Correctness Proofs of Transformation Schemas

Halime Büyükyıldız and Pierre Flener

Department of Computer Engineering and Information Science
Faculty of Engineering, Bilkent University, 06533, Bilkent, Ankara, Turkey
Email correspondence to: pf@cs.bilkent.edu.tr

Abstract

Schema-based logic program transformation has proven to be an effective technique for the optimization of programs. Some transformation schemas were given in [3]; they pre-compile some widely used transformation techniques from an input program schema that abstracts a particular family of programs into an output program schema that abstracts another family of programs.

This report presents the correctness proofs of these transformation schemas, based on a correctness definition of transformation schemas. A transformation schema is *correct* iff the templates of its input and output program schemas are equivalent wrt the specification of the top-level relation defined in these program schemas, under the applicability conditions of this transformation schema.

1 Introduction

In this introductory section, we give the definitions of the notions that are needed to prove the correctness of the transformation schemas in [3]. The transformation schemas proved in this report are pre-compilations of the accumulation strategy [2], of tupling generalization, which is a special case of structural generalization [4], of a combination of the previous two techniques, and of the first duality law of the fold operators in functional programming [1]. For a detailed explanation of these transformation schemas and examples of the definitions below, the reader is invited to consult [3].

Throughout this report, the word program (resp. procedure) is used to mean typed definite program (resp. procedure). An *open program* is a program where some of the relations appearing in the clause bodies are not appearing in any heads of clauses, and these relations are called *undefined* (or *open*) relations. If all the relations appearing in the program are *defined*, then the program is called a *closed program*. A *formal specification* of a program for a relation r of arity 2 is a first-order formula written in the format:

$$\forall X : \mathcal{X}.\ \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the sorts (or: types) of X and Y, respectively, $\mathcal{I}_r(X)$ denotes the *input condition* that must be fulfilled before the execution of the program, and $\mathcal{O}_r(X,Y)$ denotes the *output condition* that will be fulfilled after the execution. All the definitions are given only for programs in closed frameworks. So, we first give the definition of frameworks.

Definition 1 (Frameworks)
A *framework* $\mathcal{F}$ is a full first-order logical theory (with identity) with an intended model. An *open framework* consists of:

* a (many-sorted) signature of
  - both *defined* and *open* sort names;
  - function declarations, for declaring both *defined* and *open* constant and function names;
  - relation declarations, for declaring both *defined* and *open* relation names;
* a set of first-order axioms each for the (declared) *defined* and *open* function and relation names, the former possibly containing induction schemas;
* a set of theorems.

An open framework $\mathcal{F}$ is also denoted by $\mathcal{F}(\Pi)$, where Π are the open names, or parameters, of $\mathcal{F}$. The definition of a *closed framework* is the same as the definition of an open framework, except that a closed framework has no open names. Therefore, a closed framework is just an extreme case of an open one, namely where Π is empty.

Now, we give the definitions of correctness of a logic program and equivalence of two programs, which will be used in the equivalence definition of two program schemas.

Definition 2 (Correctness of a Closed Program)
Let P be a closed program for relation r in a closed framework $\mathcal{F}$. We say that P is (*totally*) *correct* wrt its specification S_r iff, for any ground term t of $\mathcal{X}$ such that $\mathcal{I}_r(t)$ holds, the following condition holds: $P \vdash r(t,u)$ iff $\mathcal{F} \models \mathcal{O}_r(t,u)$, for every ground term u of $\mathcal{Y}$.

If we replace 'iff' by 'implies' in the condition above, then P is said to be *partially correct* wrt S_r, and if we replace 'iff' by 'if', then P is said to be *complete* wrt S_r.

This kind of correctness is not entirely satisfactory, for two reasons. First, it defines the correctness of P in terms of the procedures for the relations in its clause bodies, rather than in terms of their specifications. Second, P must be a closed program, even though it might be desirable to discuss the correctness of P without having to fully implement it. So, the abstraction achieved through the introduction (and specification) of the relations in its clause bodies is wasted. This leads us to the notion of steadfastness (also known as parametric correctness) [5] (also see [4]).

Definition 3 (Steadfastness of an Open Program in a Set of Specifications)
In a closed framework $\mathcal{F}$, let:

- P be an open program for relation r
- $q_1, \dots, q_m$ be all the undefined relation names appearing in P
- $S_1, \dots, S_m$ be the specifications of $q_1, \dots, q_m$.

We say that P is *steadfast* wrt its specification S_r in $\{S_1, \dots, S_m\}$ iff the (closed) program $P \cup P_S$ is correct wrt S_r, where P_S is any closed program such that

- P_S is correct wrt each specification S_j $(1 \leq j \leq m)$
- P_S contains no occurrences of the relations defined in P.

The steadfastness definition has the following interesting property, which is actually a high-level recursive algorithm to check the steadfastness of an open program.

Property 1 In a closed framework $\mathcal{F}$, let:

- P be an open program for relation r of the specification S_r
- $p_1, \dots, p_t$ be all the defined relation names appearing in P (including r thus)
- $q_1, \dots, q_m$ be all the undefined relation names appearing in P
- $S_1, \dots, S_m$ be the specifications of $q_1, \dots, q_m$.

For $t \geq 2$, the program P is steadfast wrt S_r in $\{S_1, \dots, S_m\}$ iff every P_i $(1 \leq i \leq t)$ is steadfast wrt the specification of p_i in the set of the specifications of all undefined relations in P_i, where P_i is a program for p_i, such that $P = \bigcup_{i=1}^{t} P_i$. When $t = 1$, the definition of steadfastness is directly used, since the only defined relation is the relation r. Thus, $t = 1$ is the stopping case of this recursive algorithm.

For program equivalence, we do not require the two programs to have the same models, because this would not make much sense in some program transformation settings where the transformed program features relations that were not in the initially given program. That is why our program equivalence criterion establishes equivalence wrt the specification of a common relation (usually the root of their call-hierarchies).

Definition 4 (Equivalence of Two Open Programs)
In a closed framework $\mathcal{F}$, let P and Q be two open programs for a relation r. We say that P is *equivalent to* Q wrt the specification S_r iff the following two conditions hold:

(*a*) P is steadfast wrt S_r in $\{S_1, \dots, S_m\}$, where $S_1, \dots, S_m$ are the specifications of $p_1, \dots, p_m$, which are all the undefined relation names appearing in P

(*b*) Q is steadfast wrt S_r in $\{S'_1, \dots, S'_t\}$, where $S'_1, \dots, S'_t$ are the specifications of $q_1, \dots, q_t$, which are all the undefined relation names appearing in Q.

Since the 'is equivalent to' relation is symmetric, we also say that P and Q are *equivalent* wrt S_r.

Sometimes, in program transformation settings, there exist some conditions that have to be verified related to some parts of the initial and/or transformed program in order to have a transformed program that is equivalent to the initially given program wrt the specification of the top-level relation. Hence the following definition.

Definition 5 (Conditional Equivalence of Two Open Programs)
In a closed framework $\mathcal{F}$, let P and Q be two open programs for a relation r. We say that P is *equivalent to* Q wrt the specification S_r *under* conditions C iff P is *equivalent to* Q wrt S_r provided that C hold.

Before we define the notions of transformation schema and correctness of transformation schemas, we have to define the notions of program schema, schema pattern, and particularization.

Definition 6 In a closed framework $\mathcal{F}$, a *program schema* for a relation r is a pair $\langle T, C\rangle$, where T is an open program for r, called the *template*, and C is the set of specifications of the open relations of T in terms of each other and the input/output conditions of the closed relations of T. The specifications in C, called the *steadfastness constraints*, are such that, in $\mathcal{F}$, T is steadfast wrt its specification S_r in C.

Sometimes, a series of schemas are quite similar, in the sense that they only differ in the number of arguments of some relations, or in the number of calls to some relations, etc. For this purpose, rather than having a proliferation of similar schemas, we introduce the notions of *schema pattern* and *particularization*.

Definition 7 A *schema pattern* is a schema where term, conjunct, and disjunct ellipses are allowed in the template and in the steadfastness constraints.

For instance, $TX_1, \ldots, TX_t$ is a term ellipsis, and $\bigwedge_{i=1}^{t} r(TX_i, TY_i)$ is a conjunct ellipsis.

Definition 8 A *particularization* of a schema pattern is a schema obtained by eliminating the ellipses, i.e., by binding the (mathematical) variables denoting their lower and upper bounds to natural numbers.

Finally, we give the definition of transformation schemas and their correctness definition.

Definition 9 A *transformation schema* encoding a transformation technique is a 5-tuple $\langle S_1, S_2, A, O_{12}, O_{21}\rangle$, where S_1 and S_2 are program schemas (or schema patterns), A is a set of *applicability conditions*, which ensure the equivalence of the templates of S_1 and S_2 wrt the specification of the top-level relation, and O_{12} (respectively, O_{21}) is a set of *optimizability conditions* when S_2 (respectively, S_1) is the output program schema (or schema pattern).

If the transformation schema embodies some generalization technique, then it is called a *generalization schema*. The generalization methods that we pre-compile in our transformation schemas are *tupling generalization*, which is a special case of *structural generalization* where the structure of some parameter is generalized, and *descending generalization*, which is a special case of *computational generalization* where the general state of computation is generalized in terms of what remains to be done. We also introduce a new method, called *simultaneous tupling-and-descending generalization*, which can be thought of as applying descending generalization to a tupling generalized problem. Transformation schemas that simulate and extend a basic theorem in functional programming (the first duality law of the fold operators) for logic programs are called *duality schemas*.

Definition 10 A transformation schema $\langle S_1, S_2, A, O_{12}, O_{21}\rangle$ is *correct* iff the templates of program schemas (or schema patterns) S_1 and S_2 are equivalent wrt the specification of the top-level relation under A.

In program transformation, for proving the correctness of a transformation schema $\langle S_1, S_2, A, O_{12}, O_{21}\rangle$, we have to prove the equivalence of T_1 and T_2, which are the templates of $S_1 = \langle T_1, C_1\rangle$ and $S_2 = \langle T_2, C_2\rangle$. We assume that the template T_i of the input program schema $S_i = \langle T_i, C_i\rangle$ (where $i = 1, 2$) is steadfast wrt the specification of the top-level relation, say S_r, in C_i; then the correctness of the transformation schema is proven by establishing the steadfastness of the template T_j of the output program schema (or schema pattern) $S_j = \langle T_j, C_j\rangle$ (where $j = 1, 2$ and $j \neq i$) wrt S_r in C_j using the applicability conditions A.

In the remainder of this report, first the tupling generalization schemas are proved to be correct, in Section 2. In Section 3, the correctness proofs of the descending generalization schemas, which are a pre-compilation of the accumulation strategy, are given. The correctness proofs of the simultaneous tupling-and descending generalization schemas are given in Section 4. Before we conclude in Section 6, we will give the correctness proofs of the duality schemas in Section 5.

2 Proofs of the Tupling Generalization Schemas

Theorem 1 The generalization schema TG_1, which is given below, is correct.

TG_1 : $\langle$ $DCLR$, TG, A_{t1}, O_{t112}, O_{t121} $\rangle$ where

A_{t1} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element
(3) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
(4) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{t112} : partial evaluation of the conjunction
$process(HX, HY), compose(HY, TY, Y)$
results in the introduction of a non-recursive relation

O_{t121} : partial evaluation of the conjunction
$process(HX, HY), compose(I_{p-1}, HY, I_p)$
results in the introduction of a non-recursive relation

where the templates $DCLR$ and TG are Logic Program Templates 1 and 2 below:

Logic Program Template 1

```
r(X, Y) ←
    minimal(X),
    solve(X, Y)
r(X, Y) ←
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    r(TX1, TY1), ..., r(TXt, TYt),
    I0 = e,
    compose(I0, TY1, I1), ..., compose(Ip-2, TYp-1, Ip-1),
    process(HX, HY), compose(Ip-1, HY, Ip),
    compose(Ip, TYp, Ip+1), ..., compose(It, TYt, It+1),
    Y = It+1
```

Logic Program Template 2

```
r(X, Y) ←
    r_tupling([X], Y)
r_tupling(Xs, Y) ←
    Xs = [],
    Y = e
r_tupling(Xs, Y) ←
    Xs = [X|TXs],
    minimal(X),
    r_tupling(TXs, TY),
    solve(X, HY),
    compose(HY, TY, Y)
r_tupling(Xs, Y) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    minimal(TX1), ..., minimal(TXt),
    r_tupling(TXs, TY),
    process(HX, HY),
```

```
        compose(HY,TY,Y)
    r_tupling(Xs,Y) ←
        Xs = [X|TXs],
        nonMinimal(X),
        decompose(X,HX,TX1,...,TXt),
        minimal(TX1),...,minimal(TXp-1),
        (nonMinimal(TXp);...;nonMinimal(TXt)),
        r_tupling([TXp,...,TXt|TXs],TY),
        process(HX,HY),
        compose(HY,TY,Y)
    r_tupling(Xs,Y) ←
        Xs = [X|TXs],
        nonMinimal(X),
        decompose(X,HX,TX1,...,TXt),
        (nonMinimal(TX1);...;nonMinimal(TXp-1)),
        minimal(TXp),...,minimal(TXt),
        minimal(U1),...,minimal(Up-1),
        decompose(N,HX,U1,...,Up-1,TXp,...,TXt),
        r_tupling([TX1,...,TXp-1,N|TXs],Y)
    r_tupling(Xs,Y) ←
        Xs = [X|TXs],
        nonMinimal(X),
        decompose(X,HX,TX1,...,TXt),
        (nonMinimal(TX1);...;nonMinimal(TXp-1)),
        (nonMinimal(TXp);...;nonMinimal(TXt)),
        minimal(U1),...,minimal(Ut),
        decompose(N,HX,U1,...,Ut),
        r_tupling([TX1,...,TXp-1,N,TXp,...,TXt|TXs],Y)
```

and the specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

and the specification $S_{r\_tupling}$ of relation *r_tupling* is:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs,Y) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = e)$$
$$\vee (Xs = [X_1,\ldots,X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1},Y_i,I_i) \ \wedge Y = I_q)]$$

where $\mathcal{O}_c$ is the output condition of *compose* and $q \geq 1$.

Proof 1 To prove the correctness of the generalization schema TG_1, by Definition 10, we have to prove that templates *DCLR* and *TG* are *equivalent* wrt S_r under the applicability conditions A_{t1}. By Definition 5, the templates *DCLR* and *TG* are *equivalent* wrt S_r under the applicability conditions A_{t1} iff *DCLR* is *equivalent to TG* wrt the specification S_r provided that the conditions in A_{t1} hold. By Definition 4, *DCLR* is *equivalent to TG* wrt the specification S_r iff the following two conditions hold:

(*a*) *DCLR* is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in *DCLR*.

(*b*) *TG* is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by tupling generalization of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: TG is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$, where $P_{r\_tupling}$ is the procedure for $r\_tupling$, and P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$, where P_r is the procedure for r.

To prove that $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_tupling}$ and from which we try to obtain $P_{r\_tupling}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_tupling}$ becomes:

$$\begin{array}{l} \forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee (Xs = [X_1] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge Y = I_1) \\ \vee (Xs = [X_1, X_2, \ldots, X_q] \wedge\ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i)\ \wedge I_1 = Y_1 \wedge\ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)\ \wedge Y = I_q)] \end{array}$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\begin{array}{l} \forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee (Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge TY = e \wedge \mathcal{O}_c(I_1, TY, Y)) \\ \vee (Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge\ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i)\ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge \\ \bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)\ \wedge TY = I_q \wedge \mathcal{O}_c(I_1, TY, Y))] \end{array}$$

where $q \geq 2$.

