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 if 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:

$$\forall X : \mathcal{X}. \forall Y : \mathcal{Y}. \; I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)]$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the sorts (or: types) of $X$ and $Y$, respectively, $I_r(X)$ denotes the input condition that must be fulfilled before the execution of the program, and $O_r(X,Y)$ 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 $\Pi$ 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 $\Pi$ 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 \emph{(totally) correct wrt its specification \( S_r \)} iff, for any ground term \( t \) of \( \mathcal{X} \) such that \( I_r(t) \) holds, the following condition holds: 
\[ P \vdash r(t, u) \text{ iff } \mathcal{F} \models Q_r(t, u), \text{ for every ground term } u \text{ of } \mathcal{Y}. \]

If we replace ‘iff’ by ‘implies’ in the condition above, then \( P \) is said to be \emph{partially correct wrt \( S_r \)}, and if we replace ‘iff’ by ‘if’, then \( P \) is said to be \emph{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 \( r \)
- \( q_1, \ldots, q_m \) be all the undefined relation names appearing in \( P \)
- \( S_1, \ldots, S_m \) be the specifications of \( q_1, \ldots, q_m \)

We say that \( P \) is \emph{steadfast wrt its specification \( S_r \)} in \( \{S_1, \ldots, S_m\} \) 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 \( S_r \)
- \( p_1, \ldots, p_t \) be all the defined relation names appearing in \( P \) (including \( r \) thus)
- \( q_1, \ldots, q_m \) be all the undefined relation names appearing in \( P \)
- \( S_1, \ldots, S_m \) be the specifications of \( q_1, \ldots, q_m \)

For \( t \geq 2 \), the program \( P \) is steadfast wrt \( S_r \) in \( \{S_1, \ldots, S_m\} \) 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 \emph{equivalent to} \( Q \) wrt the specification \( S_r \) iff the following two conditions hold:

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

(b) \( Q \) is steadfast wrt \( S_r \) in \( \{S'_1, \ldots, S'_t\} \), where \( S'_1, \ldots, S'_t \) are the specifications of \( q_1, \ldots, q_t \), 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 \emph{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 \( F \), 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 \( F \), a **program schema** for a relation \( r \) is a pair \( (T, C) \), 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 \( F \), \( T \) 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, \( TX_1, \ldots, TX_i \) is a term ellipse, and \( \bigwedge_{i=1}^n r(TX_i, TY_i) \) is a conjunct ellipse.

**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 \( (S_1, S_2, A, O_{12}, O_{21}) \), 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 \( (S_1, S_2, A, O_{12}, O_{21}) \) 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 \( (S_1, S_2, A, O_{12}, O_{21}) \), we have to prove the equivalence of \( T_1 \) and \( T_2 \), which are the templates of \( S_1 = (T_1, C_1) \) and \( S_2 = (T_2, C_2) \).

We assume that the template \( T_i \) of the input program schema \( S_i = (T_i, C_i) \) (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) \( S_j = (T_j, C_j) \) (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.

\[
TG_1 : \{ DCLR, TG, A_{t1}, O_{t12}, O_{t121} \}
\]

where

\( A_{t1} : (1) \) \( \text{compose} \) is associative

\( (2) \) \( \text{compose} \) has \( e \) as the left and right identity element

\( (3) \forall X : X. \ I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e) \)

\( (4) \forall X : X. \ I_r(X) \Rightarrow [\neg \text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)] \)

\( O_{t12} : \) partial evaluation of the conjunction

\( \text{process}(HX, HY), \text{compose}(HY, TY, Y) \)

results in the introduction of a non-recursive relation

\( O_{t121} : \) partial evaluation of the conjunction

\( \text{process}(HX, HY), \text{compose}(Ip_{-1}, HY, Ip) \)

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**

\[
\begin{align*}
r(X, Y) & \leftarrow \\
& \text{minimal}(X), \\
& \text{solve}(X, Y) \\
r(X, Y) & \leftarrow \\
& \text{nonMinimal}(X), \\
& \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
& I_0 = e, \\
& \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(Ip_{-2}, TY_{p-1}, Ip_{-1}), \\
& \text{process}(HX, HY), \text{compose}(Ip_{-1}, HY, Ip), \\
& \text{compose}(Ip, TY_p, Ip_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
& Y = I_{t+1}
\end{align*}
\]

**Logic Program Template 2**

\[
\begin{align*}
r(X, Y) & \leftarrow \\
& \text{r.tupling}([X], Y) \\
r.tupling(Xs, Y) & \leftarrow \\
& Xs = []. \\
& Y = e \\
r.tupling(Xs, Y) & \leftarrow \\
& Xs = [X|TXs], \\
& \text{minimal}(X), \\
& \text{r.tupling}(TXs, TY), \\
& \text{solve}(X, HY), \\
& \text{compose}(HY, TY, Y) \\
r.tupling(Xs, Y) & \leftarrow \\
& Xs = [X|TXs], \\
& \text{nonMinimal}(X), \\
& \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t) \\
& \text{r.tupling}(TXs, TY), \\
& \text{process}(HX, HY)
\end{align*}
\]
\[ \text{compose}(HY, TY, Y) \]
\[ r \text{.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X | TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \]
\[ \langle \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \rangle, \]
\[ r \text{.tupling}([TX_p, \ldots, TX_t | TXs], TY), \]
\[ \text{process}(HX, HY), \]
\[ \text{compose}(HY, TY, Y) \]
\[ r \text{.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X | TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \langle \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \rangle, \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \]
\[ r \text{.tupling}([TX_1, \ldots, TX_{p-1}, N | TXs], Y) \]
\[ r \text{.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X | TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \langle \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \rangle, \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{t}), \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_t), \]
\[ r \text{.tupling}([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t | TXs], Y) \]

and the specification $S_r$ of relation $r$ is:
\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)] \]

and the specification $S_{r \text{.tupling}}$ of relation $r \text{.tupling}$ is:
\[ \forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \quad X \in Xs \Rightarrow I_r(X)) \Rightarrow [r \text{.tupling}(Xs, Y) \Leftrightarrow \bigwedge_{i=1}^{q} O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=q+1}^{t} O_r(I_{i-1}, Y_i, I_i) \land Y = I_t] \]

where $O_r$ is the output condition of $\text{compose}$ and $q > 1$.  

**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 $A_{11}$. By Definition 5, the templates $DCLR$ and $TG$ are equivalent wrt $S_r$ under the applicability conditions $A_{11}$ iff $DCLR$ is equivalent to $TG$ wrt the specification $S_r$ provided that the conditions in $A_{11}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}$ are the specifications of $\text{minimal, nonMinimal, solve, decompose, process, compose}$, which are all the undefined relation names appearing in $DCLR$.

(b) $TG$ is also steadfast wrt $S_r$ in $S$. 

5
Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S'_1, \ldots, S'_l\} \) in Definition 4 are equal to \( S \) 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 \( S \), 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 \( S \) if \( P_r,\text{tupling} \) is steadfast wrt \( S_r,\text{tupling} \) in \( S \), where \( P_r,\text{tupling} \) is the procedure for \( r,\text{tupling} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r,\text{tupling}\} \), where \( P_r \) is the procedure for \( r \).

To prove that \( P_r,\text{tupling} \) is steadfast wrt \( S_r,\text{tupling} \) in \( S \), we do a constructive forward proof that we begin with \( S_r,\text{tupling} \) and from which we try to obtain \( P_r,\text{tupling} \).

If we separate the cases of \( q \geq 1 \) by \( q = 1 \) or \( q \geq 2 \), then \( S_r,\text{tupling} \) becomes:

\[
\forall Xs : \text{list of } X, \forall Y : Y. \ (\forall X : X. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r,\text{tupling}(Xs,Y) \Rightarrow S_r,\text{tupling}]
\]

\[
\forall Xs = [\emptyset] \land Y = \emptyset
\]

\[
\forall (Xs = [X_1] \land \text{O_r}(X_1,Y_1) \land Y_1 = I_1 \land Y = I_1)
\]

\[
\forall (Xs = [X_1,X_2,\ldots,X_q] \land \bigwedge_{i=1}^{q} \text{O_r}(X_i,Y_i) \land Y_1 = Y_2 = I_2 \land Y = I_3)
\]

where \( q \geq 2 \).

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

\[
\forall Xs : \text{list of } X, \forall Y : Y. \ (\forall X : X. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r,\text{tupling}(Xs,Y) \Rightarrow S_r,\text{tupling}]
\]

\[
\forall (Xs = [\emptyset] \land Y = \emptyset)
\]

\[
\forall (Xs = [X_1] \land TXs \land TXs = [\emptyset \land \text{O_r}(X_1,Y_1) \land Y_1 = I_1 \land TY = \emptyset \land O_c(I_1,TY,Y))
\]

\[
\forall (Xs = [X_1] \land TXs \land TXs = [X_2,\ldots,X_q] \land \bigwedge_{i=1}^{q} \text{O_r}(X_i,Y_i) \land Y_1 = I_1 \land Y_2 = I_2 \land Y = I_3)
\]

where \( q \geq 2 \).

By folding using \( S_r,\text{tupling} \), and renaming:

\[
\forall Xs : \text{list of } X, \forall Y : Y. \ (\forall X : X. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r,\text{tupling}(Xs,Y) \Rightarrow S_r,\text{tupling}]
\]

\[
\forall (Xs = [\emptyset] \land Y = \emptyset)
\]

\[
\forall (Xs = [X] \land TXs \land TXs = [X_1] \land r,\text{tupling}(TXs,TY) \land \text{O_c}(HY,TY,Y))
\]

By taking the ‘decomposition’:

\[
\text{clause 1: } r,\text{tupling}(Xs,Y) \leftarrow
\]

\[
Xs = [\emptyset], Y = \emptyset
\]

\[
\text{clause 2: } r,\text{tupling}(Xs,Y) \leftarrow
\]

\[
Xs = [X] \land TXs \land r,\text{tupling}(TXs,TY) \land \text{O_c}(HY,TY,Y)
\]

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

\[
\text{clause 3: } r,\text{tupling}(Xs,Y) \leftarrow
\]

\[
Xs = [X] \land TXs \land \text{minimal}(X),
\]

\[
r,\text{tupling}(TXs,TY),
\]

\[
\text{solve}(X, HY), \text{compose}(HY,TY,Y)
\]

\[
\text{clause 4: } r,\text{tupling}(Xs,Y) \leftarrow
\]

\[
Xs = [X] \land TXs \land \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
\]

\[
r(TX_1, TY_1),\ldots,r(TX_t, TY_t),
\]

\[
I_0 = \emptyset,
\]

\[
\text{compose}(I_0, TY_1, I_1),\ldots,\text{compose}(I_{n-2}, TY_{p-1}, I_{p-1}),
\]

\[
\text{process}(HX, H HY), \text{compose}(I_{p-1}, H HY, I_p),
\]

\[
\text{compose}(I_p, TY_p, I_{p+1}),\ldots,\text{compose}(I_{n}, TY_1, I_{n+1}),
\]

\[
HY = I_{n+1}, r,\text{tupling}(TXs,TY), \text{compose}(HY,TY,Y)
\]

By introducing
\[
\{ \text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_t) \}\lor \\
((\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_{p-1}) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))) \lor \\
((\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1}) \land (\text{minimal}(TX_p) \land \ldots \land \text{minimal}(TX_t)) \lor \\
\{ \text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1}) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))\})
\]

in clause 4, using applicability condition (4).

\textit{clause 5: } \texttt{r.tupling}(Xs, Y) \leftarrow \\
Xs = [X][TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e. \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_I, TY_I, I_{I+1}), \\
HY = I_{I+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)

\textit{clause 6: } \texttt{r.tupling}(Xs, Y) \leftarrow \\
Xs = [X][TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\{\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)\}, \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e. \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_I, TY_I, I_{I+1}), \\
HY = I_{I+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)

\textit{clause 7: } \texttt{r.tupling}(Xs, Y) \leftarrow \\
Xs = [X][TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\}, \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e. \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_I, TY_I, I_{I+1}), \\
HY = I_{I+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)

\textit{clause 8: } \texttt{r.tupling}(Xs, Y) \leftarrow \\
Xs = [X][TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\}, \\
\{\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)\}, \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e. \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_I, TY_I, I_{I+1}), \\
HY = I_{I+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)

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):
clause 9: $$r\text{-tupling}(Xs, Y)$$ —

$$Xs = [X|TXs],$$
nonMinimal(X), decompose(X, HX, TX1, ..., TXi),
minimal(TX1), ..., minimal(TXi),
solve(TX1, TY1), ..., solve(TXi, TYi),
$$I_0 = e,$$
$$compose(I_0, TY1, I_1), ..., compose(I_{p-2}, TY_{p-1}, I_{p-1}).$$
process(HX, HHY), compose(I_{p-1}, HHY, I_{p}),
$$compose(I_p, TY_p, I_{p+1}), ..., compose(I_i, TY_i, I_{i+1}),$$
$$HY = I_{i+1}, r\text{-tupling}(TXs, TY), compose(HY, TY, Y)$$

By using applicability condition (3):

clause 10: $$r\text{-tupling}(Xs, Y)$$ —

$$Xs = [X|TXs],$$
nonMinimal(X), decompose(X, HX, TX1, ..., TXi),
minimal(TX1), ..., minimal(TXi),
solve(TX1, e), ..., solve(TXi, e),
$$I_0 = e,$$
$$compose(I_0, e, I_1), ..., compose(I_{p-2}, e, I_{p-1}).$$
process(HX, HHY), compose(I_{p-1}, HHY, I_{p}),
$$compose(I_p, e, I_{p+1}), ..., compose(I_i, e, I_{i+1}),$$
$$HY = I_{i+1}, r\text{-tupling}(TXs, TY), compose(HY, TY, Y)$$

By deleting one of the minimal(TX1), ..., minimal(TXi) atoms in clause 10:

clause 11: $$r\text{-tupling}(Xs, Y)$$ —

$$Xs = [X|TXs],$$
nonMinimal(X), decompose(X, HX, TX1, ..., TXi),
minimal(TX1), ..., minimal(TXi),
solve(TX1, e), ..., solve(TXi, e),
$$I_0 = e,$$
$$compose(I_0, e, I_1), ..., compose(I_{p-2}, e, I_{p-1}).$$
process(HX, HHY), compose(I_{p-1}, HHY, I_{p}),
$$compose(I_p, e, I_{p+1}), ..., compose(I_i, e, I_{i+1}),$$
$$HY = I_{i+1}, r\text{-tupling}(TXs, TY), compose(HY, TY, Y)$$

By using applicability condition (2):

clause 12: $$r\text{-tupling}(Xs, Y)$$ —

$$Xs = [X|TXs],$$
nonMinimal(X), decompose(X, HX, TX1, ..., TXi),
minimal(TX1), ..., minimal(TXi),
solve(TX1, e), ..., solve(TXi, e),
$$I_0 = e,$$
$$I_1 = I_0, ..., I_{p-1} = I_{p-2},$$
process(HX, HHY), compose(I_{p-1}, HHY, I_{p}),
$$I_{p+1} = I_p, ..., I_{i+1} = I_i,$$
$$HY = I_{i+1}, r\text{-tupling}(TXs, TY), compose(HY, TY, Y)$$

By simplification:

clause 13: $$r\text{-tupling}(Xs, Y)$$ —

$$Xs = [X|TXs],$$
nonMinimal(X), decompose(X, HX, TX1, ..., TXi),
minimal(TX1), ..., minimal(TXi),
$$r\text{-tupling}(TXs, TY),$$
process(HX, HHY), compose(HY, TY, Y)

By p–1 times unfolding clause 6 wrt $$r(TX1, TY1), ..., r(TX_{p-1}, TY_{p-1})$$ using DCLR, and simplifying using condition (4):
clause 14: \( r.tuling(X, Y) \) -
\[ X = [X|TX], \]
nonMinimal\( (X) \), decompose\( (X, HX, TX_{1}, \ldots, TX_{t}) \),
minimal\( (TX_{1}), \ldots, \) minimal\( (TX_{p-1}) \),
\( \text{nonMinimal}(TX_{p}), \ldots, \text{nonMinimal}(TX_{t}) \),
solve\( (TX_{1}, TY_{1}) \), \ldots, solve\( (TX_{p-1}, TY_{p-1}) \),
r\( (TX_{p}, TY_{p}), \ldots, r(TX_{t}, TY_{t}) \)
\( l_{0} = e \),
\( \text{compose}(l_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}) \),
process\( (HX, HHY) \), compose\( (l_{p-1}, HHY, I_{p}) \),
\( \text{compose}(l_{p}, TY_{p}, I_{p+1}), \ldots, \text{compose}(l_{t}, TY_{t}, I_{t+1}) \),
\( HY = I_{t+1}, r.tuling(TXs, TY), \text{compose}(HY, TY, Y) \)

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

clause 15: \( r.tuling(X, Y) \) -
\[ X = [X|TX], \]
nonMinimal\( (X) \), decompose\( (X, HX, TX_{1}, \ldots, TX_{t}) \),
minimal\( (TX_{1}), \ldots, \) minimal\( (TX_{p-1}) \),
\( \text{nonMinimal}(TX_{p}), \ldots, \text{nonMinimal}(TX_{t}) \),
solve\( (TX_{1}, TY_{1}) \), \ldots, solve\( (TX_{p-1}, TY_{p-1}) \),
r\( (TX_{p}, TY_{p}), \ldots, r(TX_{t}, TY_{t}) \)
\( l_{0} = e \),
\( \text{compose}(l_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}) \),
process\( (HX, HHY) \), compose\( (l_{p-1}, HHY, I_{p}) \),
\( \text{compose}(l_{p}, TY_{p}, I_{p+1}), \ldots, \text{compose}(l_{t}, TY_{t}, I_{t+1}) \),
\( HY = I_{t+1}, r.tuling(TXs, TY), \text{compose}(HY, TY, Y) \)

By rewriting clause 15 using applicability condition (1):

clause 16: \( r.tuling(X, Y) \) -
\[ X = [X|TX], \]
nonMinimal\( (X) \), decompose\( (X, HX, TX_{1}, \ldots, TX_{t}) \),
minimal\( (TX_{1}), \ldots, \) minimal\( (TX_{p-1}) \),
\( \text{nonMinimal}(TX_{p}), \ldots, \text{nonMinimal}(TX_{t}) \),
solve\( (TX_{1}, TY_{1}) \), \ldots, solve\( (TX_{p-1}, TY_{p-1}) \),
r\( (TX_{p}, TY_{p}), \ldots, r(TX_{t}, TY_{t}) \)
\( l_{0} = e \),
\( \text{compose}(l_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}) \),
process\( (HX, HHY) \), compose\( (l_{p-1}, HHY, I_{p}) \), \( HY = I_{p} \),
\( \text{compose}(TY_{p}, TY_{p+1}, I_{p+1}) \),
\( \text{compose}(l_{p+1}, TY_{p+2}, I_{p+2}), \ldots, \text{compose}(I_{t-1}, TY_{t}, I_{t}) \),
\( r.tuling(TXs, TY), \text{compose}(I_{t}, TTY, TY), \text{compose}(HY, TY, Y) \)

By \( t - p \) times folding clause 16 using clauses 1 and 2:

clause 17: \( r.tuling(X, Y) \) -
\[ X = [X|TX], \]
nonMinimal\( (X) \), decompose\( (X, HX, TX_{1}, \ldots, TX_{t}) \),
minimal\( (TX_{1}), \ldots, \) minimal\( (TX_{p-1}) \),
\( \text{nonMinimal}(TX_{p}), \ldots, \text{nonMinimal}(TX_{t}) \),
solve\( (TX_{1}, TY_{1}) \), \ldots, solve\( (TX_{p-1}, TY_{p-1}) \),
r\( (TX_{p}, TY_{p}), \ldots, r(TX_{t}, TY_{t}) \)
\( l_{0} = e \),
\( \text{compose}(l_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}) \),
process\( (HX, HHY) \), compose\( (l_{p-1}, HHY, I_{p}) \), \( HY = I_{p} \),
\( \text{compose}(HY, TY, Y) \)

By using applicability condition (3):
clause 18: \( \mathit{r.tupling}(Xs, Y) \) —

\[
Xs = [X | TXs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]
\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e),
\]
\[
\mathit{r.tupling}([TX_p, \ldots, TX_t] | TXs], TY),
\]
\[
I_0 = e,
\]
\[
\text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}).
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), H_Y = I_p.
\]
\[
\text{compose}(HY, TY, Y)
\]

By using applicability condition (2):

clause 19: \( \mathit{r.tupling}(Xs, Y) \) —

\[
Xs = [X | TXs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]
\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e),
\]
\[
\mathit{r.tupling}([TX_p, \ldots, TX_t] | TXs], TY),
\]
\[
I_0 = e,
\]
\[
I_1 = I_0, \ldots, I_{p-1} = I_{p-2}.
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), H_Y = I_p.
\]
\[
\text{compose}(HY, TY, Y)
\]

By simplification:

clause 20: \( \mathit{r.tupling}(Xs, Y) \) —

\[
Xs = [X | TXs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]
\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\mathit{r.tupling}([TX_p, \ldots, TX_t] | TXs], TY),
\]
\[
\text{process}(HX, HY), \text{compose}(HY, TY, Y)
\]

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

clause 21: \( \mathit{r.tupling}(Xs, Y) \) —

\[
Xs = [X | TXs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
\]
\[
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}).
\]
\[
\text{r}(TX_1, TY_1), \ldots, \text{r}(TX_t, TY_t),
\]
\[
I_0 = e,
\]
\[
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),
\]
\[
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}),
\]
\[
HY = I_{t+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\]

By using applicability condition (3):
clause 22:  \( r.tupling(X, Y) \) —  
\( X_s = [X|TXs] \), 
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \).
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \).
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \).
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_{p-1}) \).
\( r(U_1, e), \ldots, r(U_{p-1}, e) \).
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \).
\( I_0 = e \),
\( \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}) \).
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \).
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}) \).
\( HY = I_{t+1}, r.tupling(TXS, TY) \), \( \text{compose}(HY, TY, Y) \)  
By using applicability condition (2):  

clause 23:  \( r.tupling(X, Y) \) —  
\( X_s = [X|TXs] \), 
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \).
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \).
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \).
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_{p-1}) \).
\( r(U_1, e), \ldots, r(U_{p-1}, e) \).
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \).
\( I_0 = e \),
\( \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}) \).
\( \text{compose}(I_{p-1}, e, K_1), \text{compose}(K_1, e, K_2), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}) \).
\( \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, I_p) \).
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}) \).
\( HY = I_{t+1}, r.tupling(TXS, TY) \), \( \text{compose}(HY, TY, Y) \)  
By using applicability conditions (1) and (2):  

clause 24:  \( r.tupling(X, Y) \) —  
\( X_s = [X|TXs] \), 
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \).
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \).
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \).
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_{p-1}) \).
\( r(U_1, e), \ldots, r(U_{p-1}, e) \).
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \).
\( I_0 = e \),
\( \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}) \).
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \).
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}) \).
\( HY = I_{t+1}, r.tupling(TXS, TY) \), \( \text{compose}(HY, TY, TI) \).
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \).
\( \text{compose}(K_{p-2}, TI, Y) \)  
By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_{p-1} \) in place of some occurrences of \( e \):
clause 25: 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).
r(U1, YU1), ..., r(Up-1, YUp-1),
r(TX1, TY1), ..., r(TXt, TYt).
l0 = e,
compose(I0, YU1, I1), ..., compose(Ip-2, YUp-1, Ip-1),
process(HX, HHY), compose(Ip-1, HHY, Ip),
compose(Ip, TYp, Ip+1), ..., compose(It, TYt, It+1).
HY = It+1, r.tupling(TXs, TY), compose(HY, TY, TI).
compose(TY1, TY2, K1), compose(K1, TY3, K2), ..., compose(Kp-3, TYp-1, Kp-2).
compose(Kp-2, TI, Y)

