $\tilde{E}_2$  that contain variables from  $e$  such a reordering does not matter.

Note that  $v$  does not appear in  $\tilde{E}_1$ : If it were there in some literal in an output position, then  $L$  would not be the leftmost such literal. If it were there in some literal  $L'$  in an input position, then well-modedness of the sequence would require  $v$  to appear in an output position in another literal  $L''$  that is even before  $L'$ , i.e., to the left of  $L$  and it would again contradict the assumption that  $L$  is the leftmost literal containing  $v$  in an output position.

Let  $\tilde{E}_1, L[\theta], v \doteq e, \tilde{E}_2[\theta], \tilde{E}_3[\theta]$  be a sequence of all literals taken from  $\mathcal{K}_1 \wedge \mathcal{K}_2 \wedge v \doteq e$ . We distinguish two cases, depending whether  $\theta$  affects  $L$  or not.

$\theta$  affects  $L$ . Then it replaces  $v$  in  $L$  with  $e$ , i.e.,  $L[\theta] = L\theta$ . Then the variables of  $e$  appear in output positions in  $L\theta$  and, hence, placing  $v \doteq e$  after  $L\theta$  in the sequence would not destroy well-modedness. As for the  $L\theta$  itself, we have two alternatives:

1.  $L\theta$  is an equation, say  $s \doteq t\theta$ , obtained from  $L = (s \doteq t)$  by replacing occurrences of  $v$  in  $t$  by  $e$ . In this case, by well-modedness of  $\tilde{E}_1, L, v \doteq e, \tilde{E}_2, \tilde{E}_3$ , variables of  $s$  appear in  $\tilde{E}_1$  and  $s$  does not contain  $v$ . Then the same property is maintained in  $\tilde{E}_1, L\theta, v \doteq e, \tilde{E}_2[\theta], \tilde{E}_3[\theta]$ , since  $s$  remains in  $L\theta$  and  $\tilde{E}_1$  does not change.
2.  $L\theta$  is an atom. Then replacing  $v$  by  $e$  in an output position of  $L$ , which gives  $L\theta$ , does not affect well-modedness.

Hence, we got that  $\tilde{E}_1, L, v \doteq e$  is well-moded. Now we can safely extend this sequence with  $\tilde{E}_2[\theta], \tilde{E}_3[\theta]$ , because variables in new occurrences of  $e$  in  $\tilde{E}_2[\theta], \tilde{E}_3[\theta]$  are preceded by  $v \doteq e$ , and the relative order of the other variables does not change. Hence, the sequence  $\tilde{E}_1, L\theta, v \doteq e, \tilde{E}_2[\theta], \tilde{E}_3[\theta]$  is well-moded.

$\theta$  does not affect  $L$ . Then  $L[\theta] = L$ , the sequence  $\tilde{E}_1, L, v \doteq e$  is well-moded and it can

be safely extended with  $\tilde{E}_2[\theta], \tilde{E}_3[\theta]$ , obtaining the well-moded sequence  $\tilde{E}_1, L, v \doteq e, \tilde{E}_2[\theta], \tilde{E}_3[\theta]$ .

Hence, we showed also in *Case 2* that there exists a well-moded sequence of literals, namely,  $\tilde{E}_1, L[\theta], v \doteq e, \tilde{E}_2[\theta], \tilde{E}_3[\theta]$ , that corresponds to  $\mathcal{K}_1 \wedge \mathcal{K}_2 \theta \wedge v \doteq e$ . Hence,  $\mathcal{K}_1 \wedge \mathcal{K}_2 \theta \wedge v \doteq e$  is well-moded.  $\square$

*Lemma 3*

Let  $\text{Pr}$  be a well-moded CLP(H) program and  $\langle \mathbf{G} \parallel \mathcal{C} \rangle$  be a well-moded state. If  $\langle \mathbf{G} \parallel \mathcal{C} \rangle \mapsto \langle \mathbf{G}' \parallel \mathcal{C}' \rangle$  is a reduction using clauses in  $\text{Pr}$ , then  $\langle \mathbf{G}' \parallel \mathcal{C}' \rangle$  is also a well-moded state.