By folding using $S_{r\_tupling}$, and renaming:

$$\begin{array}{l} \forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee (Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge r\_tupling(TXs, TY) \wedge \mathcal{O}_c(HY, TY, Y))] \end{array}$$

By taking the 'decompletion':

$$\begin{array}{ll} \textit{clause } 1: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [\,], Y = e \\ \textit{clause } 2: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], r(X, HY), \\ & \quad r\_tupling(TXs, TY), compose(HY, TY, Y) \end{array}$$

By unfolding clause 2 wrt $r(X, HY)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} \textit{clause } 3: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad minimal(X), \\ & \quad r\_tupling(TXs, TY), \\ & \quad solve(X, HY), compose(HY, TY, Y) \\ \textit{clause } 4: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\ & \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y) \end{array}$$

By introducing

$$(minimal(TX_1) \wedge \ldots \wedge minimal(TX_t))\vee$$
$$((minimal(TX_1) \wedge \ldots \wedge minimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))\vee$$
$$((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (minimal(TX_p) \wedge \ldots \wedge minimal(TX_t)))\vee$$
$$((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))$$

in clause 4, using applicability condition (4):

$$
\begin{array}{ll}
clause\ 5: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_t), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y) \\
clause\ 6: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y)
\end{array}
$$

$$
\begin{array}{ll}
clause\ 7: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p),\ldots,minimal(TX_t), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y) \\
clause\ 8: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y)
\end{array}
$$

By t times unfolding clause 5 wrt $r(TX_1,TY_1),\ldots,r(TX_t,TY_t)$ using $DCLR$, and simplifying using condition (4):

$$
\begin{aligned}
\mathit{clause}\ 9: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
\mathit{clause}\ 10: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$$
\begin{aligned}
\mathit{clause}\ 11: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
\mathit{clause}\ 12: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad I_{p+1} = I_p, \ldots, I_{t+1} = I_t, \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
\mathit{clause}\ 13: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad r\_tupling(TXs, TY), \\
& \quad process(HX, HY), compose(HY, TY, Y)
\end{aligned}
$$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCLR$, and simplifying using condition (4):

$$
\begin{array}{ll}
clause\ 14: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad solve(TX_1,TY_1),\ldots,solve(TX_{p-1},TY_{p-1}), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t) \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y)
\end{array}
$$

By deleting one of the $minimal(TX_1),\ldots,minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{array}{ll}
clause\ 15: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad solve(TX_1,TY_1),\ldots,solve(TX_{p-1},TY_{p-1}), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t) \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs,TY), compose(HY,TY,Y)
\end{array}
$$

By rewriting clause 15 using applicability condition (1):

$$
\begin{array}{ll}
clause\ 16: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad solve(TX_1,TY_1),\ldots,solve(TX_{p-1},TY_{p-1}), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t) \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), HY = I_p, \\
& \quad compose(TY_p,TY_{p+1},I_{p+1}), \\
& \quad compose(I_{p+1},TY_{p+2},I_{p+2}),\ldots,compose(I_{t-1},TY_t,I_t), \\
& \quad r\_tupling(TXs,TTY), compose(I_t,TTY,TY), \\
& \quad compose(HY,TY,Y)
\end{array}
$$

By $t-p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 17: & r\_tupling(Xs,Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad solve(TX_1,TY_1),\ldots,solve(TX_{p-1},TY_{p-1}), \\
& \quad r\_tupling([TX_p,\ldots,TX_t|TXs],TY), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), HY = I_p, \\
& \quad compose(HY,TY,Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 18: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& \quad r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& \quad compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 19: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& \quad r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& \quad I_0 = e, \\
& \quad I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& \quad compose(HY, TY, Y)
\end{array}
$$

By simplification:

$$
\begin{array}{ll}
clause\ 20: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& \quad process(HX, HY), compose(HY, TY, Y)
\end{array}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{array}{ll}
clause\ 21: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 22: & r\_tupling(Xs,Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 23: & r\_tupling(Xs,Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\
& process(HX, HHY), compose(K_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability conditions (1) and (2):

$$
\begin{array}{ll}
clause\ 24: & r\_tupling(Xs,Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{array}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$$
\begin{aligned}
\textit{clause } 25: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \land decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$$
\begin{aligned}
\textit{clause } 26: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$
\begin{aligned}
\textit{clause } 27: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By folding clause 27 using $DCLR$:

$$
\begin{array}{ll}
clause\ 28: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI, Y)
\end{array}
$$

By folding clause 28 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 29: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& \quad r\_tupling([N|TXs], TI), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI, Y)
\end{array}
$$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 30: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r\_tupling([TX_1, \ldots, TX_{p-1}, N|TXs], Y)
\end{array}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$$
\begin{array}{ll}
clause\ 31: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 32: & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 33: & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\
& process(HX, HHY), compose(K_{p-1}, HHY, K_p), \\
& compose(K_p, e, K_{p+1}), \ldots, compose(K_t, e, K_{t+1}), compose(K_{t+1}, e, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability conditions (1) and (2):

$$
\begin{array}{ll}
clause\ 34: & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$$
\begin{aligned}
clause\ 35:\ \ & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$$
\begin{aligned}
clause\ 36:\ \ & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_t)$:

$$
\begin{aligned}
clause\ 37: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& \quad NHY = I_{t+1}, \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By folding clause 37 using $DCLR$:

$$
\begin{aligned}
clause\ 38: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (1):

$$
\begin{aligned}
clause\ 39: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, TI_2, Y), compose(NHY, TI_1, TI_2), \\
& \quad r\_tupling(TXs, TY), compose(K_{t-1}, TY, TI_1)
\end{aligned}
$$

By $t - p + 1$ times folding clause 39 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 40: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI_2, Y), compose(NHY, TI_1, TI_2), \\
& \quad r\_tupling([TX_p, \ldots, TX_t|TXs], TI_1)
\end{aligned}
$$

By folding clause 40 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 41: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI_2, Y), \\
& \quad r\_tupling([N, TX_p, \ldots, TX_t|TXs], TI_1)
\end{aligned}
$$

By $p-1$ times folding clause 41 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 42: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r\_tupling([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], TI_1)
\end{aligned}
$$

Clauses 1, 3, 13, 20, 30 and 42 are the clauses of $P_{r\_tupling}$. Therefore $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$, we do a backward proof that we begin with P_r in TG and from which we try to obtain S_r.

The procedure P_r for r in TG is:

$$r(X, Y) \leftarrow \quad r\_tupling([X], Y)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_tupling([X], Y)]$$

By unfolding the 'completion' above wrt $r\_tupling([X], Y)$ using $S_{r\_tupling}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$.
Therefore, TG is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 2 The generalization schema TG_2, which is given below, is correct.

$TG_2 : \langle\ DCRL, TG, A_{t2}, O_{t212}, O_{t221}\ \rangle$ where

A_{t2} : (1) *compose* is associative

(2) *compose* has e as the left and right identity element, where e appears in $DCRL$

(3) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \land minimal(X) \Rightarrow \mathcal{O}_r(X, e)$

(4) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{t212} : partial evaluation of the conjunction $process(HX, HY), compose(HY, TY, Y)$ results in the introduction of a non-recursive relation

O_{t221} : partial evaluation of the conjunction $process(HX, HY), compose(HY, I_p, I_{p-1})$ results in the introduction of a non-recursive relation

where the template TG is Logic Program Template 2 in Theorem 1 and the template $DCRL$ is Logic Program Template 3 below:

Logic Program Template 3

```
r(X, Y) ←
    minimal(X),
    solve(X, Y)
r(X, Y) ←
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    r(TX1, TY1), ..., r(TXt, TYt),
    It+1 = e,
    compose(TYt, It+1, It), ..., compose(TYp, Ip+1, Ip),
    process(HX, HY), compose(HY, Ip, Ip-1),
    compose(TYp-1, Ip-1, Ip-2), ..., compose(TY1, I1, I0),
    Y = I0
```

and the specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

and the specification $S_{r\_tupling}$ of relation *r_tupling* is:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ \ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow$$
$$(Xs = [\,] \land Y = e)$$
$$\lor (Xs = [X_1, \ldots, X_q] \land \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \ \land I_1 = Y_1 \land \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \ \land Y = I_q)]$$

Proof 2 To prove the correctness of the generalization schema TG_2, by Definition 10, we have to prove that templates $DCRL$ and TG are *equivalent* wrt S_r under the applicability conditions A_{t2}. By Definition 5, the templates $DCRL$ and TG are *equivalent* wrt S_r under the applicability conditions A_{t2} iff $DCRL$ is *equivalent to* TG wrt the specification S_r provided that the conditions in A_{t2} hold. By Definition 4, $DCRL$ is *equivalent to* TG wrt the specification S_r iff the following two conditions hold:

(a) $DCRL$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCRL$.

(b) TG is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by tupling generalization of P.

In program transformation, we assume that the input program, here template $DCRL$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: TG is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$, where $P_{r\_tupling}$ is the procedure for $r\_tupling$, and P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$, where P_r is the procedure for r.

To prove that $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_tupling}$ and from which we try to obtain $P_{r\_tupling}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_tupling}$ becomes:

$$\begin{array}{l}\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee(Xs = [X_1] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge Y = I_1) \\ \vee(Xs = [X_1, X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge Y = I_q)]\end{array}$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\begin{array}{l}\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee(Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge TY = e \wedge \mathcal{O}_c(I_1, TY, Y)) \\ \vee(Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge \\ \bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge TY = I_q \wedge \mathcal{O}_c(I_1, TY, Y))]\end{array}$$

where $q \geq 2$.

By folding using $S_{r\_tupling}$, and renaming:

$$\begin{array}{l}\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_tupling(Xs, Y) \Leftrightarrow \\ (Xs = [\,] \wedge Y = e) \\ \vee(Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge r\_tupling(TXs, TY) \wedge \mathcal{O}_c(HY, TY, Y))]\end{array}$$

By taking the 'decompletion':

$$\begin{array}{ll}clause\ 1: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [\,], Y = e \\ clause\ 2: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], r(X, HY), \\ & \quad r\_tupling(TXs, TY), compose(HY, TY, Y)\end{array}$$

By unfolding clause 2 wrt $r(X, HY)$ using $DCRL$, and using the assumption that $DCRL$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll}clause\ 3: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad minimal(X), \\ & \quad r\_tupling(TXs, TY), \\ & \quad solve(X, HY), compose(HY, TY, Y)\end{array}$$

$$\begin{array}{ll}clause\ 4: & r\_tupling(Xs, Y) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_{t+1} = e, \\ & \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\ & \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\ & \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\ & \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)\end{array}$$

By introducing

$$\begin{array}{c}(minimal(TX_1) \wedge \ldots \wedge minimal(TX_t)) \vee \\ ((minimal(TX_1) \wedge \ldots \wedge minimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t))) \vee \\ ((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (minimal(TX_p) \wedge \ldots \wedge minimal(TX_t))) \vee \\ ((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))\end{array}$$

in clause 4, using applicability condition (4):

$$
\begin{array}{ll}
clause\ 5: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y) \\
clause\ 6: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

$$
\begin{array}{ll}
clause\ 7: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, TX_2), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y) \\
clause\ 8: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, TX_2), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By t times unfolding clause 5 wrt $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$ using $DCRL$, and simplifying using condition (4):

$$
\begin{array}{ll}
clause\ 9: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By using applicability condition (3):

$$\begin{aligned}
clause\ 10:\ \ & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& HY = I_0, r_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$$\begin{aligned}
clause\ 11:\ \ & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& HY = I_0, r_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}$$

By using applicability condition (2):

$$\begin{aligned}
clause\ 12:\ \ & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& I_t = I_{t+1}, \ldots, I_p = I_{p+1}, \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& I_{p-2} = I_{p-1}, \ldots, I_0 = I_1, \\
& HY = I_0, r_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}$$

By simplification:

$$\begin{aligned}
clause\ 13:\ \ & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& r_tupling(TXs, TY), \\
& process(HX, HY), compose(HY, TY, Y)
\end{aligned}$$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCRL$, and simplifying using condition (4):

$$
\begin{aligned}
\mathit{clause}\ 14: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HY), compose(HY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{aligned}
\mathit{clause}\ 15: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HY), compose(HY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By rewriting clause 15 using applicability conditions (1) and (2):

$$
\begin{aligned}
\mathit{clause}\ 16: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(TY_p, TY_{p+1}, I_{p+1}), \\
& compose(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, compose(I_{t-1}, TY_t, I_t), \\
& r\_tupling(TXs, TTY), compose(I_t, TTY, TY), \\
& compose(HY, TY, Y)
\end{aligned}
$$

By $t - p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{aligned}
\mathit{clause}\ 17: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 18: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 19: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& I_0 = e, \\
& I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(HY, TY, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 20: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r\_tupling([TX_p, \ldots, TX_t|TXs], TY), \\
& process(HX, HY), compose(HY, TY, Y)
\end{aligned}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{aligned}
clause\ 21: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HY), compose(HY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 22: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 23: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, K_1), compose(e, K_1, K_2), \ldots, compose(e, K_{p-2}, K_{p-1}), \\
& compose(TY_{p-1}, K_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$$
\begin{aligned}
clause\ 24: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$$
\begin{aligned}
\textit{clause } 25: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \land decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$$
\begin{aligned}
\textit{clause } 26: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$
\begin{aligned}
\textit{clause } 27: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}
$$

By folding clause 27 using $DCRL$:

$$\begin{aligned}
clause\ 28:\quad & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& r_tupling(TXs, TY), compose(HY, TY, TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}$$

By folding clause 28 using clauses 1 and 2:

$$\begin{aligned}
clause\ 29:\quad & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& r_tupling([N|TXs], TI), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, TI, Y)
\end{aligned}$$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$$\begin{aligned}
clause\ 30:\quad & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r_tupling([TX_1, \ldots, TX_{p-1}, N|TXs], Y)
\end{aligned}$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$$\begin{aligned}
clause\ 31:\quad & r_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, TX_2), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}$$

By using applicability condition (3):

$$
\begin{aligned}
\mathit{clause}\ 32: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, TX_2), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
\mathit{clause}\ 33: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& compose(e, I_p, K_{t+1}), \\
& compose(e, K_{t+1}, K_t), \ldots, compose(e, K_{p+1}, K_p), \\
& process(HX, HHY), compose(HHY, K_p, K_{p-1}), \\
& compose(e, K_{p-1}, K_{p-2}), \ldots, compose(e, K_1, K_0), \\
& compose(e, K_0, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$$
\begin{aligned}
\mathit{clause}\ 34: \quad & r\_tupling(Xs, Y) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$$
\begin{array}{ll}
\mathit{clause}\ 35: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad NHY = I_0, \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$$
\begin{array}{ll}
\mathit{clause}\ 36: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad NHY = I_0, \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{array}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_t)$:

$$
\begin{aligned}
\textit{clause } 37: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad NHY = I_0, \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By folding clause 37 using $DCRL$:

$$
\begin{aligned}
\textit{clause } 38: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, NHY, TI), compose(TI, K_{t-1}, HY), \\
& \quad r\_tupling(TXs, TY), compose(HY, TY, Y)
\end{aligned}
$$

By using applicability condition (1):

$$
\begin{aligned}
\textit{clause } 39: \quad & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{p-2}, TI_2, Y), compose(NHY, TI_1, TI_2), \\
& \quad r\_tupling(TXs, TY), compose(K_{t-1}, TY, TI_1)
\end{aligned}
$$

By $t - p + 1$ times folding clause 39 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 40: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI_2, Y), compose(NHY, TI_1, TI_2), \\
& \quad r\_tupling([TX_p, \ldots, TX_t|TXs], TI_1)
\end{array}
$$

By folding clause 40 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 41: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, TI_2, Y), \\
& \quad r\_tupling([N, TX_p, \ldots, TX_t|TXs], TI_1)
\end{array}
$$

By $p-1$ times folding clause 41 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 42: & r\_tupling(Xs, Y) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r\_tupling([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], TI_1)
\end{array}
$$

Clauses 1, 3, 13, 20, 30 and 42 are the clauses of $P_{r\_tupling}$. Therefore $P_{r\_tupling}$ is steadfast wrt $S_{r\_tupling}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$, we do a backward proof that we begin with P_r in TG and from which we try to obtain S_r.
The procedure P_r for r in TG is:

$$r(X, Y) \leftarrow \quad r\_tupling([X], Y)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_tupling([X], Y)]$$

By unfolding the 'completion' above wrt $r\_tupling([X], Y)$ using $S_{r\_tupling}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_tupling}\}$.
Therefore, TG is also steadfast wrt S_r in $\mathcal{S}$. □

3 Proofs of the Descending Generalization Schemas

Theorem 3 The generalization schema DG_1, which is given below, is correct.

DG_1 : $\langle$ $DCLR$, $DGLR$, A_{dg1}, O_{dg112}, O_{dg121} $\rangle$ where

A_{dg1} : (1) *compose* is associative
(2) *compose* has e as the left identity element,
where e appears in $DCLR$ and $DGLR$

O_{dg112} : - *compose* has e as the right identity element,
where e appears in $DCLR$ and $DGLR$
and $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
- partial evaluation of the conjunction
$process(HX, HY), compose(A_{p-1}, HY, A_p)$
results in the introduction of a non-recursive relation

O_{dg121} : - partial evaluation of the conjunction
$process(HX, HY), compose(I_{p-1}, HY, I_p)$
results in the introduction of a non-recursive relation

where the template $DCLR$ is Logic Program Template 1 in Section 2 and the template $DGLR$ is Logic Program Template 4 below:

Logic Program Template 4

```
r(X,Y) ←
    r_descending1(X,Y,e)
r_descending1(X,Y,A) ←
    minimal(X),
    solve(X,S), compose(A,S,Y)
r_descending1(X,Y,A) ←
    nonMinimal(X),
    decompose(X,HX,TX1,...,TXt),
    compose(A,e,A0),
    r_descending1(TX1,A1,A0),...,r_descending1(TXp-1,Ap-1,Ap-2),
    process(HX,HY), compose(Ap-1,HY,Ap),
    r_descending1(TXp,Ap+1,Ap),...,r_descending1(TXt,At+1,At),
    Y = At+1
```

and the specification S_r of relation r is:

$$\forall X : \mathcal{X}.\ \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

and the specification $S_{r\_descending_1}$ of relation *r_descending*$_1$ is:

$$\forall X : \mathcal{X}.\ \forall Y, A : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r\_descending_1(X, Y, A)\ \Leftrightarrow\ \exists S : \mathcal{Y}.\ \mathcal{O}_r(X, S) \wedge \mathcal{O}_c(A, S, Y)]$$

Proof 3 To prove the correctness of the generalization schema DG_1, by Definition 10, we have to prove that templates $DCLR$ and $DGLR$ are *equivalent* wrt S_r under the applicability conditions A_{dg1}. By Definition 5, the templates $DCLR$ and $DGLR$ are *equivalent* wrt S_r under the applicability conditions A_{dg1} iff $DCLR$ is *equivalent to* $DGLR$ wrt the specification S_r provided that the conditions in A_{dg1} hold. By Definition 4, $DCLR$ is *equivalent to* $DGLR$ wrt the specification S_r iff the following two conditions hold:

(a) $DCLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}, S_{nonMinimal}, S_{solve}, S_{process}, S_{decompose}, S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCLR$.

(b) $DGLR$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by descending generalization of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGLR$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$, where $P_{r\_descending_1}$ is the procedure for $r\_descending_1$, and P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$, where P_r is the procedure for r.

To prove that $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_descending_1}$ and from which we try to obtain $P_{r\_descending_1}$.