By introducing nonMinimal(N) and decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt), since

\[ \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, ..., U_{p-1}, TX_p, ..., TX_t) \]

always holds (because N is existentially quantified)

clause 26: 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).
r(U1, YU1), ..., r(Up-1, YUp-1),
nonMinimal(N), decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt)
r(TX1, TY1), ..., r(TXt, TYt).
l0 = e,
compose(I0, YU1, I1), ..., compose(Ip-2, YUp-1, Ip-1),
process(HX, HHY), compose(Ip-1, HHY, Ip),
compose(Ip, TYp, Ip+1), ..., compose(It, TYt, It+1).
HY = It+1, r.tupling(TXs, TY), compose(HY, TY, TI).
compose(TY1, TY2, K1), compose(K1, TY3, K2), ..., compose(Kp-3, TYp-1, Kp-2).
compose(Kp-2, TI, Y)

By duplicating goal decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt):

clause 27: 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).
r(U1, YU1), ..., r(Up-1, YUp-1),
nonMinimal(N), decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt)
decompose(N, HX, U1, ..., Up-1, TXp, ..., TXt),
r(TX1, TY1), ..., r(TXt, TYt).
l0 = e,
compose(I0, YU1, I1), ..., compose(Ip-2, YUp-1, Ip-1),
process(HX, HHY), compose(Ip-1, HHY, Ip),
compose(Ip, TYp, Ip+1), ..., compose(It, TYt, It+1).
HY = It+1, r.tupling(TXs, TY), compose(HY, TY, TI).
compose(TY1, TY2, K1), compose(K1, TY3, K2), ..., compose(Kp-3, TYp-1, Kp-2).
compose(Kp-2, TI, Y)

By folding clause 27 using DCLR.
clause 28: \texttt{r.tupling}(X, Y) =
\begin{align*}
X &= [X][TXs], \\
\text{nonMinimal}(X), &\text{decompose}(X, HX, TX_{1}, \ldots, TX_{t}), \\
\text{(nonMinimal}(TX_{1}); &\ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_{p}), &\ldots, \text{minimal}(TX_{t}). \\
\text{minimal}(U_{1}), &\ldots, \text{minimal}(U_{p-1}). \\
\text{decompose}(N, HX, U_{1}, \ldots, U_{p-1}, TX_{p}, \ldots, TX_{t}), \\
\text{r}(TX_{1}, TY_{1}), &\ldots, \text{r}(TX_{p-1}, TY_{p-1}), \text{r}(N, HY), \\
\text{r.tupling}(TXs, TY), &\text{compose}(HY, TY, TI), \\
\text{compose}(TY_{1}, TY_{2}, K_{1}), &\text{compose}(K_{1}, TY_{3}, K_{2}), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(K_{p-2}, TI, Y). \\
\end{align*}

By folding clause 28 using clauses 1 and 2:

clause 29: \texttt{r.tupling}(X, Y) =
\begin{align*}
X &= [X][TXs], \\
\text{nonMinimal}(X), &\text{decompose}(X, HX, TX_{1}, \ldots, TX_{t}), \\
\text{(nonMinimal}(TX_{1}); &\ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_{p}), &\ldots, \text{minimal}(TX_{t}). \\
\text{minimal}(U_{1}), &\ldots, \text{minimal}(U_{p-1}). \\
\text{decompose}(N, HX, U_{1}, \ldots, U_{p-1}, TX_{p}, \ldots, TX_{t}), \\
\text{r}(TX_{1}, TY_{1}), &\ldots, \text{r}(TX_{p-1}, TY_{p-1}), \\
\text{r.tupling}(N[TXs], TI), \\
\text{compose}(TY_{1}, TY_{2}, K_{1}), &\text{compose}(K_{1}, TY_{3}, K_{2}), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(K_{p-2}, TI, Y). \\
\end{align*}

By \( p - 1 \) times folding clause 29 using clauses 1 and 2:

clause 30: \texttt{r.tupling}(X, Y) =
\begin{align*}
X &= [X][TXs], \\
\text{nonMinimal}(X), &\text{decompose}(X, HX, TX_{1}, \ldots, TX_{t}), \\
\text{(nonMinimal}(TX_{1}); &\ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_{p}), &\ldots, \text{minimal}(TX_{t}). \\
\text{minimal}(U_{1}), &\ldots, \text{minimal}(U_{p-1}). \\
\text{decompose}(N, HX, U_{1}, \ldots, U_{p-1}, TX_{p}, \ldots, TX_{t}), \\
\text{r.tupling}(TX_{1}, \ldots, TX_{p-1}, N[TXs], Y). \\
\end{align*}

By introducing atoms \text{minimal}(U_{1}), \ldots, \text{minimal}(U_{t}) (with new, i.e. existentially quantified, variables \( U_{1}, \ldots, U_{t} \)) in clause 8:

clause 31: \texttt{r.tupling}(X, Y) =
\begin{align*}
X &= [X][TXs], \\
\text{nonMinimal}(X), &\text{decompose}(X, HX, TX_{1}, \ldots, TX_{t}), \\
\text{(nonMinimal}(TX_{1}); &\ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{(nonMinimal}(TX_{p}); &\ldots; \text{nonMinimal}(TX_{t})), \\
\text{minimal}(U_{1}), &\ldots, \text{minimal}(U_{t}). \\
\text{r}(TX_{1}, TY_{1}), &\ldots, \text{r}(TX_{t}, TY_{t}), \\
l_{0} &= e, \\
\text{compose}(I_{0}, TY_{1}, I_{1}), &\ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{process}(HX, HHY), &\text{compose}(I_{p-1}, HHY, I_{p}), \\
\text{compose}(I_{p}, TY_{p}, I_{p+1}), &\ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}), \\
HY &= I_{t+1}, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, TI). \\
\end{align*}

By using applicability condition (3):
\text{clause 32:} \ r_{\text{tupling}}(Xs, Y) - \\
Xs = [X|TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
(non\text{Minimal}(TX_p); \ldots; non\text{Minimal}(TX_i)), \\
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), \\
l_0 = e, \\
\text{compose}(l_0, TY_1, I_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}). \\
HY = I_{t+1}, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, Y). \\

By using applicability condition (2):

\text{clause 33:} \ r_{\text{tupling}}(Xs, Y) - \\
Xs = [X|TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
(non\text{Minimal}(TX_p); \ldots; non\text{Minimal}(TX_i)), \\
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), \\
l_0 = e, \\
\text{compose}(l_0, TY_1, I_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{compose}(I_{p-1}, e, K_1), \text{compose}(K_1, e, K_2), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}), \\
\text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, K_p), \\
\text{compose}(K_p, e, K_{p+1}), \ldots, \text{compose}(K_t, e, K_{t+1}), \text{compose}(K_{t+1}, e, I_p). \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}). \\
HY = I_{t+1}, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, Y). \\

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

\text{clause 34:} \ r_{\text{tupling}}(Xs, Y) - \\
Xs = [X|TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
(non\text{Minimal}(TX_p); \ldots; non\text{Minimal}(TX_i)), \\
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), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \\
l_0 = e, \\
\text{compose}(l_0, e, I_1), \ldots, \text{compose}(l_{p-2}, e, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\text{compose}(I_p, e, I_{p+1}), \ldots, \text{compose}(I_t, e, I_{t+1}). \\
\text{HY} = I_{t+1}, \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \text{compose}(K_{t-1}, HY, I_t), \text{compose}(HY, TY, Y). \\
r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, Y). \\

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

14
clause 35: \( r \cdot \text{tupling}(Xs, Y) \rightarrow \\
Xs = [X | Xs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
r(U_1, YU_1, \ldots, r(U_t, YU_t). \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
I_0 = e, \\
\text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \\
\text{compose}(I_p, YU_p, I_{p+1}), \ldots, \text{compose}(I_t, YU_t, I_{t+1}). \\
NHY = I_{t+1}. \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(K_{t-2}, NHY, TI), \text{compose}(TI, K_{t-1}, HY). \\
r \cdot \text{tupling}(TXs, TY), \text{compose}(HY, TY, Y). \\
\)

By introducing \text{nonMinimal}(N) and \text{decompose}(N, HX, U_1, \ldots, U_t), since
\[
\exists N : X \cdot \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_t)
\]
always holds (because \( N \) is existentially quantified)

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

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

By folding clause 37 using DCLR:

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

By using applicability condition (1):

clause 39: \( r.tupling(Xs, Y) = \\
Xs = [X | TXs]. \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \)
\( \text{minimal}(U_1); \ldots, \text{minimal}(U_t). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( r(N, NHY). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \)
\( \text{compose}(K_{t-2}, TII, Y), \text{compose}(NHY, TII, TII). \)
\( r.tupling(TXs, TY), \text{compose}(K_{t-1}, TII, TII) \)

By \( t - p + 1 \) times folding clause 39 using clauses 1 and 2:
clause 40 : \( r\text{-tupling}(X_s, Y) \rightarrow X_s = [X][T]X_s \),
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \),
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})) \),
\( (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)) \),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \)
\( \text{decompose}(N, HX, U_1, \ldots, U_t) \),
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY) \),
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \),
\( \text{compose}(K_{p-2}, TL_2, Y), \text{compose}(NHY, TL_1, TL_2) \),
\( r\text{-tupling}([TX_p, \ldots, TX_t][T]X_s, TL_1) \)

By folding clause 40 using clauses 1 and 2:

\begin{align*}
\text{clause 41 : } & r\text{-tupling}(X_s, Y) \\
& X_s = [X][T]X_s \\
& \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \\
& (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})) \\
& (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)) \\
& \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \\
& \text{decompose}(N, HX, U_1, \ldots, U_t) \\
& r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}) \\
& \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \\
& \text{compose}(K_{p-2}, TL_2, Y) \\
& r\text{-tupling}([N, TX_p, \ldots, TX_t][T]X_s, TL_1) \\
\end{align*}

By \( p - 1 \) times folding clause 41 using clauses 1 and 2:

\begin{align*}
\text{clause 42 : } & r\text{-tupling}(X_s, Y) \\
& X_s = [X][T]X_s \\
& \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \\
& (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})) \\
& (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)) \\
& \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \\
& \text{decompose}(N, HX, U_1, \ldots, U_t) \\
& r\text{-tupling}([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t][T]X_s, TL_1) \\
\end{align*}

Clauses 1, 3, 13, 20, 30 and 42 are the clauses of \( P_{r\text{-tupling}} \). Therefore \( P_{r\text{-tupling}} \) is steadfast wrt \( S_{r\text{-tupling}} \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r\text{-tupling}}\} \), 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:

\[ r(X, Y) \leftarrow r\text{-tupling}([X], Y) \]

By taking the ‘completion’:

\[ \forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r\text{-tupling}([X], Y)] \]

By unfolding the ‘completion’ above wrt \( r\text{-tupling}([X], Y) \) using \( S_{r\text{-tupling}} \):

\[ \forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : Y. \ O_r(X, Y_1) \& I_1 = Y_1 \& Y = I_1] \]

By simplification:

\[ \forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)] \]

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r\text{-tupling}}\} \).

Therefore, \( TG \) is also steadfast wrt \( S_r \) in \( S \). \[ \square \]

**Theorem 2** The generalization schema \( TG_2 \), which is given below, is correct.
$TG_2 : \{ DCRL, TG, A_{t_2}, O_{t_2 12}, O_{t_2 221} \}$ where

1. $compose$ is associative
2. $compose$ has $e$ as the left and right identity element, where $e$ appears in $DCRL$
3. $\forall X : \mathcal{X}. \mathcal{I}_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e)$
4. $\forall X : \mathcal{X}. \mathcal{I}_r(X) \Rightarrow [\neg \text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)]$

$O_{t_2 121} :$ partial evaluation of the conjunction

- $\text{process}(H_X, H_Y), \text{compose}(H_Y, T_Y, Y)$ results in the introduction of a non-recursive relation

$O_{t_2 221} :$ partial evaluation of the conjunction

- $\text{process}(H_X, H_Y), \text{compose}(H_Y, I_p, I_{p-1})$ 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**

\[ r(X, Y) \leftarrow \]
\[ \text{minimal}(X), \text{solve}(X, Y) \]
\[ r(X, Y) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H_X, T_X_1, \ldots, T_X_t), \]
\[ r(T_X_1, T_Y_1), \ldots, r(T_X_t, T_Y_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(T_Y_t, I_{t+1}, I_t), \ldots, \text{compose}(T_Y_p, I_{p+1}, I_p), \]
\[ \text{process}(H_X, H_Y), \text{compose}(H_Y, I_p, I_{p-1}), \]
\[ \text{compose}(T_Y_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(T_Y_1, I_1, I_0), \]
\[ Y = I_0 \]

and the specification $S_r$ of relation $r$ is:

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

and the specification $S_{r\text{-tupling}}$ of relation $r\text{-tupling}$ is:

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

**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 $A_{t_2}$. By Definition 5, the templates $DCRL$ and $TG$ are equivalent wrt $S_r$ under the applicability conditions $A_{t_2}$ iff $DCRL$ is equivalent to $TG$ wrt the specification $S_r$ provided that the conditions in $A_{t_2}$ 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 $S = \{ S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}$ are the specifications of $\text{minimal, nonMinimal, solve, decompose, process, compose}$, which are all the undefined relation names appearing in $DCRL$.

(b) $TG$ is also steadfast wrt $S_r$ in $S$.

Note that the sets $\{ S_1, \ldots, S_m \}$ and $\{ S'_1, \ldots, S'_n \}$ in Definition 4 are equal to $S$ 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 $S$, so condition (a) always holds.
To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $T \triangleright G$ is steadfast wrt $S_r$ in $S$ if $P_{r,\text{tupling}}$ is steadfast wrt $S_{r,\text{tupling}}$ in $S$, where $P_{r,\text{tupling}}$ is the procedure for $r,\text{tupling}$, and $P_r$ is steadfast wrt $S_r$ in $\{S_{r,\text{tupling}}\}$, where $P_r$ is the procedure for $r$.

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

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

$$\forall X, Y : \forall Y : \exists Y. (X, X \in X) \Rightarrow \exists X = I_r(X) \Rightarrow [r,\text{tupling}(X, Y) \Rightarrow \langle X = [] \land Y = \epsilon \rangle$$(1)

$$\forall X = [X_1, \ldots, X_q] \land \forall Y = [Y_1, \ldots, Y_q] \land \forall I_1 \land Y = I_1$$

$$\forall X = [X_1, X_2, \ldots, X_q] \land \forall I_1 \land Y = I_1$$

where $q \geq 2$.

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

$$\forall X, Y : \forall Y : \exists Y. (X, X \in X \Rightarrow \exists X = I_r(X) \Rightarrow [r,\text{tupling}(X, Y) \Rightarrow \langle X = [] \land Y = \epsilon \rangle$$(1)

$$\forall X = [X_1, \ldots, X_q] \land \forall Y = [Y_1, \ldots, Y_q] \land \forall I_1 \land Y = I_1$$

where $q > 2$.

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

$$\forall X, Y : \forall Y : \exists Y. (X, X \in X \Rightarrow \exists X = I_r(X) \Rightarrow [r,\text{tupling}(X, Y) \Rightarrow \langle X = [X[TX], O_r(X, HY) \land \forall I_1 \land Y = I_1$$

By taking the 'decompection':

**clause 1:** $r,\text{tupling}(X, Y) \leftarrow$ $X = [X[TX], O_r(X, HY) \land \forall I_1 \land Y = I_1$

**clause 2:** $r,\text{tupling}(X, Y) \leftarrow$ $X = [X[TX], r(X, HY)$

$r,\text{tupling}(TX, TY), \text{compose}(HY, TY, Y)$

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

**clause 3:** $r,\text{tupling}(X, Y) \leftarrow$ $X = [X[TX], \text{minimal}(X),$ $\text{r,\text{tupling}(TX, TY), solve(X, HY), \text{compose}(HY, TY, Y)$

**clause 4:** $r,\text{tupling}(X, Y) \leftarrow$ $X = [X[TX], \text{nonMinimal}(X), \text{decompose}(X, HY, TX_1, \ldots, TX_t),$ $r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$ $I_1 = \epsilon, \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),$ $\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}),$ $\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0),$ $HY = I_0, r,\text{tupling}(TX, TY), \text{compose}(HY, HY, Y)$

By introducing

$$\forall [\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t)] \land$$

$$[(\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))] \land$$

$$[(\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{minimal}(TX_p) \land \ldots \land \text{minimal}(TX_t))] \land$$

$$[(\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))]$$

in clause 4, using applicability condition (4).
clause 5: \[ \text{r.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X][TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_t+1 = e. \]
\[ \text{compose}(TY_1, I_t+1, I_t), \ldots, \text{compose}(TY_p, I_p+1, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_p-1). \]
\[ \text{compose}(TY_p-1, I_p-1, I_p-2), \ldots, \text{compose}(TY_1, I_1, I_0). \]
\[ HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 6: \[ \text{r.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X][TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_p-1), \]
\[ \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_t+1 = e. \]
\[ \text{compose}(TY_1, I_t+1, I_t), \ldots, \text{compose}(TY_p, I_p+1, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_p-1). \]
\[ \text{compose}(TY_p-1, I_p-1, I_p-2), \ldots, \text{compose}(TY_1, I_1, I_0). \]
\[ HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 7: \[ \text{r.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X][TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, TX_2), \]
\[ \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_p-1), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_t+1 = e. \]
\[ \text{compose}(TY_1, I_t+1, I_t), \ldots, \text{compose}(TY_p, I_p+1, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_p-1). \]
\[ \text{compose}(TY_p-1, I_p-1, I_p-2), \ldots, \text{compose}(TY_1, I_1, I_0). \]
\[ HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 8: \[ \text{r.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X][TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, TX_2), \]
\[ \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_p-1), \]
\[ \text{minimal}(TX_p), \ldots, \text{nonMinimal}(TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_t+1 = e. \]
\[ \text{compose}(TY_1, I_t+1, I_t), \ldots, \text{compose}(TY_p, I_p+1, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_p-1). \]
\[ \text{compose}(TY_p-1, I_p-1, I_p-2), \ldots, \text{compose}(TY_1, I_1, I_0). \]
\[ HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y) \]

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):

clause 9: \[ \text{r.tupling}(Xs, Y) \leftarrow \]
\[ Xs = [X][TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_t, TY_t), \]
\[ I_t+1 = e. \]
\[ \text{compose}(TY_1, I_t+1, I_t), \ldots, \text{compose}(TY_p, I_p+1, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_p-1). \]
\[ \text{compose}(TY_p-1, I_p-1, I_p-2), \ldots, \text{compose}(TY_1, I_1, I_0). \]
\[ HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y) \]
By using applicability condition (3):

\textit{clause 10: } \texttt{r.tupling(Xs, Y) —}

\hspace{1em} \texttt{Xs = [X|TXs],}

\hspace{1em} \texttt{nonMinimal(X), decompose(X, HX, TX1, ..., TXl).}

\hspace{1em} \texttt{minimal(TX1), ..., minimal(TXl).}

\hspace{1em} \texttt{solve(TX1, e), ..., solve(TXl, e).}

\hspace{1em} \texttt{l_{i+1} = e,}

\hspace{1em} \texttt{compose(e, l_{i+1}, I_i), ..., compose(e, l_{p+1}, I_p).}

\hspace{1em} \texttt{process(HX, HHY), compose(HHY, I_p, I_{p-1}),}

\hspace{1em} \texttt{compose(e, I_{p-1}, I_{p-2}), ..., compose(e, I_1, I_0).}

\hspace{1em} \texttt{HY = I_0, r.tupling(TXS, TY), compose(HY, TY, Y).}

By deleting one of the \texttt{minimal(TX1), ..., minimal(TXl)} atoms in clause 10:

\textit{clause 11: } \texttt{r.tupling(Xs, Y) —}

\hspace{1em} \texttt{Xs = [X|TXs],}

\hspace{1em} \texttt{nonMinimal(X), decompose(X, HX, TX1, ..., TXl).}

\hspace{1em} \texttt{minimal(TX1), ..., minimal(TXl).}

\hspace{1em} \texttt{solve(TX1, e), ..., solve(TXl, e).}

\hspace{1em} \texttt{l_{i+1} = e,}

\hspace{1em} \texttt{compose(e, l_{i+1}, I_i), ..., compose(e, l_{p+1}, I_p).}

\hspace{1em} \texttt{process(HX, HHY), compose(HHY, I_p, I_{p-1}),}

\hspace{1em} \texttt{compose(e, I_{p-1}, I_{p-2}), ..., compose(e, I_1, I_0).}

\hspace{1em} \texttt{HY = I_0, r.tupling(TXS, TY), compose(HY, TY, Y).}

By using applicability condition (2):

\textit{clause 12: } \texttt{r.tupling(Xs, Y) —}

\hspace{1em} \texttt{Xs = [X|TXs],}

\hspace{1em} \texttt{nonMinimal(X), decompose(X, HX, TX1, ..., TXl).}

\hspace{1em} \texttt{minimal(TX1), ..., minimal(TXl).}

\hspace{1em} \texttt{solve(TX1, e), ..., solve(TXl, e).}

\hspace{1em} \texttt{l_{i+1} = e,}

\hspace{1em} \texttt{l_i = l_{i+1}, ..., l_p = l_{p+1},}

\hspace{1em} \texttt{process(HX, HHY), compose(HHY, I_p, I_{p-1}),}

\hspace{1em} \texttt{l_{p-1} = l_{p-2}, ..., l_0 = l_1.}

\hspace{1em} \texttt{HY = I_0, r.tupling(TXS, TY), compose(HY, TY, Y).}

By simplification:

\textit{clause 13: } \texttt{r.tupling(Xs, Y) —}

\hspace{1em} \texttt{Xs = [X|TXs],}

\hspace{1em} \texttt{nonMinimal(X), decompose(X, HX, TX1, ..., TXl).}

\hspace{1em} \texttt{minimal(TX1), ..., minimal(TXl).}

\hspace{1em} \texttt{r.tupling(TXS, TY).}

\hspace{1em} \texttt{process(HX, HHY), compose(HY, TY, Y).}

By \(p-1\) times unfolding clause 6 wrt \(r(TX_1, TY_1), ..., r(TX_{p-1}, TY_{p-1})\) using DCRIL, and simplifying using condition (4):
clause 14: \[ r.tupling(X_s, Y) = \]
\[ X_s = [X|TX_s], \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ minimal(TX_1), \ldots, minimal(TX_{p-1}). \]
\[ (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \]
\[ minimal(TX_1), \ldots, minimal(TX_{p-1}). \]
\[ solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HY), compose(HY, I_p, I_{p-1}), \]
\[ compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \]
\[ HY = I_0, r.tupling(TX_s, TY), compose(HY, TY, Y). \]

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

clause 15: \[ r.tupling(X_s, Y) = \]
\[ X_s = [X|TX_s], \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ minimal(TX_1), \ldots, minimal(TX_{p-1}). \]
\[ (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \]
\[ solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HY), compose(HY, I_p, I_{p-1}), \]
\[ compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0), \]
\[ HY = I_0, r.tupling(TX_s, TY), compose(HY, TY, Y). \]

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

clause 16: \[ r.tupling(X_s, Y) = \]
\[ X_s = [X|TX_s], \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ minimal(TX_1), \ldots, minimal(TX_{p-1}). \]
\[ (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \]
\[ solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_n = e, \]
\[ compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \]
\[ compose(TY_p, TY_{p+1}, I_{p+1}), \]
\[ compose(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, compose(I_{t-1}, TY_t, I_t), \]
\[ r.tupling(TX_s, TY), compose(I_t, TTY, TY), \]
\[ compose(HY, TY, Y). \]