*Proof*

Let  $\mathbf{G} = L_1, \dots, L_i, \dots, L_n, \mathcal{C} = \mathcal{K}_1 \vee \dots \vee \mathcal{K}_m$ , and  $\langle \mathbf{G} \parallel \mathcal{C} \rangle$  be a well-moded state. We will use the notation  $\hat{\mathbf{G}}$  for the conjunction of all literals in  $\mathbf{G}$ , i.e.,  $\hat{\mathbf{G}} = L_1 \wedge \dots \wedge L_i \wedge \dots \wedge L_n$ . Assume that  $L_i$  is the selected literal in reduction that gives  $\langle \mathbf{G}' \parallel \mathcal{C}' \rangle$  from  $\langle \mathbf{G} \parallel \mathcal{C} \rangle$ . We consider four possible cases, according to the definition of operational semantics:

*Case 1.* Let  $L_i$  be a primitive constraint and  $\mathcal{C}' \neq \text{false}$ . Let  $\mathcal{D}$  denote the DNF of  $\mathcal{C} \wedge L_i$ .

In order to prove that  $\langle \mathbf{G}' \parallel \mathcal{C}' \rangle$  is well-moded, by the definition of *solve*, it is sufficient to prove that  $\langle \mathbf{G}' \parallel \text{step}(\mathcal{D}) \rangle$  is well-moded. Since, obviously,  $\langle \mathbf{G}' \parallel \mathcal{D} \rangle$  is a well-moded state, we have to show that state well-modedness is preserved by each rule of the solver.

Since  $\mathcal{C}' \neq \text{false}$ , the step is not performed by any of the failure rules of the solver. For the rules M1–M8, M11–M12, D1, and D2, it is pretty easy to verify that  $\langle \mathbf{G}' \parallel \text{step}(\mathcal{D}) \rangle$  is well-moded. Therefore, we consider the other rules in more detail. We denote the disjunct of  $\mathcal{D}$  on which the rule is applied by  $\mathcal{K}_{\mathcal{D}}$ . The cases below are distinguished by the rules:

**Del.** Here the same variable is removed from both sides of the selected equation. Assume  ${}_1, s \doteq t, {}_2$  is a well-moded sequence corresponding to  $\hat{\mathbf{G}}' \wedge \mathcal{K}_{\mathcal{D}}$ , and  $s \doteq t$  is the selected equation affected by one of the deletion rules. Well-modedness of  ${}_1, s \doteq t, {}_2$  requires that the variable deleted at this step from  $s \doteq t$  should occur in an output position in some other literal in  ${}_1$ . Let  $s' \doteq t'$  be the equation obtained by the deletion step from  $s \doteq t$ . Then  ${}_1, s' \doteq t', {}_2$  is again well-moded, which implies that  $\hat{\mathbf{G}}' \wedge \text{step}(\mathcal{K}_{\mathcal{D}})$  is well-moded and, therefore, that  $\langle \mathbf{G}' \parallel \text{step}(\mathcal{D}) \rangle$  is well-moded.

**M9.** Let  $\hat{\mathbf{G}}' \wedge \mathcal{K}_{\mathcal{D}}$  be represented as  $\hat{\mathbf{G}}' \wedge \bar{x}$  in  $f(\mathbf{R}) \wedge \mathcal{K}'$ , where  $\bar{x}$  in  $f(\mathbf{R})$  is the membership atom affected by the rule. Note that then  $\hat{\mathbf{G}}' \wedge \bar{x} \doteq x \wedge x$  in  $f(\mathbf{R}) \wedge \mathcal{K}'$  is also well-moded. Applying Lemma 2, we get that  $\hat{\mathbf{G}}' \wedge \bar{x} \doteq x \wedge x$  in  $f(\mathbf{R}) \wedge \mathcal{K}' \theta$  is well-moded, where  $\theta = \{\bar{x} \mapsto x\}$ . Then we get well-modedness of  $\hat{\mathbf{G}}' \wedge \text{step}(\mathcal{K}_{\mathcal{D}})$ , which implies well-modedness of  $\langle \mathbf{G}' \parallel \text{step}(\mathcal{D}) \rangle$ .