By taking the 'decompletion' of $S_{r\_descending_1}$:

$$clause\ 1:\quad r\_descending_1(X,Y,A) \leftarrow r(X,S), compose(A,S,Y)$$

By unfolding clause 1 wrt $r(X,S)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll}
clause\ 2: & r_descending_1(X,Y,A) \leftarrow \\
& \quad minimal(X), \\
& \quad solve(X,S), compose(A,S,Y) \\
clause\ 3: & r_descending_1(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0,TS_1,I_1),\ldots,compose(I_{p-2},TS_{p-1},I_{p-1}), \\
& \quad process(HX,HS), compose(I_{p-1},HS,I_p), \\
& \quad compose(I_p,TS_p,I_{p+1}),\ldots,compose(I_t,TS_t,I_{t+1}), \\
& \quad S = I_{t+1}, compose(A,S,Y)
\end{array}$$

By using applicability condition (1) on clause 3:

$$\begin{array}{ll}
clause\ 4: & r_descending_1(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\
& \quad compose(A,e,A_0), \\
& \quad compose(A_0,TS_1,A_1),\ldots,compose(A_{p-2},TS_{p-1},A_{p-1}), \\
& \quad process(HX,HS), compose(A_{p-1},HS,A_p), \\
& \quad compose(A_p,TS_p,A_{p+1}),\ldots,compose(A_t,TS_t,A_{t+1}), \\
& \quad Y = A_{t+1}
\end{array}$$

By t times folding clause 4 using clause 1:

$$\begin{array}{ll}
clause\ 5: & r_descending_1(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad compose(A,e,A_0), \\
& \quad r_descending_1(TX_1,A_1,A_0),\ldots,r_descending_1(TX_{p-1},A_{p-1},A_{p-2}), \\
& \quad process(HX,HY), compose(A_{p-1},HY,A_p), \\
& \quad r_descending_1(TX_p,A_{p+1},A_p),\ldots,r_descending_1(TX_t,A_{t+1},A_t), \\
& \quad Y = A_{t+1}
\end{array}$$

Clauses 2 and 5 are the clauses of the $P_{r\_descending_1}$. Therefore $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$, we do a backward proof that we begin with P_r in $DGLR$ and from which we try to obtain S_r.

The procedure P_r for r in $DGLR$ is:

$$r(X,Y) \leftarrow \quad r\_descending_1(X,Y,e)$$

By taking the 'completion':

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_descending_1(X,Y,e)]$$

By unfolding the 'completion' above wrt $r\_descending_1(X,Y,e)$ using $S_{r\_descending_1}$:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(e,S,Y)]$$

By using applicability condition (2):

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge S = Y]$$

By simplification:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$.
Therefore, $DGLR$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 4 The generalization schema DG_2, which is given below, is correct.

DG_2 : ⟨ $DCLR$, $DGRL$, A_{dg2}, O_{dg212}, O_{dg221} ⟩ where
A_{dg2} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element,
where e appears in $DCLR$ and $DGRL$
O_{dg212} : - $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X,e)$
- partial evaluation of the conjunction
$process(HX,HY), compose(HY,A_p,A_{p-1})$
results in the introduction of a non-recursive relation
O_{dg221} : - partial evaluation of the conjunction
$process(HX,HY), compose(I_{p-1},HY,I_p)$
results in the introduction of a non-recursive relation

where the template $DCLR$ is Logic Program Template 1 in Section 2 and the template $DGRL$ is Logic Program Template 5 below:

Logic Program Template 5

```
r(X,Y) ←
    r_descending2(X,Y,e)
r_descending2(X,Y,A) ←
    minimal(X),
    solve(X,S), compose(S,A,Y)
r_descending2(X,Y,A) ←
    nonMinimal(X),
    decompose(X,HX,TX1,...,TXt),
    compose(e,A,At+1),
    r_descending2(TXt,At,At+1),...,r_descending2(TXp,Ap,Ap+1),
    process(HX,HY), compose(HY,Ap,Ap-1),
    r_descending2(TXp-1,Ap-2,Ap-1),...,r_descending2(TX1,A0,A1),
    Y = A0
```

and the specification S_r of relation r is:

$$\forall X:\mathcal{X}.\ \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

and the specification $S_{r\_descending_2}$ of relation $r\_descending_2$ is:

$$\forall X:\mathcal{X}.\ \forall Y,A:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r\_descending_2(X,Y,A)\ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,A,Y)]$$

Proof 4 To prove the correctness of the generalization schema DG_2, by Definition 10, we have to prove that templates $DCLR$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dg2}. By Definition 5, the templates $DCLR$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dg2} iff $DCLR$ is *equivalent to* $DGRL$ wrt the specification S_r provided that the conditions in A_{dg2} hold. By Definition 4, $DCLR$ is *equivalent to* $DGRL$ wrt the specification S_r iff the following two conditions hold:

(*a*) $DCLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{process}$, $S_{decompose}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCLR$.

(b) $DGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by descending generalization of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, where $P_{r\_descending_2}$ is the procedure for $r\_descending_2$, and P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_descending_2}$ and from which we try to obtain $P_{r\_descending_2}$.

By taking the 'decompletion' of $S_{r\_descending_2}$:

$$clause\ 1: \quad r\_descending_2(X, Y, A) \leftarrow r(X, S), compose(S, A, Y)$$

By unfolding clause 1 wrt $r(X, S)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} clause\ 2: & r\_descending_2(X, Y, A) \leftarrow \\ & \quad minimal(X), \\ & \quad solve(X, S), compose(S, A, Y) \\ clause\ 3: & r\_descending_2(X, Y, A) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TS_1, I_1), \ldots, compose(I_{p-2}, TS_{p-1}, I_{p-1}), \\ & \quad process(HX, HS), compose(I_{p-1}, HS, I_p), \\ & \quad compose(I_p, TS_p, I_{p+1}), \ldots, compose(I_t, TS_t, I_{t+1}), \\ & \quad S = I_{t+1}, compose(S, A, Y) \end{array}$$

By using applicability condition (1) on clause 3:

$$\begin{array}{ll} clause\ 4: & r\_descending_2(X, Y, A) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\ & \quad compose(TS_t, A, A_t), \ldots, compose(TS_p, A_{p+1}, A_p), \\ & \quad process(HX, HY), compose(HY, A_p, A_{p-1}), \\ & \quad compose(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TS_1, A_1, A_0), \\ & \quad compose(e, A_0, Y) \end{array}$$

By using applicability condition (2) on clause 4:

$$\begin{array}{ll} clause\ 5: & r\_descending_2(X, Y, A) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\ & \quad compose(TS_t, A, A_t), \ldots, compose(TS_p, A_{p+1}, A_p), \\ & \quad process(HX, HY), compose(HY, A_p, A_{p-1}), \\ & \quad compose(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TS_1, A_1, A_0), \\ & \quad Y = A_0 \end{array}$$

By using applicability condition (2) on clause 5 and introducing a new, i.e. existentially quantified, variable A_{t+1}:

$$\begin{array}{ll} clause\ 6: & r\_descending_2(X, Y, A) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\ & \quad compose(e, A, A_{t+1}), \\ & \quad compose(TS_t, A_{t+1}, A_t), \ldots, compose(TS_p, A_{p+1}, A_p), \\ & \quad process(HX, HY), compose(HY, A_p, A_{p-1}), \\ & \quad compose(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TS_1, A_1, A_0), \\ & \quad Y = A_0 \end{array}$$

By t times folding clause 6 using clause 1:

$$
\begin{aligned}
\mathit{clause}\ 7:\quad & r\_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad compose(e,A,A_{t+1}), \\
& \quad r\_descending_2(TX_t,A_t,A_{t+1}),\ldots,r\_descending_2(TX_p,A_p,A_{p+1}), \\
& \quad process(HX,HY), compose(HY,A_p,A_{p-1}), \\
& \quad r\_descending_2(TX_{p-1},A_{p-2},A_{p-1}),\ldots,r\_descending_2(TX_1,A_0,A_1), \\
& \quad Y = A_0
\end{aligned}
$$

Clauses 2 and 7 are the clauses of $P_{r\_descending_2}$. Therefore $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, we do a backward proof that we begin with P_r in $DGRL$ and from which we try to obtain S_r.

The procedure P_r for r in $DGRL$ is:

$$r(X,Y) \leftarrow \quad r\_descending_2(X,Y,e)$$

By taking the 'completion':

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_descending_2(X,Y,e)]$$

By unfolding the 'completion' above wrt $r\_descending_2(X,Y,e)$ using $S_{r\_descending_2}$:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,e,Y)]$$

By using applicability condition (2):

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge S = Y]$$

By simplification:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$.

Therefore, $DGRL$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 5 The generalization schema DG_3, which is given below, is correct.

DG_3 : ⟨ $DCRL$, $DGRL$, A_{dg3}, O_{dg312}, O_{dg321} ⟩ where

A_{dg3} : (1) *compose* is associative
(2) *compose* has e as the right identity element,
where e appears in $DCRL$ and $DGRL$

O_{dg312} : - *compose* has e as the left identity element,
where e appears in $DCLR$ and $DGRL$
and $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X,e)$
- partial evaluation of the conjunction
$process(HX,HY), compose(HY,A_p,A_{p-1})$
results in the introduction of a non-recursive relation

O_{dg321} : - partial evaluation of the conjunction
$process(HX,HY), compose(HY,I_p,I_{p-1})$
results in the introduction of a non-recursive relation

where the template $DGRL$ is Logic Program Template 5 in Theorem 4 and the template $DCRL$ is Logic Program Template 3 in Section 2.

The specification S_r of relation r is:

$$\forall X:\mathcal{X}.\ \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

The specification $S_{r\_descending_2}$ of relation $r\_descending_2$ is:

$$\forall X:\mathcal{X}.\ \forall Y,A:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r\_descending_2(X,Y,A) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,A,Y)]$$

Proof 5 To prove the correctness of the generalization schema DG_3, by Definition 10, we have to prove that templates $DCRL$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dg3}. By Definition 5, the templates $DCRL$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dg3} iff $DCRL$ is *equivalent to* $DGRL$ wrt the specification S_r provided that the conditions in A_{dg3} hold. By Definition 4, $DCRL$ is *equivalent to* $DGRL$ wrt the specification S_r iff the following two conditions hold:

(a) $DCRL$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}, S_{nonMinimal}, S_{solve}, S_{process}, S_{decompose}, S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCRL$.

(b) $DGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by descending generalization of P.

In program transformation, we assume that the input program, here template $DCRL$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, where $P_{r\_descending_2}$ is the procedure for $r\_descending_2$, and P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_descending_2}$ and from which we try to obtain $P_{r\_descending_2}$.

By taking the 'decompletion' of $S_{r\_descending_2}$:

$$clause\ 1:\quad r\_descending_2(X,Y,A) \leftarrow r(X,S), compose(S,A,Y)$$

By unfolding clause 1 wrt $r(X,S)$ using $DCRL$, and using the assumption that $DCRL$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll}
clause\ 2: & r_descending_2(X,Y,A) \leftarrow \\
& \quad minimal(X), \\
& \quad solve(X,S), compose(S,A,Y) \\
clause\ 3: & r_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TS_t,I_{t+1},I_t),\ldots,compose(TS_p,I_{p+1},I_p), \\
& \quad process(HX,HS), compose(HS,I_p,I_{p-1}), \\
& \quad compose(TS_{p-1},I_{p-1},I_{p-2}),\ldots,compose(TS_1,I_1,I_0), \\
& \quad S = I_0, compose(S,A,Y)
\end{array}$$

By using applicability condition (1) on clause 3:

$$\begin{array}{ll}
clause\ 4: & r_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\
& \quad compose(e,A,A_{t+1}), \\
& \quad compose(TS_t,A_{t+1},A_t),\ldots,compose(TS_p,A_{p+1},A_p), \\
& \quad process(HX,HY), compose(HY,A_p,A_{p-1}), \\
& \quad compose(TS_{p-1},A_{p-1},A_{p-2}),\ldots,compose(TS_1,A_1,A_0), \\
& \quad Y = A_0
\end{array}$$

By t times folding clause 4 using clause 1:

$$\begin{array}{ll}
clause\ 5: & r_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad compose(e,A,A_{t+1}), \\
& \quad r_descending_2(TX_t,A_t,A_{t+1}),\ldots,r_descending_2(TX_p,A_p,A_{p+1}), \\
& \quad process(HX,HY), compose(HY,A_p,A_{p-1}), \\
& \quad r_descending_2(TX_{p-1},A_{p-2},A_{p-1}),\ldots,r_descending_2(TX_1,A_0,A_1), \\
& \quad Y = A_0
\end{array}$$

Clauses 2 and 5 are the clauses of $P_{r\_descending_2}$. Therefore $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, we do a backward proof that we begin with P_r in $DGRL$ and from which we try to obtain S_r.
The procedure P_r for r in $DGRL$ is:

$$r(X,Y) \leftarrow \quad r\_descending_2(X,Y,e)$$

By taking the 'completion':

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_descending_2(X,Y,e)]$$

By unfolding the 'completion' above wrt $r\_descending_2(X,Y,e)$ using $S_{r\_descending_2}$:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,e,Y)]$$

By using applicability condition (2):

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge S = Y]$$

By simplification:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$.
Therefore, $DGRL$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 6 The generalization schema DG_4, which is given below, is correct.

DG_4 : $\langle$ $DCRL$, $DGLR$, A_{dg4}, O_{dg412}, O_{dg421} $\rangle$ where
A_{dg4} : - *compose* is associative
- *compose* has e as the left and right identity element,
where e appears in $DCRL$ and $DGLR$
O_{dg412} : - $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X,e)$
- partial evaluation of the conjunction
$process(HX,HY), compose(A_{p-1},HY,A_p)$
results in the introduction of a non-recursive relation
O_{dg421} : - partial evaluation of the conjunction
$process(HX,HY), compose(HY,I_p,I_{p-1})$
results in the introduction of a non-recursive relation

where the template $DCRL$ is Logic Program Template 3 in Section 2 and the template $DGLR$ is Logic Program Template 4 in Theorem 3.
The specification S_r of relation r is:

$$\forall X:\mathcal{X}.\ \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

The specification $S_{r\_descending_1}$ of relation $r\_descending_1$ is:

$$\forall X:\mathcal{X}.\ \forall Y,A:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r\_descending_1(X,Y,A) \ \Leftrightarrow\ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(A,S,Y)]$$

Proof 6 To prove the correctness of the generalization schema DG_4, by Definition 10, we have to prove that templates $DCRL$ and $DGLR$ are *equivalent* wrt S_r under the applicability conditions A_{dg4}. By Definition 5, the templates $DCRL$ and $DGLR$ are *equivalent* wrt S_r under the applicability conditions A_{dg4} iff $DCRL$ is *equivalent to* $DGLR$ wrt the specification S_r provided that the conditions in A_{dg4} hold. By Definition 4, $DCRL$ is *equivalent to* $DGLR$ wrt the specification S_r iff the following two conditions hold:

(a) $DCRL$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{process}$, $S_{decompose}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCRL$.

(*b*) $DGLR$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ to $\mathcal{S}$ when Q is obtained by descending generalization of P.

In program transformation, we assume that the input program, here template $DCRL$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGLR$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$, where $P_{r\_descending_1}$ is the procedure for $r\_descending_1$, and P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$, where P_r is the procedure for r.

To prove that $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_descending_1}$ and from which we try to obtain $P_{r\_descending_1}$.

By taking the 'decompletion' of $S_{r\_descending_1}$:

$$clause\ 1: \quad r\_descending_1(X,Y,A) \leftarrow r(X,S), compose(A,S,Y)$$

By unfolding clause 1 wrt $r(X,S)$ using $DCRL$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} clause\ 2: & r\_descending_1(X,Y,A) \leftarrow \\ & \quad minimal(X), \\ & \quad solve(X,S), compose(A,S,Y) \\ clause\ 3: & r\_descending_1(X,Y,A) \leftarrow \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\ & \quad I_{t+1} = e, \\ & \quad compose(TS_t,I_{t+1},I_t),\ldots,compose(TS_p,I_{p+1},I_p), \\ & \quad process(HX,HS), compose(HS,I_p,I_{p-1}), \\ & \quad compose(TS_{p-1},I_{p-1},I_{p-2}),\ldots,compose(TS_1,I_1,I_0), \\ & \quad S = I_0, compose(A,S,Y) \end{array}$$

By using applicability condition (1) on clause 3:

$$\begin{array}{ll} clause\ 4: & r\_descending_1(X,Y,A) \leftarrow \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\ & \quad compose(A,TS_1,A_1),\ldots,compose(A_{p-2},TS_{p-1},A_{p-1}), \\ & \quad process(HX,HS), compose(A_{p-1},HS,A_p), \\ & \quad compose(A_p,TS_p,A_{p+1}),\ldots,compose(A_t,TS_t,A_{t+1}), \\ & \quad compose(A_{t+1},e,Y) \end{array}$$

By using applicability condition (2) on clause 4:

$$\begin{array}{ll} clause\ 5: & r\_descending_1(X,Y,A) \leftarrow \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\ & \quad compose(A,TS_1,A_1),\ldots,compose(A_{p-2},TS_{p-1},A_{p-1}), \\ & \quad process(HX,HS), compose(A_{p-1},HS,A_p), \\ & \quad compose(A_p,TS_p,A_{p+1}),\ldots,compose(A_t,TS_t,A_{t+1}), \\ & \quad Y = A_{t+1} \end{array}$$

By using applicability condition (2) on clause 5 and introducing a new, i.e. existentially quantified, variable A_0:

$$\begin{array}{ll} clause\ 6: & r\_descending_1(X,Y,A) \leftarrow \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad r(TX_1,TS_1),\ldots,r(TX_t,TS_t), \\ & \quad compose(A,e,A_0), \\ & \quad compose(A_0,TS_1,A_1),\ldots,compose(A_{p-2},TS_{p-1},A_{p-1}), \\ & \quad process(HX,HS), compose(A_{p-1},HS,A_p), \\ & \quad compose(A_p,TS_p,A_{p+1}),\ldots,compose(A_t,TS_t,A_{t+1}), \\ & \quad Y = A_{t+1} \end{array}$$

By t times folding clause 6 using clause 1:

$$\begin{aligned}
\mathit{clause}\ 7: \quad & r_descending_1(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad compose(A,e,A_0), \\
& \quad r_descending_1(TX_1,A_1,A_0),\ldots,r_descending_1(TX_{p-1},A_{p-1},A_{p-2}), \\
& \quad process(HX,HY), compose(A_{p-1},HY,A_p), \\
& \quad r_descending_1(TX_p,A_{p+1},A_p),\ldots,r_descending_1(TX_t,A_{t+1},A_t), \\
& \quad Y = A_{t+1}
\end{aligned}$$

Clauses 2 and 7 are the clauses of the $P_{r\_descending_1}$. Therefore $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$, we do a backward proof that we begin with P_r in $DGLR$ and from which we try to obtain S_r.