By \( t - p \) times folding clause 16 using clauses 1 and 2:

clause 17: \[ r.tupling(X_s, Y) = \]
\[ X_s = [X|TX_s], \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ minimal(TX_1), \ldots, minimal(TX_{p-1}). \]
\[ (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \]
\[ solve(TX_1, TY_1), \ldots, solve(TX_{p-1}, TY_{p-1}), \]
\[ r.tupling([TX_p, \ldots, TX_t]TX_s, TY), \]
\[ I_n = e, \]
\[ compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ process(HX, HHY), compose(I_{p-1}, HHY, I_p), HY = I_p, \]
\[ compose(HY, TY, Y). \]

By using applicability condition (3):
clause 18:  \( r_tupling(X_s, Y) \) — 
\( X_s = [X[TXs]] \),
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\((TX_1, e), \ldots, solve(TX_{p-1}, e)\),
\( r_tupling([TX_p, \ldots, TX_t[TXs]], TY)\),
\( I_0 = e\),
\( compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1})\),
\( process(HX, HHY), compose(I_{p-1}, HHY, I_p)\), \( HY = I_p\).
\( compose(HY, TY, Y)\)

By using applicability condition (2):

clause 19:  \( r_tupling(X_s, Y) \) — 
\( X_s = [X[TXs]] \),
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\((TX_1, e), \ldots, solve(TX_{p-1}, e)\),
\( r_tupling([TX_p, \ldots, TX_t[TXs]], TY)\),
\( I_0 = e\),
\( I_1 = I_0, \ldots, I_{p-1} = I_{p-2}\).
\( process(HX, HHY), compose(I_{p-1}, HHY, I_p)\), \( HY = I_p\).
\( compose(HY, TY, Y)\)

By simplification:

clause 20:  \( r_tupling(X_s, Y) \) — 
\( X_s = [X[TXs]] \),
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\( r_tupling([TX_p, \ldots, TX_t[TXs]], TY)\),
\( process(HX, HHY), compose(HHY, TY, Y)\)

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:

clause 21:  \( r_tupling(X_s, Y) \) — 
\( X_s = [X[TXs]] \),
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\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t)\),
\( I_{t+1} = e\),
\( compose(TY_t, I_{t+1}, I_r), \ldots, compose(TY_p, I_{p+1}, I_p)\)
\( process(HX, HHY), compose(HHY, I_p, I_{p-1})\),
\( compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0)\),
\( HY = I_0, r_tupling(TXs, TY), compose(HY, TY, Y)\)

By using applicability condition (3):
clause 22: \[ r.tupling(Xs, Y) \rightarrow \]
\[ Xs = [X|TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0). \]
\[ HY = I_0, r.tupling(TXs, TY), compose(HY, TY, Y) \]

By using applicability condition (2):

clause 23: \[ r.tupling(Xs, Y) \rightarrow \]
\[ Xs = [X|TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(e, I_{p-1}, K_1), \ldots, compose(e, K_{p-2}, K_{p-1}), \]
\[ compose(TY_p, K_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0). \]
\[ HY = I_0, r.tupling(TXs, TY), compose(HY, TY, Y) \]

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

clause 24: \[ r.tupling(Xs, Y) \rightarrow \]
\[ Xs = [X|TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(e, I_{p-1}, I_{p-2}), \ldots, compose(e, I_1, I_0). \]
\[ HY = I_0, r.tupling(TXs, TY), compose(HY, TY, TT), \]
\[ compose(TY_1, TY_2, K_1), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(K_{p-2}, TT, Y) \]

By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_{p-1} \) in place of some occurrences of \( e \):
clause 25: \[ r.tupleing(X,Y) \] \[ X_s = [X][TX_s]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_H, I_{p-1}), \]
\[ \text{compose}(YU_{p-1}, I_{p-1}, I_p), \ldots, \text{compose}(YU_1, I_1, I_0), \]
\[ HY = I_0, r.tupleing(TX_s, TY), \text{compose}(HY, TY, TI). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(K_{p-2}, TI, Y) \]

By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \), since
\[ \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \]
always holds (because \( N \) is existentially quantified)

clause 26: \[ r.tupleing(X,Y) \] \[ X_s = [X][TX_s]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_H, I_{p-1}), \]
\[ \text{compose}(YU_{p-1}, I_{p-1}, I_p), \ldots, \text{compose}(YU_1, I_1, I_0), \]
\[ HY = I_0, r.tupleing(TX_s, TY), \text{compose}(HY, TY, TI). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(K_{p-2}, TI, Y) \]

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

clause 27: \[ r.tupleing(X,Y) \] \[ X_s = [X][TX_s]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_H, I_{p-1}), \]
\[ \text{compose}(YU_{p-1}, I_{p-1}, I_p), \ldots, \text{compose}(YU_1, I_1, I_0), \]
\[ HY = I_0, r.tupleing(TX_s, TY), \text{compose}(HY, TY, TI). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(K_{p-2}, TI, Y) \]

By folding clause 27 using DCRIL
clause 28: \(\text{r.tupling}(Xs, Y)\) —
\[Xs = [X | TXs],\]
nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt),
(nonMinimal(TX1); \ldots; nonMinimal(TXp−1)).
minimal(TXp), \ldots, minimal(TXt),
minimal(U1), \ldots, minimal(Up−1).
\(r(TX1, TY1), \ldots, r(TXp−1, TYp−1), r(N, HY),\)
\(r.tupling(TXs, TY), \text{compose}(HY, TY, TI),\)
\(\text{compose}(TY1, TY2, K1), \text{compose}(K1, TY3, K2), \ldots, \text{compose}(Kp−3, TYp−1, Kp−2),\)
\(\text{compose}(Kp−2, TI, Y)\)

By folding clause 28 using clauses 1 and 2:

clause 29: \(\text{r.tupling}(Xs, Y)\) —
\[Xs = [X | TXs],\]
nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt),
(nonMinimal(TX1); \ldots; nonMinimal(TXp−1)).
minimal(TXp), \ldots, minimal(TXt),
minimal(U1), \ldots, minimal(Up−1).
\(r(TX1, TY1), \ldots, r(TXp−1, TYp−1),\)
\(r.tupling([N | TXs], TI),\)
\(\text{compose}(TY1, TY2, K1), \text{compose}(K1, TY3, K2), \ldots, \text{compose}(Kp−3, TYp−1, Kp−2),\)
\(\text{compose}(Kp−2, TI, Y)\)

By \(p−1\) times folding clause 29 using clauses 1 and 2:

clause 30: \(\text{r.tupling}(Xs, Y)\) —
\[Xs = [X | TXs],\]
nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt),
(nonMinimal(TX1); \ldots; nonMinimal(TXp−1)).
minimal(TXp), \ldots, minimal(TXt),
minimal(U1), \ldots, minimal(Up−1).
\(r(TX1, TY1), \ldots, r(TXp−1, TYp−1),\)
\(r.tupling([TX1, \ldots, TXp−1, N | TXs], Y)\)

By introducing atoms \(\text{minimal}(U1), \ldots, \text{minimal}(U_t)\) (with new, i.e. existentially quantified, variables \(U_1, \ldots, U_t\) in clause 8:

clause 31: \(\text{r.tupling}(Xs, Y)\) —
\[Xs = [X | TXs],\]
nonMinimal(X), decompose(X, HX, TX1, TX2),
(nonMinimal(TX1); \ldots; nonMinimal(TXp−1)).
\(\text{compose}(TY1, I_{p+1}, I_t), \ldots, \text{compose}(TYp, I_{p+1}, I_p),\)
\(\text{compose}(HY, I_0, I_{p−1}),\)
\(\text{compose}(TYp−1, I_{p−1}, I_{p−2}), \ldots, \text{compose}(TY1, I_0, I_0),\)
\(HY = I_0, \text{r.tupling}(TXs, TY), \text{compose}(HY, TY, Y)\)

By using applicability condition (3)
clause 32:  \( r.tupling(Xs, Y) \) —
\[
Xs = [X | TXs],
\]
nonMinimal(\( X \)), decompose(\( X, HX, TX_1, TX_2 \)),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)),
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_1) \)),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_1) \)),
\( r(U_1, e), \ldots, r(U_t, e) \)),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)),
\( I_{t+1} = e \)),
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)),
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}) \)),
\( \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}) \), \ldots, \text{compose}(TY_1, I_1, I_0) \)),
\( HY = I_0 \), \( r.tupling(TXS, TY) \)), \text{compose}(HY, TY, Y) \).

By using applicability condition (2):

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

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

clause 34:  \( r.tupling(Xs, Y) \) —
\[
Xs = [X | TXs],
\]
nonMinimal(\( X \)), decompose(\( X, HX, TX_1, \ldots, TX_1 \)),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)),
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_1) \)),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_1) \)),
\( r(U_1, e), \ldots, r(U_t, e) \)),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)),
\( I_{t+1} = e \)),
\( \text{compose}(e, I_{t+1}, I_t), \ldots, \text{compose}(e, I_{p+1}, I_p) \)),
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}) \)),
\( \text{compose}(e, I_{p-1}, I_{p-2}) \), \ldots, \text{compose}(e, I_1, I_0) \)),
\( NHY = I_0 \)),
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)),
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}) \), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}) \)),
\( \text{compose}(K_{t-2}, NHY, TY), \text{compose}(TT, K_{t-1}, HY) \)),
\( r.tupling(TXS, TY) \)), \text{compose}(HY, TY, Y) \).

By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_t \) in place of some occurrences of \( e \):
clause 35:  \( r.tupling(X_s, Y) \) —  
\( X_s = [X][T X_s] \)  
\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \)  
\( \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \)  
\( \text{(nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)) \)  
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \)  
\( r(U_1, Y U_1), \ldots, r(U_t, Y U_t) \)  
\( r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t) \)  
\( I_{t+1} = \epsilon \)  
\( \text{compose}(Y U_1, I_{t+1}, I_t), \ldots, \text{compose}(Y U_p, I_{p+1}, I_p) \)  
\( \text{process}(H X, H H Y), \text{compose}(H H Y, I_p, I_{p-1}) \)  
\( \text{compose}(Y U_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(Y U_1, I_1, I_0) \)  
\( N H Y = I_0 \)  
\( \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}) \)  
\( \text{compose}(T Y_p, T Y_{p+1}, K_p), \text{compose}(K_p, T Y_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, T Y_t, K_{t-1}) \)  
\( \text{compose}(K_{p-2}, N H Y, T T), \text{compose}(T T, K_{t-1}, H Y) \)  
\( r.tupling(T X_s, T Y), \text{compose}(H Y, T Y, Y) \)  
By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N, H X, U_1, \ldots, U_t) \), since  
\( \exists N : X . \text{nonMinimal}(N) \land \text{decompose}(N, H X, U_1, \ldots, U_t) \)  
always holds (because \( N \) is existentially quantified)

clause 36:  \( r.tupling(X_s, Y) \) —  
\( X_s = [X][T X_s] \)  
\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \)  
\( \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \)  
\( \text{(nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)) \)  
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \)  
\( r(U_1, Y U_1), \ldots, r(U_t, Y U_t) \)  
\( \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_t) \)  
\( r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t) \)  
\( I_{t+1} = \epsilon \)  
\( \text{compose}(Y U_1, I_{t+1}, I_t), \ldots, \text{compose}(Y U_p, I_{p+1}, I_p) \)  
\( \text{process}(H X, H H Y), \text{compose}(H H Y, I_p, I_{p-1}) \)  
\( \text{compose}(Y U_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(Y U_1, I_1, I_0) \)  
\( N H Y = I_0 \)  
\( \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}) \)  
\( \text{compose}(T Y_p, T Y_{p+1}, K_p), \text{compose}(K_p, T Y_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, T Y_t, K_{t-1}) \)  
\( \text{compose}(K_{p-2}, N H Y, T T), \text{compose}(T T, K_{t-1}, H Y) \)  
\( r.tupling(T X_s, T Y), \text{compose}(H Y, T Y, Y) \)  
By duplicating goal \( \text{decompose}(N, H X, U_1, \ldots, U_t) \):
clause 37: \[\text{rutpling}(X_s, Y) \rightarrow \]
\[X_s = [X | TX_s].\]
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ... ; nonMinimal(TX_{p-1})),
(nonMinimal(TX_p); ... ; nonMinimal(TX_t)),
minimal(U_1), ..., minimal(U_t),
decompose(N, HX, U_1, ..., U_t),
r(U_1, YU_1), ..., r(U_t, YU_t),
nonMinimal(N), decompose(N, HX, U_1, ..., U_t),
r(TX_1, TY_1), ..., r(TX_t, TY_t),
I_{t+1} = e,
 \begin{align*}
& \text{compose}(YU_1, I_{t+1}, I_t), \ldots , \text{compose}(YU_p, I_{p+1}, I_p),
& \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}),
& \text{compose}(Y_{p-1}, I_{p-1}, I_{p-2}), \ldots , \text{compose}(YU_1, I_1, I_0),
\end{align*}
NHY = I_0,
\begin{align*}
& \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots , \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}),
& \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots , \text{compose}(K_{t-2}, TY_t, K_{t-1}),
& \text{compose}(K_{p-2}, NHY, TI), \text{compose}(TI, K_{t-1}, HY),
\end{align*}
r\text{utpling}(TX_s, TY), \text{compose}(HY, TY, Y)

By folding clause 37 using \(DCRL\):

clause 38: \[\text{rutpling}(X_s, Y) \rightarrow \]
\[X_s = [X | TX_s].\]
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ... ; nonMinimal(TX_{p-1})),
(nonMinimal(TX_p); ... ; nonMinimal(TX_t)),
minimal(U_1), ..., minimal(U_t),
decompose(N, HX, U_1, ..., U_t),
r(TX_1, TY_1), ..., r(TX_t, TY_t), r(N, NHY),
\begin{align*}
& \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots , \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}),
& \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots , \text{compose}(K_{t-2}, TY_t, K_{t-1}),
& \text{compose}(K_{p-2}, NHY, TI), \text{compose}(TI, K_{t-1}, HY),
\end{align*}
r\text{utpling}(TX_s, TY), \text{compose}(HY, TY, Y)

By using applicability condition (1):

clause 39: \[\text{rutpling}(X_s, Y) \rightarrow \]
\[X_s = [X | TX_s].\]
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ... ; nonMinimal(TX_{p-1})),
(nonMinimal(TX_p); ... ; nonMinimal(TX_t)),
minimal(U_1), ..., minimal(U_t),
decompose(N, HX, U_1, ..., U_t),
r(TX_1, TY_1), ..., r(TX_t, TY_t), r(N, NHY),
\begin{align*}
& \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots , \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}),
& \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots , \text{compose}(K_{t-2}, TY_t, K_{t-1}),
& \text{compose}(K_{p-2}, TI_1, TI_2, Y), \text{compose}(NHY, TI_1, TI_2),
\end{align*}
r\text{utpling}(TX_s, TY), \text{compose}(K_{t-1}, TY, TI_1)

By \(t - p + 1\) times folding clause 39 using clauses 1 and 2:
clause 40: \( r_{\text{tupling}}(X_s, Y) \) 
\[ X_s = [X | TX_s], \]
nonMinimal(X), \( \text{decompose}(X, HX, TX_1, \ldots, TX_t) \),
nonMinimal(TX_1), \ldots, nonMinimal(TX_{p-1}),
nonMinimal(TX_p), \ldots, nonMinimal(TX_{t1}),
minimal(U_1), \ldots, minimal(U_t),
decompose(N, HX, U_1, \ldots, U_t),
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}) \), \( r(N, NHY) \),
\( \text{compose}(TY_1, TY_2, K_1) \), \( \text{compose}(K_1, TY_3, K_2) \), \ldots, \( \text{compose}(K_{p-3}, TY_{p-1}, K_p-2) \),
\( \text{compose}(K_{p-2}, TI_2, Y) \), \( \text{compose}(NHY, TI_1, TI_2) \),
r_{\text{tupling}}([TX_p, \ldots, TX_t | TX_s], TI_1)

By folding clause 40 using clauses 1 and 2:

clause 41: \( r_{\text{tupling}}(X_s, Y) \) 
\[ X_s = [X | TX_s], \]
nonMinimal(X), \( \text{decompose}(X, HX, TX_1, \ldots, TX_t) \),
nonMinimal(TX_1), \ldots, nonMinimal(TX_{p-1}),
nonMinimal(TX_p), \ldots, nonMinimal(TX_{t1}),
minimal(U_1), \ldots, minimal(U_t),
decompose(N, HX, U_1, \ldots, U_t),
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}) \), \( r(N, NHY) \),
\( \text{compose}(TY_1, TY_2, K_1) \), \( \text{compose}(K_1, TY_3, K_2) \), \ldots, \( \text{compose}(K_{p-3}, TY_{p-1}, K_p-2) \),
\( \text{compose}(K_{p-2}, TI_2, Y) \),
r_{\text{tupling}}([N, TX_p, \ldots, TX_t | TX_s], TI_1)

By \( p - 1 \) times folding clause 41 using clauses 1 and 2:

clause 42: \( r_{\text{tupling}}(X_s, Y) \) 
\[ X_s = [X | TX_s], \]
nonMinimal(X), \( \text{decompose}(X, HX, TX_1, \ldots, TX_t) \),
nonMinimal(TX_1), \ldots, nonMinimal(TX_{p-1}),
nonMinimal(TX_p), \ldots, nonMinimal(TX_{t1}),
minimal(U_1), \ldots, minimal(U_t),
decompose(N, HX, U_1, \ldots, U_t),
r_{\text{tupling}}([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t | TX_s], TI_1)

Clauses 1, 3, 13, 20, 30 and 42 are the clauses of \( P_{r_{\text{tupling}}} \). Therefore \( P_{r_{\text{tupling}}} \) is steadfast wrt \( S_{r_{\text{tupling}}} \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{ S_{r_{\text{tupling}}} \} \), 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:
\( r(X, Y) \leftarrow r_{\text{tupling}}([X], Y) \)

By taking the ‘completion’:
\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r_{\text{tupling}}([X], Y)] \]

By unfolding the ‘completion’ above wrt \( r_{\text{tupling}}([X], Y) \) using \( S_{r_{\text{tupling}}} \):
\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)] \]

By simplification:
\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)] \]

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{ S_{r_{\text{tupling}}} \} \).
Therefore, \( TG \) is also steadfast wrt \( S_r \) in \( S \).

\[ \square \]

3 Proofs of the Descending Generalization Schemas

Theorem 3 The generalization schema \( DG_1 \), which is given below, is correct.
$DG_1 : \{ DCLR, DGLR, A_{d1}, O_{d12}, O_{d121} \}$ where

$A_{d1} : (1) \text{ compose is associative}
(2) \text{ compose has } e \text{ as the left identity element,}
where e appears in DCLR and DGLR$

$O_{d12} : - \text{ compose has } e \text{ as the right identity element,}
where e appears in DCLR and DGLR
and $I_e(X) \land \text{minimal}(X) \Rightarrow O_e(X, e)$
- partial evaluation of the conjunction
$\text{process}(H_X, H_Y), \text{compose}(A_{p-1}, H_Y, A_p)$
results in the introduction of a non-recursive relation

$O_{d121} : - \text{ partial evaluation of the conjunction}
\text{process}(H_X, H_Y), \text{compose}(I_{p-1}, HY, I_p)$
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**

\[
\begin{align*}
&x(X, Y) \leftarrow \\
&\quad r_{\text{descending}}(X, Y, e) \\
&r_{\text{descending}}(X, Y, A) \leftarrow \\
&\quad \text{minimal}(X), \\
&\quad \text{solve}(X, S), \text{compose}(A, S, Y) \\
&r_{\text{descending}}(X, Y, A) \leftarrow \\
&\quad \text{nonMinimal}(X), \\
&\quad \text{decompose}(X, H_X, T_X, \ldots, T_X), \\
&\quad \text{compose}(A, e, A_0), \\
&\quad r_{\text{descending}}(T_{X_1}, A_1, A_0), \ldots, r_{\text{descending}}(T_{X_{p-1}}, A_{p-1}, A_{p-2}), \\
&\quad \text{process}(H_X, H_Y), \text{compose}(A_{p-1}, H_Y, A_p), \\
&\quad r_{\text{descending}}(T_{X_p}, A_{p+1}, A_p), \ldots, r_{\text{descending}}(T_{X_t}, A_{t+1}, A_t) \\
&y = A_{t+1}
\end{align*}
\]

and the specification $S_r$ of relation $r$ is:

$\forall X : X. \forall Y : Y. \quad I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)]$

and the specification $S_{r_{\text{descending}}}$ of relation $r_{\text{descending}}$ is:

$\forall X : X. \forall Y, A : Y. \quad I_r(X) \Rightarrow [r_{\text{descending}}(X, Y, A) \Leftrightarrow \exists S : Y. \quad O_r(X, S) \land O_c(A, S, Y)]$

**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 $A_{d1}$. By Definition 5, the templates DCLR and DGLR are equivalent wrt $S_r$ under the applicability conditions $A_{d1}$ iff DCLR is equivalent to DGLR wrt the specification $S_r$ provided that the conditions in $A_{d1}$ 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 $S = \{ S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}$ 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 $S$.

Note that the sets $\{ S_1, \ldots, S_m \}$ and $\{ S'_1, \ldots, S'_l \}$ in Definition 4 are equal to $S$ 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 $S$, so condition (a) always holds.
To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DCLR$ is steadfast wrt $S_r$ in $S$ if $P_{r,\text{descending}_1}$ is steadfast wrt $S_{r,\text{descending}_1}$ in $S$, where $P_{r,\text{descending}_1}$ is the procedure for $r_{\text{descending}_1}$, and $P_r$ is steadfast wrt $S_r$ in $\{S_{r,\text{descending}_1}\}$, where $P_r$ is the procedure for $r$.

To prove that $P_{r,\text{descending}_1}$ is steadfast wrt $S_{r,\text{descending}_1}$ in $S$, we do a constructive forward proof that we begin with $S_{r,\text{descending}_1}$ and from which we try to obtain $P_{r,\text{descending}_1}$.

By taking the ‘decomposition’ of $S_{r,\text{descending}_1}$:

\begin{align*}
\text{clause 1:} & \quad r_{\text{descending}_1}(X, Y, A) \leftarrow r(X, S), \text{compose}(A, S, Y) \\
\text{By unfolding clause 1 wrt } r(X, S) \text{ using DCLR, and using the assumption that DCLR is steadfast wrt } S_r \text{ in } S:} \\
\text{clause 2:} & \quad r_{\text{descending}_1}(X, Y, A) \leftarrow \\
& \quad \text{minimal}(X), \text{solve}(X, S), \text{compose}(A, S, Y) \\
\text{clause 3:} & \quad r_{\text{descending}_1}(X, Y, A) \leftarrow \\
& \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\
& \quad l_0 = e, \text{compose}(I_0, TS_1, I_1), \ldots, \text{compose}(I_{p-1}, TS_{p-1}, I_p), \\
& \quad \text{process}(HX, HS), \text{compose}(I_{p-1}, HS, I_p), \\
& \quad \text{compose}(I_p, TS_p, I_{p+1}), \ldots, \text{compose}(I_t, TS_t, I_{t+1}), \\
& \quad S = I_{t+1}, \text{compose}(A, S, Y) \\
\text{By using applicability condition (1) on clause 3:} \\
\text{clause 4:} & \quad r_{\text{descending}_1}(X, Y, A) \leftarrow \\
& \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\
& \quad \text{compose}(A, e, A_0), \\
& \quad \text{compose}(A_0, TS_1, A_1), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-1}), \\
& \quad \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \\
& \quad \text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \\
& \quad Y = A_{t+1} \\
\text{By } t \text{ times folding clause 4 using clause 1:} \\
\text{clause 5:} & \quad r_{\text{descending}_1}(X, Y, A) \leftarrow \\
& \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \quad \text{compose}(A, e, A_0), \\
& \quad \text{compose}(A_0, TX_1, A_1, A_0), \ldots, \text{compose}(TX_{p-1}, A_{p-1}, A_{p-2}), \\
& \quad \text{process}(HX, HY), \text{compose}(A_{p-1}, HY, A_p), \\
& \quad r_{\text{descending}_1}(TX_1, A_{p+1}, A_p), \ldots, r_{\text{descending}_1}(TX_t, A_{t+1}, A_{t}), \\
& \quad Y = A_{t+1} \\
\text{Clauses 2 and 5 are the clauses of the } P_{r,\text{descending}_1}. \quad \text{Therefore } P_{r,\text{descending}_1} \text{ is steadfast wrt } S_{r,\text{descending}_1} \text{ in } S.
\end{align*}

To prove that $P_r$ is steadfast wrt $S_r$ in $\{S_{r,\text{descending}_1}\}$, we do a backward proof that we begin with $P_r$ in $DCLR$ and from which we try to obtain $S_r$.