**M10.** Let  $\hat{\mathbf{G}}' \wedge \mathcal{K}_{\mathcal{D}}$  be represented as  $\hat{\mathbf{G}}' \wedge X(H)$  in  $f(\mathbf{R}) \wedge \mathcal{K}'$ , where  $X(H)$  in  $f(\mathbf{R})$  is the membership atom affected by the rule. Note that then  $\hat{\mathbf{G}}' \wedge X(H)$  in  $f(\mathbf{R}) \wedge X \doteq f \wedge \mathcal{K}'$  is also well-moded. Applying Lemma 2, we get that  $\hat{\mathbf{G}}' \wedge X(H) \theta$  in  $f(\mathbf{R}) \wedge X \doteq f \wedge \mathcal{K}' \theta$  is well-moded, where  $\theta = \{X \mapsto f\}$ . But it means that  $\hat{\mathbf{G}}' \wedge \text{step}(\mathcal{K}_{\mathcal{D}})$  is well-moded, which implies that  $\langle \mathbf{G}' \parallel \text{step}(\mathcal{D}) \rangle$  is well-moded.

**E1, E2.** For these rules, well-modedness of  $\hat{\mathbf{G}}' \wedge \text{step}(\mathcal{K}_{\mathcal{D}})$  is a direct consequence of Lemma 2.

**E3.** Let  $\hat{\mathbf{G}}' \wedge \mathcal{K}_{\mathcal{D}}$  be represented as  $\hat{\mathbf{G}}' \wedge (\bar{x}, H_1) \simeq H_2 \wedge \mathcal{K}'$ , where  $(\bar{x}, H_1) \simeq H_2$  is the equation affected by the rule and  $\bar{x} \notin \text{var}(H_2)$ . Then  $\hat{\mathbf{G}}' \wedge \bar{x} \doteq H' \wedge H_1 \doteq H'' \wedge \mathcal{K}'$  is

also well-moded for some  $H'$  and  $H''$  with  $(H', H'') = H_2$ . Applying Lemma 2, we get that  $\hat{G}' \wedge \bar{x} \doteq H' \wedge H_1\theta \doteq H'' \wedge \mathcal{K}'\theta$  is well-moded, where  $\theta = \{\bar{x} \mapsto H'\}$ . Since  $H'$  and  $H''$  were arbitrary, it implies that  $\hat{G}' \wedge \text{step}(\mathcal{K}_{\mathcal{D}})$  and, therefore,  $\langle G' \parallel \text{step}(\mathcal{D}) \rangle$  is well-moded.

E4. Similar to the case of the rule E3.

*Case 2.* Let  $L_i$  be a primitive constraint and  $C' = \text{false}$ , where  $C' = \text{solve}(C \wedge L_i)$ . Then by the operational semantics we have  $G' = \square$  and the theorem trivially holds, since the state  $\langle \square \parallel \text{false} \rangle$  is well-moded.

*Case 3.* Let  $L_i$  be an atom  $p(t_1, \dots, t_k, \dots, t_l)$ . Assume that Pr contains a clause of the form  $p(r_1, \dots, r_k, \dots, r_l) \leftarrow B$ , where  $B$  denotes the body of the clause. Assume also that for the predicate  $p$ , the set  $\{1, \dots, k\}$  is the set of the input positions and  $\{k+1, \dots, l\}$  is the set of the output ones. Then we have

$$\begin{aligned} G &= L_1, \dots, L_{i-1}, p(t_1, \dots, t_k, \dots, t_l), L_{i+1}, \dots, L_n, \\ G' &= L_1, \dots, L_{i-1}, t_1 \doteq r_1, \dots, t_k \doteq r_k, \dots, t_l \doteq r_l, B, L_{i+1}, \dots, L_n, \\ C' &= C = \mathcal{K}_1 \vee \dots \vee \mathcal{K}_m. \end{aligned}$$