The procedure P_r for r in $DGLR$ is:

$$r(X,Y) \leftarrow \quad r\_descending_1(X,Y,e)$$

By taking the 'completion':

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_descending_1(X,Y,e)]$$

By unfolding the 'completion' above wrt $r\_descending_1(X,Y,e)$ using $S_{r\_descending_1}$:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow \ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(e,S,Y)]$$

By using applicability condition (2):

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow \ \exists S:\mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge S = Y]$$

By simplification:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow \ \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_descending_1}\}$.
Therefore, $DGLR$ is also steadfast wrt S_r in $\mathcal{S}$. □

4 Proofs of the Simultaneous Tupling-and-Descending Generalization Schemas

Theorem 7 The generalization schema TDG_1, which is given below, is correct.

TDG_1 : ⟨ $DCLR$, $TDGLR$, A_{td1}, O_{td112}, O_{td121} ⟩ where

A_{td1} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element
(3) $\forall X:\mathcal{X}.\ \mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X,e)$
(4) $\forall X:\mathcal{X}.\ \mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{td112} : partial evaluation of the conjunction
$process(HX,HY), compose(A,HY,NewA)$
results in the introduction of a non-recursive relation

O_{td121} : partial evaluation of the conjunction
$process(HX,HY), compose(I_{p-1},HY,I_p)$
results in the introduction of a non-recursive relation

where the template $DCLR$ is Logic Program Template 1 in Section 2 and the template $TDGLR$ is Logic Program Template 6 below:

Logic Program Template 6

```
r(X, Y) ←
    r_td1([X], Y, e)
r_td1(Xs, Y, A) ←
    Xs = [ ],
    Y = A
r_td1(Xs, Y, A) ←
    Xs = [X|TXs],
    minimal(X),
    solve(X, HY),
    compose(A, HY, NewA),
    r_td1(TXs, Y, NewA)
r_td1(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    minimal(TX1), ..., minimal(TXt),
    process(HX, HY), compose(A, HY, NewA),
    r_td1(TXs, Y, NewA)
r_td1(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    minimal(TX1), ..., minimal(TXp-1),
    (nonMinimal(TXp); ...; nonMinimal(TXt)),
    process(HX, HY), compose(A, HY, NewA),
    r_td1([TXp, ..., TXt|TXs], Y, NewA)
r_td1(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    (nonMinimal(TX1); ...; nonMinimal(TXp-1)),
    minimal(TXp), ..., minimal(TXt),
    minimal(U1), ..., minimal(Up-1),
    decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt),
    r_td1([TX1, ..., TXp-1, N|TXs], Y, A)
r_td1(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    (nonMinimal(TX1); ...; nonMinimal(TXp-1)),
    (nonMinimal(TXp); ...; nonMinimal(TXt)),
    minimal(U1), ..., minimal(Ut),
    decompose(N, HX, U1, ..., Ut),
    r_td1([TX1, ..., TXp-1, N, TXp, ..., TXt|TXs], Y, A)
```

and the specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

and the specification $S_{r\_td_1}$:

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs,Y,A) \Leftrightarrow (Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge\ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i)\ \wedge I_1 = Y_1 \wedge\ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1},Y_i,I_i)$$
$$\wedge \mathcal{O}_c(A, I_q, I_{q+1}) \wedge Y = I_{q+1})]$$

Proof 7 To prove the correctness of the generalization schema TDG_1, by Definition 10, we have to prove that templates $DCLR$ and $TDGLR$ are *equivalent* wrt S_r under the applicability conditions A_{td1}. By Definition 5, the templates $DCLR$ and $TDGLR$ are *equivalent* wrt S_r under the applicability conditions A_{td1} iff $DCLR$ is *equivalent to* $TDGLR$ wrt the specification S_r provided that the conditions in A_{td1} hold. By Definition 4, $DCLR$ is *equivalent to* $TDGLR$ wrt the specification S_r iff the following two conditions hold:

(a) $DCLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCLR$.

(b) $TDGLR$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by simultaneous tupling-and-descending generalization of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $TDGLR$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$, where $P_{r\_td_1}$ is the procedure for r_td_1, and P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$, where P_r is the procedure for r.

To prove that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_td_1}$ and from which we try to obtain $P_{r\_td_1}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_td_1}$ becomes:

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs,Y,A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1] \wedge \mathcal{O}_r(X_1,Y_1) \wedge Y_1 = I_1 \wedge \mathcal{O}_c(A,I_1,I_2) \wedge Y = I_2)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge\ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i)\ \wedge I_1 = Y_1 \wedge\ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1},Y_i,I_i)\ \wedge$$
$$\mathcal{O}_c(A, I_q, I_{q+1}) \wedge Y = I_{q+1})]$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs,Y,A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1,Y_1) \wedge Y_1 = I_1 \wedge TY = A \wedge \mathcal{O}_c(A,I_1,NA) \wedge \mathcal{O}_c(NA,TY,Y))$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge\ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i)\ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge$$
$$\bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1},Y_i,I_i)\ \wedge TY = I_q \wedge \mathcal{O}_c(A,I_1,NA) \wedge \mathcal{O}_c(NA,TY,Y))]$$

where $q \geq 2$.

By folding using $S_{r\_td_1}$, and renaming:

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs,Y,A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X|TXs] \wedge \mathcal{O}_r(X,HY) \wedge \mathcal{O}_c(A,HY,NA) \wedge r\_td_1(TXs,Y,NA))]$$

By taking the 'decompletion':

clause 1 : $r\_td_1(Xs,Y,A) \leftarrow$
$Xs = [\,], Y = A$

clause 2 : $r\_td_1(Xs,Y,A) \leftarrow$
$Xs = [X|TXs], r(X,HY),$
$compose(A,HY,NA), r\_td_1(TXs,Y,NA)$

By unfolding clause 2 wrt $r(X, HY)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$
\begin{array}{ll}
clause\ 3: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad minimal(X), \\
& \quad solve(X, HY), compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA) \\
clause\ 4: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By introducing

$$
\begin{array}{c}
(minimal(TX_1) \wedge \ldots \wedge minimal(TX_t)) \vee \\
((minimal(TX_1) \wedge \ldots \wedge minimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t))) \vee \\
((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (minimal(TX_p) \wedge \ldots \wedge minimal(TX_t))) \vee \\
((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))
\end{array}
$$

in clause 4, using applicability condition (4):

$$
\begin{array}{ll}
clause\ 5: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA) \\
clause\ 6: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

$$
\begin{aligned}
\mathit{clause}\ 7:\quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA) \\
\mathit{clause}\ 8:\quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By t times unfolding clause 5 wrt $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$ using $DCLR$, and simplifying using condition (4):

$$
\begin{aligned}
\mathit{clause}\ 9:\quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
\mathit{clause}\ 10:\quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$$
\begin{array}{ll}
clause\ 11: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 12: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, e), \ldots, solve(TX_t, e), \\
& \quad I_0 = e, \\
& \quad I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad I_{p+1} = I_p, \ldots, I_{t+1} = I_t, \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By simplification:

$$
\begin{array}{ll}
clause\ 13: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad process(HX, HY), compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCLR$, and simplifying using condition (4):

$$
\begin{array}{ll}
clause\ 14: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& \quad r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{aligned}
\textit{clause } 15: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, compose(A, HY, NA), \\
& r\_d_1(TXs, Y, NA)
\end{aligned}
$$

By rewriting clause 15 using applicability condition (1):

$$
\begin{aligned}
\textit{clause } 16: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& compose(TY_p, TY_{p+1}, I_{p+1}), \\
& compose(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, compose(I_{t-1}, TY_t, I_t), \\
& compose(NA, I_t, NNA), \\
& r\_d_1(TXs, Y, NNA)
\end{aligned}
$$

By $t-p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 17: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& r\_d_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
\textit{clause } 18: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& r\_d_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 19: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& \quad I_0 = e, \\
& \quad I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& \quad compose(A, HY, NA), \\
& \quad r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{array}
$$

By simplification:

$$
\begin{array}{ll}
clause\ 20: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad process(HX, HY), compose(A, HY, NA), \\
& \quad r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{array}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{array}{ll}
clause\ 21: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 22: & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(A, HY, NA), \\
& \quad r\_td_1(TXs, Y, NA)
\end{array}
$$

By using applicability condition (2):

$$\begin{array}{ll} \textit{clause } 23: & r\_td_1(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\ & \quad minimal(TX_p), \ldots, minimal(TX_t), \\ & \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\ & \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\ & \quad compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\ & \quad process(HX, HHY), compose(K_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, compose(A, HY, NA), \\ & \quad r\_td_1(TXs, Y, NA) \end{array}$$

By using applicability conditions (1) and (2):

$$\begin{array}{ll} \textit{clause } 24: & r\_td_1(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\ & \quad minimal(TX_p), \ldots, minimal(TX_t), \\ & \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\ & \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\ & \quad compose(A, K_{p-2}, NA), \\ & \quad I_0 = e, \\ & \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\ & \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, compose(NA, HY, NNA), \\ & \quad r\_td_1(TXs, Y, NNA) \end{array}$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$$\begin{array}{ll} \textit{clause } 25: & r\_td_1(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\ & \quad minimal(TX_p), \ldots, minimal(TX_t), \\ & \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\ & \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\ & \quad compose(A, K_{p-2}, NA), \\ & \quad I_0 = e, \\ & \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\ & \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, compose(NA, HY, NNA), \\ & \quad r\_td_1(TXs, Y, NNA) \end{array}$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \land decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$$
\begin{aligned}
clause\ 26: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(NA, HY, NNA), \\
& \quad r\_td_1(TXs, Y, NNA)
\end{aligned}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$
\begin{aligned}
clause\ 27: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, compose(NA, HY, NNA), \\
& \quad r\_td_1(TXs, Y, NNA)
\end{aligned}
$$

By folding clause 27 using $DCLR$:

$$
\begin{aligned}
clause\ 28: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \\
& \quad r\_td_1(TXs, Y, NNA)
\end{aligned}
$$

By folding clause 28 using clauses 1 and 2:

$clause\ 29:\quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),$
$r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}),$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(A, K_{p-2}, NA),$
$r\_td_1([N|TXs], Y, NA)$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$clause\ 30:\quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),$
$r\_td_1([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A)$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$clause\ 31:\quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)),$
$minimal(U_1), \ldots, minimal(U_t),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_0 = e,$
$compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}),$
$process(HX, HHY), compose(I_{p-1}, HHY, I_p),$
$compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}),$
$HY = I_{t+1}, compose(A, HY, NA),$
$r\_td_1(TXs, Y, NA)$

By using applicability condition (3):

$clause\ 32:\quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)),$
$minimal(U_1), \ldots, minimal(U_t),$
$r(U_1, e), \ldots, r(U_t, e),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_0 = e,$
$compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}),$
$process(HX, HHY), compose(I_{p-1}, HHY, I_p),$
$compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}),$
$HY = I_{t+1}, compose(A, HY, NA),$
$r\_td_1(TXs, Y, NA)$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 33: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\
& process(HX, HHY), compose(K_{p-1}, HHY, K_p), \\
& compose(K_p, e, K_{p+1}), \ldots, compose(K_t, e, K_{t+1}), compose(K_{t+1}, e, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, compose(A, HY, NA), \\
& r\_d_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$$
\begin{aligned}
clause\ 34: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(A, K_{p-2}, NA), compose(NA, NHY, NA_1), \\
& compose(NA_1, K_{t-1}, NA_2), r\_d_1(TXs, Y, NA_2)
\end{aligned}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$$
\begin{aligned}
clause\ 35: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(A, K_{p-2}, NA), compose(NA, NHY, NA_1), \\
& compose(NA_1, K_{t-1}, NA_2), r\_d_1(TXs, Y, NA_2)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$$\begin{array}{ll} \mathit{clause}\ 36: & r\_td_1(Xs,Y,A) \leftarrow \\ & \quad Xs=[X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad r(U_1,YU_1),\ldots,r(U_t,YU_t), \\ & \quad nonMinimal(N), decompose(N,HX,U_1,\ldots,U_t), \\ & \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\ & \quad compose(TY_1,TY_2,K_1), compose(K_1,TY_3,K_2),\ldots,compose(K_{p-3},TY_{p-1},K_{p-2}), \\ & \quad I_0=e, \\ & \quad compose(I_0,YU_1,I_1),\ldots,compose(I_{p-2},YU_{p-1},I_{p-1}), \\ & \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\ & \quad compose(I_p,YU_p,I_{p+1}),\ldots,compose(I_t,YU_t,I_{t+1}), \\ & \quad NHY=I_{t+1}, \\ & \quad compose(TY_p,TY_{p+1},K_p), compose(K_p,TY_{p+2},K_{p+1}),\ldots,compose(K_{t-2},TY_t,K_{t-1}), \\ & \quad compose(A,K_{p-2},NA), compose(NA,NHY,NA_1), \\ & \quad compose(NA_1,K_{t-1},NA_2), r\_td_1(TXs,Y,NA_2) \end{array}$$

By duplicating goal $decompose(N,HX,U_1,\ldots,U_t)$:

$$\begin{array}{ll} \mathit{clause}\ 37: & r\_td_1(Xs,Y,A) \leftarrow \\ & \quad Xs=[X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad r(U_1,YU_1),\ldots,r(U_t,YU_t), \\ & \quad nonMinimal(N), decompose(N,HX,U_1,\ldots,U_t), \\ & \quad decompose(N,HX,U_1,\ldots,U_t), \\ & \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\ & \quad compose(TY_1,TY_2,K_1), compose(K_1,TY_3,K_2),\ldots,compose(K_{p-3},TY_{p-1},K_{p-2}), \\ & \quad I_0=e, \\ & \quad compose(I_0,YU_1,I_1),\ldots,compose(I_{p-2},YU_{p-1},I_{p-1}), \\ & \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\ & \quad compose(I_p,YU_p,I_{p+1}),\ldots,compose(I_t,YU_t,I_{t+1}), \\ & \quad NHY=I_{t+1}, \\ & \quad compose(TY_p,TY_{p+1},K_p), compose(K_p,TY_{p+2},K_{p+1}),\ldots,compose(K_{t-2},TY_t,K_{t-1}), \\ & \quad compose(A,K_{p-2},NA), compose(NA,NHY,NA_1), \\ & \quad compose(NA_1,K_{t-1},NA_2), r\_td_1(TXs,Y,NA_2) \end{array}$$

By folding clause 37 using $DCLR$:

$$\begin{array}{ll} \mathit{clause}\ 38: & r\_td_1(Xs,Y,A) \leftarrow \\ & \quad Xs=[X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad decompose(N,HX,U_1,\ldots,U_t), \\ & \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), r(N,NHY), \\ & \quad compose(TY_1,TY_2,K_1), compose(K_1,TY_3,K_2),\ldots,compose(K_{p-3},TY_{p-1},K_{p-2}), \\ & \quad compose(TY_p,TY_{p+1},K_p), compose(K_p,TY_{p+2},K_{p+1}),\ldots,compose(K_{t-2},TY_t,K_{t-1}), \\ & \quad compose(A,K_{p-2},NA), compose(NA,NHY,NA_1), \\ & \quad compose(NA_1,K_{t-1},NA_2), r\_td_1(TXs,Y,NA_2) \end{array}$$

By $t-p+1$ times folding clause 38 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 39: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(A, K_{p-2}, NA), compose(NA, NHY, NA_1), \\
& r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA_1)
\end{aligned}
$$

By folding clause 39 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 40: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(A, K_{p-2}, NA), \\
& r\_td_1([N, TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By $p-1$ times folding clause 40 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 41: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r\_td_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], Y, A)
\end{aligned}
$$

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of $P_{r\_td_1}$. Therefore $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$, we do a backward proof that we begin with P_r in $TDGLR$ and from which we try to obtain S_r.

The procedure P_r for r in $TDGLR$ is:

$$r(X, Y) \leftarrow \quad r\_td_1([X], Y, e)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_td_1([X], Y, e)]$$

By unfolding the 'completion' above wrt $r\_td_1([X], Y, e)$ using $S_{r\_td_1}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge \mathcal{O}_c(e, I_1, Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$.
Therefore, $TDGLR$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 8 The generalization schema TDG_2, which is given below, is correct.

TDG_2 : $\langle$ $DCLR$, $TDGRL$, A_{td2} O_{td212}, O_{td221} $\rangle$ where

A_{td2} : (1) *compose* is associative

(2) *compose* has e as the left and right identity element

(3) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \land minimal(X) \Rightarrow \mathcal{O}_r(X, e)$

(4) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{td212} : partial evaluation of the conjunction
$process(HX, HY), compose(HY, A, NewA)$
results in the introduction of a non-recursive relation

O_{td221} : partial evaluation of the conjunction
$process(HX, HY), compose(I_{p-1}, HY, I_p)$
results in the introduction of a non-recursive relation

where the template $DCLR$ is Logic Program Template 1 in Section 2 and the template $TDGRL$ is Logic Program Template 7 below.