The procedure $P_r$ for $r$ in $DCLR$ is:

\begin{align*}
\text{r}(X, Y) & \leftarrow \ r_{\text{descending}_1}(X, Y, e) \\
\text{By taking the ‘completion’:} \\
\forall X : X, \forall Y : Y. \quad I_r(X) \Rightarrow [r(X, Y) \iff r_{\text{descending}_1}(X, Y, e)] \\
\text{By unfolding the ‘completion’ above wrt } r_{\text{descending}_1}(X, Y, e) \text{ using } S_{r,\text{descending}_1}:
\forall X : X, \forall Y : Y. \quad I_r(X) \Rightarrow [r(X, Y) \iff \exists S : Y. \quad \phi_r(X, S) \land \phi_e(e, S, Y)]
\text{By using applicability condition (2):}
\forall X : X, \forall Y : Y. \quad I_r(X) \Rightarrow [r(X, Y) \iff \exists S : Y. \quad \phi_r(X, S) \land S = Y]
\end{align*}
By simplification:

\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)] \]

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{ S_{\text{descending}_2} \} \).
Therefore, \( DGLR \) is also steadfast wrt \( S_r \) in \( S \). \( \square \)

**Theorem 4** The generalization schema \( DG_2 \), which is given below, is correct.

\[
DG_2 : \{ DCLR, DGLR, A_{d2}, O_{d212}, O_{d222} \} \text{ where double}\n\]

\( A_{d2} : \) (1) \( \text{compose} \) is associative

\( (2) \) \( \text{compose} \) has \( e \) as the left and right identity element.

where \( e \) appears in \( DCLR \) and \( DGLR \).

\( O_{d212} : - \quad \text{I}_r(X) \land \text{minimal}(X) \Rightarrow \text{O}_r(X,e) \)

- partial evaluation of the conjunction

\[ \text{process}(H X, H Y), \text{compose}(H Y, A_p, A_{p-1}) \]

results in the introduction of a non-recursive relation

\( O_{d222} : \) partial evaluation of the conjunction

\[ \text{process}(H X, H Y), \text{compose}(I_{p-1}, H Y, I_p) \]

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 5 below:

**Logic Program Template 5**

\[
\begin{align*}
\text{r}(X,Y) & \leftarrow \\
\text{r}_{\text{descending}_2}(X,Y,e) & \\
\text{r}_{\text{descending}_2}(X,Y,A) & \leftarrow \\
\text{minimal}(X), & \\
\text{solve}(X,S), \text{compose}(S,A,Y) & \\
\text{r}_{\text{descending}_2}(X,Y,A) & \leftarrow \\
\text{nonMinimal}(X), & \\
\text{decompose}(X,H X,T X_1,\ldots,T X_t), & \\
\text{compose}(e,A,A_{t+1}), & \\
\text{r}_{\text{descending}_2}(T X_t,A_t,A_{t+1}),\ldots,r_{\text{descending}_2}(T X_p,A_p,A_{p+1}) & \\
\text{process}(H X,H Y), \text{compose}(H Y,A_p,A_{p-1}) & \\
\text{r}_{\text{descending}_2}(T X_{p-1},A_{p-2},A_{p-1}),\ldots,r_{\text{descending}_2}(T X_1,A_0,A_1) & \\
Y & = A_0
\end{align*}
\]

and the specification \( S_r \) of relation \( r \) is:

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

and the specification \( S_{r_{\text{descending}_2}} \) of relation \( r_{\text{descending}_2} \) is:

\[
\forall X : \mathcal{X}. \forall Y, A : \mathcal{Y}. \quad \text{I}_r(X) \Rightarrow [r_{\text{descending}_2}(X,Y,A) \Leftrightarrow \exists S : \mathcal{Y}. \quad \text{O}_r(X,S) \land \text{O}_s(S,A,Y)]
\]

**Proof 4** To prove the correctness of the generalization schema \( DG_2 \), by Definition 10, we have to prove that templates \( DCLR \) and \( DGLR \) are equivalent wrt \( S_r \) under the applicability conditions \( A_{d2} \). By Definition 5, the templates \( DCLR \) and \( DGLR \) are equivalent wrt \( S_r \) under the applicability conditions \( A_{d2} \) iff \( DCLR \) is equivalent to \( DGLR \) wrt the specification \( S_r \) provided that the conditions in \( A_{d2} \) 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 \( S = \{ S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \} \),

where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{process}}, S_{\text{decompose}}, S_{\text{compose}} \) are the specifications of \( \text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose} \), which are all the undefined relation names appearing in \( DCLR \).

33
(b) DGRL is also steadfast wrt $S_r$ in $S$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_t\}$ in Definition 4 are equal to $S$ 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 $S$, 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 $S$ if $P_{r, \text{descending}_{a_2}}$ is steadfast wrt $S_{r, \text{descending}_{a_2}}$, in $S$, where $P_{r, \text{descending}_{a_2}}$ is the procedure for $r_{\text{descending}_{a_2}}$, and $P_r$ is steadfast wrt $S_r$ in $\{S_{r, \text{descending}_{a_2}}\}$, where $P_r$ is the procedure for $r$.

To prove that $P_{r, \text{descending}_{a_2}}$ is steadfast wrt $S_{r, \text{descending}_{a_2}}$ in $S$, we do a constructive forward proof that we begin with $S_{r, \text{descending}_{a_2}}$ and from which we try to obtain $P_{r, \text{descending}_{a_2}}$.

By taking the ‘decompletion’ of $S_{r, \text{descending}_{a_2}}$:

clause 1: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow r(X, S), \text{compose}(S, A, Y)$

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

clause 2: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow$

minimal(X),
\text{solve}(X, S), \text{compose}(S, A, Y)

clause 3: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow$
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
r(TX_1, TS_1), \ldots, r(TX_t, TS_t),
l_0 = e,
\text{compose}(I_0, TS_1, I_1), \ldots, \text{compose}(I_{p-2}, TS_{p-1}, I_{p-1}),
\text{process}(HX, HS), \text{compose}(I_{p-1}, HS, I_p),
\text{compose}(I_p, TS_p, I_{p+1}), \ldots, \text{compose}(I_t, TS_t, I_{t+1}),
S = I_{t+1}, \text{compose}(S, A, Y)$

By using applicability condition (1) on clause 3:

clause 4: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow$
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
r(TX_1, TS_1), \ldots, r(TX_t, TS_t),
\text{compose}(TS_1, A, A_1), \ldots, \text{compose}(TS_p, A_{p+1}, A_p),
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}),
\text{compose}(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TS_1, A_1, A_0),
\text{compose}(e, A_0, Y)$

By using applicability condition (2) on clause 4:

clause 5: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow$
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
r(TX_1, TS_1), \ldots, r(TX_t, TS_t),
\text{compose}(TS_1, A, A_1), \ldots, \text{compose}(TS_p, A_{p+1}, A_p),
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}),
\text{compose}(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TS_1, A_1, A_0),
Y = A_0$

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

clause 6: $r_{\text{descending}_{a_2}}(X, Y, A) \leftarrow$
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
r(TX_1, TS_1), \ldots, r(TX_t, TS_t),
\text{compose}(e, A, A_{t+1}),
\text{compose}(TS_1, A_{t+1}, A_1), \ldots, \text{compose}(TS_p, A_{p+1}, A_p),
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}),
\text{compose}(TS_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TS_1, A_1, A_0),
Y = A_0$
By $t$ times folding clause 6 using clause 1:

**clause 7**: $r_{\text{descending}}_2(X, Y, A) \leftarrow$
\begin{itemize}
  \item \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
  \item \text{compose}(e, A, A_{t+1}),
  \item $r_{\text{descending}}_2(TX_t, A, A_{t+1})$, \ldots, $r_{\text{descending}}_2(TX_p, A_p, A_{p+1}),$
  \item process($HX, HY$), \text{compose}(HY, A_p, A_{p-1}).
\end{itemize}

$r_{\text{descending}}_2(TX_{p-1}, A_{p-2}, A_{p-1})$, \ldots, $r_{\text{descending}}_2(TX_1, A_0, A_1)$.

$Y = A_e$

Clauses 2 and 7 are the clauses of $P_{r_{\text{descending}}_2}$. Therefore $P_{r_{\text{descending}}_2}$ is steadfast wrt $S_{r_{\text{descending}}_2}$ in $S$.

To prove that $P_r$ is steadfast wrt $S_r$ in $\{S_{r_{\text{descending}}_2}\}$, 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:

$$r(X,Y) \leftarrow r_{\text{descending}}_2(X, Y, e)$$

By taking the ‘completion’:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r_{\text{descending}}_2(X, Y, e)]$$

By unfolding the ‘completion’ above wrt $r_{\text{descending}}_2(X, Y, e)$ using $S_{r_{\text{descending}}_2}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists S : \mathcal{Y}. \ O_r(X, S) \land O_r(S, e, Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists S : \mathcal{Y}. \ O_r(X, S) \land S = Y]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X, Y)]$$

We obtain $S_r$, so $P_r$ is steadfast wrt $S_r$ in $\{S_{r_{\text{descending}}_2}\}$.

Therefore, $DGRL$ is also steadfast wrt $S_r$ in $S$. 

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

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

- **$Adg3$** (1) \text{compose} is associative
  - where $e$ appears in $DCRL$ and $DGRL$

- **$O_{dg312}$** - \text{compose} has $e$ as the right identity element.
  - where $e$ appears in $DCRL$ and $DGRL$
  - and $I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e)$
  - partial evaluation of the conjunction
  - $\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1})$
  - results in the introduction of a non-recursive relation

- **$O_{dg321}$** - partial evaluation of the conjunction
  - $\text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1})$
  - 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:

$$\forall X : \mathcal{X}. \ \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X, Y)]$$

The specification $S_{r_{\text{descending}}_2}$ of relation $r_{\text{descending}}_2$ is:

$$\forall X : \mathcal{X}. \ \forall Y, A : \mathcal{Y}. \ I_r(X) \Rightarrow [r_{\text{descending}}_2(X, Y, A) \Leftrightarrow \exists S : \mathcal{Y}. \ O_r(X, S) \land O_r(S, A, Y)]$$
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 $A_{d3}$. By Definition 5, the templates $DCRL$ and $DGRL$ are equivalent wrt $S_r$ under the applicability conditions $A_{d3}$ iff $DCRL$ is equivalent to $DGRL$ wrt the specification $S_r$ provided that the conditions in $A_{d3}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}$ are the specifications of $\text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose}$, which are all the undefined relation names appearing in $DCRL$.

(b) $DGRL$ is also steadfast wrt $S_r$ in $S$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_l\}$ in Definition 4 are equal to $S$ 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 $S$, 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 $S$ if $P_{r, \text{descending}_2}$ is steadfast wrt $S_{r, \text{descending}_2}$ in $S$, where $P_{r, \text{descending}_2}$ is the procedure for $r, \text{descending}_2$, and $P_r$ is steadfast wrt $S_r$ in $\{S_{r, \text{descending}_2}\}$, where $P_r$ is the procedure for $r$.

To prove that $P_{r, \text{descending}_2}$ is steadfast wrt $S_{r, \text{descending}_2}$ in $S$, we do a constructive forward proof that we begin with $S_{r, \text{descending}_2}$ and from which we try to obtain $P_{r, \text{descending}_2}$.

By taking the ‘decomposition’ of $S_{r, \text{descending}_2}$:

\[
\text{clause 1: } r, \text{descending}_2(X, Y, A) \leftarrow r(X, S), \text{compose}(S, A, Y)
\]

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

\[
\text{clause 2: } r, \text{descending}_2(X, Y, A) \leftarrow \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, S), \text{compose}(S, A, Y)
\]

\[
\text{clause 3: } r, \text{descending}_2(X, Y, A) \leftarrow \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\
\quad 1_{t+1} = e, \\
\quad \text{compose}(TS_1, I_{t+1}, I_1), \ldots, \text{compose}(TS_p, I_{p+1}, I_p), \\
\quad \text{process}(HX, HS), \text{compose}(HS, I_0, I_{p-1}), \\
\quad \text{compose}(TS_{p-1}, I_{p-1}, I_p), \ldots, \text{compose}(TS_1, I_1, I_0), \\
\quad S = I_0, \text{compose}(S, A, Y)
\]

By using applicability condition (1) on clause 3:

\[
\text{clause 4: } r, \text{descending}_2(X, Y, A) \leftarrow \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\
\quad \text{compose}(e, A, A_{t+1}), \\
\quad \text{compose}(TS_t, A_{t+1}, A_t), \ldots, \text{compose}(TS_p, A_{p+1}, A_p), \\
\quad \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \\
\quad \text{compose}(TS_{p-1}, A_{p-1}, A_p), \ldots, \text{compose}(TS_1, A_1, A_0), \\
\quad Y = A_e
\]

By $t$ times folding clause 4 using clause 1:

\[
\text{clause 5: } r, \text{descending}_2(X, Y, A) \leftarrow \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad \text{compose}(e, A, A_{t+1}), \\
\quad r, \text{descending}_2(TX_1, A_1, A_{t+1}), \ldots, r, \text{descending}_2(TX_p, A_p, A_{p+1}), \\
\quad \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \\
\quad r, \text{descending}_2(TX_{p-1}, A_{p-1}, A_p), \ldots, r, \text{descending}_2(TX_1, A_0, A_1), \\
\quad Y = A_e
\]
Clauses 2 and 5 are the clauses of $P_{\text{descending}_2}$. Therefore $P_{\text{descending}_2}$ is steadfast wrt $S_{\text{descending}_2}$ in $S$.

To prove that $P_r$ is steadfast wrt $S_r$ in $\{S_{\text{descending}_2}\}$, 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:

$$r(X,Y) \leftarrow r_{\text{descending}_2}(X,Y,e)$$

By taking the ‘completion’:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r_{\text{descending}_2}(X,Y,e)]$$

By unfolding the ‘completion’ above wrt $r_{\text{descending}_2}(X,Y,e)$ using $S_{\text{descending}_2}$:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists S : \mathcal{Y}. \quad O_r(X,S) \land O_e(S,e,Y)]$$

By using applicability condition (2):

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists S : \mathcal{Y}. \quad O_r(X,S) \land S = Y]$$

By simplification:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)]$$

We obtain $S_r$, so $P_r$ is steadfast wrt $S_r$ in $\{S_{\text{descending}_2}\}$. Therefore, DGRL is also steadfast wrt $S_r$ in $S$.

Theorem 6 The generalization schema $\text{DG}_4$, which is given below, is correct.

$$\text{DG}_4 : \{ \text{DCRL}, \text{DGLR}, A_{\text{dglr}}, O_{\text{dglr1}}, O_{\text{dglr2}} \}$$

where

- $A_{\text{dglr}}$ is associative
- $O_{\text{dglr1}}$ is $I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X,e)$
- $O_{\text{dglr2}}$ is $\text{process}(H_X,H_Y), \text{compose}(A_{p-1},H_Y,A_p)$
- $O_{\text{dglr2}}$ results in the introduction of a non-recursive relation

The specification $S_r$ of relation $r$ is:

$$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)]$$

The specification $S_{\text{descending}_2}$ of relation $r_{\text{descending}_2}$ is:

$$\forall X : \mathcal{X}, \forall Y, A : \mathcal{Y}. \quad I_r(X) \Rightarrow [r_{\text{descending}_2}(X,Y,A) \Leftrightarrow \exists S : \mathcal{Y}. \quad O_r(X,S) \land O_e(A,S,Y)]$$

Proof 6 To prove the correctness of the generalization schema $\text{DG}_4$, by Definition 10, we have to prove that templates DCRL and DGLR are equivalent wrt $S_r$ under the applicability conditions $A_{\text{dglr}}$. By Definition 5, the templates DCRL and DGLR are equivalent wrt $S_r$ under the applicability conditions $A_{\text{dglr}}$ iff DCRL is equivalent to DGLR wrt the specification $S_r$ provided that the conditions in $A_{\text{dglr}}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{process}}, S_{\text{decompose}}, S_{\text{compose}}$ 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 \( S \).

Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S'_1, \ldots, S'_t\} \) in Definition 4 are equal to \( S \) in \( S \) 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 \( S \), 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 \( S \) if \( P_{r, \text{descending}} \) is steadfast wrt \( S_{r, \text{descending}} \) in \( S \), where \( P_{r, \text{descending}} \) is the procedure for \( r_{\text{descending}} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r, \text{descending}}\} \), where \( P_r \) is the procedure for \( r \).

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

By taking the ‘decompletion’ of \( S_{r, \text{descending}} \):

\[ \text{clause 1: } r_{\text{descending}}(X, Y, A) \leftarrow r(X, S), \text{compose}(A, S, Y) \]

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

\[ \text{clause 2: } r_{\text{descending}}(X, Y, A) \leftarrow \]
\[ \text{minimal}(X), \]
\[ \text{solve}(X, S), \text{compose}(A, S, Y) \]

\[ \text{clause 3: } r_{\text{descending}}(X, Y, A) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TS_1, I_{t+1}, I_1), \ldots, \text{compose}(TS_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HS), \text{compose}(HS, I_p, I_{p-1}), \]
\[ \text{compose}(TS_{p-1}, I_{p-2}, I_{p-1}), \ldots, \text{compose}(TS_1, I_1, I_0), \]
\[ S = I_0, \text{compose}(A, S, Y) \]

By using applicability condition (1) on clause 3:

\[ \text{clause 4: } r_{\text{descending}}(X, Y, A) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \]
\[ \text{compose}(A, TS_1, A_1), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-1}), \]
\[ \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \]
\[ \text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \]
\[ \text{compose}(A_{t+1}, e, Y) \]

By using applicability condition (2) on clause 4:

\[ \text{clause 5: } r_{\text{descending}}(X, Y, A) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \]
\[ \text{compose}(A, TS_1, A_1), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-1}), \]
\[ \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \]
\[ \text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \]
\[ Y = A_{t+1} \]

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

\[ \text{clause 6: } r_{\text{descending}}(X, Y, A) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \]
\[ \text{compose}(A, e, A_0), \]
\[ \text{compose}(A_0, TS_1, A_1), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-1}), \]
\[ \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \]
\[ \text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \]
\[ Y = A_{t+1} \]
By \( t \) times folding clause 6 using clause 1:

**Clause 7:** \( r_{\text{descending}}(X, Y, A) \leftarrow 
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), 
\text{compose}(A, e, A_0), 
\text{r}_{\text{descending}}(TX_1, A_1, A_0), \ldots, \text{r}_{\text{descending}}(TX_{p-1}, A_{p-1}, A_{p-2}), 
\text{process}(HX, HY), \text{compose}(A_{p-1}, HY, A_p), 
\text{r}_{\text{descending}}(TX_p, A_{p+1}, A_p), \ldots, \text{r}_{\text{descending}}(TX_t, A_t, A_{t+1}), 
Y = A_{t+1} \)

Clauses 2 and 7 are the clauses of the \( P_{\text{r}_{\text{descending}}} \). Therefore \( P_{\text{r}_{\text{descending}}} \) is steadfast wrt \( S_{\text{r}_{\text{descending}}} \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{\text{r}_{\text{descending}}}\} \), 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:

\[
r(X, Y) \leftarrow r_{\text{descending}}(X, Y, e)
\]

By taking the ‘completion’:

\[
\forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow \langle r(X, Y) \leftrightarrow r_{\text{descending}}(X, Y, e) \rangle
\]

By unfolding the ‘completion’ above wrt \( r_{\text{descending}}(X, Y, e) \) using \( S_{\text{r}_{\text{descending}}} \):

\[
\forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow \langle r(X, Y) \leftrightarrow \exists S : Y. \ O_r(X, S) \wedge O_r(e, S, Y) \rangle
\]

By using applicability condition (2):

\[
\forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow \langle r(X, Y) \leftrightarrow \exists S : Y. \ O_r(X, S) \wedge S = Y \rangle
\]

By simplification:

\[
\forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow \langle r(X, Y) \leftrightarrow O_r(X, Y) \rangle
\]

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{\text{r}_{\text{descending}}}\} \).

Therefore, \( DGLR \) is also steadfast wrt \( S_r \) in \( S \).

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

**Theorem 7** The generalization schema \( TDG_1 \), which is given below, is correct.

\[
TDG_1 : \{ DC^L R, TDGLR, A_{td1}, O_{td12}, O_{td121} \}
\]

where

- \( A_{td1} : (1) \text{ compose is associative} 
(2) \text{ compose has } e \text{ as the left and right identity element} 
(3) \forall X : X. \ I_r(X) \wedge \text{ minimal}(X) \Rightarrow O_r(X, e) 
(4) \forall X : X. \ I_r(X) \Rightarrow \langle-\text{minimal}(X) \leftrightarrow \text{nonMinimal}(X) \rangle 
\]

- \( O_{td12} : \text{ partial evaluation of the conjunction} 
  \text{process}(HX, HY), \text{compose}(A, HY, NewA) 
  \text{ results in the introduction of a non-recursive relation} 
\]

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

where the template \( DC^L R \) is Logic Program Template 1 in Section 2 and the template \( TDGLR \) is Logic Program Template 6 below.
Logic Program Template 6

\[ r(X, Y) \rightarrow \\
\quad r_{td_1}(X, Y, e) \]

\[ r_{td_2}(Xs, Y, A) \rightarrow \\
\quad Xs = [], \\
\quad Y = A \]

\[ r_{td_1}(Xs, Y, A) \rightarrow \\
\quad Xs = [X|TXs], \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, HY), \\
\quad \text{compose}(A, HY, NewA), \\
\quad r_{td_1}(TXs, Y, NewA) \]

\[ r_{td_1}(Xs, Y, A) \rightarrow \\
\quad Xs = [X|TXs], \\
\quad \text{nonMinimal}(X), \\
\quad \text{decompose}(X, HX, TX_1, \ldots, TX_T), \\
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_T), \\
\quad \text{process}(HX, HY), \text{compose}(A, HY, NewA), \\
\quad r_{td_1}(TXs, Y, NewA) \]

\[ r_{td_2}(Xs, Y, A) \rightarrow \\
\quad Xs = [X|TXs], \\
\quad \text{nonMinimal}(X), \\
\quad \text{decompose}(X, HX, TX_1, \ldots, TX_T), \\
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\quad \text{minimal}(TX_{p}), \ldots, \text{minimal}(TX_T), \\
\quad \text{process}(HX, HY), \text{compose}(A, HY, NewA), \\
\quad r_{td_1}([TX_p, \ldots, TX_T|TXs], Y, NewA) \]

\[ r_{td_1}(Xs, Y, A) \rightarrow \\
\quad Xs = [X|TXs], \\
\quad \text{nonMinimal}(X), \\
\quad \text{decompose}(X, HX, TX_1, \ldots, TX_T), \\
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\quad \text{minimal}(TX_{p}), \ldots, \text{minimal}(TX_T), \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
\quad \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_T), \\
\quad r_{td_1}([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A) \]

\[ r_{td_2}(Xs, Y, A) \rightarrow \\
\quad Xs = [X|TXs], \\
\quad \text{nonMinimal}(X), \\
\quad \text{decompose}(X, HX, TX_1, \ldots, TX_T), \\
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\quad \text{minimal}(TX_{p}), \ldots, \text{minimal}(TX_T), \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_T), \\
\quad \text{decompose}(N, HX, U_1, \ldots, U_T), \\
\quad r_{td_1}([TX_1, \ldots, TX_T, N, TX_p, \ldots, TX_T|TXs], Y, A) \]
and the specification \( S_r \) of relation \( r \) is:

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

and the specification \( S_{r,tdl} \):

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r_{tdl}(Xs, Y, A) \Leftrightarrow (Xs = [] \land Y = A) \\
\lor (Xs = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \land Y_1 = Y_1 \land \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \\
\land \mathcal{O}_c(A, I_q, I_{q+1}) \land Y = Y_{q+1})]
\]

**Proof 7** To prove the correctness of the generalization schema \( TDGLR \), by Definition 10, we have to prove that templates \( DCLR \) and \( TDGLR \) are equivalent wrt \( S_r \) under the applicability conditions \( A_{tdl} \). By Definition 5, the templates \( DCLR \) and \( TDGLR \) are equivalent wrt \( S_r \) under the applicability conditions \( A_{tdl} \) iff \( DCLR \) is equivalent to \( TDGLR \) wrt the specification \( S_r \) provided that the conditions in \( A_{tdl} \) 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 \( S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \)

where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \) are the specifications of \( \text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose} \), which are all the undefined relation names appearing in \( DCLR \).