From well-modedness of the state  $\langle G \parallel C \rangle$  we know that for all  $1 \leq j \leq m$ , the literals from  $L_1, \dots, L_{i-1}, L_{i+1}, \dots, L_n$  and  $\mathcal{K}_j$  can be reordered in two sequences of literals  $\frac{1}{j}$  and  $\frac{2}{j}$  in such a way that the sequence  $\frac{1}{j}, p(t_1, \dots, t_k, \dots, t_l), \frac{2}{j}$  is well-moded. Then we have  $\text{var}(t_1, \dots, t_k) \subseteq \text{outvar}(\frac{1}{j})$ . Therefore, we obtain that the sequence

$$\frac{1}{j}, t_1 \doteq r_1, \dots, t_k \doteq r_k, \frac{2}{j} \tag{A1}$$

is well-moded for all  $1 \leq j \leq m$ .

From well-modedness of  $p(r_1, \dots, r_k, \dots, r_l) \leftarrow B$  we know that  $\text{var}(r_{k+1}, \dots, r_l) \subseteq \text{outvar}(B) \cup \text{var}(r_1, \dots, r_k)$ . By the definition of program well-modedness, the literals of  $B$  can be put into a well-moded sequence, written, say, as  $B_1, \dots, B_q$ , such that for each  $1 \leq u \leq q$  and  $v \in \text{invar}(B_u)$  we have  $v \in \text{outvar}(B_{u'})$  for some  $u' < u$ , or  $v \in \text{var}(r_1, \dots, r_k)$ . From then we can say that the sequence

$$t_1 \doteq r_1, \dots, t_k \doteq r_k, B_1, \dots, B_q, t_{k+1} \doteq r_{k+1}, \dots, t_l \doteq r_l \tag{A2}$$

is well-moded.

From (A1) and (A2), by the definition of well-modedness, we can conclude that

$$\frac{1}{j}, t_1 \doteq r_1, \dots, t_k \doteq r_k, B_1, \dots, B_q, t_{k+1} \doteq r_{k+1}, \dots, t_l \doteq r_l, \frac{2}{j} \tag{A3}$$

is well-moded for all  $1 \leq j \leq m$ . By construction, the literals in (A3) are exactly those from  $\hat{G}' \wedge \mathcal{K}_j$  for  $1 \leq j \leq m$ . It means that  $\langle G' \parallel \mathcal{K}_j \rangle$  is well-moded for all  $1 \leq j \leq m$ , which implies that  $\langle G' \parallel C' \rangle$  is well-moded.

*Case 4.* If  $\text{defn}_P(L_i) = \emptyset$ , then  $G' = \square$ ,  $C' = \text{false}$ , and the theorem trivially holds.  $\square$

### Corollary 1

If  $C$  is a well-moded constraint, then  $\text{solve}(C)$  is also well-moded.

### Proof

By the definition of well-modedness, since  $C$  is well-moded, the state  $\langle a \doteq a \parallel C \rangle$  is also well-moded, where  $a$  is an arbitrary function symbol. By the operational semantics, we

have the reduction  $\langle a \doteq a \parallel \mathcal{C} \rangle \rightsquigarrow \langle \square \parallel \text{solve}(a \doteq a \wedge \mathcal{C}) \rangle$ . By Lemma 3, we get that  $\langle \square \parallel \text{solve}(a \doteq a \wedge \mathcal{C}) \rangle$  is also well-moded and, hence,  $\text{solve}(a \doteq a \wedge \mathcal{C})$  is well-moded. By the definition of  $\text{solve}$  and the rules of the solver, it is straightforward to see that  $\text{solve}(a \doteq a \wedge \mathcal{C}) = \text{solve}(\mathcal{C})$ . Hence,  $\text{solve}(\mathcal{C})$  is well-moded.  $\square$

*Theorem 4*

Let  $\mathcal{C}$  be a well-moded constraint and  $\text{solve}(\mathcal{C}) = \mathcal{C}'$ , where  $\mathcal{C}' \neq \text{false}$ . Then  $\mathcal{C}'$  is solved.