Logic Program Template 7

```
r(X, Y) ←
    r_td2([X], Y, e)
r_td2(Xs, Y, A) ←
    Xs = [],
    Y = A
r_td2(Xs, Y, A) ←
    Xs = [X|TXs],
    minimal(X),
    r_td2(TXs, NewA, A),
    solve(X, HY),
    compose(HY, NewA, Y)
r_td2(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    minimal(TX1), ..., minimal(TXt),
    r_td2(TXs, NewA, A),
    process(HX, HY), compose(HY, NewA, Y)
r_td2(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    minimal(TX1), ..., minimal(TXp-1),
    (nonMinimal(TXp); ...; nonMinimal(TXt)),
    r_td2([TXp, ..., TXt|TXs], NewA, A),
    process(HX, HY), compose(HY, NewA, Y)
r_td2(Xs, Y, A) ←
    Xs = [X|TXs],
    nonMinimal(X),
    decompose(X, HX, TX1, ..., TXt),
    (nonMinimal(TX1); ...; nonMinimal(TXp-1)),
    minimal(TXp), ..., minimal(TXt),
    minimal(U1), ..., minimal(Up-1),
```

```
        decompose(N,HX,U1,...,Up-1,TXp,...,TXt),
        r_td2([TX1,...,TXp-1,N|TXs],Y,A)
    r_td2(Xs,Y,A) ←
        Xs = [X|TXs],
        nonMinimal(X),
        decompose(X,HX,TX1,...,TXt),
        (nonMinimal(TX1);...;nonMinimal(TXp-1)),
        (nonMinimal(TXp);...;nonMinimal(TXt)),
        minimal(U1),...,minimal(Ut),
        decompose(N,HX,U1,...,Ut),
        r_td2([TX1,...,TXp-1,N,TXp,...,TXt|TXs],Y,A)
```

and the specification S_r of relation r is:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

and the specification of r_td_2, namely $S_{r\_td_2}$, is:

$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs,Y,A) \Leftrightarrow (Xs = [\,] \wedge Y = A)$
$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)$
$\wedge \mathcal{O}_c(I_q, A, I_{q+1}) \wedge Y = I_{q+1})]$

Proof 8 To prove the correctness of the generalization schema TDG_2, by Definition 10, we have to prove that templates $DCLR$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{td2}. By Definition 5, the templates $DCLR$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{td2} iff $DCLR$ is *equivalent to* $TDGRL$ wrt the specification S_r provided that the conditions in A_{td2} hold. By Definition 4, $DCLR$ is *equivalent to* $TDGRL$ wrt the specification S_r iff the following two conditions hold:

(*a*) $DCLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCLR$.

(*b*) $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by simultaneous tupling-and-descending generalization of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (*a*) always holds.

To prove equivalence, we have to prove condition (*b*). We will use the following property of steadfastness: $TDGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, where $P_{r\_td_2}$ is the procedure for r_td_2, and P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_td_2}$ and from which we try to obtain $P_{r\_td_2}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_td_2}$ becomes:

$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs,Y,A) \Leftrightarrow$
$(Xs = [\,] \wedge Y = A)$
$\vee (Xs = [X_1] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge \mathcal{O}_c(I_1, A, I_2) \wedge Y = I_2)$
$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge$
$\mathcal{O}_c(I_q, A, I_{q+1}) \wedge Y = I_{q+1})]$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs,Y,A) \Leftrightarrow$
$(Xs = [\,] \wedge Y = A)$
$\vee (Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge TY = A \wedge \mathcal{O}_c(TY, A, NA) \wedge \mathcal{O}_c(I_1, NA, Y))$
$\vee (Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge$
$\bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge TY = I_q \wedge \mathcal{O}_c(TY, A, NA) \wedge \mathcal{O}_c(I_1, NA, Y))]$

where $q \geq 2$.

By folding using $S_{r\_td_2}$, and renaming:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee(Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge r\_td_2(TXs, NA, A) \wedge \mathcal{O}_c(HY, NA, Y))]$$

By taking the 'decompletion':

$$\begin{array}{ll} \textit{clause } 1: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [\,], Y = A \\ \textit{clause } 2: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], r(X, HY), \\ & \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y) \end{array}$$

By unfolding clause 2 wrt $r(X, HY)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} \textit{clause } 3: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad minimal(X), \\ & \quad solve(X, HY), \\ & \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y) \\ \textit{clause } 4: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\ & \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, \\ & \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y) \end{array}$$

By introducing

$$(minimal(TX_1) \wedge \ldots \wedge minimal(TX_t)) \vee$$
$$((minimal(TX_1) \wedge \ldots \wedge minimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t))) \vee$$
$$((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (minimal(TX_p) \wedge \ldots \wedge minimal(TX_t))) \vee$$
$$((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))$$

in clause 4, using applicability condition (4):

$$\begin{array}{ll} \textit{clause } 5: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad minimal(TX_1), \ldots, minimal(TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\ & \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad HY = I_{t+1}, \\ & \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y) \end{array}$$

$$
\begin{array}{ll}
clause\ 6: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs=[X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0=e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY=I_{t+1}, \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y) \\
clause\ 7: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs=[X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p),\ldots,minimal(TX_t), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0=e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY=I_{t+1}, \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y) \\
clause\ 8: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs=[X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \\
& \quad I_0=e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY=I_{t+1}, \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y)
\end{array}
$$

By t times unfolding clause 5 wrt $r(TX_1,TY_1),\ldots,r(TX_t,TY_t)$ using $DCLR$, and simplifying using condition (4):

$$
\begin{array}{ll}
clause\ 9: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs=[X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_t), \\
& \quad solve(TX_1,TY_1),\ldots,solve(TX_t,TY_t), \\
& \quad I_0=e, \\
& \quad compose(I_0,TY_1,I_1),\ldots,compose(I_{p-2},TY_{p-1},I_{p-1}), \\
& \quad process(HX,HHY), compose(I_{p-1},HHY,I_p), \\
& \quad compose(I_p,TY_p,I_{p+1}),\ldots,compose(I_t,TY_t,I_{t+1}), \\
& \quad HY=I_{t+1}, \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{aligned}
\mathit{clause}\ 10:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$$
\begin{aligned}
\mathit{clause}\ 11:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
\mathit{clause}\ 12:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_0 = e, \\
& I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& I_{p+1} = I_p, \ldots, I_{t+1} = I_t, \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
\mathit{clause}\ 13:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& r\_td_2(TXs, NA, A), \\
& process(HX, HY), compose(HY, NA, Y)
\end{aligned}
$$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCLR$, and simplifying using condition (4):

$$
\begin{aligned}
\mathit{clause}\ 14: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{aligned}
\mathit{clause}\ 15: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By rewriting clause 15 using applicability condition (1):

$$
\begin{aligned}
\mathit{clause}\ 16: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, compose(HY, NA, Y), \\
& compose(TY_p, I_{p+1}, NA), \\
& compose(TY_{p+1}, I_{p+2}, I_{p+1}), \ldots, compose(TY_{t-1}, I_t, I_{t-1}), \\
& compose(TY_t, NNA, I_t), \\
& r\_td_2(TXs, NNA, A)
\end{aligned}
$$

By $t - p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 17: \quad & r\_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_d_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 18: \quad & r\_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_d_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 19: \quad & r\_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_d_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 20: \quad & r\_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r\_d_2([TX_p, \ldots, TX_t|TXs], NA, A), \\
& process(HX, HY), compose(HY, NA, Y)
\end{aligned}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{array}{ll}
clause\ 21: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 22: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 23: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_0 = e, \\
& \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& \quad compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\
& \quad process(HX, HHY), compose(K_{p-1}, HHY, I_p), \\
& \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& \quad HY = I_{t+1}, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability conditions (1) and (2):

$clause\ 24:\quad r\_td_2(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, e), \ldots, r(U_{p-1}, e),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(K_{p-2}, NA, Y),$
$I_0 = e,$
$compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}),$
$process(HX, HHY), compose(I_{p-1}, HHY, I_p),$
$compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}),$
$HY = I_{t+1}, compose(HY, NNA, NA),$
$r\_td_2(TXs, NNA, A)$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$clause\ 25:\quad r\_td_2(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(K_{p-2}, NA, Y),$
$I_0 = e,$
$compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}),$
$process(HX, HHY), compose(I_{p-1}, HHY, I_p),$
$compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}),$
$HY = I_{t+1}, compose(HY, NNA, NA),$
$r\_td_2(TXs, NNA, A)$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \land decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$clause\ 26:\quad r\_td_2(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(K_{p-2}, NA, Y),$
$I_0 = e,$
$compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}),$
$process(HX, HHY), compose(I_{p-1}, HHY, I_p),$
$compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}),$
$HY = I_{t+1}, compose(HY, NNA, NA),$
$r\_td_2(TXs, NNA, A)$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$\begin{aligned}
clause\ 27: \quad & r_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, NA, Y), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, compose(HY, NNA, NA), \\
& r_d_2(TXs, NNA, A)
\end{aligned}$$

By folding clause 27 using $DCLR$:

$$\begin{aligned}
clause\ 28: \quad & r_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, NA, Y), compose(HY, NNA, NA), \\
& r_d_2(TXs, NNA, A)
\end{aligned}$$

By folding clause 28 using clauses 1 and 2:

$$\begin{aligned}
clause\ 29: \quad & r_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, NA, Y), \\
& r_d_2([N|TXs], NA, A)
\end{aligned}$$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$$\begin{aligned}
clause\ 30: \quad & r_d_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& r_d_2([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A)
\end{aligned}$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$$
\begin{aligned}
clause\ 31: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 32: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 33: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \\
& process(HX, HHY), compose(K_{p-1}, HHY, K_p), \\
& compose(K_p, e, K_{p+1}), \ldots, compose(K_t, e, K_{t+1}), compose(K_{t+1}, e, I_p), \\
& compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\
& HY = I_{t+1}, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$$
\begin{aligned}
clause\ 34: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, e), \ldots, r(U_t, e), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad I_0 = e, \\
& \quad compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}), \\
& \quad NHY = I_{t+1}, \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{t-1}, NA, NA_1), compose(NHY, NA_1, NA_2), \\
& \quad compose(K_{p-2}, NA_2, Y), r\_td_2(TXs, NA, A)
\end{aligned}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$$
\begin{aligned}
clause\ 35: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad I_0 = e, \\
& \quad compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& \quad process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& \quad compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& \quad NHY = I_{t+1}, \\
& \quad compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& \quad compose(K_{t-1}, NA, NA_1), compose(NHY, NA_1, NA_2), \\
& \quad compose(K_{p-2}, NA_2, Y), r\_td_2(TXs, NA, A)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \land decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$$
\begin{array}{ll}
clause\ 36: & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{t-1}, NA, NA_1), compose(NHY, NA_1, NA_2), \\
& compose(K_{p-2}, NA_2, Y), r\_td_2(TXs, NA, A)
\end{array}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_t)$:

$$
\begin{array}{ll}
clause\ 37: & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& I_0 = e, \\
& compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& compose(I_p, YU_p, I_{p+1}), \ldots, compose(I_t, YU_t, I_{t+1}), \\
& NHY = I_{t+1}, \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{t-1}, NA, NA_1), compose(NHY, NA_1, NA_2), \\
& compose(K_{p-2}, NA_2, Y), r\_td_2(TXs, NA, A)
\end{array}
$$

By folding clause 37 using $DCLR$:

$$
\begin{array}{ll}
clause\ 38: & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(K_{t-1}, NA, NA_1), compose(NHY, NA_1, NA_2), \\
& compose(K_{p-2}, NA_2, Y), r\_td_2(TXs, NA, A)
\end{array}
$$

By $t - p + 1$ times folding clause 38 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 39: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(NHY, NA_1, NA_2), \\
& compose(K_{p-2}, NA_2, Y), r\_td_2([TX_p, \ldots, TX_t|TXs], NA_1, A)
\end{aligned}
$$

By folding clause 39 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 40: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(K_{p-2}, NA_2, Y), r\_td_2([N, TX_p, \ldots, TX_t|TXs], NA_2, A)
\end{aligned}
$$

By $p-1$ times folding clause 40 using clauses 1 and 2:

$$
\begin{aligned}
\textit{clause } 41: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r\_td_2([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], Y, A)
\end{aligned}
$$

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of $P_{r\_td_2}$. Therefore $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, we do a backward proof that we begin with P_r in $TDGRL$ and from which we try to obtain S_r.
The procedure P_r for r in $TDGRL$ is:

$$r(X, Y) \leftarrow \quad r\_td_2([X], Y, e)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_td_2([X], Y, e)]$$

By unfolding the 'completion' above wrt $r\_td_2([X], Y, e)$ using $S_{r\_td_2}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \land I_1 = Y_1 \land \mathcal{O}_c(I_1, e, Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \land I_1 = Y_1 \land Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$.
Therefore, $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 9 The generalization schema TDG_3, which is given below, is correct.

TDG_3 : $\langle$ $DCRL$, $TDGRL$, A_{td3} O_{td312}, O_{td321} $\rangle$ where

A_{td3} : (1) *compose* is associative

(2) *compose* has e as the left and right identity element

(3) $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$

(4) $\mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{td312} : partial evaluation of the conjunction
$process(HX, HY), compose(HY, A, NewA)$
results in the introduction of a non-recursive relation

O_{td321} : partial evaluation of the conjunction
$process(HX, HY), compose(HY, I_p, I_{p-1})$
results in the introduction of a non-recursive relation

where the template of $DCRL$ is Logic Program Template 3 in Section 2 and the template $TDGRL$ is Logic Program Template 7 in Theorem 8.

The specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

The specification of r_td_2, namely $S_{r\_td_2}$, is:

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow (Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)$$
$$\wedge \mathcal{O}_c(I_q, A, I_{q+1}) \wedge Y = I_{q+1})]$$

Proof 9 To prove the correctness of the generalization schema TDG_3, by Definition 10, we have to prove that templates $DCRL$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{td3}. By Definition 5, the templates $DCRL$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{td3} iff $DCRL$ is *equivalent to* $TDGRL$ wrt the specification S_r provided that the conditions in A_{td3} hold. By Definition 4, $DCRL$ is *equivalent to* $TDGRL$ wrt the specification S_r iff the following two conditions hold:

(a) $DCRL$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCRL$.

(b) $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by simultaneous tupling-and-descending generalization of P.

In program transformation, we assume that the input program, here template $DCRL$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $TDGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, where $P_{r\_td_2}$ is the procedure for r_td_2, and P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_td_2}$ and from which we try to obtain $P_{r\_td_2}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_td_2}$ becomes:

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge \mathcal{O}_c(I_1, A, I_2) \wedge Y = I_2)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge I_1 = Y_1 \wedge \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge$$
$$\mathcal{O}_c(I_q, A, I_{q+1}) \wedge Y = I_{q+1})]$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\forall Xs : \textit{list of}\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge TY = A \wedge \mathcal{O}_c(TY, A, NA) \wedge \mathcal{O}_c(I_1, NA, Y))$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge$$
$$\bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge TY = I_q \wedge \mathcal{O}_c(TY, A, NA) \wedge \mathcal{O}_c(I_1, NA, Y))]$$

where $q \geq 2$.

By folding using $S_{r\_td_2}$, and renaming:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge r\_td_2(TXs, NA, A) \wedge \mathcal{O}_c(HY, NA, Y))]$$

By taking the 'decompletion':

$$\begin{array}{ll} \textit{clause } 1: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [\,], Y = A \\ \textit{clause } 2: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], r(X, HY), \\ & \quad r\_td_2(TXs, NA, A), \textit{compose}(HY, NA, Y) \end{array}$$

By unfolding clause 2 wrt $r(X, HY)$ using $DCRL$, and using the assumption that $DCRL$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} \textit{clause } 3: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad \textit{minimal}(X), \\ & \quad \textit{solve}(X, HY), \\ & \quad r\_td_2(TXs, NA, A), \textit{compose}(HY, NA, Y) \\ \textit{clause } 4: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad \textit{nonMinimal}(X), \textit{decompose}(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_{t+1} = e, \\ & \quad \textit{compose}(TY_t, I_{t+1}, I_t), \ldots, \textit{compose}(TY_p, I_{p+1}, I_p), \\ & \quad \textit{process}(HX, HHY), \textit{compose}(HHY, I_p, I_{p-1}), \\ & \quad \textit{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \textit{compose}(TY_1, I_1, I_0), \\ & \quad HY = I_0, \\ & \quad r\_td_2(TXs, NA, A), \textit{compose}(HY, NA, Y) \end{array}$$

By introducing

$$\begin{array}{c} (\textit{minimal}(TX_1) \wedge \ldots \wedge \textit{minimal}(TX_t)) \vee \\ ((\textit{minimal}(TX_1) \wedge \ldots \wedge \textit{minimal}(TX_{p-1})) \wedge (\textit{nonMinimal}(TX_p) \vee \ldots \vee \textit{nonMinimal}(TX_t))) \vee \\ ((\textit{nonMinimal}(TX_1) \vee \ldots \vee \textit{nonMinimal}(TX_{p-1})) \wedge (\textit{minimal}(TX_p) \wedge \ldots \wedge \textit{minimal}(TX_t))) \vee \\ ((\textit{nonMinimal}(TX_1) \vee \ldots \vee \textit{nonMinimal}(TX_{p-1})) \wedge (\textit{nonMinimal}(TX_p) \vee \ldots \vee \textit{nonMinimal}(TX_t))) \end{array}$$

in clause 4, using applicability condition (4):

$$\begin{array}{ll} \textit{clause } 5: & r\_td_2(Xs, Y, A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad \textit{nonMinimal}(X), \textit{decompose}(X, HX, TX_1, \ldots, TX_t), \\ & \quad \textit{minimal}(TX_1), \ldots, \textit{minimal}(TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_{t+1} = e, \\ & \quad \textit{compose}(TY_t, I_{t+1}, I_t), \ldots, \textit{compose}(TY_p, I_{p+1}, I_p), \\ & \quad \textit{process}(HX, HHY), \textit{compose}(HHY, I_p, I_{p-1}), \\ & \quad \textit{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \textit{compose}(TY_1, I_1, I_0), \\ & \quad HY = I_0, \\ & \quad r\_td_2(TXs, NA, A), \textit{compose}(HY, NA, Y) \end{array}$$

$$\begin{array}{ll}
clause\ 6: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y) \\
clause\ 7: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y) \\
clause\ 8: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}$$