(b) \( TDGLR \) is also steadfast wrt \( S_r \) in \( S \).

Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S_1', \ldots, S_l\} \) in Definition 4 are equal to \( S \) when \( Q \) is obtained by simultaneous tupling-and-descending generalization of \( P_r \).

In program transformation, we assume that the input program, here template \( DCLR \), is steadfast wrt \( S_r \) in \( S \), 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 \( S \) if \( P_{r,tdl} \) is steadfast wrt \( S_{r,tdl} \) in \( S \), where \( P_{r,tdl} \) is the procedure for \( r_{tdl} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r,tdl}\} \), where \( P_r \) is the procedure for \( r \).

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

If we separate the cases of \( q \geq 1 \) by \( q = 1 \lor q = 2 \), then \( S_{r,tdl} \) becomes:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r_{tdl}(Xs, Y, A) \Leftrightarrow \\
\lor (Xs = [X_1] \land Y = A) \\
\lor (Xs = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \land Y_1 = Y_1 \land \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \\
\land \mathcal{O}_c(A, I_q, I_{q+1}) \land Y = Y_{q+1})]
\]

where \( q > 2 \).

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

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r_{tdl}(Xs, Y, A) \Leftrightarrow \\
\lor (Xs = [X_1] \land Y = A) \\
\lor (Xs = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \land Y_1 = Y_1 \land \bigwedge_{i=2}^{q} \mathcal{O}_c(I_{i-1}, Y_i, I_i) \\
\land \mathcal{O}_c(A, I_q, I_{q+1}) \land Y = Y_{q+1})]
\]

where \( q > 2 \).

By folding using \( S_{r,tdl} \), and renaming:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r_{tdl}(Xs, Y, A) \Leftrightarrow \\
\lor (Xs = [X] \land Y = A) \\
\lor (Xs = [X][TXs] \land \mathcal{O}_c(X, HY) \land \mathcal{O}_c(A, HY, NA) \land r_{tdl}(TXs, Y, NA))]
\]

By taking the ‘decomposition’:

- **clause 1:** \( r_{tdl}(Xs, Y, A) \to \\
\quad Xs = [], Y = A \\
- **clause 2:** \( r_{tdl}(Xs, Y, A) \to \\
\quad Xs = [X][TXs], r(X, HY), compose(A, HY, NA), r_{tdl}(TXs, Y, NA) \)
By unfolding clause 2 wrt $r(X, HY)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt $S_r$ in $S$:

**clause 3**: $r.d_1(Xs, Y, A) \leftarrow$
- $Xs = [X][TXs]$
- $\text{minimal}(X)$
- $\text{solve}(X, HY), \text{compose}(A, HY, NA)$
- $r.d_1(TXs, Y, NA)$

**clause 4**: $r.d_1(Xs, Y, A) \leftarrow$
- $Xs = [X][TXs]$
- $\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t)$
- $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$
- $I_0 = e$
- $\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1})$
- $\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p)$
- $\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1})$
- $HY = I_{t+1}, \text{compose}(A, HY, NA)$
- $r.d_1(TXs, Y, NA)$

By introducing

$$
[\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_t)] \lor
((\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t)))
\lor
((\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{minimal}(TX_p) \land \ldots \land \text{minimal}(TX_t)))
\lor
((\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t)))
$$

in clause 4, using applicability condition (4):

**clause 5**: $r.d_1(Xs, Y, A) \leftarrow$
- $Xs = [X][TXs]$
- $\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t)$
- $\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t)$
- $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$
- $I_0 = e$
- $\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1})$
- $\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p)$
- $\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1})$
- $HY = I_{t+1}, \text{compose}(A, HY, NA)$
- $r.d_1(TXs, Y, NA)$

**clause 6**: $r.d_1(Xs, Y, A) \leftarrow$
- $Xs = [X][TXs]$
- $\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t)$
- $\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1})$
- $\text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t)$
- $r(TX_1, TY_1), \ldots, r(TX_t, TY_t)$
- $I_0 = e$
- $\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1})$
- $\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p)$
- $\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1})$
- $HY = I_{t+1}, \text{compose}(A, HY, NA)$
- $r.d_1(TXs, Y, NA)$
clause 7:  \( r \cdot t d_1(X, Y, A) \leftarrow \)
\[
X_s = [X|TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\},
\]
\[
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_0 = e,
\]
\[
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),
\]
\[
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}),
\]
\[
HY = I_{t+1}, \text{compose}(A, HY, NA),
\]
\[
r \cdot t d_1(TX_s, Y, NA)
\]

clause 8:  \( r \cdot t d_1(X, Y, A) \leftarrow \)
\[
X_s = [X|TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\},
\]
\[
\{\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)\},
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_0 = e,
\]
\[
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),
\]
\[
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}),
\]
\[
HY = I_{t+1}, \text{compose}(A, HY, NA),
\]
\[
r \cdot t d_1(TX_s, Y, NA)
\]

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):

clause 9:  \( r \cdot t d_1(X, Y, A) \leftarrow \)
\[
X_s = [X|TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t),
\]
\[
solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t),
\]
\[
l_0 = e,
\]
\[
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),
\]
\[
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}),
\]
\[
HY = I_{t+1}, \text{compose}(A, HY, NA),
\]
\[
r \cdot t d_1(TX_s, Y, NA)
\]

By using applicability condition (3):

clause 10:  \( r \cdot t d_1(X, Y, A) \leftarrow \)
\[
X_s = [X|TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t),
\]
\[
solve(TX_1, e), \ldots, solve(TX_t, e),
\]
\[
l_0 = e,
\]
\[
\text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}),
\]
\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),
\]
\[
\text{compose}(I_p, e, I_{p+1}), \ldots, \text{compose}(I_t, e, I_{t+1}),
\]
\[
HY = I_{t+1}, \text{compose}(A, HY, NA),
\]
\[
r \cdot t d_1(TX_s, Y, NA)
\]

By deleting one of the \( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t) \) atoms in clause 10:
clause 11:  \( r tersdy(Xs, Y, A) \leftarrow \)
\( Xs = [X \cup TXs] \),
\( nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt) \),
\( minimal(TX1), \ldots, minimal(TXt) \),
\( solve(TX1, e), \ldots, solve(TXt, e) \),
\( I0 = e \),
\( compose(I0, e, I1), \ldots, compose(I_{p-2}, e, I_{p-1}) \),
\( process(HX, HHY), compose(I_{p-1}, HHY, Ip) \),
\( compose(Ip, e, I_{p+1}), \ldots, compose(I_t, e, I_{t+1}) \).
\( HY = I_{t+1}, compose(A, HYY, NA) \).
\( r tersdy(TXs, Y, NA) \). 

By using applicability condition (2):

clause 12:  \( r tersdy(Xs, Y, A) \leftarrow \)
\( Xs = [X \cup TXs] \),
\( nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt) \),
\( minimal(TX1), \ldots, minimal(TXt) \),
\( solve(TX1, e), \ldots, solve(TXt, e) \),
\( I0 = e \),
\( I1 = I0, \ldots, Ip-1 = Ip-2 \),
\( process(HX, HHY), compose(I_{p-1}, HHY, Ip) \),
\( Ip+1 = Ip, \ldots, It+1 = It \),
\( HY = It+1, compose(A, HYY, NA) \).
\( r tersdy(TXs, Y, NA) \). 

By simplification:

clause 13:  \( r tersdy(Xs, Y, A) \leftarrow \)
\( Xs = [X \cup TXs] \),
\( nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt) \),
\( minimal(TX1), \ldots, minimal(TXt) \),
\( process(HX, HYY), compose(A, HYY, NA) \).
\( r tersdy(TXs, Y, NA) \). 

By \( p-1 \) times unfolding clause 6 wrt \( r(TX1, TY1), \ldots, r(TX_{p-1}, TY_{p-1}) \) using DCLR, and simplifying using condition (4):

clause 14:  \( r tersdy(Xs, Y, A) \leftarrow \)
\( Xs = [X \cup TXs] \),
\( nonMinimal(X), decompose(X, HX, TX1, \ldots, TXt) \),
\( minimal(TX1), \ldots, minimal(TXp-1) \),
\( nonMinimal(TXp) \),
\( minimal(TX1), \ldots, minimal(TXp-1) \),
\( solve(TX1, TY1), \ldots, solve(TX_{p-1}, TY_{p-1}) \),
\( r(TXp, TYp), \ldots, r(TXt, TYt) \),
\( I0 = e \),
\( compose(I0, TY1, I1), \ldots, compose(I_{p-2}, TY_{p-1}, Ip-1) \),
\( process(HX, HYY), compose(I_{p-1}, HHY, Ip) \),
\( compose(Ip, TYp, Ip+1), \ldots, compose(It, TYt, It+1) \).
\( HY = It+1, compose(A, HYY, NA) \).
\( r tersdy(TXs, Y, NA) \). 

By deleting one of the minimal\((TX1), \ldots, minimal(TXp-1) \) atoms in clause 14:
clause 15: \[r.td_1(X_s, Y, A) \leftarrow\]
\[X_s = [X[TXs]].\]
\[\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i).\]
\[\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).\]
\[(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i)),\]
\[\text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}),\]
\[r(TX_p, TY_p), \ldots, r(TX_i, TY_i).\]
\[I_0 = e,\]
\[\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),\]
\[\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p),\]
\[\text{compose}(I_p, TY_i, I_{p+1}), \ldots, \text{compose}(I_{t-1}, TY_{i+1}).\]
\[HY = I_{t+1}, \text{compose}(A, HY, NA).\]
\[r.td_1([TXs], Y, NA)\]

By rewriting clause 15 using applicability condition (1):

clause 16: \[r.td_1(X_s, Y, A) \leftarrow\]
\[X_s = [X[TXs]].\]
\[\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i).\]
\[\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).\]
\[(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i)),\]
\[\text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}),\]
\[r(TX_p, TY_p), \ldots, r(TX_i, TY_i).\]
\[I_0 = e,\]
\[\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),\]
\[\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), HY = I_p,\]
\[\text{compose}(A, HY, NA),\]
\[\text{compose}(TY_p, TY_{p+1}, I_{p+1}),\]
\[\text{compose}(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, \text{compose}(I_{t-1}, TY_{i+1}).\]
\[\text{compose}(NA, I_t, NA),\]
\[r.td_1([TXs], Y, NNA)\]

By \(t-p\) times folding clause 16 using clauses 1 and 2:

clause 17: \[r.td_1(X_s, Y, A) \leftarrow\]
\[X_s = [X[TXs]].\]
\[\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i).\]
\[\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).\]
\[(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i)),\]
\[\text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}),\]
\[I_0 = e,\]
\[\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),\]
\[\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), HY = I_p,\]
\[\text{compose}(A, HY, NA),\]
\[r.td_1([TXs], Y, NA)\]

By using applicability condition (3):

clause 18: \[r.td_1(X_s, Y, A) \leftarrow\]
\[X_s = [X[TXs]].\]
\[\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i).\]
\[\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).\]
\[(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i)),\]
\[\text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e),\]
\[I_0 = e,\]
\[\text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}),\]
\[\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), HY = I_p,\]
\[\text{compose}(A, HY, NA),\]
\[r.td_1([TXs], Y, NA)\]

By using applicability condition (2):
clause 19: \( r_1.d_1(Xs,Y,A) \)
\[
Xs = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_1), \\
minimal(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_1)), \\
solve(TX_1, \varepsilon), \ldots, \varepsilon, \text{solve}(TX_{p-1}, \varepsilon), \\
l_0 = \varepsilon, \\
l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \\
\text{process}(HX, HGY), \text{compose}(l_{p-1}, HGY, I_p), HY = I_p, \\
\text{compose}(A, HGY, NA), \\
r_1.d_1([TX_p, \ldots, TX_1[TXs], Y, NA])
\]

By simplification:

clause 20: \( r_1.d_1(Xs,Y,A) \)
\[
Xs = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_1), \\
minimal(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_1)), \\
\text{process}(HX, HGY), \text{compose}(A, HGY, NA), \\
r_1.d_1([TX_p, \ldots, TX_1[TXs], Y, NA])
\]

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

clause 21: \( r_1.d_1(Xs,Y,A) \)
\[
Xs = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_1), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
r(TX_1, TY_1), \ldots, r(TX_1, TY_1), \\
l_0 = \varepsilon, \\
\text{compose}(l_0, TY_1, I_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
\text{process}(HX, HGY), \text{compose}(l_{p-1}, HGY, I_p), \\
\text{compose}(l_p, TY_p, I_{p+1}), \ldots, \text{compose}(l_{p+1}, TY_{p+1}, I_{p+1}), HGY = I_{p+1}, \text{compose}(A, HGY, NA), \\
r_1.d_1([TXs, Y, NA])
\]

By using applicability condition (3):

clause 22: \( r_1.d_1(Xs,Y,A) \)
\[
Xs = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_1), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
r(U_1, \varepsilon), \ldots, r(U_{p-1}, \varepsilon), \\
r(TX_1, TY_1), \ldots, r(TX_1, TY_1), \\
l_0 = \varepsilon, \\
\text{compose}(l_0, TY_1, I_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
\text{process}(HX, HGY), \text{compose}(l_{p-1}, HGY, I_p), \\
\text{compose}(l_p, TY_p, I_{p+1}), \ldots, \text{compose}(l_{p+1}, TY_{p+1}, I_{p+1}), HGY = I_{p+1}, \text{compose}(A, HGY, NA), \\
r_1.d_1([TXs, Y, NA])
\]

By using applicability condition (2):
clause 23:  \( r \cdot td_1(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, e), \ldots, r(U_{p-1}, e). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( l_0 = e. \)
\( \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \)
\( \text{compose}(I_{p-1}, e, K_1), \text{compose}(K_1, e, K_2), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}). \)
\( \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, I_p). \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}). \)
\( HY = I_{t+1}; \text{compose}(A, HHY, NA). \)
\( r \cdot td_1(TXs, Y, NA). \)

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

clause 24:  \( r \cdot td_1(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, e), \ldots, r(U_{p-1}, e). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(A, K_{p-2}, NA). \)
\( l_0 = e. \)
\( \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}). \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}). \)
\( HY = I_{t+1}; \text{compose}(NA, HHY, NNA). \)
\( r \cdot td_1(TXs, Y, NNA). \)

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

clause 25:  \( r \cdot td_1(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(A, K_{p-2}, NA). \)
\( l_0 = e. \)
\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}). \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p). \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}). \)
\( HY = I_{t+1}; \text{compose}(NA, HHY, NNA). \)
\( r \cdot td_1(TXs, Y, NNA). \)

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

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

always holds (because \( N \) is existentially quantified)
clause 26: \[ r \cdot d_1 (X, Y, A) \leftarrow \]
\[ X_s = [X] \cdot T \cdot X_s. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}). \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(A, K_{p-2}, NA). \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}). \]
\[ \text{compose}(H, HHY, Y, H, K_1), \text{compose}(H, HHY, I_p). \]
\[ \text{compose}(I_0, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}). \]
\[ HY = I_{t+1}, \text{compose}(NA, HY, NNA). \]
\[ r \cdot d_1 (TX_s, Y, NNA). \]

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

clause 27: \[ r \cdot d_1 (X, Y, A) \leftarrow \]
\[ X_s = [X] \cdot T \cdot X_s. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}). \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(A, K_{p-2}, NA). \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}). \]
\[ \text{compose}(H, HHY, Y, H, K_1), \text{compose}(H, HHY, I_p). \]
\[ \text{compose}(I_0, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}). \]
\[ HY = I_{t+1}, \text{compose}(NA, HY, NNA). \]
\[ r \cdot d_1 (TX_s, Y, NNA). \]

By folding clause 27 using \( DCLR \):

clause 28: \[ r \cdot d_1 (X, Y, A) \leftarrow \]
\[ X_s = [X] \cdot T \cdot X_s. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ \text{compose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \]
\[ r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}). \]
\[ r(N, HY). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, HY, NNA). \]
\[ r \cdot d_1 (TX_s, Y, NNA). \]

By folding clause 28 using clauses 1 and 2:
clause 29: \( r.td_1(X_s, Y, A) \leftarrow \)
\[ X_s = [X|TX_s], \]
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),
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \)
\( compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( compose(A, K_{p-2}, NA), \)
\( r.td_1([N|TX_s], Y, NA) \)

By \( p - 1 \) times folding clause 29 using clauses 1 and 2:

clause 30: \( r.td_1(X_s, Y, A) \leftarrow \)
\[ X_s = [X|TX_s], \]
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),
\( r.td_1([TX_1, \ldots, TX_{p-1}, N|TX_s], Y, A) \)

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: \( r.td_1(X_s, Y, A) \leftarrow \)
\[ X_s = [X|TX_s], \]
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),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( \lambda = e, \)
\( compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \)
\( process(HX, HHY), compose(I_{p-1}, HHY, I_p), \)
\( compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \)
\( HY = I_{t+1}, compose(A, HY, NA) \)
\( r.td_1([TX_s, Y, NA) \]

By using applicability condition (3):

clause 32: \( r.td_1(X_s, Y, A) \leftarrow \)
\[ X_s = [X|TX_s], \]
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),
\( r(U_1, e), \ldots, r(U_t, e), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( \lambda = e, \)
\( compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \)
\( process(HX, HHY), compose(I_{p-1}, HHY, I_p), \)
\( compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_t, TY_t, I_{t+1}), \)
\( HY = I_{t+1}, compose(A, HY, NA) \)
\( r.td_1([TX_s, Y, NA) \]

By using applicability condition (2):
\[ r_{td}(X_s, Y, A) = \]
\[ X_s = [X | T X_s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i), \]
\[ \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}) \]
\[ \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_{i+1}) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \]
\[ r(U_1, e), \ldots, r(U_i, e) \]
\[ r(TX_1, TY_1), \ldots, r(TX_i, TY_i) \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}) \]
\[ \text{compose}(I_{p-1}, e, K_1), \text{compose}(K_1, e, K_2), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}) \]
\[ \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, K_p) \]
\[ \text{compose}(K_p, e, K_{p+1}), \ldots, \text{compose}(K_{i-1}, e, K_i), \text{compose}(K_{i+1}, e, I_p) \]
\[ \text{compose}(I_p, TY_i, I_{p+1}), \ldots, \text{compose}(I_i, TY_i, I_{i+1}) \]
\[ HY = I_{i+1}, \text{compose}(A, HY, NA) \]
\[ r_{td}(TX_s, Y, NA) \]

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

\[ r_{td}(X_s, Y, A) = \]
\[ X_s = [X | T X_s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i), \]
\[ \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}) \]
\[ \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_{i+1}) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \]
\[ r(U_1, e), \ldots, r(U_i, e) \]
\[ r(TX_1, TY_1), \ldots, r(TX_i, TY_i) \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}) \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \]
\[ \text{compose}(I_p, e, I_{p+1}), \ldots, \text{compose}(I_i, e, I_{i+1}) \]
\[ NH Y = I_{i+1}, \]
\[ \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{i-2}, TY_i, K_{i-1}) \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NH Y, NA_1) \]
\[ \text{compose}(NA_1, K_{i-1}, NA_2), r_{td}(TX_s, Y, NA_2) \]

By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_i \) in place of some occurrences of \( e \):

\[ r_{td}(X_s, Y, A) = \]
\[ X_s = [X | T X_s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i), \]
\[ \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}) \]
\[ \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_{i+1}) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \]
\[ r(U_1, YU_1), \ldots, r(U_i, YU_i) \]
\[ r(TX_1, TY_1), \ldots, r(TX_i, TY_i) \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}) \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \]
\[ \text{compose}(I_p, YU_p, I_{p+1}), \ldots, \text{compose}(I_i, YU_i, I_{i+1}) \]
\[ NH Y = I_{i+1}, \]
\[ \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{i-2}, TY_i, K_{i-1}) \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NH Y, NA_1) \]
\[ \text{compose}(NA_1, K_{i-1}, NA_2), r_{td}(TX_s, Y, NA_2) \]

By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N, HX, U_1, \ldots, U_i) \), since

\[ \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_i) \]
always holds (because $N$ is existentially quantified)

**Clause 36**: \( r.\text{ld}_1(X, Y, A) \leftarrow \)

\( X = [X|TXs] \)

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)

\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)

\( \text{minimal}(U_1); \ldots, \text{minimal}(U_i) \)

\( r(U_1, YU_1), \ldots, r(U_t, YU_t) \)

\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t) \)

\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)

\( I_0 = e, \)

\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}) \)

\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \)

\( \text{compose}(I_p, YU_p, I_{p+1}), \ldots, \text{compose}(I_t, YU_t, I_{t+1}) \)

\( NHY = I_{t+1} \)

\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}) \)

\( \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NHY, NA_1) \)

\( \text{compose}(NA_1, K_{t-1}, NA_2), \text{r.\text{ld}_1}(TXs, Y, NA_2) \)

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_t) \):

**Clause 37**: \( r.\text{ld}_1(X, Y, A) \leftarrow \)

\( X = [X|TXs] \)

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)

\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)

\( \text{minimal}(U_1); \ldots, \text{minimal}(U_i) \)

\( r(U_1, YU_1), \ldots, r(U_t, YU_t) \)

\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t) \)

\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)

\( I_0 = e, \)

\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}) \)

\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p) \)

\( \text{compose}(I_p, YU_p, I_{p+1}), \ldots, \text{compose}(I_t, YU_t, I_{t+1}) \)

\( NHY = I_{t+1} \)

\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}) \)

\( \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NHY, NA_1) \)

\( \text{compose}(NA_1, K_{t-1}, NA_2), \text{r.\text{ld}_1}(TXs, Y, NA_2) \)

By folding clause 37 using \( DCLR \):

**Clause 38**: \( r.\text{ld}_1(X, Y, A) \leftarrow \)

\( X = [X|TXs] \)

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)

\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)

\( \text{minimal}(U_1); \ldots, \text{minimal}(U_i) \)

\( \text{compose}(N, HX, U_1, \ldots, U_t) \)

\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)

\( r(N, NHY), \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)

\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}) \)

\( \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NHY, NA_1) \)

\( \text{compose}(NA_1, K_{t-1}, NA_2), \text{r.\text{ld}_1}(TXs, Y, NA_2) \)

By \( t - p + 1 \) times folding clause 38 using clauses 1 and 2:
clause 39: \( r.td_1(X_5, Y, A) \rightarrow \)
\( X_5 = [X][T][X_5] \),
\( \text{nonMinimal}(X), \) \( \text{decompose}(X, H, X_1, \ldots, X_{i-1}) \),
\( (\text{nonMinimal}(X_1); \ldots; \text{nonMinimal}(X_{i-1})) \),
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_i) \),
\( \text{decompose}(N, H, X, U_1, \ldots, U_i) \),
\( r(TX_1, TY_1), \ldots, r(TX_{i-1}, TY_{i-1}), r(N, NH, Y) \),
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TX_3, K_2), \ldots, \text{compose}(K_{i-2}, TY_{i-1}, K_{i-1}) \),
\( \text{compose}(A, K_{i-1}, NA), \) \( \text{compose}(NA, NH, NA_1) \),
\( r.td_1([TX_5, \ldots, TX_i][T][X_5], Y, NA_1) \)

By folding clause 39 using clauses 1 and 2:

clause 40: \( r.td_1(X_5, Y, A) \rightarrow \)
\( X_5 = [X][T][X_5] \),
\( \text{nonMinimal}(X), \) \( \text{decompose}(X, H, X_1, \ldots, X_{i-1}) \),
\( (\text{nonMinimal}(X_1); \ldots; \text{nonMinimal}(X_{i-1})) \),
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_i) \),
\( \text{decompose}(N, H, X, U_1, \ldots, U_i) \),
\( r(TX_1, TY_1), \ldots, r(TX_{i-1}, TY_{i-1}) \),
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{i-2}, TY_{i-1}, K_{i-1}) \),
\( \text{compose}(A, K_{i-1}, NA) \),
\( r.td_1([N, TX_5, \ldots, TX_i][T][X_5], Y, NA) \)

By \( p - 1 \) times folding clause 40 using clauses 1 and 2:

clause 41: \( r.td_1(X_5, Y, A) \rightarrow \)
\( X_5 = [X][T][X_5] \),
\( \text{nonMinimal}(X), \) \( \text{decompose}(X, H, X_1, \ldots, X_{i-1}) \),
\( (\text{nonMinimal}(X_1); \ldots; \text{nonMinimal}(X_{i-1})) \),
\( \text{minimal}(U_1); \ldots; \text{minimal}(U_i) \),
\( \text{decompose}(N, H, X, U_1, \ldots, U_i) \),
\( r.td_1([TX_5, \ldots, TX_i][T][X_5], Y, A) \)

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

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:

\( r(X, Y) \leftarrow r.td_1([X], Y, e) \)

By taking the ‘completion’:

\( \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r.td_1([X], Y, e)] \)

By unfolding the ‘completion’ above wrt \( r.td_1([X], Y, e) \) using \( S_{r.td} \):

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

By using applicability condition (2):

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

By simplification:

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

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 \( S \).
Theorem 8 The generalization schema $TDG_2$, which is given below, is correct.