*Proof*

By the Corollary 1, the constraint  $\mathcal{C}'$  is well-moded. If  $\mathcal{C}'$  is true then it is already solved. Consider the case when  $\mathcal{C}'$  is not false. Let  $\mathcal{C}' = \mathcal{K}_1 \vee \dots \vee \mathcal{K}_m$ . Since  $\mathcal{C}' \neq \text{false}$ , by the Theorem 3  $\mathcal{C}'$  is partially solved. It means that each  $\mathcal{K}_j$ ,  $1 \leq j \leq m$ , is partially solved and well-moded. By definition,  $\mathcal{K}_j$  is well-moded if there exists a permutation of its literals  $\mathbf{c}_1, \dots, \mathbf{c}_i, \dots, \mathbf{c}_n$  which satisfies the well-modedness property. Assume  $\mathbf{c}_1, \dots, \mathbf{c}_{i-1}$  are solved. By this assumption and the definition of well-modedness, each of  $\mathbf{c}_1, \dots, \mathbf{c}_{i-1}$  is an equation whose one side is a variable that occurs neither in its other side nor in any other primitive constraint. Then well-modedness of  $\mathcal{K}_j$  guarantees that the other sides of these equations are ground terms. Assume by contradiction that  $\mathbf{c}_i$  is partially solved, but not solved. If  $\mathbf{c}_i$  is a membership constraint, well-modedness of  $\mathcal{K}_j$  implies that  $\mathbf{c}_i$  does not contain variables and, therefore, can not be partially solved. Now let  $\mathbf{c}_i$  be an equation. Since all variables in  $\mathbf{c}_1, \dots, \mathbf{c}_{i-1}$  are solved, they can not appear in  $\mathbf{c}_i$ . From this fact and well-modedness of  $\mathcal{K}_j$ ,  $\mathbf{c}_i$  should have at least one ground side. But then it can not be partially solved. The obtained contradiction shows that  $\mathcal{C}'$  is solved.  $\square$

*Theorem 5*

Let  $\langle \mathbf{G} \parallel \text{true} \rangle \rightsquigarrow \dots \rightsquigarrow \langle \square \parallel \mathcal{C} \rangle$  be a finished derivation with respect to a well-moded CLP(H) program, starting from a well-moded goal  $\mathbf{G}$ . If  $\mathcal{C} \neq \text{false}$ , then  $\mathcal{C}$  is solved.

*Proof*

We prove a slightly more general statement: Let  $\langle \mathbf{G} \parallel \text{true} \rangle \rightsquigarrow \dots \rightsquigarrow \langle \mathbf{G}' \parallel \mathcal{C}' \rangle$  be a derivation with respect to a well-moded program, starting from a well-moded goal  $\mathbf{G}$  and ending with  $\mathbf{G}'$  that is either  $\square$  or consists only of atomic formulas without arguments (propositional constants). If  $\mathcal{C}' \neq \text{false}$ , then  $\mathcal{C}'$  is solved.

To prove this statement, we use induction on the length  $n$  of the derivation. When  $n = 0$ , then  $\mathcal{C}' = \text{true}$  and it is solved. Assume the statement holds when the derivation length is  $n$ , and prove it for the derivation with the length  $n + 1$ . Let such a derivation be  $\langle \mathbf{G} \parallel \text{true} \rangle \rightsquigarrow \dots \rightsquigarrow \langle \mathbf{G}_n \parallel \mathcal{C}_n \rangle \rightsquigarrow \langle \mathbf{G}_{n+1} \parallel \mathcal{C}_{n+1} \rangle$ . Assume that  $\mathbf{G}_{n+1}$  that is either  $\square$  or consists only of propositional constants. According to the operational semantics, there are two possibilities how the last step is made:

1.  $\mathbf{G}_n$  has a form (modulo permutation)  $L, p_1, \dots, p_m$ ,  $m \geq 0$ , where  $L$  is primitive constraint, the  $p$ 's are propositional constants,  $\mathbf{G}_{n+1} = p_1, \dots, p_m$ , and  $\mathcal{C}_{n+1} = \text{solve}(\mathcal{C}_n \wedge L)$ .
2.  $\mathbf{G}_n$  has a form (modulo permutation)  $q, p_1, \dots, p_m$ ,  $m \geq 0$ , where  $q$  and  $p$ 's are propositional constants, the program contains a clause  $q \leftarrow q_1, \dots, q_k$ ,  $k \geq 0$ , where all  $q_i$ ,  $1 \leq i \leq k$ , are propositional constants,  $\mathbf{G}_{n+1} = q_1, \dots, q_k, p_1, \dots, p_m$ , and  $\mathcal{C}_{n+1} = \mathcal{C}_n$ .