By t times unfolding clause 5 wrt $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$ using $DCRL$, and simplifying using condition (4):

$$\begin{array}{ll}
clause\ 9: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 10:\ \ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$$
\begin{aligned}
clause\ 11:\ \ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 12:\ \ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& solve(TX_1, e), \ldots, solve(TX_t, e), \\
& I_{t+1} = e, \\
& I_t = I_{t+1}, \ldots, I_p = I_{p+1}, \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& I_{p-2} = I_{p-1}, \ldots, I_0 = I_1, \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 13:\ \ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_t), \\
& r\_td_2(TXs, NA, A), \\
& process(HX, HY), compose(HY, NA, Y)
\end{aligned}
$$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCRL$, and simplifying using condition (4):

$$
\begin{aligned}
\mathit{clause}\ 14: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{aligned}
\mathit{clause}\ 15: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By rewriting clause 15 using applicability conditions (1) and (2):

$$
\begin{aligned}
\mathit{clause}\ 16: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, compose(HY, NA, Y), \\
& compose(TY_p, I_{p+1}, NA), \\
& compose(TY_{p+1}, I_{p+2}, I_{p+1}), \ldots, compose(TY_{t-1}, I_t, I_{t-1}), \\
& compose(TY_t, NNA, I_t), \\
& r\_td_2(TXs, NNA, A)
\end{aligned}
$$

By $t - p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 17: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_td_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 18: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_td_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 19: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), \\
& HY = I_p, \\
& r\_td_2([TX_p, \ldots, TX_t|TXs], NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 20: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r\_td_2([TX_p, \ldots, TX_t|TXs], NA, A), \\
& process(HX, HY), compose(HY, NA, Y)
\end{aligned}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{aligned}
clause\ 21: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 22: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 23: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, K_1), compose(e, K_1, K_2), \ldots, compose(e, K_{p-2}, K_{p-1}), \\
& compose(TY_{p-1}, K_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$$
\begin{array}{ll}
\mathit{clause}\ 24: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, e), \ldots, r(U_{p-1}, e), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& \quad HY = I_0, r\_td_2(TXs, NA, A), compose(HY, NA, NNA), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{array}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$$
\begin{array}{ll}
\mathit{clause}\ 25: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad HY = I_0, r\_td_2(TXs, NA, A), compose(HY, NA, NNA), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{array}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$$
\begin{array}{ll}
\mathit{clause}\ 26: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad HY = I_0, r\_td_2(TXs, NA, A), compose(HY, NA, NNA), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{array}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$
\begin{aligned}
clause\ 27:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad HY = I_0, r\_td_2(TXs, NA, A), compose(HY, NA, NNA), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{aligned}
$$

By folding clause 27 using $DCRL$:

$$
\begin{aligned}
clause\ 28:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, NNA), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{aligned}
$$

By folding clause 28 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 29:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& \quad r\_td_2([N|TXs], NNA, A), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(K_{p-2}, NNA, Y)
\end{aligned}
$$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 30:\quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r\_td_2([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A)
\end{aligned}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$$
\begin{array}{ll}
clause\ 31: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability condition (3):

$$
\begin{array}{ll}
clause\ 32: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, e), \ldots, r(U_t, e), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
clause\ 33: & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad r(U_1, e), \ldots, r(U_t, e), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad compose(e, I_p, K_{t+1}), \\
& \quad compose(e, K_{t+1}, K_t), \ldots, compose(e, K_{p+1}, K_p), \\
& \quad process(HX, HHY), compose(HHY, K_p, K_{p-1}), \\
& \quad compose(e, K_{p-1}, K_{p-2}), \ldots, compose(e, K_1, K_0), \\
& \quad compose(e, K_0, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}
$$

By using applicability conditions (1) and (2):

$$
\begin{aligned}
clause\ 34: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& I_{t+1} = e, \\
& compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& r\_td_2(TXs, NA, A), compose(K_{t-1}, NA, NA_1), \\
& compose(NHY, NA_1, NA_2), compose(K_{p-2}, NA_2, Y)
\end{aligned}
$$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$$
\begin{aligned}
clause\ 35: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& I_{t+1} = e, \\
& compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& r\_td_2(TXs, NA, A), compose(K_{t-1}, NA, NA_1), \\
& compose(NHY, NA_1, NA_2), compose(K_{p-2}, NA_2, Y)
\end{aligned}
$$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$$
\begin{aligned}
clause\ 36: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& I_{t+1} = e, \\
& compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& r\_td_2(TXs, NA, A), compose(K_{t-1}, NA, NA_1), \\
& compose(NHY, NA_1, NA_2), compose(K_{p-2}, NA_2, Y)
\end{aligned}
$$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_t)$:

$$
\begin{aligned}
clause\ 37: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& I_{t+1} = e, \\
& compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& r\_td_2(TXs, NA, A), compose(K_{t-1}, NA, NA_1), \\
& compose(NHY, NA_1, NA_2), compose(K_{p-2}, NA_2, Y)
\end{aligned}
$$

By folding clause 37 using $DCRL$:

$$
\begin{aligned}
clause\ 38: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& r\_td_2(TXs, NA, A), compose(K_{t-1}, NA, NA_1), \\
& compose(NHY, NA_1, NA_2), compose(K_{p-2}, NA_2, Y)
\end{aligned}
$$

By $t - p + 1$ times folding clause 38 using clauses 1 and 2:

$$\begin{array}{ll} clause\ 39: & r\_td_2(Xs,Y,A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad r(TX_1,TY_1),\ldots,r(TX_{p-1},TY_{p-1}), r(N,NHY), \\ & \quad decompose(N,HX,U_1,\ldots,U_t), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad compose(TY_1,TY_2,K_1), compose(K_1,TY_3,K_2),\ldots,compose(K_{p-3},TY_{p-1},K_{p-2}), \\ & \quad r\_td_2([TX_p,\ldots,TX_t|TXs],NA_1,A), \\ & \quad compose(NHY,NA_1,NA_2), compose(K_{p-2},NA_2,Y) \end{array}$$

By folding clause 39 using clauses 1 and 2:

$$\begin{array}{ll} clause\ 40: & r\_td_2(Xs,Y,A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad r(TX_1,TY_1),\ldots,r(TX_{p-1},TY_{p-1}), \\ & \quad decompose(N,HX,U_1,\ldots,U_t), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad compose(TY_1,TY_2,K_1), compose(K_1,TY_3,K_2),\ldots,compose(K_{p-3},TY_{p-1},K_{p-2}), \\ & \quad r\_td_2([N,TX_p,\ldots,TX_t|TXs],NA_2,A), \\ & \quad compose(K_{p-2},NA_2,Y) \end{array}$$

By $p-1$ times folding clause 40 using clauses 1 and 2:

$$\begin{array}{ll} clause\ 41: & r\_td_2(Xs,Y,A) \leftarrow \\ & \quad Xs = [X|TXs], \\ & \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\ & \quad (nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})), \\ & \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\ & \quad decompose(N,HX,U_1,\ldots,U_t), \\ & \quad minimal(U_1),\ldots,minimal(U_t), \\ & \quad r\_td_2([TX_1,\ldots,TX_{p-1},N,TX_p,\ldots,TX_t|TXs],Y,A) \end{array}$$

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of $P_{r\_td_2}$. Therefore $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, we do a backward proof that we begin with P_r in $TDGRL$ and from which we try to obtain S_r.
The procedure P_r for r in $TDGRL$ is:

$$r(X,Y) \leftarrow \quad r\_td_2([X],Y,e)$$

By taking the 'completion':

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_td_2([X],Y,e)]$$

By unfolding the 'completion' above wrt $r\_td_2([X],Y,e)$ using $S_{r\_td_2}$:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists Y_1,I_1:\mathcal{Y}.\ \mathcal{O}_r(X,Y_1) \wedge I_1 = Y_1 \wedge \mathcal{O}_c(I_1,e,Y)]$$

By using applicability condition (2):

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists Y_1,I_1:\mathcal{Y}.\ \mathcal{O}_r(X,Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X:\mathcal{X}, \forall Y:\mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$.
Therefore, $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 10 The generalization schema TDG_4, which is given below, is correct.

TDG_4 : $\langle$ $DCRL$, $TDGLR$, A_{td4}, O_{td412}, O_{td421} $\rangle$ where

A_{td4} : (1) *compose* is associative

(2) *compose* has e as the left and right identity element

(3) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$

(4) $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \Rightarrow [\neg minimal(X) \Leftrightarrow nonMinimal(X)]$

O_{td412} : partial evaluation of the conjunction
$process(HX, HY), compose(A, HY, NewA)$
results in the introduction of a non-recursive relation

O_{td421} : partial evaluation of the conjunction
$process(HX, HY), compose(HY, I_p, I_{p-1})$
results in the introduction of a non-recursive relation

where the template $DCRL$ is Logic Program Template 3 in Section 2 and the template $TDGLR$ is Logic Program Template 6 in Theorem 7.

The specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

The specification $S_{r\_td_1}$:

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ \ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs, Y, A) \Leftrightarrow (Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \ \textstyle\bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \ \wedge I_1 = Y_1 \wedge \ \ \textstyle\bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)$$
$$\wedge \mathcal{O}_c(A, I_q, I_{q+1}) \wedge Y = I_{q+1})]$$

Proof 10 To prove the correctness of the generalization schema TDG_4, by Definition 10, we have to prove that templates $DCRL$ and $TDGLR$ are *equivalent* wrt S_r under the applicability conditions A_{td4}. By Definition 5, the templates $DCRL$ and $TDGLR$ are *equivalent* wrt S_r under the applicability conditions A_{td4} iff $DCRL$ is *equivalent to* $TDGLR$ wrt the specification S_r provided that the conditions in A_{td4} hold. By Definition 4, $DCRL$ is *equivalent to* $TDGLR$ wrt the specification S_r iff the following two conditions hold:

(a) $DCRL$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{decompose}$, $S_{process}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCRL$.

(b) $TDGLR$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by simultaneous tupling-and-descending generalization of P.

In program transformation, we assume that the input program, here template $DCRL$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $TDGLR$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$, where $P_{r\_td_1}$ is the procedure for r_td_1, and P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$, where P_r is the procedure for r.

To prove that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_td_1}$ and from which we try to obtain $P_{r\_td_1}$.

If we separate the cases of $q \geq 1$ by $q = 1 \vee q \geq 2$, then $S_{r\_td_1}$ becomes:

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ \ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge \mathcal{O}_c(A, I_1, I_2) \wedge Y = I_2)$$
$$\vee (Xs = [X_1, X_2, \ldots, X_q] \wedge \ \ \textstyle\bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \ \wedge I_1 = Y_1 \wedge \ \ \textstyle\bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \ \wedge$$
$$\mathcal{O}_c(A, I_q, I_{q+1}) \wedge Y = I_{q+1})]$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [\,] \wedge \mathcal{O}_r(X_1, Y_1) \wedge Y_1 = I_1 \wedge TY = A \wedge \mathcal{O}_c(A, I_1, NA) \wedge \mathcal{O}_c(NA, TY, Y))$$
$$\vee (Xs = [X_1|TXs] \wedge TXs = [X_2, \ldots, X_q] \wedge \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \wedge Y_1 = I_1 \wedge Y_2 = I_2 \wedge$$
$$\bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \wedge TY = I_q \wedge \mathcal{O}_c(A, I_1, NA) \wedge \mathcal{O}_c(NA, TY, Y))]$$

where $q \geq 2$.

By folding using $S_{r\_td_1}$, and renaming:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee (Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge \mathcal{O}_c(A, HY, NA) \wedge r\_td_1(TXs, Y, NA))]$$

By taking the 'decompletion':

$$
\begin{aligned}
\textit{clause } 1: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [\,], Y = A \\
\textit{clause } 2: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], r(X, HY), \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By unfolding clause 2 wrt $r(X, HY)$ using $DCRL$, and using the assumption that $DCRL$ is steadfast wrt S_r in $\mathcal{S}$:

$$
\begin{aligned}
\textit{clause } 3: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad minimal(X), \\
& \quad solve(X, HY), \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA) \\
\textit{clause } 4: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By introducing

$$
\begin{gathered}
(minimal(TX_1) \wedge \ldots \wedge minimal(TX_t)) \vee \\
((minimal(TX_1) \wedge \ldots \wedge minimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t))) \vee \\
((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (minimal(TX_p) \wedge \ldots \wedge minimal(TX_t))) \vee \\
((nonMinimal(TX_1) \vee \ldots \vee nonMinimal(TX_{p-1})) \wedge (nonMinimal(TX_p) \vee \ldots \vee nonMinimal(TX_t)))
\end{gathered}
$$

in clause 4, using applicability condition (4):

$$
\begin{aligned}
\textit{clause } 5: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

$$
\begin{aligned}
clause\ 6: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA) \\
clause\ 7: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA) \\
clause\ 8: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By t times unfolding clause 5 wrt $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$ using $DCRL$, and simplifying using condition (4):

$$
\begin{aligned}
clause\ 9: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$clause\ 10: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$minimal(TX_1), \ldots, minimal(TX_t),$
$minimal(TX_1), \ldots, minimal(TX_t),$
$solve(TX_1, e), \ldots, solve(TX_t, e),$
$I_{t+1} = e,$
$compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0),$
$HY = I_0,$
$compose(A, HY, NA), r\_td_1(TXs, Y, NA)$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_t)$ atoms in clause 10:

$clause\ 11: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$minimal(TX_1), \ldots, minimal(TX_t),$
$solve(TX_1, e), \ldots, solve(TX_t, e),$
$I_{t+1} = e,$
$compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0),$
$HY = I_0,$
$compose(A, HY, NA), r\_td_1(TXs, Y, NA)$

By using applicability condition (2):

$clause\ 12: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$minimal(TX_1), \ldots, minimal(TX_t),$
$solve(TX_1, e), \ldots, solve(TX_t, e),$
$I_{t+1} = e,$
$I_t = I_{t+1}, \ldots, I_p = I_{p+1},$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$I_{p-2} = I_{p-1}, \ldots, I_0 = I_1,$
$HY = I_0,$
$compose(A, HY, NA), r\_td_1(TXs, Y, NA)$

By simplification:

$clause\ 13: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$minimal(TX_1), \ldots, minimal(TX_t),$
$process(HX, HY), compose(A, HY, NA),$
$r\_td_1(TXs, Y, NA)$

By $p-1$ times unfolding clause 6 wrt $r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1})$ using $DCRL$, and simplifying using condition (4):

$$
\begin{aligned}
clause\ 14: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By deleting one of the $minimal(TX_1), \ldots, minimal(TX_{p-1})$ atoms in clause 14:

$$
\begin{aligned}
clause\ 15: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By rewriting clause 15 using applicability conditions (1) and (2):

$$
\begin{aligned}
clause\ 16: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& compose(TY_p, TY_{p+1}, I_{p+1}), \\
& compose(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, compose(I_{t-1}, TY_t, I_t), \\
& compose(NA, I_t, NNA), \\
& r\_td_1(TXs, Y, NNA)
\end{aligned}
$$

By $t - p$ times folding clause 16 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 17:\ \ & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \\
& I_0 = e, \\
& compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 18:\ \ & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 19:\ \ & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& solve(TX_1, e), \ldots, solve(TX_{p-1}, e), \\
& I_0 = e, \\
& I_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \\
& process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \\
& compose(A, HY, NA), \\
& r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 20:\ \ & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& process(HX, HY), compose(A, HY, NA), \\
& r\_td_1([TX_p, \ldots, TX_t|TXs], Y, NA)
\end{aligned}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_{p-1})$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_{p-1}$) in clause 7:

$$
\begin{aligned}
clause\ 21: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 22: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 23: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& minimal(TX_p), \ldots, minimal(TX_t), \\
& minimal(U_1), \ldots, minimal(U_{p-1}), \\
& r(U_1, e), \ldots, r(U_{p-1}, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(e, I_{p-1}, K_1), compose(e, K_1, K_2), \ldots, compose(e, K_{p-2}, K_{p-1}), \\
& compose(TY_{p-1}, K_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$clause\ 24: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, e), \ldots, r(U_{p-1}, e),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_{t+1} = e,$
$compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0),$
$HY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(A, K_{p-2}, NA), compose(NA, HY, NNA),$
$r\_td_1(TXs, Y, NNA)$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_{p-1}$ in place of some occurrences of e:

$clause\ 25: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_{t+1} = e,$
$compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0),$
$HY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(A, K_{p-2}, NA), compose(NA, HY, NNA),$
$r\_td_1(TXs, Y, NNA)$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$$

always holds (because N is existentially quantified):

$clause\ 26: \quad r\_td_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$minimal(TX_p), \ldots, minimal(TX_t),$
$minimal(U_1), \ldots, minimal(U_{p-1}),$
$r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}),$
$nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_{t+1} = e,$
$compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0),$
$HY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(A, K_{p-2}, NA), compose(NA, HY, NNA),$
$r\_td_1(TXs, Y, NNA)$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t)$:

$$
\begin{array}{ll}
clause\ 27: & r\_d_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
& \quad nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& \quad I_{t+1} = e, \\
& \quad compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& \quad process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& \quad compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& \quad HY = I_0, \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \\
& \quad r\_d_1(TXs, Y, NNA)
\end{array}
$$

By folding clause 27 using $DCRL$:

$$
\begin{array}{ll}
clause\ 28: & r\_d_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, HY), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \\
& \quad r\_d_1(TXs, Y, NNA)
\end{array}
$$

By folding clause 28 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 29: & r\_d_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& \quad compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& \quad compose(A, K_{p-2}, NA), \\
& \quad r\_d_1([N|TXs], Y, NA)
\end{array}
$$

By $p-1$ times folding clause 29 using clauses 1 and 2:

$$
\begin{array}{ll}
clause\ 30: & r\_d_1(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r\_d_1([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A)
\end{array}
$$

By introducing atoms $minimal(U_1), \ldots, minimal(U_t)$ (with new, i.e. existentially quantified, variables $U_1, \ldots, U_t$) in clause 8:

$$
\begin{aligned}
clause\ 31: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (3):

$$
\begin{aligned}
clause\ 32: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 33: \quad & r\_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, e), \ldots, r(U_t, e), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p), \\
& compose(e, I_p, K_{t+1}), \\
& compose(e, K_{t+1}, K_t), \ldots, compose(e, K_{p+1}, K_p), \\
& process(HX, HHY), compose(HHY, K_p, K_{p-1}), \\
& compose(e, K_{p-1}, K_{p-2}), \ldots, compose(e, K_1, K_0), \\
& compose(e, K_0, I_{p-1}), \\
& compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \\
& HY = I_0, \\
& compose(A, HY, NA), r\_td_1(TXs, Y, NA)
\end{aligned}
$$