$TDG_2 : \{ DCLR, TDGLR, A_{td2}, O_{td212}, O_{td221} \}$ where

- $A_{td2}$: (1) compose is associative
  (2) compose has $\epsilon$ as the left and right identity element
  (3) $\forall X : X. \tau(X) \land \text{minimal}(X) \Rightarrow \text{O}(X, \epsilon)$
  (4) $\forall X : X. \tau(X) \Rightarrow [-\text{minimal}(X) \iff \text{nonMinimal}(X)]$

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

- $O_{td221}$: partial evaluation of the conjunction
  $\text{process}(HX, HY), \text{compose}(Ip-1, HY, Ip)$
  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 7 below.

Logic Program Template 7

\begin{align*}
\text{r}(X, Y) & :- \\
& \quad \text{r}\_td2([X], Y, e) \\
\text{r}\_td2(Xs, Y, A) & :- \\
& \quad Xs = [], \\
& \quad Y = A \\
\text{r}\_td2(Xs, Y, A) & :- \\
& \quad Xs = [X]\_TXs, \\
& \quad \text{minimal}(X), \\
& \quad \text{r}\_td2(TXs, NewA, A), \\
& \quad \text{solve}(X, HY), \\
& \quad \text{compose}(HY, NewA, Y) \\
\text{r}\_td2(Xs, Y, A) & :- \\
& \quad Xs = [X]\_TXs, \\
& \quad \text{nonMinimal}(X), \\
& \quad \text{decompose}(X,HX,TX_1,\ldots,TX_t), \\
& \quad \text{minimal}(TX_1),\ldots,\text{minimal}(TX_t), \\
& \quad \text{r}\_td2(TXs, NewA, A), \\
& \quad \text{process}(HX, HY), \text{compose}(HY, NewA, Y) \\
\text{r}\_td2(Xs, Y, A) & :- \\
& \quad Xs = [X]\_TXs, \\
& \quad \text{nonMinimal}(X), \\
& \quad \text{decompose}(X,HX,TX_1,\ldots,TX_t), \\
& \quad \text{minimal}(TX_1),\ldots,\text{minimal}(TX_{p-1}), \\
& \quad (\text{nonMinimal}(TX_p);\ldots;\text{nonMinimal}(TX_t)), \\
& \quad \text{r}\_td2([TX_p,\ldots,TX_t]\_TXs, NewA, A), \\
& \quad \text{process}(HX, HY), \text{compose}(HY, NewA, Y) \\
\text{r}\_td2(Xs, Y, A) & :- \\
& \quad Xs = [X]\_TXs, \\
& \quad \text{nonMinimal}(X), \\
& \quad \text{decompose}(X,HX,TX_1,\ldots,TX_t), \\
& \quad (\text{nonMinimal}(TX_1);\ldots;\text{nonMinimal}(TX_{p-1})), \\
& \quad \text{minimal}(TX_p),\ldots,\text{minimal}(TX_t), \\
& \quad \text{minimal}(U_1),\ldots,\text{minimal}(U_{p-1}), \\
\end{align*}
\[
\text{decompose}(N, \mathbf{H}, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
r_{td2}([TX_1, \ldots, TX_{p-1}, N|TXs], Y, A) \\
r_{td2}(X, Y, A) \leftarrow \\
\begin{align*}
&Xs = [X|TXs], \\
&\text{nonMinimal}(X), \\
&\text{decompose}(X, \mathbf{H}, TX_1, \ldots, TX_t), \\
&(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
&(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
&\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
&\text{decompose}(N, \mathbf{H}, U_1, \ldots, U_t), \\
&r_{td2}([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t|TXs], Y, A)
\end{align*}
\]

and the specification \(S_r\) of relation \(r\) is:

\[
\forall X : \mathcal{X}, \forall Y : \mathcal{Y}, \quad I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow O_r(X, Y)]
\]

and the specification of \(r_{td2}\), namely \(S_{r_{td2}}\), is:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}, \forall A : \mathcal{Y}, \quad (X(Y : \mathcal{X}, X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{td2}(Xs, Y, A) \Leftrightarrow (Xs = [] \land Y = A) \\
\forall (Xs = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{q} O_r(I_{i-1}, Y_i, I_i) \\
\land O_r(I_q, A, I_{q+1}) \land Y = I_{q+1}])]
\]

**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 \(S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}\), where \(S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\) are the specifications of \(\text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose}\), which are all the undefined relation names appearing in \(DCLR\).

(b) \(TDGRL\) is also steadfast wrt \(S_r\) in \(S\).

Note that the sets \(\{S_1, \ldots, S_m\}\) and \(\{S_1', \ldots, S'_l\}\) in Definition 4 are equal to \(S\) 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 \(S\), 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 \(S\) if \(P_{r_{td2}}\) is steadfast wrt \(S_{r_{td2}}\) in \(S\), where \(P_{r_{td2}}\) is the procedure for \(r_{td2}\), and \(P_r\) is steadfast wrt \(S_r\) in \(\{S_{r_{td2}}\}\), where \(P_r\) is the procedure for \(r\).

To prove that \(P_{r_{td2}}\) is steadfast wrt \(S_{r_{td2}}\) in \(S\), we do a constructive forward proof that we begin with \(S_{r_{td2}}\), and from which we try to obtain \(P_{r_{td2}}\).

If we separate the cases of \(q \geq 1\) by \(q = 1 \lor q \geq 2\), then \(S_{r_{td2}}\) becomes:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}, \quad (X(Y : \mathcal{X}, X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{td2}(Xs, Y, A) \Leftrightarrow \\
\forall (Xs = [] \land Y = A) \\
\forall (Xs = [X_1, X_2, \ldots, X_q] \land Y_1 = I_1 \land O_r(I_1, A, I_2) \land Y = I_2) \\
\forall (Xs = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{q} O_r(I_{i-1}, Y_i, I_i) \\
\land O_r(I_q, A, I_{q+1}) \land Y = I_{q+1}])
\]

where \(q \geq 2\).

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

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}, \quad (X(Y : \mathcal{X}, X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{td2}(Xs, Y, A) \Leftrightarrow \\
\forall (Xs = [] \land Y = A) \\
\forall (Xs = [X_1|TXs] \land TXs = [] \land O_r(X_1, Y_1) \land I_1 = Y_1 \land TY = A \land O_r(TY, A, NA) \land O_r(I_1, NA, Y)) \\
\forall (Xs = [X_1|TXs] \land TXs = [X_2, \ldots, X_q] \land \bigwedge_{i=1}^{q} O_r(X_i, Y_i) \land Y_1 = I_1 \land Y_2 = I_2 \land \\
\bigwedge_{i=2}^{q} O_r(I_{i-1}, Y_i, I_i) \land TY = I_q \land O_r(TY, A, NA) \land O_r(I_1, NA, Y))]
\]

54
where \( q \geq 2 \).

By folding using \( S_{r,t_{d_{2}}} \), and renaming:

\[
\forall Xs : \text{list of } \mathcal{X}, YY : \mathcal{Y}. (XY : \mathcal{X}. X \in Xs \Rightarrow \mathcal{Y}(X)) \Rightarrow [r_{td_{2}}(Xs, Y, A) \Leftrightarrow \\
\{ Xs = [] \land Y = A \} \land \\
\forall (Xs = [X[TXs] \land \mathcal{O}_{r}(X, HY) \land r_{td_{2}}(TXs, NA, A) \land \mathcal{O}_{r}(HY, NA, Y))]\]

By taking the ‘decompletion’:

\text{clause 1: } r_{td_{2}}(Xs, Y, A) \leftarrow \\
\quad Xs = [], Y = A

\text{clause 2: } r_{td_{2}}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], r(X, HY). \\
\quad r_{td_{2}}(TXs, NA, A), \text{compose}(HY, NA, Y)]

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

\text{clause 3: } r_{td_{2}}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, HY), \\
\quad r_{td_{2}}(TXs, NA, A), \text{compose}(HY, NA, Y)]

\text{clause 4: } r_{td_{2}}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TXt_{1}, \ldots, TXt_{t}), \\
\quad r(TX_{1}, TY_{1}), \ldots, r(TX_{t}, TY_{t}), \\
\quad l_{0} = e, \\
\quad \text{compose}(I_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\quad \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_{p}), \\
\quad \text{compose}(I_{p}, TY_{p}, I_{p+1}), \ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}), \\
\quad HY = I_{t+1}, \\
\quad r_{td_{2}}(TXs, NA, A), \text{compose}(HY, NA, Y)]

By introducing

\[
\begin{align*}
&\{\text{minimal}(TX_{1}) \land \ldots \land \text{minimal}(TX_{t})\} \lor \\
&((\text{minimal}(TX_{1}) \land \ldots \land \text{minimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_{p}) \lor \ldots \lor \text{nonMinimal}(TX_{t})) \lor \\
&((\text{nonMinimal}(TX_{1}) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{minimal}(TX_{p}) \land \ldots \land \text{minimal}(TX_{t})) \lor \\
&\{\text{nonMinimal}(TX_{1}) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})\} \land (\text{nonMinimal}(TX_{p}) \lor \ldots \lor \text{nonMinimal}(TX_{t}))\}
\end{align*}
\]

in clause 4, using applicability condition (4):

\text{clause 5: } r_{td_{2}}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TXt_{1}, \ldots, TXt_{t}), \\
\quad \text{minimal}(TX_{1}), \ldots, \text{minimal}(TX_{t}), \\
\quad r(TX_{1}, TY_{1}), \ldots, r(TX_{t}, TY_{t}), \\
\quad l_{0} = e, \\
\quad \text{compose}(I_{0}, TY_{1}, I_{1}), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\quad \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_{p}), \\
\quad \text{compose}(I_{p}, TY_{p}, I_{p+1}), \ldots, \text{compose}(I_{t}, TY_{t}, I_{t+1}), \\
\quad HY = I_{t+1}, \\
\quad r_{td_{2}}(TXs, NA, A), \text{compose}(HY, NA, Y)\]
clause 6: \( rld_2(Xs, Y, A) \leftarrow \)
\[
\begin{align*}
Xs &= [X][TXs], \\
\nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \\
\{\nonMinimal(TX_1) ; \ldots \nonMinimal(TX_t)\}.
\end{align*}
\]
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}.
\]
\( rld_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

clause 7: \( rld_2(Xs, Y, A) \leftarrow \)
\[
\begin{align*}
Xs &= [X][TXs], \\
\nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\nonMinimal(TX_1) ; \ldots \nonMinimal(TX_{p-1})\}.
\end{align*}
\]
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}.
\]
\( rld_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

clause 8: \( rld_2(Xs, Y, A) \leftarrow \)
\[
\begin{align*}
Xs &= [X][TXs], \\
\nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\nonMinimal(TX_1) ; \ldots \nonMinimal(TX_{p-1})\}.
\end{align*}
\]
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_0 = e, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}.
\]
\( rld_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

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):

clause 9: \( rld_2(Xs, Y, A) \leftarrow \)
\[
\begin{align*}
Xs &= [X][TXs], \\
\nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
solve(TX_1, TY_1), \ldots, \text{solve}(TX_t, TY_t), \\
I_0 = e, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}). \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}.
\]
\( rld_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):
clause 10: \( r \cdot t \cdot d_2(X_s, Y, A) \)
\[
X_s = [X][T X_s].
\]
\( \nonMinimal(X) \), \( \text{decompose}(X, H X, T X_1, \ldots, T X_t) \),
\( \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) \),
\( \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e) \),
\( l_0 = e \),
\( \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}) \),
\( \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p) \),
\( \text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}) \),
\( H Y = l_{t+1} \),
\( r \cdot t \cdot d_2(T X_s, N A, A), \text{compose}(H Y, N A, Y) \)

By deleting one of the \( \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) \) atoms in clause 10:

clause 11: \( r \cdot t \cdot d_2(X_s, Y, A) \)
\[
X_s = [X][T X_s].
\]
\( \nonMinimal(X) \), \( \text{decompose}(X, H X, T X_1, \ldots, T X_t) \),
\( \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) \),
\( \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e) \),
\( l_0 = e \),
\( \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}) \),
\( \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p) \),
\( \text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}) \),
\( H Y = l_{t+1} \),
\( r \cdot t \cdot d_2(T X_s, N A, A), \text{compose}(H Y, N A, Y) \)

By using applicability condition (2):

clause 12: \( r \cdot t \cdot d_2(X_s, Y, A) \)
\[
X_s = [X][T X_s].
\]
\( \nonMinimal(X) \), \( \text{decompose}(X, H X, T X_1, \ldots, T X_t) \),
\( \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) \),
\( \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e) \),
\( l_0 = e \),
\( l_1 = l_0 \), \( \ldots, l_{p-1} = l_{p-2} \),
\( \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p) \),
\( l_{p+1} = l_p \), \( \ldots, l_{t+1} = l_t \),
\( H Y = l_{t+1} \),
\( r \cdot t \cdot d_2(T X_s, N A, A), \text{compose}(H Y, N A, Y) \)

By simplification:

clause 13: \( r \cdot t \cdot d_2(X_s, Y, A) \)
\[
X_s = [X][T X_s].
\]
\( \nonMinimal(X) \), \( \text{decompose}(X, H X, T X_1, \ldots, T X_t) \),
\( \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) \),
\( r \cdot t \cdot d_2(T X_s, N A, A) \),
\( \text{process}(H X, H Y), \text{compose}(H Y, N A, Y) \)

By \( p-1 \) times unfolding clause 6 wrt \( r(T X_1, T Y_1), \ldots, r(T X_{p-1}, T Y_{p-1}) \) using DCLR, and simplifying using condition (4):
clause 14: \[ r.td_2(Xs, Y, A) \rightarrow\]
\[ Xs = [X|TXs], \]
\[ \text{nonMinimal}\langle X \rangle, \text{decompose}\langle X, HX, TX_1, \ldots, TX_t \rangle, \]
\[ \text{minimal}\langle TX_1 \rangle, \ldots, \text{minimal}\langle TX_{p-1} \rangle. \]
\[ (\text{nonMinimal}\langle TX_p \rangle; \ldots; \text{nonMinimal}\langle TX_t \rangle), \]
\[ \text{minimal}\langle TX_1 \rangle, \ldots, \text{minimal}\langle TX_{p-1} \rangle. \]
\[ \text{solve}\langle TX_1, TY_1 \rangle, \ldots, \text{solve}\langle TX_{p-1}, TY_{p-1} \rangle, \]
\[ r(TX_p, TY_p), \ldots, r(TX_1, TY_1). \]
\[ l_0 = e, \]
\[ \text{compose}\langle I_0, TY_1, I_1 \rangle, \ldots, \text{compose}\langle I_{p-2}, TY_{p-1}, I_{p-1} \rangle, \]
\[ \text{process}\langle HX, HHY \rangle, \text{compose}\langle I_{p-1}, HHY, I_p \rangle, \]
\[ \text{compose}\langle I_p, TY_p, I_{p+1} \rangle, \ldots, \text{compose}\langle I_t, TY_t, I_{t+1} \rangle. \]
\[ HY = I_{t+1}; \]
\[ r.td_2(TXs, NA, A), \text{compose}\langle HY, NA, Y \rangle. \]

By deleting one of the \text{minimal}\langle TX_1 \rangle, \ldots, \text{minimal}\langle TX_{p-1} \rangle atoms in clause 14:

clause 15: \[ r.td_2(Xs, Y, A) \rightarrow\]
\[ Xs = [X|TXs], \]
\[ \text{nonMinimal}\langle X \rangle, \text{decompose}\langle X, HX, TX_1, \ldots, TX_t \rangle, \]
\[ \text{minimal}\langle TX_1 \rangle, \ldots, \text{minimal}\langle TX_{p-1} \rangle. \]
\[ (\text{nonMinimal}\langle TX_p \rangle; \ldots; \text{nonMinimal}\langle TX_t \rangle), \]
\[ \text{solve}\langle TX_1, TY_1 \rangle, \ldots, \text{solve}\langle TX_{p-1}, TY_{p-1} \rangle, \]
\[ r(TX_p, TY_p), \ldots, r(TX_1, TY_1). \]
\[ l_0 = e, \]
\[ \text{compose}\langle I_0, TY_1, I_1 \rangle, \ldots, \text{compose}\langle I_{p-2}, TY_{p-1}, I_{p-1} \rangle, \]
\[ \text{process}\langle HX, HHY \rangle, \text{compose}\langle I_{p-1}, HHY, I_p \rangle, \]
\[ \text{compose}\langle I_p, TY_p, I_{p+1} \rangle, \ldots, \text{compose}\langle I_t, TY_t, I_{t+1} \rangle. \]
\[ HY = I_{t+1}; \]
\[ r.td_2(TXs, NA, A), \text{compose}\langle HY, NA, Y \rangle. \]

By rewriting clause 15 using applicability condition (1):

clause 16: \[ r.td_2(Xs, Y, A) \rightarrow\]
\[ Xs = [X|TXs], \]
\[ \text{nonMinimal}\langle X \rangle, \text{decompose}\langle X, HX, TX_1, \ldots, TX_t \rangle, \]
\[ \text{minimal}\langle TX_1 \rangle, \ldots, \text{minimal}\langle TX_{p-1} \rangle. \]
\[ (\text{nonMinimal}\langle TX_p \rangle; \ldots; \text{nonMinimal}\langle TX_t \rangle), \]
\[ \text{solve}\langle TX_1, TY_1 \rangle, \ldots, \text{solve}\langle TX_{p-1}, TY_{p-1} \rangle, \]
\[ r(TX_p, TY_p), \ldots, r(TX_1, TY_1). \]
\[ l_0 = e, \]
\[ \text{compose}\langle I_0, TY_1, I_1 \rangle, \ldots, \text{compose}\langle I_{p-2}, TY_{p-1}, I_{p-1} \rangle, \]
\[ \text{process}\langle HX, HHY \rangle, \text{compose}\langle I_{p-1}, HHY, I_p \rangle, \]
\[ HY = I_p, \text{compose}\langle HY, NA, Y \rangle, \]
\[ \text{compose}\langle TY_p, I_{p+1}, NA \rangle, \]
\[ \text{compose}\langle TY_{p+1}, I_{p+2}, I_{p+1} \rangle, \ldots, \text{compose}\langle TY_{t-1}, I_t, I_{t-1} \rangle, \]
\[ \text{compose}\langle TY_t, NNA, I_t \rangle, \]
\[ r.td_2(TXs, NNA, A) \]

By \( t - p \) times folding clause 16 using clauses 1 and 2:
clause 17: \[ r.\text{td}_2(X_s, Y, A) = \]
\[ X_s = [X[TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \]
\[ HY = I_p, \]
\[ r.\text{td}_2([TX_p, \ldots, TX_t]TX_s], NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (3):

clause 18: \[ r.\text{td}_2(X_s, Y, A) = \]
\[ X_s = [X[TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e), \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \]
\[ HY = I_p, \]
\[ r.\text{td}_2([TX_p, \ldots, TX_t]TX_s], NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (2):

clause 19: \[ r.\text{td}_2(X_s, Y, A) = \]
\[ X_s = [X[TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e), \]
\[ I_0 = e, \]
\[ l_1 = I_0, \ldots, I_{p-1} = I_{p-2}, \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \]
\[ HY = I_p, \]
\[ r.\text{td}_2([TX_p, \ldots, TX_t]TX_s], NA, A), \text{compose}(HY, NA, Y) \]

By simplification:

clause 20: \[ r.\text{td}_2(X_s, Y, A) = \]
\[ X_s = [X[TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ r.\text{td}_2([TX_p, \ldots, TX_t]TX_s], NA, A), \]
\[ \text{process}(HX, HYY), \text{compose}(HY, NA, Y) \]

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:
\textbf{clause 21}: \quad r.\text{td}_2(Xs, Y, A) \leftarrow \\
\qquad Xs = [X[TXs]]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \dotsc, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \dotsc; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \dotsc, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \dotsc, \text{minimal}(U_{p-1}). \\
\quad r(TX_1, TY_1), \dotsc, r(TX_t, TY_t). \\
\quad I_0 = \epsilon, \\
\quad \text{compose}(I_0, TY_1, I_1), \dotsc, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\quad \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\quad \text{compose}(I_p, TY_p, I_{p+1}), \dotsc, \text{compose}(I_t, TY_t, I_{t+1}). \\
\quad HY = I_{t+1}, \\
\quad r.\text{td}_2(TXs, NA, A), \text{compose}(HY, NA, Y). \\

By using applicability condition (3):