In the first case, by the  $n$ -fold application of Lemma 3 we get that  $\langle G_n \parallel C_n \rangle$  is well-moded. Since the  $p$ 's have no influence on well-modedness (they are just propositional constants),  $C_n \wedge L$  is well-moded and hence it is solvable. By Theorem 4 we get that if  $C_{n+1} = \text{solve}(C_n \wedge L) \neq \text{false}$ , then  $C_{n+1}$  is solved.

In the second case, since  $G_n$  consists of propositional constants only, by the induction hypothesis we have that if  $C_n$  is not **false**, then it is solved. But  $C_n = C_{n+1}$ . It finishes the proof.  $\square$

*Lemma 4*

Any partially solved KIF constraint is solved.

*Proof*

Let  $\mathcal{K}$  be a partially solved conjunction of primitive constraints. Then, by the definition, each primitive constraint  $\mathbf{c}$  from  $\mathcal{K}$  should be either solved in  $\mathcal{K}$ , or should have one of the following forms:

- Membership atom:
  - $f_u(H_1, \bar{x}, H_2)$  in  $f_u(R)$ .
  - $(\bar{x}, H)$  in  $R$  where  $H \neq \epsilon$  and  $R$  has the form  $R_1 \cdot R_2$  or  $R_1^*$ .
- Equation:
  - $(\bar{x}, H_1) \doteq (\bar{y}, H_2)$  where  $\bar{x} \neq \bar{y}$ ,  $H_1 \neq \epsilon$  and  $H_2 \neq \epsilon$ .
  - $(\bar{x}, H_1) \doteq (T, \bar{y}, H_2)$ , where  $\bar{x} \notin \text{var}(T)$ ,  $H_1 \neq \epsilon$ , and  $T \neq \epsilon$ . The variables  $\bar{x}$  and  $\bar{y}$  are not necessarily distinct.
  - $f_u(H_1, \bar{x}, H_2) \doteq f_u(H_3, \bar{y}, H_4)$  where  $(H_1, \bar{x}, H_2)$  and  $(H_3, \bar{y}, H_4)$  are disjoint.

However,  $\mathbf{c}$  is also a KIF constraint. By the definition of KIF form, none of the above mentioned forms for membership atoms and equations are permitted. Hence,  $\mathbf{c}$  is solved in  $\mathcal{K}$  and, therefore,  $\mathcal{K}$  is solved. It implies the lemma.  $\square$

*Theorem 6*

Let  $\mathcal{C}$  be a KIF constraint and  $\text{solve}(\mathcal{C}) = \mathcal{C}'$ , where  $\mathcal{C}' \neq \text{false}$ . Then  $\mathcal{C}'$  is solved.

*Proof*

By Theorem 3,  $\mathcal{C}'$  should be in a partially solved form. It is also in the KIF form, as we noted above. Then, by Lemma 4,  $\mathcal{C}'$  is solved.  $\square$

*Theorem 7*

Let  $\langle G \parallel \text{true} \rangle \rightsquigarrow \dots \rightsquigarrow \langle \square \parallel \mathcal{C}' \rangle$  be a finished derivation with respect to a KIF program, starting from a KIF goal  $G$ . If  $\mathcal{C}' \neq \text{false}$ , then  $\mathcal{C}'$  is solved.

*Proof*

Since the reduction preserves KIF states,  $\mathcal{C}'$  is in the KIF form. Since the derivation is finished and  $\mathcal{C}' \neq \text{false}$ , by the definition of finished derivation,  $\mathcal{C}'$  is partially solved. By Lemma 4, we conclude that  $\mathcal{C}'$  is solved.  $\square$