By using applicability conditions (1) and (2):

$clause\ 34: \quad r\_d_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)),$
$minimal(U_1), \ldots, minimal(U_t),$
$r(U_1, e), \ldots, r(U_t, e),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$compose(e, I_{t+1}, I_t), \ldots, compose(e, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0),$
$NHY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}),$
$compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2),$
$compose(NA_2, K_{t-1}, NA), r\_d_1(TXs, Y, NA)$

By introducing new, i.e. existentially quantified, variables $YU_1, \ldots, YU_t$ in place of some occurrences of e:

$clause\ 35: \quad r\_d_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)),$
$minimal(U_1), \ldots, minimal(U_t),$
$r(U_1, YU_1), \ldots, r(U_t, YU_t),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_{t+1} = e,$
$compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0),$
$NHY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}),$
$compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2),$
$compose(NA_2, K_{t-1}, NA), r\_d_1(TXs, Y, NA)$

By introducing $nonMinimal(N)$ and $decompose(N, HX, U_1, \ldots, U_t)$, since

$$\exists N : \mathcal{X}.nonMinimal(N) \wedge decompose(N, HX, U_1, \ldots, U_t)$$

always holds (because N is existentially quantified):

$clause\ 36: \quad r\_d_1(Xs, Y, A) \leftarrow$
$Xs = [X|TXs],$
$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$
$(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})),$
$(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)),$
$minimal(U_1), \ldots, minimal(U_t),$
$r(U_1, YU_1), \ldots, r(U_t, YU_t),$
$nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t),$
$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$
$I_{t+1} = e,$
$compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p),$
$process(HX, HHY), compose(HHY, I_p, I_{p-1}),$
$compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0),$
$NHY = I_0,$
$compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}),$
$compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}),$
$compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2),$
$compose(NA_2, K_{t-1}, NA), r\_d_1(TXs, Y, NA)$

By duplicating goal $decompose(N, HX, U_1, \ldots, U_t)$:

$$
\begin{aligned}
clause\ 37: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
& nonMinimal(N), decompose(N, HX, U_1, \ldots, U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_{t+1} = e, \\
& compose(YU_t, I_{t+1}, I_t), \ldots, compose(YU_p, I_{p+1}, I_p), \\
& process(HX, HHY), compose(HHY, I_p, I_{p-1}), \\
& compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0), \\
& NHY = I_0, \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \\
& compose(NA_2, K_{t-1}, NA), r\_d_1(TXs, Y, NA)
\end{aligned}
$$

By folding clause 37 using $DCRL$:

$$
\begin{aligned}
clause\ 38: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), \ldots, compose(K_{t-2}, TY_t, K_{t-1}), \\
& compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \\
& compose(NA_2, K_{t-1}, NA), r\_d_1(TXs, Y, NA)
\end{aligned}
$$

By $t - p + 1$ times folding clause 38 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 39: \quad & r\_d_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \\
& r\_d_1([TX_p, \ldots, TX_t|TXs], Y, NA_2)
\end{aligned}
$$

By folding clause 39 using clauses 1 and 2:

$$\begin{aligned}
clause\ 40: \quad & r_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \\
& compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \\
& compose(A, K_{p-2}, NA_1), \\
& r_td_1([N, TX_p, \ldots, TX_t|TXs], Y, NA_1)
\end{aligned}$$

By $p-1$ times folding clause 40 using clauses 1 and 2:

$$\begin{aligned}
clause\ 41: \quad & r_td_1(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r_td_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], Y, A)
\end{aligned}$$

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of $P_{r\_td_1}$. Therefore $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$, we do a backward proof that we begin with P_r in $TDGLR$ and from which we try to obtain S_r.

The procedure P_r for r in $TDGLR$ is:

$$r(X, Y) \leftarrow \quad r\_td_1([X], Y, e)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_td_1([X], Y, e)]$$

By unfolding the 'completion' above wrt $r\_td_1([X], Y, e)$ using $S_{r\_td_1}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge \mathcal{O}_c(e, I_1, Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_td_1}\}$.
Therefore, $TDGLR$ is also steadfast wrt S_r in $\mathcal{S}$. □

5 Proofs of the Duality Schemas

Theorem 11 The duality schema D_{dc}, which is given below, is correct.

D_{dc} : $\langle$ $DCLR$, $DCRL$, A_{ddc}, O_{ddc12}, $O_{ddc21}\rangle$ where

A_{ddc} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element, where e appears in $DCLR$ and $DCRL$

O_{ddc12} : - partial evaluation of the conjunction $process(HX, HY), compose(HY, I_p, I_{p-1})$ results in the introduction of a non-recursive relation

O_{ddc21} : - partial evaluation of the conjunction $process(HX, HY), compose(I_{p-1}, HY, I_p)$ results in the introduction of a non-recursive relation where the template $DCLR$ is Logic Program Template 1 in Section 2 and the template $DCRL$ is Logic Program Template 3 in Section 3.

The specification S_r of both a $DCLR$ program and a $DCRL$ program for relation r is:

$$\forall X : \mathcal{X}.\ \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

Proof 11 To prove the correctness of the duality schema D_{dc}, by Definition 10, we have to prove that templates $DCLR$ and $DCRL$ are *equivalent* wrt S_r under the applicability conditions A_{ddc}. By Definition 5, the templates $DCLR$ and $DCRL$ are *equivalent* wrt S_r under the applicability conditions A_{ddc} iff $DCLR$ is *equivalent to* $DCRL$ wrt the specification S_r provided that the conditions in A_{ddc} hold. By Definition 4, $DCLR$ is *equivalent to* $DCRL$ wrt the specification S_r iff the following two conditions hold:

(a) $DCLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{process}$, $S_{decompose}$, $S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DCLR$.

(b) $DCRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by duality transformation of P.

In program transformation, we assume that the input program, here template $DCLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use Definition 3: $DCRL$ is steadfast wrt S_r in $\mathcal{S}$ iff $DCRL \cup P_S$ is correct wrt S_r, where P_S is any closed program such that P_S is correct wrt each specification in $\mathcal{S}$ and P_S contains no occurrences of the relation r.

To prove that $DCRL$ is steadfast wrt S_r in $\mathcal{S}$, we do a constructive forward proof that we begin with S_r and from which we try to obtain the open program $DCRL$.

By taking the 'decompletion' of S_r:

$$clause\ 1:\quad r(X,Y) \leftarrow r(X,Y)$$

By unfolding clause 1 wrt the atom $r(X,Y)$ on the right-hand side of $\leftarrow$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll} clause\ 2: & r(X,Y) \leftarrow \\ & \quad minimal(X), \\ & \quad solve(X,Y) \\ clause\ 3: & r(X,Y) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad I_0 = e, \\ & \quad compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \\ & \quad process(HX, HY), compose(I_{p-1}, HY, I_p), \\ & \quad compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \\ & \quad Y = I_{t+1} \end{array}$$

By using applicability condition (1) on clause 3:

$$\begin{array}{ll} clause\ 4: & r(X,Y) \leftarrow \\ & \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\ & \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\ & \quad compose(TY_{t-1}, TY_t, A_{t-1}), \\ & \quad compose(TY_{t-2}, A_{t-1}, A_{t-2}), \ldots, compose(TY_p, A_{p+1}, A_p), \\ & \quad process(HX, HY), compose(HY, A_p, A_{p-1}), \\ & \quad compose(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TY_1, A_1, A_0), \\ & \quad compose(e, A_0, Y) \end{array}$$

By using applicability conditions (1) and (2) on clause 4:

$clause\ 5: \quad r(X,Y) \leftarrow$
$$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$$
$$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$$
$$compose(TY_t, e, A_t),$$
$$compose(TY_{t-1}, A_t, A_{t-1}), \ldots, compose(TY_p, A_{p+1}, A_p),$$
$$process(HX, HY), compose(HY, A_p, A_{p-1}),$$
$$compose(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TY_1, A_1, A_0),$$
$$Y = A_0$$

By introducing a new, i.e. existentially quantified, variable A_{t+1}:

$clause\ 6: \quad r(X,Y) \leftarrow$
$$nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),$$
$$r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$$
$$A_{t+1} = e,$$
$$compose(TY_t, A_{t+1}, A_t), \ldots, compose(TY_p, A_{p+1}, A_p),$$
$$process(HX, HY), compose(HY, A_p, A_{p-1}),$$
$$compose(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, compose(TY_1, A_1, A_0),$$
$$Y = A_0$$

Clauses 2 and 6 are the clauses of $DCRL$.
Therefore $DCRL$ is steadfast wrt S_r in $\mathcal{S}$. □

Theorem 12 The duality schema D_{dg}, which is given below, is correct.

D_{dg} : $\langle$ $DGLR$, $DGRL$, A_{ddg}, O_{ddg12}, $O_{ddg21}\rangle$ where
A_{ddg} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element,
O_{ddg12}: - $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
- partial evaluation of the conjunction
$process(HX, HY), compose(HY, A_p, A_{p-1})$
results in the introduction of a non-recursive relation
O_{ddg21}: - $\mathcal{I}_r(X) \wedge minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
- partial evaluation of the conjunction
$process(HX, HY), compose(A_{p-1}, HY, A_p)$
results in the introduction of a non-recursive relation

and the templates $DGLR$ and $DGRL$ are Logic Program Templates 4 and 5 in Section 3.
The specification S_r of both a $DGLR$ program and a $DGRL$ program for relation r is:

$$\forall X : \mathcal{X}.\ \forall Y : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

The specification $S_{r\_descending_1}$ of relation $r\_descending_1$ is:

$$\forall X : \mathcal{X}.\ \forall Y, A : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r\_descending_1(X,Y,A) \ \Leftrightarrow\ \exists S : \mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(A,S,Y)]$$

The specification $S_{r\_descending_2}$ of relation $r\_descending_2$ is:

$$\forall X : \mathcal{X}.\ \forall Y, A : \mathcal{Y}.\ \ \mathcal{I}_r(X) \Rightarrow [r\_descending_2(X,Y,A) \ \Leftrightarrow\ \exists S : \mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,A,Y)]$$

Proof 12 To prove the correctness of the duality schema D_{dg}, by Definition 10, we have to prove that templates $DGLR$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{ddg}. By Definition 5, the templates $DGLR$ and $DGRL$ are *equivalent* wrt S_r under the applicability conditions A_{ddg} iff $DGLR$ is *equivalent to* $DGRL$ wrt the specification S_r provided that the conditions in A_{ddg} hold. By Definition 4, $DGLR$ is *equivalent to* $DGRL$ wrt the specification S_r iff the following two conditions hold:

(a) $DGLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}, S_{nonMinimal}, S_{solve}, S_{process}, S_{decompose}, S_{compose}$ are the specifications of *minimal*, *nonMinimal*, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $DGLR$.

(b) $DGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by duality transformation of P.

In program transformation, we assume that the input program, here template $DGLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, where $P_{r\_descending_2}$ is the procedure for $r\_descending_2$, and P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_descending_2}$ and from which we try to obtain $P_{r\_descending_2}$.

By taking the 'decompletion' of $S_{r\_descending_2}$:

$$clause\ 1: \quad r\_descending_2(X, Y, A) \leftarrow r(X, S), compose(S, A, Y)$$

By unfolding clause 1 wrt $r(X, S)$ using $DGLR$, and using the assumption that $DGLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$clause\ 2: \quad r\_descending_2(X, Y, A) \leftarrow r\_descending_1(X, S, e), compose(S, A, Y)$$

By unfolding clause 2 wrt $r\_descending_1(X, S, e)$ using $DGLR$, and using the assumption that $P_{r\_descending_1}$ is steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$, since $P_{r\_descending_1}$ must be steadfast wrt $S_{r\_descending_1}$ in $\mathcal{S}$ for $DGLR$ to be steadfast wrt S_r in $\mathcal{S}$:

$$\begin{aligned}
clause\ 3: \quad & r_descending_2(X, Y, A) \leftarrow \\
& \quad minimal(X), \\
& \quad solve(X, S'), compose(e, S', S), compose(S, A, Y) \\
clause\ 4: \quad & r_descending_2(X, Y, A) \leftarrow \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad compose(e, e, A_0), \\
& \quad r_descending_1(TX_1, A_1, A_0), \ldots, r_descending_1(TX_{p-1}, A_{p-1}, A_{p-2}), \\
& \quad process(HX, HS), compose(A_{p-1}, HS, A_p), \\
& \quad r_descending_1(TX_p, A_{p+1}, A_p), \ldots, r_descending_1(TX_t, A_{t+1}, A_t), \\
& \quad S = A_{t+1}, compose(S, A, Y)
\end{aligned}$$

By using applicability condition (2) on clause 3:

$$\begin{aligned}
clause\ 5: \quad & r_descending_2(X, Y, A) \leftarrow \\
& \quad minimal(X), \\
& \quad solve(X, S'), S = S, compose(S, A, Y)
\end{aligned}$$

By simplification:

$$\begin{aligned}
clause\ 6: \quad & r_descending_2(X, Y, A) \leftarrow \\
& \quad minimal(X), \\
& \quad solve(X, S), compose(S, A, Y)
\end{aligned}$$

By t times unfolding clause 4 wrt the atoms
$r\_descending_1(TX_1, A_1, A_0), \ldots, r\_descending_1(TX_{p-1}, A_{p-1}, A_{p-2}),$
$r\_descending_1(TX_p, A_{p+1}, A_p), \ldots, r\_descending_1(TX_t, A_{t+1}, A_t)$
using the 'decompletion' of $S_{r\_descending_1}$ (refer to Proofs 3 and 6):

$$\begin{aligned}
clause\ 7: \quad & r_descending_2(X, Y, A) \leftarrow \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad compose(e, e, A_0), \\
& \quad r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \\
& \quad compose(A_0, TS_1, A_1), \ldots, compose(A_{p-2}, TS_{p-1}, A_{p-1}), \\
& \quad process(HX, HS), compose(A_{p-1}, HS, A_p), \\
& \quad r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \\
& \quad compose(A_p, TS_p, A_{p+1}), \ldots, compose(A_t, TS_t, A_{t+1}), \\
& \quad S = A_{t+1}, compose(S, A, Y)
\end{aligned}$$

By using applicability condition (1) on clause 7:

$$
\begin{aligned}
\textit{clause } 8: \quad & r\_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad compose(e, I, Y), compose(e, A_0, I), \\
& \quad r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \\
& \quad compose(TS_1, A_1, A_0), \ldots, compose(TS_{p-1}, A_{p-1}, A_{p-2}), \\
& \quad process(HX, HS), compose(HS, A_p, A_{p-1}), \\
& \quad r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \\
& \quad compose(TS_p, A_{p+1}, A_p), \ldots, compose(TS_t, A, A_t)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
\textit{clause } 9: \quad & r\_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad Y = A_0, \\
& \quad r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \\
& \quad compose(TS_1, A_1, A_0), \ldots, compose(TS_{p-1}, A_{p-1}, A_{p-2}), \\
& \quad process(HX, HS), compose(HS, A_p, A_{p-1}), \\
& \quad r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \\
& \quad compose(TS_p, A_{p+1}, A_p), \ldots, compose(TS_t, A_{t+1}, A_t), \\
& \quad compose(e, A, A_{t+1})
\end{aligned}
$$

By t times folding clause 9 using clause 1:

$$
\begin{aligned}
\textit{clause } 10: \quad & r\_descending_2(X,Y,A) \leftarrow \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad compose(e, A, A_{t+1}), \\
& \quad r\_descending_2(TX_t, A_t, A_{t+1}), \ldots, r\_descending_2(TX_p, A_p, A_{p+1}), \\
& \quad process(HX, HY), compose(HY, A_p, A_{p-1}), \\
& \quad r\_descending_2(TX_{p-1}, A_{p-2}, A_{p-1}), \ldots, r\_descending_2(TX_1, A_0, A_1), \\
& \quad Y = A_0
\end{aligned}
$$

Clauses 2 and 10 are the clauses of $P_{r\_descending_2}$. Therefore $P_{r\_descending_2}$ is steadfast wrt $S_{r\_descending_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$, we do a backward proof that we begin with P_r in *DGRL* and from which we try to obtain S_r.
The procedure P_r for r in *DGRL* is:

$$r(X,Y) \leftarrow \quad r\_descending_2(X,Y,e)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r\_descending_2(X,Y,e)]$$

By unfolding the 'completion' above wrt $r\_descending_2(X,Y,e)$ using $S_{r\_descending_2}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S : \mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge \mathcal{O}_c(S,e,Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \exists S : \mathcal{Y}.\ \mathcal{O}_r(X,S) \wedge S = Y]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \ \Leftrightarrow\ \mathcal{O}_r(X,Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_descending_2}\}$.
Therefore, *DGRL* is also steadfast wrt S_r in $\mathcal{S}$. □

Theorem 13 The duality schema D_{tdg}, which is given below, is correct.

D_{tdg} : $\langle$ *TDGLR*, *TDGRL*, A_{dtdg}, O_{dtdg12}, $O_{dtdg21}\rangle$ where
A_{dtdg} : (1) *compose* is associative
(2) *compose* has e as the left and right identity element,

where e appears in $TDGLR$ and $TDGRL$

O_{dtdg12}: - $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \land minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
- partial evaluation of the conjunction
$process(HX, HY), compose(HY, NewA, F)$
results in the introduction of a non-recursive relation

O_{dtdg21}: - $\forall X : \mathcal{X}.\ \mathcal{I}_r(X) \land minimal(X) \Rightarrow \mathcal{O}_r(X, e)$
- partial evaluation of the conjunction
$process(HX, HY), compose(A, HY, NewA)$
results in the introduction of a non-recursive relation

where the templates $TDGLR$ and $TDGRL$ are Logic Program Templates 6 and 7 in Section 4.