\textbf{clause 22}: \quad r.\text{td}_2(Xs, Y, A) \leftarrow \\
\qquad Xs = [X[TXs]]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \dotsc, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \dotsc; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \dotsc, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \dotsc, \text{minimal}(U_{p-1}). \\
\quad r(U_1, \epsilon), \dotsc, r(U_{p-1}, \epsilon), \\
\quad r(TX_1, TY_1), \dotsc, r(TX_t, TY_t). \\
\quad I_0 = \epsilon, \\
\quad \text{compose}(I_0, TY_1, I_1), \dotsc, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\quad \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \\
\quad \text{compose}(I_p, TY_p, I_{p+1}), \dotsc, \text{compose}(I_t, TY_t, I_{t+1}). \\
\quad HY = I_{t+1}, \\
\quad r.\text{td}_2(TXs, NA, A), \text{compose}(HY, NA, Y). \\

By using applicability condition (2):

\textbf{clause 23}: \quad r.\text{td}_2(Xs, Y, A) \leftarrow \\
\qquad Xs = [X[TXs]]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \dotsc, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \dotsc; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \dotsc, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \dotsc, \text{minimal}(U_{p-1}). \\
\quad r(U_1, \epsilon), \dotsc, r(U_{p-1}, \epsilon), \\
\quad r(TX_1, TY_1), \dotsc, r(TX_t, TY_t). \\
\quad I_0 = \epsilon, \\
\quad \text{compose}(I_0, TY_1, I_1), \dotsc, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\quad \text{compose}(I_{p-1}, \epsilon, K_1), \text{compose}(K_1, \epsilon, K_2), \dotsc, \text{compose}(K_{p-2}, \epsilon, K_{p-1}), \\
\quad \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, I_p), \\
\quad \text{compose}(I_p, TY_p, I_{p+1}), \dotsc, \text{compose}(I_t, TY_t, I_{t+1}). \\
\quad HY = I_{t+1}, \\
\quad r.\text{td}_2(TXs, NA, A), \text{compose}(HY, NA, Y). \\

By using applicability conditions (1) and (2):
clause 24: \[ r_{\text{td}}(X_s, Y, A) = \]
\[ X_s = [X|T|X_s]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ \text{compose}(K_{p-2}, N, A, Y). \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}). \]
\[ \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \]
\[ \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}). \]
\[ HY = I_{t+1}, \text{compose}(HY, NNA, NA), \]
\[ r_{\text{td}}(TX_s, NNA, A) \]

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

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

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

\[ \exists N: X_0 \cdot \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \]

always holds (because \( N \) is existentially quantified)

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

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \):
clause 27: \( r.jd_2(X_s, Y, A) \) —

\[
X_s = [X | T X_s].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t). \)

\( \text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1}). \)

\( \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t). \)

\( \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t). \)

\( \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( r(U_1, Y U_1), \ldots, r(U_{p-1}, Y U_{p-1}). \)

\( r(T X_1, Y T Y_1), \ldots, r(T X_t, Y T Y_t). \)

\( \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N A, Y). \)

\( l_0 = e, \)

\( \text{compose}(I_0, Y U_1, I_1), \ldots, \text{compose}(I_{p-2}, Y U_{p-1}, I_{p-1}). \)

\( \text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p). \)

\( \text{compose}(I_p, Y T Y_p, I_{p+1}), \ldots, \text{compose}(I_t, Y T Y_t, I_{t+1}). \)

\( H Y = I_{t+1}, \text{compose}(H Y, N N A, N A). \)

\( r.jd_2(T X_s, N N A, A) \)

By folding clause 27 using \( DCLR \):

clause 28: \( r.jd_2(X_s, Y, A) \) —

\[
X_s = [X | T X_s].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t). \)

\( \text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1}). \)

\( \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t). \)

\( \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( r(T X_1, Y T Y_1), \ldots, r(T X_{p-1}, Y T Y_{p-1}), r(N, H Y). \)

\( \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N A, Y). \)

\( \text{compose}(H Y, N N A, N A). \)

\( r.jd_2(T X_s, N N A, A) \)

By folding clause 28 using clauses 1 and 2:

clause 29: \( r.jd_2(X_s, Y, A) \) —

\[
X_s = [X | T X_s].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t). \)

\( \text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1}). \)

\( \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t). \)

\( \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( r(T X_1, Y T Y_1), \ldots, r(T X_{p-1}, Y T Y_{p-1}), r(N, H Y). \)

\( \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N A, Y). \)

\( r.jd_2([N | T X_s], N N A, A) \)

By \( p-1 \) times folding clause 29 using clauses 1 and 2:

clause 30: \( r.jd_2(X_s, Y, A) \) —

\[
X_s = [X | T X_s].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t). \)

\( \text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1}). \)

\( \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t). \)

\( r.jd_2(T X_1, T X_{p-1}, [N | T X_s], Y, A) \)

By introducing atoms \( \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_t \) in clause 8:}
clause 31: \[ r \cup \tilde{d}_2(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_0 = e, \]
\[ \text{compose}(I_{l_0}, TY_1, I_1), \ldots, \text{compose}(I_{l_{p-2}}, TY_{p-1}, I_{l_{p-1}}), \]
\[ \text{process}(HX, HHY), \text{compose}(I_{l_{p-1}}, HHY, I_p), \]
\[ \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_{l_t}, TY_t, I_{l_{t+1}}). \]
\[ HY = I_{l_{t+1}}, \]
\[ r \cup \tilde{d}_2(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (3):

clause 32: \[ r \cup \tilde{d}_2(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \]
\[ r(U_1, e), \ldots, r(U_t, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_0 = e, \]
\[ \text{compose}(I_{l_0}, TY_1, I_1), \ldots, \text{compose}(I_{l_{p-2}}, TY_{p-1}, I_{l_{p-1}}), \]
\[ \text{process}(HX, HHY), \text{compose}(I_{l_{p-1}}, HHY, I_p), \]
\[ \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_{l_t}, TY_t, I_{l_{t+1}}). \]
\[ HY = I_{l_{t+1}}, \]
\[ r \cup \tilde{d}_2(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (2):

clause 33: \[ r \cup \tilde{d}_2(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t). \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \]
\[ r(U_1, e), \ldots, r(U_t, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_0 = e, \]
\[ \text{compose}(I_{l_0}, TY_1, I_1), \ldots, \text{compose}(I_{l_{p-2}}, TY_{p-1}, I_{l_{p-1}}), \]
\[ \text{compose}(I_{l_{p-1}}, e, K_1), \text{compose}(K_1, e, K_2), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, K_p), \]
\[ \text{compose}(K_p, e, K_{p+1}), \ldots, \text{compose}(K_{l_t}, e, K_{l_{t+1}}), \text{compose}(I_{l_{t+1}}, e, I_p). \]
\[ HY = I_{l_{t+1}}, \]
\[ r \cup \tilde{d}_2(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability conditions (1) and (2):
clause 34: \( r \cdot td_2(X_s, Y, A) = \)
\[ X_s = [X[TXs]]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \)
\( r(U_1, e), \ldots, r(U_t, e). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( I_0 = e, \)
\( \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}). \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, e, I_{p+1}), \ldots, \text{compose}(I_t, e, I_{t+1}). \)
\( NHY = I_{t+1}, \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \)
\( \text{compose}(K_{t-1}, NA, NA_1), \text{compose}(NHY, NA_1, NA_2), \)
\( \text{compose}(K_{p-2}, NA_2, Y), r \cdot td_2(TXs, NA, A) \)

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

clause 35: \( r \cdot td_2(Xs, Y, A) = \)
\[ X_s = [X[TXs]]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}). \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \)
\( r(U_1, YU_1), \ldots, r(U_t, YU_t). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( I_0 = e, \)
\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}). \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, YU_p, I_{p+1}), \ldots, \text{compose}(I_t, YU_t, I_{t+1}). \)
\( NHY = I_{t+1}, \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \)
\( \text{compose}(K_{t-1}, NA, NA_1), \text{compose}(NHY, NA_1, NA_2), \)
\( \text{compose}(K_{p-2}, NA_2, Y), r \cdot td_2(TXs, NA, A) \)

By introducing \text{nonMinimal}(N) and \text{decompose}(N, HX, U_1, \ldots, U_t), since
\[ \exists N: X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_t) \]
always holds (because \( N \) is existentially quantified)
clause 36: \( r.td_2(Xs, Y, A) \rightarrow \\
Xs = [X | TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t). \\
r(U_1, YU_1), \ldots, r(U_t, YU_t). \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
l_0 = e, \\
\text{compose}(I_{p_0}, YU_1, I_1), \ldots, \text{compose}(I_{p_{p-2}}, YU_{p-1}, I_{p_{p-1}}). \\
\text{process}(HX, HHY), \text{compose}(I_{p_{p-1}}, HHY, I_{p_t}). \\
\text{compose}(I_{p_t}, YU_1, I_{p_1+1}), \ldots, \text{compose}(I_{l_t}, YU_{l_t}, I_{l_{t+1}}). \\
NHY = l_{t+1}. \\
\text{compose}(TY_t, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \\
\text{compose}(K_{t-1}, N, A, N_A), \text{compose}(NHY, N_{A_1}, N_A). \\
\text{compose}(K_{p-2}, N_{A_2}, Y), r.td_2(TXs, N, A, A). \\
\)

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_t): \)

clause 37: \( r.td_2(Xs, Y, A) \rightarrow \\
Xs = [X | TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t). \\
\text{decompose}(N, HX, U_1, \ldots, U_t). \\
r(U_1, YU_1), \ldots, r(U_t, YU_t). \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
l_0 = e, \\
\text{compose}(I_{p_0}, YU_1, I_1), \ldots, \text{compose}(I_{p_{p-2}}, YU_{p-1}, I_{p_{p-1}}). \\
\text{process}(HX, HHY), \text{compose}(I_{p_{p-1}}, HHY, I_{p_t}). \\
\text{compose}(I_{p_t}, YU_1, I_{p_1+1}), \ldots, \text{compose}(I_{l_t}, YU_{l_t}, I_{l_{t+1}}). \\
NHY = l_{t+1}. \\
\text{compose}(TY_t, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \\
\text{compose}(K_{t-1}, N, A, N_A), \text{compose}(NHY, N_{A_1}, N_A). \\
\text{compose}(K_{p-2}, N_{A_2}, Y), r.td_2(TXs, N, A, A). \\
\)

By folding clause 37 using \( DCLR \):

clause 38: \( r.td_2(Xs, Y, A) \rightarrow \\
Xs = [X | TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{decompose}(N, HX, U_1, \ldots, U_t). \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
\text{compose}(TY_t, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}). \\
\text{compose}(K_{t-1}, N, A, N_A), \text{compose}(NHY, N_{A_1}, N_A). \\
\text{compose}(K_{p-2}, N_{A_2}, Y), r.td_2(TXs, N, A, A). \\
\)

By \( t = p + 1 \) times folding clause 38 using clauses 1 and 2:
clause 39: \( r \cdot td_2(Xs, Y, A) \leftarrow \\
Xs = [X[Txs]]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_{t})). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{decompose}(N, HX, U_1, \ldots, U_t). \\
r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY). \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
\text{compose}(NY, NA_1, NA_2). \\
\text{compose}(K_{p-2}, NA_2, Y), r \cdot td_2([TX_1, \ldots, TX_t[Txs]], NA_1, A). \)

By folding clause 39 using clauses 1 and 2:

clause 40: \( r \cdot td_2(Xs, Y, A) \leftarrow \\
Xs = [X[Txs]]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_{t})). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{decompose}(N, HX, U_1, \ldots, U_t). \\
r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}). \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \\
\text{compose}(K_{p-2}, NA_2, Y), r \cdot td_2([N, TX_p, \ldots, TX_t[Txs]], NA_2, A). \)

By \( p-1 \) times folding clause 40 using clauses 1 and 2:

clause 41: \( r \cdot td_2(Xs, Y, A) \leftarrow \\
Xs = [X[Txs]]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})). \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_{t})). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t). \\
\text{decompose}(N, HX, U_1, \ldots, U_t). \\
r \cdot td_2([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t[Txs]], Y, A). \)

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of \( P_{r \cdot td_2} \). Therefore \( P_{r \cdot td_2} \) is strongest-wrt \( S_{r \cdot td_2} \) in \( S \).

To prove that \( P_r \) is strongest-wrt \( S_r \) in \( \{S_{r \cdot td_2}\} \), we do a backward proof that we begin with \( P_r \) in \( \text{TDGRL} \) and from which we try to obtain \( S_r \).

The procedure \( P_r \) for \( r \) in \( \text{TDGRL} \) is:

\[
r(X, Y) \leftarrow r \cdot td_2([X], Y, e) \]

By taking the 'completion':

\[
\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad \Upsilon_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r \cdot td_2([X], Y, e)]
\]

By unfolding the 'completion' above wrt \( r \cdot td_2([X], Y, e) \) using \( S_{r \cdot td_2} \):

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

By using applicability condition (2):

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

By simplification:

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

We obtain \( S_r \), so \( P_r \) is strongest-wrt \( S_r \) in \( \{S_{r \cdot td_2}\} \).

Therefore, \( \text{TDGRL} \) is also strongest-wrt \( S_r \) in \( S \).

\[ \square \]

**Theorem 9** The generalization schema \( \text{TDGRL} \), which is given below, is correct.
\[
TDG_3 : \{ DCRL, TDGRL, A_{td3}, O_{td312}, O_{td321} \} \text{ where }
\]
\[
A_{td3} : (1) \text{ compose is associative}
\]
\[
(2) \text{ compose has } e \text{ as the left and right identity element}
\]
\[
(3) T_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e)
\]
\[
(4) T_r(X) \Rightarrow \neg \text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)
\]
\[
O_{td312} : \text{ partial evaluation of the conjunction }
\]
\[
\text{process}(H X, H Y), \text{compose}(H Y, A, N e w A)
\]
\[
\text{results in the introduction of a non-recursive relation}
\]
\[
O_{td321} : \text{ partial evaluation of the conjunction }
\]
\[
\text{process}(H X, H Y), \text{compose}(H Y, I_p, I_{p-1})
\]
\[
\text{results in the introduction of a non-recursive relation}
\]

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:
\[
\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ T_r(X) \Rightarrow |r(X, Y) \Rightarrow O_r(X, Y)|
\]

The specification of \( r_{td2} \), namely \( S_{r_{td2}} \), is:
\[
\forall X s : \text{list of } X, Y, X s \in X s \Rightarrow T_r(X) \Rightarrow |r_{td2}(X s, Y, A) \Leftrightarrow (X s = [] \land Y = A)
\]
\[
\forall (X s = [X_1, X_2, \ldots, X_q]) \land \bigwedge_{i=1}^q O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^q O_r(I_{i-1}, Y_i, I_i)
\]
\[
\land O_r(I_1, A, I_{q+1}) \land Y = I_{q+1}]
\]

**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 \( A_{td3} \) if \( DCRL \) 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 \( S = \{ S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \} \), where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \) are the specifications of \( \text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose} \), which are all the undefined relation names appearing in \( DCRL \).

(b) \( TDGRL \) is also steadfast \( wrt \) \( S_r \) in \( S \).

Note that the sets \( \{ S_1, \ldots, S_m \} \) and \( \{ S_1', \ldots, S_s' \} \) in Definition 4 are equal to \( S \) 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 \( S \), 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 \( S \) if \( P_{r_{td2}} \) is steadfast \( wrt \) \( S_{r_{td2}} \) in \( S \), where \( P_{r_{td2}} \) is the procedure for \( r_{td2} \), and \( P_r \) is steadfast \( wrt \) \( S_r \) in \( S_{r_{td2}} \), where \( P_r \) is the procedure for \( r \).

To prove that \( P_{r_{td2}} \) is steadfast \( wrt \) \( S_{r_{td2}} \) in \( S \), we do a constructive forward proof that we begin with \( S_{r_{td2}} \), and from which we try to obtain \( P_{r_{td2}} \).

If we separate the cases of \( q \geq 1 \) by \( q = 1 \lor q \geq 2 \), then \( S_{r_{td2}} \) becomes:
\[
\forall X s : \text{list of } X, Y Y : \mathcal{Y}. \ (X \times X. \ X \in X s \Rightarrow T_r(X)) \Rightarrow |r_{td2}(X s, Y, A) \Leftrightarrow
\]
\[
|X s = [] \land Y = A|
\]
\[
|X s = [X_1] \land O_r(X_1, Y_1) \land Y = I_1 \land O_r(I_1, A, I_2) \land Y = I_2|
\]
\[
|X s = [X_1, X_2, \ldots, X_q] \land \bigwedge_{i=1}^q O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^q O_r(I_{i-1}, Y_i, I_i) \land
\]
\[
O_r(I_1, A, I_{q+1}) \land Y = I_{q+1}]
\]

where \( q \geq 2 \).

By using applicability conditions (1) and (2):
\[
\forall X s : \text{list of } X, Y Y : \mathcal{Y}. \ (X \times X. \ X \in X s \Rightarrow T_r(X)) \Rightarrow |r_{td2}(X s, Y, A) \Leftrightarrow
\]
\[
|X s = [] \land Y = A|
\]
\[
|X s = [X_1] \times X s \land TX s = [] \land O_r(X_1, Y_1) \land Y = I_1 \land TY = A \land O_r(TY, A, N A) \land O_r(I_1, N A, Y)|
\]
\[
|X s = [X_1] \times X s \land TX s = [X_2, \ldots, X_q] \land \bigwedge_{i=1}^q O_r(X_i, Y_i) \land Y = I_1 \land Y_2 = I_2 \land
\]
\[
\bigwedge_{i=3}^q O_r(I_{i-1}, Y_i, I_i) \land TY = I_q \land O_r(TY, A, N A) \land O_r(I_1, N A, Y)]
\]
where \( q \geq 2 \).

By folding using \( S_{r,td_2} \), and renaming:

\[
\forall Xs : \text{list of } \mathcal{X}, Y : \mathcal{Y}. \ (X : \mathcal{X}. \ X \in Xs \Rightarrow \mathcal{I}_r(X)) \Rightarrow [r_{td_2}(Xs, Y, A) \Leftrightarrow \\
[Xs = [] \land Y = A] \\
\lor (Xs = [X[TXs] \land \mathcal{O}_s(X, HY) \land r_{td_2}(TXs, NA, A) \land \mathcal{O}_c(HY, NA, Y))] \\
\]

By taking the 'decompletion':

\[
\text{clause 1: } \quad r_{td_2}(Xs, Y, A) \leftarrow \\
\quad Xs = [], Y = A
\]

\[
\text{clause 2: } \quad r_{td_2}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], r(X, HY), \\
\quad r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)]
\]

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

\[
\text{clause 3: } \quad r_{td_2}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, HY), \\
\quad r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)]
\]

\[
\text{clause 4: } \quad r_{td_2}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\quad l_{t+1} = e, \\
\quad \text{compose}(TY_1, l_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \\
\quad \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\quad \text{compose}(TY_p-1, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_t, I_0), \\
\quad HY = I_0, \\
\quad r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)]
\]

By introducing

\[
[\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_t)] \lor \\
((\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_{p-1})) \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))) \lor \\
((\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})) \land (\text{minimal}(TX_p) \land \ldots \land \text{minimal}(TX_t))) \lor \\
[\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_{p-1})] \land (\text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_t))
\]

in clause 4, using applicability condition (4):

\[
\text{clause 5: } \quad r_{td_2}(Xs, Y, A) \leftarrow \\
\quad Xs = [X[TXs], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
\quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\quad l_{t+1} = e, \\
\quad \text{compose}(TY_1, l_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \\
\quad \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\quad \text{compose}(TY_p-1, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_t, I_0), \\
\quad HY = I_0, \\
\quad r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)]
\]
clause 6: \( r_{ld2}(Xs, Y, A) \) —

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i),
\]

\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1})
\]

\[
\{\text{nonMinimal}(TX_p) ; \ldots ; \text{nonMinimal}(TX_t)\}
\]

\[
r(TX_1, TY_1), \ldots, r(TX_1, TY_i),
\]

\[
I_{t+1} = e.
\]

\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]

\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1})
\]

\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0)
\]

\[
HY = I_0.
\]

\( r_{ld2}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

clause 7: \( r_{ld2}(Xs, Y, A) \) —

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i),
\]

\[
\{\text{nonMinimal}(TX_1) ; \ldots ; \text{nonMinimal}(TX_{p-1})\}
\]

\[
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t),
\]

\[
r(TX_1, TY_1), \ldots, r(TX_1, TY_i),
\]

\[
I_{t+1} = e.
\]

\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]

\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1})
\]

\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0)
\]

\[
HY = I_0.
\]

\( r_{ld2}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

clause 8: \( r_{ld2}(Xs, Y, A) \) —

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i),
\]

\[
\{\text{nonMinimal}(TX_1) ; \ldots ; \text{nonMinimal}(TX_{p-1})\}
\]

\[
\{\text{nonMinimal}(TX_p) ; \ldots ; \text{nonMinimal}(TX_t)\}
\]

\[
r(TX_1, TY_1), \ldots, r(TX_1, TY_i),
\]

\[
I_{t+1} = e.
\]

\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]

\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1})
\]

\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0)
\]

\[
HY = I_0.
\]

\( r_{ld2}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

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

clause 9: \( r_{ld2}(Xs, Y, A) \) —

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i),
\]

\[
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}),
\]

\[
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t),
\]

\[
solve(TX_1, TY_1), \ldots, \text{solve}(TX_1, TY_i),
\]

\[
I_{t+1} = e.
\]

\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]

\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1})
\]

\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0)
\]

\[
HY = I_0.
\]

\( r_{ld2}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):
\textit{clause 10}: \ \ r.td_2(Xs, Y, A) = \\
\hspace{1em} Xs = [X]TXs. \\
\hspace{1em} \nonminimal(X), \ \ \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\hspace{1em} \minimal(TX_1), \ldots, \ \ \minimal(TX_t). \\
\hspace{1em} \solve(TX_1, e), \ldots, \ \ \solve(TX_t, e). \\
\hspace{1em} I_{t+1} = e. \\
\hspace{1em} \compose(e, I_{t+1}, I_t), \ldots, \ \ \compose(e, I_{p+1}, I_p). \\
\hspace{1em} \process(HX, HHY), \ \ \compose(HHY, I_p, I_{p-1}). \\
\hspace{1em} \compose(e, I_{p-1}, I_{p-2}), \ldots, \ \ \compose(e, I_1, I_0). \\
\hspace{1em} HY = I_0. \\
\hspace{1em} r.td_2(TXs, NA, A), \ \ \compose(HY, NA, Y). \\

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

\textit{clause 11}: \ \ r.td_2(Xs, Y, A) = \\
\hspace{1em} Xs = [X]TXs. \\
\hspace{1em} \nonminimal(X), \ \ \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\hspace{1em} \minimal(TX_1), \ldots, \ \ \minimal(TX_t). \\
\hspace{1em} \solve(TX_1, e), \ldots, \ \ \solve(TX_t, e). \\
\hspace{1em} I_{t+1} = e. \\
\hspace{1em} \compose(e, I_{t+1}, I_t), \ldots, \ \ \compose(e, I_{p+1}, I_p). \\
\hspace{1em} \process(HX, HHY), \ \ \compose(HHY, I_p, I_{p-1}). \\
\hspace{1em} \compose(e, I_{p-1}, I_{p-2}), \ldots, \ \ \compose(e, I_1, I_0). \\
\hspace{1em} HY = I_0. \\
\hspace{1em} r.td_2(TXs, NA, A), \ \ \compose(HY, NA, Y). \\

By using applicability condition (2):