The specification S_r of relation r is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \mathcal{O}_r(X,Y)]$$

The specification of r_td_1, namely $S_{r\_td_1}$, is:

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_1(Xs, Y, A) \Leftrightarrow (Xs = [\,] \land Y = A)$$
$$\lor (Xs = [X_1, X_2, \ldots, X_q] \land \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \land I_1 = Y_1 \land \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)$$
$$\land \mathcal{O}_c(A, I_q, I_{q+1}) \land Y = I_{q+1})]$$

The specification of r_td_2, namely $S_{r\_td_2}$, is:

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y, A : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow (Xs = [\,] \land Y = A)$$
$$\lor (Xs = [X_1, X_2, \ldots, X_q] \land \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \land I_1 = Y_1 \land \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i)$$
$$\land \mathcal{O}_c(I_q, A, I_{q+1}) \land Y = I_{q+1})]$$

Proof 13 To prove the correctness of the duality schema D_{tdg}, by Definition 10, we have to prove that templates $TDGLR$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dtdg}. By Definition 5, the templates $TDGLR$ and $TDGRL$ are *equivalent* wrt S_r under the applicability conditions A_{dtdg} iff $TDGLR$ is *equivalent to* $TDGRL$ wrt the specification S_r provided that the conditions in A_{dtdg} hold. By Definition 4, $TDGLR$ is *equivalent to* $TDGRL$ wrt the specification S_r iff the following two conditions hold:

(a) $TDGLR$ is steadfast wrt S_r in $\mathcal{S} = \{S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose}\}$, where $S_{minimal}$, $S_{nonMinimal}$, S_{solve}, $S_{process}$, $S_{decompose}$, $S_{compose}$ are the specifications of $minimal$, $nonMinimal$, *solve*, *decompose*, *process*, *compose*, which are all the undefined relation names appearing in $TDGLR$.

(b) $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $\mathcal{S}$ when Q is obtained by duality transformation of P.

In program transformation, we assume that the input program, here template $TDGLR$, is steadfast wrt S_r in $\mathcal{S}$, so condition (a) always holds.

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $TDGRL$ is steadfast wrt S_r in $\mathcal{S}$ if $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, where $P_{r\_td_2}$ is the procedure for r_td_2, and P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, where P_r is the procedure for r.

To prove that $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r\_td_2}$ and from which we try to obtain $P_{r\_td_2}$.

If we separate the cases of $q \geq 1$ by $q = 1 \lor q \geq 2$, then $S_{r\_td_2}$ becomes:

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \land Y = A)$$
$$\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, Y_1) \land Y_1 = I_1 \land \mathcal{O}_c(I_1, A, I_2) \land Y = I_2)$$
$$\lor (Xs = [X_1, X_2, \ldots, X_q] \land \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \land I_1 = Y_1 \land \ \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \land \mathcal{O}_c(I_q, A, I_{q+1}) \land Y = I_{q+1})]$$

where $q \geq 2$.

By using applicability conditions (1) and (2):

$$\forall Xs : list\ of\ \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \land Y = A)$$
$$\lor (Xs = [X_1|TXs] \land TXs = [\,] \land \mathcal{O}_r(X_1, Y_1) \land Y_1 = I_1 \land TY = A \land \mathcal{O}_c(TY, A, NA) \land \mathcal{O}_c(I_1, NA, Y))$$
$$\lor (Xs = [X_1|TXs] \land TXs = [X_2, \ldots, X_q] \land \ \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \ \land Y_1 = I_1 \land Y_2 = I_2 \land$$
$$\bigwedge_{i=3}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \ \land TY = I_q \land \mathcal{O}_c(TY, A, NA) \land \mathcal{O}_c(I_1, NA, Y))]$$

where $q \geq 2$.

By folding using $S_{r\_td_2}$, and renaming:

$$\forall Xs : \textit{list of } \mathcal{X}, \forall Y : \mathcal{Y}.\ (\forall X : \mathcal{X}.\ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r\_td_2(Xs, Y, A) \Leftrightarrow$$
$$(Xs = [\,] \wedge Y = A)$$
$$\vee(Xs = [X|TXs] \wedge \mathcal{O}_r(X, HY) \wedge r\_td_2(TXs, NA, A) \wedge \mathcal{O}_c(HY, NA, Y))]$$

By taking the 'decompletion':

$$\begin{array}{ll}
\textit{clause } 1: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [\,], Y = A \\
\textit{clause } 2: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], r(X, HY), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}$$

By unfolding clause 2 wrt $r(X, HY)$ using $TDGLR$, and using the assumption that $DCLR$ is steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll}
\textit{clause } 3: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], r_td1([X], HY, e), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}$$

By unfolding clause 3 wrt $r\_td_1([X], HY, e)$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$, since $P_{r\_td_1}$ must be steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$ for $TDGLR$ to be steadfast wrt S_r in $\mathcal{S}$:

$$\begin{array}{ll}
\textit{clause } 4: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad minimal(X), solve(X, HY'), \\
& \quad compose(e, HY', NA'), r_td1(TXs', HY, NA'), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y) \\
\textit{clause } 5: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad process(HX, HY'), compose(e, HY', NA'), \\
& \quad r_td_1(TXs', HY, NA'), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y) \\
\textit{clause } 6: & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad process(HX, HY'), compose(e, HY', NA'), \\
& \quad r_td_1([TX_p, \ldots, TX_t|TXs'], HY, NA'), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{array}$$

$$
\begin{aligned}
clause\ 7: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r\_td_1([TX_1, \ldots, TX_{p-1}, N|TXs'], HY, e), \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y) \\
clause\ 8: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r\_td_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs'], HY, e), \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By unfolding clause 4 wrt $r\_td_1(TXs', HY, NA')$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$:

$$
\begin{aligned}
clause\ 9: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad minimal(X), solve(X, HY'), \\
& \quad compose(e, HY', NA'), \\
& \quad TXs' = [\,], HY = NA', \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{aligned}
clause\ 10: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad minimal(X), solve(X, HY'), \\
& \quad HY' = NA', \\
& \quad TXs' = [\,], HY = NA', \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By simplification:

$$
\begin{aligned}
clause\ 11: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad minimal(X), solve(X, HY), \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By unfolding clause 5 wrt $r\_td_1(TXs', HY, NA')$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$:

$$
\begin{aligned}
clause\ 12: \quad & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_t), \\
& \quad process(HX, HY'), compose(e, HY', NA'), \\
& \quad TXs' = [\,], HY = NA', \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability condition (2):

$$
\begin{array}{ll}
\mathit{clause}\ 13: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_t), \\
& \quad process(HX,HY'), HY' = NA', \\
& \quad TXs' = [\,], HY = NA', \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y)
\end{array}
$$

By simplification:

$$
\begin{array}{ll}
\mathit{clause}\ 14: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_t), \\
& \quad r\_td_2(TXs,NA,A), \\
& \quad process(HX,HY), compose(HY,NA,Y)
\end{array}
$$

By t times unfolding clause 6 wrt

$$
r\_td_1([TX_p,\ldots,TX_t|TXs'],HY,NA'),\ldots,r\_td_1([TX_t|TXs'],HY,NA_{t-1})
$$

using the "decompletion" of $S_{r\_td_1}$ in Section 4:

$$
\begin{array}{ll}
\mathit{clause}\ 15: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad process(HX,HY'), compose(e,HY',NA'), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t), \\
& \quad compose(NA',TY_p,NA_p),\ldots,compose(NA_{t-1},TY_t,NA_t), \\
& \quad r\_td_1(TXs',HY,NA_t), \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y)
\end{array}
$$

By unfolding clause 15 wrt $r\_td_1(TXs',HY,NA_t)$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$:

$$
\begin{array}{ll}
\mathit{clause}\ 16: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad process(HX,HY'), compose(e,HY',NA'), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t), \\
& \quad compose(NA',TY_p,NA_p),\ldots,compose(NA_{t-1},TY_t,NA_t), \\
& \quad TXs' = [\,], HY = NA_t, \\
& \quad r\_td_2(TXs,NA,A), compose(HY,NA,Y)
\end{array}
$$

By using applicability conditions (1) and (2), and simplification:

$$
\begin{array}{ll}
\mathit{clause}\ 17: & r\_td_2(Xs,Y,A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X,HX,TX_1,\ldots,TX_t), \\
& \quad minimal(TX_1),\ldots,minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p);\ldots;nonMinimal(TX_t)), \\
& \quad process(HX,HY), compose(HY,I_p,Y), \\
& \quad r(TX_p,TY_p),\ldots,r(TX_t,TY_t), \\
& \quad compose(TY_p,I_{p+1},I_p),\ldots,compose(TY_t,NA,I_t), \\
& \quad r\_td_2(TXs,NA,A)
\end{array}
$$

By t times folding clause 17 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 18:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad minimal(TX_1), \ldots, minimal(TX_{p-1}), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad r\_td_2([TX_p, \ldots, TX_t|TXs], NA, A), \\
& \quad process(HX, HY), compose(HY, NA, Y)
\end{aligned}
$$

By p times unfolding clause 7 wrt

$$r\_td_1([TX_1, \ldots, TX_{p-1}, N|TXs'], HY, NA'), \ldots, r\_td_1([N|TXs'], HY, NA_{p-1})$$

using the "decompletion" of $S_{r\_td_1}$ in Section 4:

$$
\begin{aligned}
clause\ 19:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \\
& \quad compose(e, TY_1, NA_1), \\
& \quad compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
& \quad compose(NA_{p-1}, YN, NA_p), \\
& \quad r\_td_1(TXs', HY, NA_p), \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By unfolding clause 19 wrt $r\_td_1(TXs', HY, NA_p)$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$:

$$
\begin{aligned}
clause\ 20:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \\
& \quad compose(e, TY_1, NA_1), \\
& \quad compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
& \quad compose(NA_{p-1}, YN, NA_p), \\
& \quad TXs' = [\,], HY = NA_p, \\
& \quad r\_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}
$$

By using applicability conditions (1) and (2), and simplification:

$$
\begin{aligned}
clause\ 21:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \\
& \quad compose(TY_1, I_1, Y), \\
& \quad compose(TY_2, I_2, I_1), \ldots, compose(TY_{p-1}, I_p, I_{p-1}), \\
& \quad compose(YN, NA, I_p), \\
& \quad r\_td_2(TXs, NA, A)
\end{aligned}
$$

By p times folding clause 21 using clauses 1 and 2:

$$\begin{aligned}
clause\ 22: \quad & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad minimal(TX_p), \ldots, minimal(TX_t), \\
& \quad minimal(U_1), \ldots, minimal(U_{p-1}), \\
& \quad decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
& \quad r_td_2([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A)
\end{aligned}$$

By $t + 1$ times unfolding clause 8 wrt

$$r\_td_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs'], HY, NA'), \ldots, r\_td_1([TX_t|TXs'], HY, NA_t)$$

using the "decompletion" of $S_{r\_td_1}$ in Section 4:

$$\begin{aligned}
clause\ 23: \quad & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, YN), \\
& \quad compose(e, TY_1, NA_1), \\
& \quad compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
& \quad compose(NA_{p-1}, YN, NA_p), \\
& \quad compose(NA_p, TY_p, NA_{p+1}), \ldots, compose(NA_t, TY_t, NA_{t+1}), \\
& \quad r_td_1(TXs', HY, NA_{t+1}), \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}$$

By unfolding clause 23 wrt $r\_td_1(TXs', HY, NA_{t+1})$ using $TDGLR$, and using the assumption that $P_{r\_td_1}$ is steadfast wrt $S_{r\_td_1}$ in $\mathcal{S}$:

$$\begin{aligned}
clause\ 24: \quad & r_td_2(Xs, Y, A) \leftarrow \\
& \quad Xs = [X|TXs], \\
& \quad Xs' = [X|TXs'], TXs' = [\,], \\
& \quad nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& \quad (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& \quad (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& \quad minimal(U_1), \ldots, minimal(U_t), \\
& \quad decompose(N, HX, U_1, \ldots, U_t), \\
& \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, YN), \\
& \quad compose(e, TY_1, NA_1), \\
& \quad compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
& \quad compose(NA_{p-1}, YN, NA_p), \\
& \quad compose(NA_p, TY_p, NA_{p+1}), \ldots, compose(NA_t, TY_t, NA_{t+1}), \\
& \quad TXs' = [\,], HY = NA_{t+1}, \\
& \quad r_td_2(TXs, NA, A), compose(HY, NA, Y)
\end{aligned}$$

By using applicability conditions (1) and (2), and simplification:

$$
\begin{aligned}
clause\ 25:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, YN), \\
& compose(TY_1, I_1, Y), \\
& compose(TY_2, I_2, I_1), \ldots, compose(TY_{p-1}, I_p, I_{p-1}), \\
& compose(YN, I_{p+1}, I_p), \\
& compose(TY_p, I_{p+2}, I_{p+1}), \ldots, compose(TY_{t-1}, I_{t+1}, I_t), \\
& compose(TY_t, NA, I_{t+1}), \\
& r\_td_2(TXs, NA, A)
\end{aligned}
$$

By $t+1$ times folding clause 25 using clauses 1 and 2:

$$
\begin{aligned}
clause\ 26:\ & r\_td_2(Xs, Y, A) \leftarrow \\
& Xs = [X|TXs], \\
& nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
& (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
& (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
& minimal(U_1), \ldots, minimal(U_t), \\
& decompose(N, HX, U_1, \ldots, U_t), \\
& r\_td_2([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], Y, A)
\end{aligned}
$$

Clauses 1, 11, 14, 18, 22 and 26 are the clauses of $P_{r\_td_2}$. Therefore $P_{r\_td_2}$ is steadfast wrt $S_{r\_td_2}$ in $\mathcal{S}$.

To prove that P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$, we do a backward proof that we begin with P_r in $TDGRL$ and from which we try to obtain S_r.
The procedure P_r for r in $TDGRL$ is:

$$r(X, Y) \leftarrow \quad r\_td_2([X], Y, e)$$

By taking the 'completion':

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\_td_2([X], Y, e)]$$

By unfolding the 'completion' above wrt $r\_td_2([X], Y, e)$ using $S_{r\_td_2}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge \mathcal{O}_c(I_1, e, Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}.\ \mathcal{O}_r(X, Y_1) \wedge I_1 = Y_1 \wedge Y = I_1]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}.\ \mathcal{I}_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)]$$

We obtain S_r, so P_r is steadfast wrt S_r in $\{S_{r\_td_2}\}$.
Therefore, $TDGRL$ is also steadfast wrt S_r in $\mathcal{S}$. □

6 Conclusion

In this report, we have proven the correctness of the 13 transformation schemas in [3]. The transformation schemas and their schema patterns can be given as the graph in Figure 1 below, where the schema patterns are the nodes of the graph, and the transformation schemas are the edges. The arrow indicates in what way the transformation schema is proved (i.e., the arrow is printed from the assumed input program

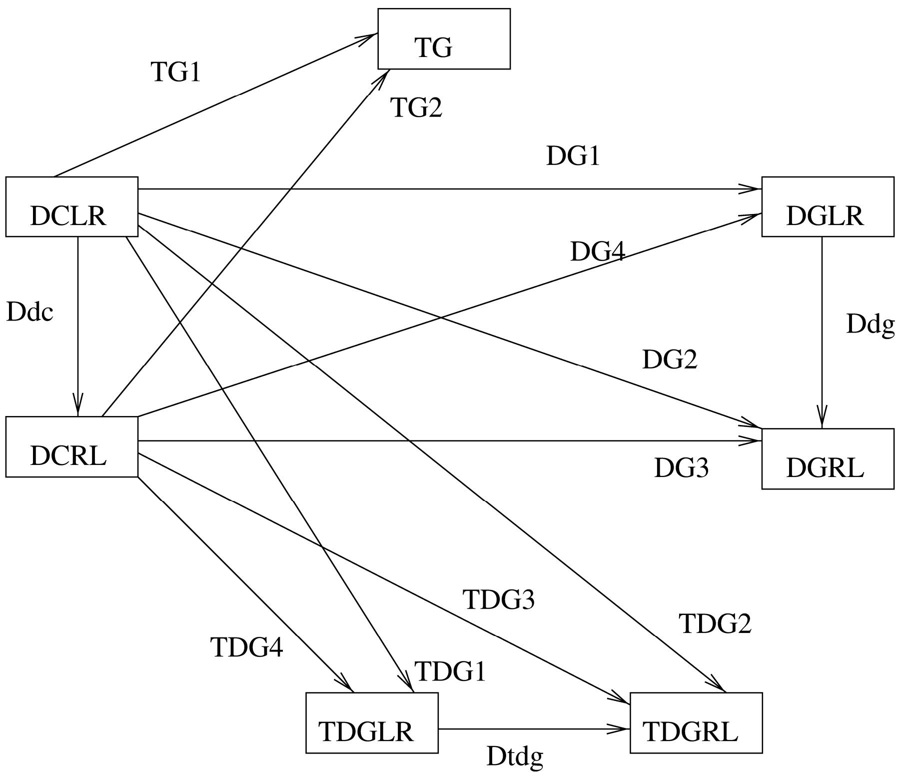

Figure 1: A Graph to Represent the Correctness Proofs of Transformation Schemas

schema pattern to the output program schema pattern in the proof of the corresponding transformation schema). Each of these transformation schemas can of course be proven in the other direction, since these transformation schemas are applicable in both directions.
Therefore, the transformation schemas proved in this report are a successful pre-compilation of the corresponding transformation techniques.

References

[1] R.S. Bird and P. Wadler. *Introduction to Functional Programming.* Prentice Hall, 1988.

[2] R.S. Bird. The promotion and accumulation strategies in transformational programming. *ACM Transactions on Programming Languages and Systems* 6(4):487–504, 1984.

[3] H. Büyükyıldız. *Schema-based Logic Program Transformation.* M.Sc. Thesis, Bilkent University, Department of Computer Science, 1997.

[4] Y. Deville. *Logic Programming: Systematic Program Development.* Addison Wesley, 1990.

[5] P. Flener, K.-K. Lau, and M. Ornaghi. On correct program schemas. In: N.E. Fuchs (ed), *Proc. of LOPSTR'97*, LNCS. Springer-Verlag, forthcoming.