\textit{clause 12}: \ \ r.td_2(Xs, Y, A) = \\
\hspace{1em} Xs = [X]TXs. \\
\hspace{1em} \nonminimal(X), \ \ \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\hspace{1em} \minimal(TX_1), \ldots, \ \ \minimal(TX_t). \\
\hspace{1em} \solve(TX_1, e), \ldots, \ \ \solve(TX_t, e). \\
\hspace{1em} I_{t+1} = e. \\
\hspace{1em} I_t = I_{t+1}, \ldots, I_p = I_{p+1}. \\
\hspace{1em} \process(HX, HHY), \ \ \compose(HHY, I_p, I_{p-1}). \\
\hspace{1em} I_{p-2} = I_{p-1}, \ldots, I_0 = I_1. \\
\hspace{1em} HY = I_0. \\
\hspace{1em} r.td_2(TXs, NA, A), \ \ \compose(HY, NA, Y). \\

By simplification:

\textit{clause 13}: \ \ r.td_2(Xs, Y, A) = \\
\hspace{1em} Xs = [X]TXs. \\
\hspace{1em} \nonminimal(X), \ \ \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\hspace{1em} \minimal(TX_1), \ldots, \ \ \minimal(TX_t). \\
\hspace{1em} r.td_2(TXs, NA, A), \\
\hspace{1em} \process(HX, HHY), \ \ \compose(HY, NA, Y). \\

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):
clause 14:  \( r \cdot d_2(Xs, Y, A) \)

\[ Xs = [X|TXs] \]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \)

\( \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}) \)

\( \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}) \)

\( r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \)

\( I_{t+1} = e \)

\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)

\( r \cdot d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

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

clause 15:  \( r \cdot d_2(Xs, Y, A) \)

\[ Xs = [X|TXs] \]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \)

\( \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}) \)

\( \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}) \)

\( r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \)

\( I_{t+1} = e \)

\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)

\( r \cdot d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

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

clause 16:  \( r \cdot d_2(Xs, Y, A) \)

\[ Xs = [X|TXs] \]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)

\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \)

\( \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}) \)

\( \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}) \)

\( r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \)

\( I_0 = e \)

\( \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}) \)

\( \text{compose}(TY_{p+1}, I_{p+1}, NA) \)

\( r \cdot d_2(TXs, NA, A) \)

By \( t - p \) times folding clause 16 using clauses 1 and 2:
clause 17: \[ r.td_2(Xs, Y, A) \]

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]

\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}),
\]

\[
l_0 = e,
\]

\[
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}),
\]

\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \text{HY} = I_p,
\]

\[
r.td_2([TX_p], \ldots, TX_t[TXs], NA, A), \text{compose}(HY, NA, Y)
\]

By using applicability condition (3):

clause 18: \[ r.td_2(Xs, Y, A) \]

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]

\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e),
\]

\[
l_0 = e,
\]

\[
\text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}),
\]

\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \text{HY} = I_p,
\]

\[
r.td_2([TX_p], \ldots, TX_t[TXs], NA, A), \text{compose}(HY, NA, Y)
\]

By using applicability condition (2):

clause 19: \[ r.td_2(Xs, Y, A) \]

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]

\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e),
\]

\[
l_0 = e,
\]

\[
l_1 = l_0, \ldots, l_{p-1} = l_{p-2},
\]

\[
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \text{HY} = I_p,
\]

\[
r.td_2([TX_p], \ldots, TX_t[TXs], NA, A), \text{compose}(HY, NA, Y)
\]

By simplification:

clause 20: \[ r.td_2(Xs, Y, A) \]

\[
Xs = [X][TXs],
\]

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}).
\]

\[
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \text{process}(HX, HHY), \text{compose}(HY, NA, Y)
\]

By introducing atoms \( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_{p-1} \)) in clause 7:
\text{clause 21: } r \text{ld}_2(XS, Y, A) = \\
xS = [X|TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \\
I_{t+1} = e, \\
\text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \\
HY = I_0, \\
r \text{ld}_2(TXS, NA, A), \text{compose}(HY, NA, Y). \\

\text{By using applicability condition (3):}

\text{clause 22: } r \text{ld}_2(XS, Y, A) = \\
xS = [X|TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\text{minimal}(U_1), \ldots, \text{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, \\
\text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \\
HY = I_0, \\
r \text{ld}_2(TXS, NA, A), \text{compose}(HY, NA, Y). \\

\text{By using applicability condition (2):}

\text{clause 23: } r \text{ld}_2(XS, Y, A) = \\
xS = [X|TXs]. \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\text{minimal}(U_1), \ldots, \text{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, \\
\text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\text{compose}(e, I_{p-1}, K_1), \text{compose}(e, K_1, K_2), \ldots, \text{compose}(e, K_{p-2}, K_{p-1}), \\
\text{compose}(TY_{p-1}, K_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0). \\
HY = I_0, \\
r \text{ld}_2(TXS, NA, A), \text{compose}(HY, NA, Y). \\

\text{By using applicability conditions (1) and (2):}
clause 24:  \( r \cdot \text{td}_2(X_s, Y, A) \rightarrow \)
\( X_s = [X|TX_s], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, e), \ldots, r(U_{p-1}, e), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( I_{t+1} = e. \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \)
\( \text{compose}(e, I_{p-1}, I_{p-2}), \ldots, \text{compose}(e, I_1, I_0). \)
\( HY = I_0, r \cdot \text{td}_2(TX_s, N A, A), \text{compose}(HY, N A, N N A), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(K_{p-2}, N N A, Y). \)

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

clause 25:  \( r \cdot \text{td}_2(X_s, Y, A) \rightarrow \)
\( X_s = [X|TX_s], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( I_{t+1} = e, \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \)
\( \text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0). \)
\( HY = I_0, r \cdot \text{td}_2(TX_s, N A, A), \text{compose}(HY, N A, N N A), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(K_{p-2}, N N A, Y). \)

By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \), since
\( \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \)
always holds (because \( N \) is existentially quantified).

clause 26:  \( r \cdot \text{td}_2(X_s, Y, A) \rightarrow \)
\( X_s = [X|TX_s], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)
\( r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \)
\( \text{nonMinimal}(N), \text{compose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( I_{t+1} = e, \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \)
\( \text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0). \)
\( HY = I_0, r \cdot \text{td}_2(TX_s, N A, A), \text{compose}(HY, N A, N N A), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)
\( \text{compose}(K_{p-2}, N N A, Y). \)

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \):
clause 27: \( rtd_2(Xs, Y, A) \)

\[
Xs = [X | TXs].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)

\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)

\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}). \)

\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \)

\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \)

\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)

\( I_{t+1} = e. \)

\( \text{compose}(TY_1, I_{t+1}, I_1), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \)

\( \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}). \)

\( \text{compose}(UY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(UY_1, I_1, I_0). \)

\( HY = I_0, rtd_2(TXs, N A, A), \text{compose}(HY, N A, N NA). \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N NA, Y). \)

By folding clause 27 using DCRL:

clause 28: \( rtd_2(Xs, Y, A) \)

\[
Xs = [X | TXs].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)

\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)

\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \)

\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}). r(N, HY). \)

\( rtd_2(TXs, N A, A), \text{compose}(HY, N A, N NA). \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N NA, Y). \)

By folding clause 28 using clauses 1 and 2:

clause 29: \( rtd_2(Xs, Y, A) \)

\[
Xs = [X | TXs].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)

\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)

\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \)

\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}). \)

\( rtd_2([N | TXs], N NA, A). \)

\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}). \)

\( \text{compose}(K_{p-2}, N NA, Y). \)

By \( p - 1 \) times folding clause 29 using clauses 1 and 2:

clause 30: \( rtd_2(Xs, Y, A) \)

\[
Xs = [X | TXs].
\]

\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)

\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \)

\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \)

\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \)

\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \)

\( rtd_2([TX_1, \ldots, TX_{p-1}, N | TXs], Y, A). \)

By introducing atoms \( \text{minimal}(U_1), \ldots, \text{minimal}(U_t) \) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_t \)) in clause 8:
clause 31: \( r \, \Delta d_2(X_s, Y, A) \leftarrow \)
\( X_s = [X | TXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_1) \)
\( I_{t+1} = e, \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}) \)
\( \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0) \)
\( HY = I_0, \)
\( r \, \Delta d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):

clause 32: \( r \, \Delta d_2(X_s, Y, A) \leftarrow \)
\( X_s = [X | TXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_1) \)
\( r(U_1, e), \ldots, r(U_t, e) \)
\( I_{t+1} = e, \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}) \)
\( \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0) \)
\( HY = I_0, \)
\( r \, \Delta d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (2):

clause 33: \( r \, \Delta d_2(X_s, Y, A) \leftarrow \)
\( X_s = [X | TXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_1) \)
\( r(U_1, e), \ldots, r(U_t, e) \)
\( I_{t+1} = e, \)
\( \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p) \)
\( \text{compose}(e, I_p, K_{t+1}) \)
\( \text{compose}(e, K_{p+1}, K_t), \ldots, \text{compose}(e, K_{p+1}, K_0) \)
\( \text{process}(HX, HHY), \text{compose}(HHY, K_p, K_{p-1}) \)
\( \text{compose}(e, K_{p-1}, K_{p-2}), \ldots, \text{compose}(e, K_1, K_0) \)
\( \text{compose}(e, K_0, I_{p-1}) \)
\( \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0) \)
\( HY = I_0, \)
\( r \, \Delta d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability conditions (1) and (2):
clause 34: \( r.td_2(Xs,Y,A) \) —
\[
Xs = [X|TXs].
\]
\( \text{nonMinimal}(X) \), \( \text{decompose}(X,HX,TX_1,\ldots,TX_t) \),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \),
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \),
\( r(TX_1,TY_1),\ldots,r(TX_t,TY_t) \),
\( \text{minimal}(U_1),\ldots,\text{minimal}(U_t) \),
\( r(U_1,\epsilon),\ldots,r(U_t,\epsilon) \),
\( I_{t+1} = \epsilon \),
\( \text{compose}(e,I_{t+1},I_t),\ldots,\text{compose}(e,I_{p+1},I_p) \),
\( \text{process}(HX,HHY), \text{compose}(HHY,I_p,I_{p-1}) \),
\( \text{compose}(e,I_{p-1},I_{p-2}),\ldots,\text{compose}(e,I_1,I_0) \),
\( NHY = I_0 \),
\( \text{compose}(TY_1,TY_2,K_1),\ldots,\text{compose}(K_p,TY_{p+2},K_{p+1}),\ldots,\text{compose}(K_{t-2},TY_t,K_{t-1}) \),
\( r.td_2(TXs,NA,A),\text{compose}(K_{t-1},NA,NA_1) \),
\( \text{compose}(NHY,NA_1,NA_2),\text{compose}(K_p,NA_2,Y) \).

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

clause 35: \( r.td_2(Xs,Y,A) \) —
\[
Xs = [X|TXs].
\]
\( \text{nonMinimal}(X) \), \( \text{decompose}(X,HX,TX_1,\ldots,TX_t) \),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \),
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t) \),
\( r(TX_1,TY_1),\ldots,r(TX_t,TY_t) \),
\( \text{minimal}(U_1),\ldots,\text{minimal}(U_t) \),
\( r(U_1,YU_1),\ldots,r(U_t,YU_t) \),
\( I_{t+1} = \epsilon \),
\( \text{compose}(YU_1,I_{t+1},I_t),\ldots,\text{compose}(YU_{p+1},I_p) \),
\( \text{process}(HX,HHY), \text{compose}(HHY,I_{p-1},I_p) \),
\( \text{compose}(YU_{p-1},I_{p-1},I_{p-2}),\ldots,\text{compose}(YU_1,I_1,I_0) \),
\( NHY = I_0 \),
\( \text{compose}(TY_1,TY_2,K_1),\ldots,\text{compose}(K_p,TY_{p+2},K_{p+1}),\ldots,\text{compose}(K_{t-2},TY_t,K_{t-1}) \),
\( r.td_2(TXs,N,A),\text{compose}(K_{t-1},N,NA_1) \),
\( \text{compose}(NHY,NA_1,NA_2),\text{compose}(K_p,NA_2,Y) \).

By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N,HX,U_1,\ldots,U_t) \), since
\[
\exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N,HX,U_1,\ldots,U_t)
\]
always holds (because \( N \) is existentially quantified)
clause 36: \( r.\text{ld}_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}). \)
\( \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \)
\( r(U_1, YU_1), \ldots, r(U_t, YU_t). \)
\( I_{t+1} = e, \)
\( \text{compose}(YU_1, I_{t+1}, I_t), \ldots, \text{compose}(YU_p, I_{p+1}, I_p). \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \)
\( \text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0), \)
\( NHY = I_0, \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \)
\( r.\text{ld}_2(TXs, NA, A), \text{compose}(K_{t-1}, NA, NA_1), \)
\( \text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y) \)

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_t) \):

clause 37: \( r.\text{ld}_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}). \)
\( \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \)
\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t). \)
\( \text{decompose}(N, HX, U_1, \ldots, U_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \)
\( r(U_1, YU_1), \ldots, r(U_t, YU_t). \)
\( I_{t+1} = e, \)
\( \text{compose}(YU_1, I_{t+1}, I_t), \ldots, \text{compose}(YU_p, I_{p+1}, I_p). \)
\( \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \)
\( \text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0), \)
\( NHY = I_0, \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \)
\( r.\text{ld}_2(TXs, NA, A), \text{compose}(K_{t-1}, NA, NA_1), \)
\( \text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y) \)

By folding clause 37 using DCRCL:

clause 38: \( r.\text{ld}_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs]. \]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \)
\( \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}). \)
\( \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t). \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NHY), \)
\( \text{decompose}(N, HX, U_1, \ldots, U_t). \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t). \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \)
\( r.\text{ld}_2(TXs, NA, A), \text{compose}(K_{t-1}, NA, NA_1), \)
\( \text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y) \)

By \( t - p + 1 \) times folding clause 38 using clauses 1 and 2:
clause 39: \( r.td_2(Xs, Y, A) \leftarrow \)
\( Xs = [X \in TxXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_{x_i}) \)
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY) \)
\( \text{decompose}(N, HX, U_1, \ldots, U_i) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)
\( \text{compose}(N, TY_1, \ldots, TX_i, TX \in TxXs), N_{A_1}, A) \)
\( \text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y) \)

By folding clause 39 using clauses 1 and 2:

clause 40: \( r.td_2(Xs, Y, A) \leftarrow \)
\( Xs = [X \in TxXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_{x_i}) \)
\( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}) \)
\( \text{decompose}(N, HX, U_1, \ldots, U_i) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}) \)
\( \text{compose}(N, TX_1, \ldots, TX_i, TX \in TxXs), N_{A_2}, A) \)
\( \text{compose}(K_{p-2}, NA_2, Y) \)

By \( p-1 \) times folding clause 40 using clauses 1 and 2:

clause 41: \( r.td_2(Xs, Y, A) \leftarrow \)
\( Xs = [X \in TxXs] \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i) \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_{x_i}) \)
\( \text{decompose}(N, HX, U_1, \ldots, U_i) \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \)
\( r.td_2([N, TX_1, \ldots, TX_i, TX \in TxXs], N_{A_2}, A) \)
\( \text{compose}(K_{p-2}, NA_2, Y) \)

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 \( S \).

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:

\( r(X, Y) \leftarrow r.td_2([X], Y, e) \)

By taking the ‘completion’:

\( \forall X : x, \forall Y : y \cdot \mathcal{I}_r(X) \Rightarrow \lvert r(X, Y) \Rightarrow r.td_2([X], Y, e) \rvert \)

By unfolding the ‘completion’ above wrt \( r.td_2([X], Y, e) \) using \( S_{r.td_2} \):

\( \forall X : x, \forall Y : y \cdot \mathcal{I}_r(X) \Rightarrow \lvert r(X, Y) \Rightarrow \exists Y_1, I_1 : y \cdot \mathcal{O}_r(X, Y_1) \land I_1 = Y_1 \land \mathcal{O}(I_1, e, y) \rvert \)

By using applicability condition (2):

\( \forall X : x, \forall Y : y \cdot \mathcal{I}_r(X) \Rightarrow \lvert r(X, Y) \Rightarrow \exists Y_1, I_1 : y \cdot \mathcal{O}_r(X, Y_1) \land I_1 = Y_1 \land Y = I_1 \rvert \)

By simplification:

\( \forall X : x, \forall Y : y \cdot \mathcal{I}_r(X) \Rightarrow \lvert r(X, Y) \Rightarrow \mathcal{O}_r(X, Y) \rvert \)

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 \( S \).

\( \square \)
Theorem 10 The generalization schema $TDG_4$, which is given below, is correct.

$TDG_4 : \{ DCRL, TDGLR, A_{td1}, O_{td12}, O_{td21} \}$ where 
$A_{td1} : (1) \text{ compose is associative}$
$(2) \text{ compose has e as the left and right identity element}$
$(3) \forall X : \mathcal{X}, I_r(X) \land \text{minimal}(X) \Rightarrow \mathcal{O}_r(X, e)$
$(4) \forall X : \mathcal{X}, I_r(X) \Rightarrow \neg\text{minimal}(X) \Rightarrow \text{nonMinimal}(X)$

$O_{td12} : \text{ partial evaluation of the conjunction}$
$\text{process}(H X, H Y), \text{compose}(A, H Y, \text{New} A)$
$\text{results in the introduction of a non-recursive relation}$

$O_{td21} : \text{ partial evaluation of the conjunction}$
$\text{process}(H X, H Y), \text{compose}(H Y, I_p, I_{p-1})$
$\text{results in the introduction of a non-recursive relation}$

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:

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

The specification $S_{r,td1}:

$$\forall x : \text{list of } X, Y : A, Y : A. \ (\forall X : \mathcal{X}, X \in X \Rightarrow I_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow (X = [] \land Y = A) \lor (X = X_1 \land Y = Y_1 \land I_r(X_1, Y_1) \land I = Y_1 \land \bigwedge_{i=1}^{q} \mathcal{O}_r(I_{p=1}, I_{p=1}) \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=1}^{q} \mathcal{O}_r(I_{p=1}, I_{p=1})]$$

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 $A_{td1}$. By Definition 5, the templates $DCRL$ and $TDGLR$ are equivalent wrt $S_r$ under the applicability conditions $A_{td1}$ iff $DCRL$ is equivalent to $TDGLR$ wrt the specification $S_r$ provided that the conditions in $A_{td1}$ 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 $S = \{ S_{minimal}, S_{nonMinimal}, S_{solve}, S_{decompose}, S_{process}, S_{compose} \}$, 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 $S$.

Note that the sets $\{ S_{1,1}, \ldots, S_{m} \}$ and $\{ S'_{1,1}, \ldots, S'_{m} \}$ in Definition 4 are equal to $S$ 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 $S$, 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 $S$ if $P_{r,td1}$ is steadfast wrt $S_{r,td1}$ in $S$, where $P_{r,td1}$ is the procedure for $r_{td1}$, and $P_r$ is steadfast wrt $S_r$ in $\{ S_{r,td1} \}$, where $P_r$ is the procedure for $r$.

To prove that $P_{r,td1}$ is steadfast wrt $S_{r,td1}$ in $S$, we do a constructive forward proof that we begin with $S_{r,td1}$ and from which we try to obtain $P_{r,td1}$.

If we separate the cases of $q \geq 1$ by $q = 1 \lor q \geq 2$, then $S_{r,td1}$ becomes:

$$\forall X : \text{list of } X, Y : A. \ (\forall X : \mathcal{X}, X \in X \Rightarrow I_r(X)) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y) \land I_1 = Y_1 \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=1}^{q} \mathcal{O}_r(I_{p=1}, I_{p=1}) \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i, Y_i)]$$

where $q \geq 2$.

By using applicability conditions (1) and (2):
∀Xs : list of X, ∀Y : Y. (∀X : X. X ∈ Xs ⇒ I_r(X)) ⇒ [r.⊤d_1(Xs, Y, A) ⇔
\langle Xs = | | \land Y = A \rangle\]

∀(Xs = [X][T,Xs] ∧ TXs = | | ∧ O_r(X_1, Y_1) ∧ Y_1 = I_1 ∧ TY = A ∧ O_r(A, I_1, N_A) ∧ \bigwedge_{\nu=0}^{\infty} O_r(I_{\nu+1}, Y_{\nu+1}) ∧ Y_{\nu+1} = I_{\nu+1} ∧ Y_2 = I_2 ∧
\bigwedge_{\nu=0}^{\infty} O_r(I_{\nu+1}, Y_{\nu+1}) ∧ TY = I_0 ∧ O_r(A, I_1, N_A) ∧ \bigwedge_{\nu=0}^{\infty} O_r(I_{\nu+1}, Y_{\nu+1})]

where \( q > 2 \).

By folding using S_r.⊤d, and renaming:

∀Xs : list of X, ∀Y : Y. (∀X : X. X ∈ Xs ⇒ I_r(X)) ⇒ [r.⊤d_1(Xs, Y, A) ⇔
\langle Xs = | | \land Y = A \rangle\]

∀(Xs = [X][T,Xs] ∧ O_r(X, HY) \land O_r(A, HY, N_A) \land r.⊤d_1(T,Xs, Y, N_A)).

By taking the ‘decomposition’:

\text{clause 1 : } r.⊤d_1(Xs, Y, A) ←
\text{Xs = | |, Y = A}

\text{clause 2 : } r.⊤d_1(Xs, Y, A) ←
\text{Xs = [X][T,Xs], r(X, HY),}
\text{compose(A, HY, N_A), r.⊤d_1(T,Xs, Y, N_A)}

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

\text{clause 3 : } r.⊤d_1(Xs, Y, A) ←
\text{Xs = [X][T,Xs],}
\text{minimal(X),}
\text{solve(X, HY),}
\text{compose(A, HY, N_A), r.⊤d_1(T,Xs, Y, N_A)}

\text{clause 4 : } r.⊤d_1(Xs, Y, A) ←
\text{Xs = [X][T,Xs],}
\text{nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_i),}
\text{r(TX_1, TY_1), \ldots, r(TX_i, TY_i),}
\text{I_{i+1} = e,}
\text{compose(TY_1, I_{i+1}, I_1), \ldots, compose(TY_p, I_{p+1}, I_p),}
\text{process(HX, HHY), compose(HHY, I_p, I_{p-1}),}
\text{compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0),}
\text{HY = I_0,}
\text{compose(A, HY, N_A), r.⊤d_1(T,Xs, Y, N_A)}

By introducing

\((\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_i))\lor\)
\((\text{nonMinimal}(TX_1) \land \ldots \land \text{nonMinimal}(TX_{p-1})) \lor \text{nonMinimal}(TX_p) \lor \ldots \lor \text{nonMinimal}(TX_i))\lor\)
\((\text{nonMinimal}(TX_1) \land \ldots \land \text{nonMinimal}(TX_{p-1})) \land \text{nonMinimal}(TX_p) \land \ldots \land \text{nonMinimal}(TX_i))\lor\)
\((\text{nonMinimal}(TX_1) \land \ldots \land \text{nonMinimal}(TX_{p-1})) \land \text{nonMinimal}(TX_p) \land \ldots \land \text{nonMinimal}(TX_i))\)

in clause 4, using applicability condition (4):

\text{clause 5 : } r.⊤d_1(Xs, Y, A) ←
\text{Xs = [X][T,Xs],}
\text{nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_i),}
\text{minimal(TX_1), \ldots, minimal(TX_i),}
\text{r(TX_1, TY_1), \ldots, r(TX_i, TY_i),}
\text{I_{i+1} = e,}
\text{compose(TY_1, I_{i+1}, I_1), \ldots, compose(TY_p, I_{p+1}, I_p),}
\text{process(HX, HHY), compose(HHY, I_p, I_{p-1}),}
\text{compose(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(TY_1, I_1, I_0),}
\text{HY = I_0,}
\text{compose(A, HY, N_A), r.⊤d_1(T,Xs, Y, N_A)}
\textbf{clause 6 :} \( \mathcal{R} \mathcal{D}_1(X, Y, A) \leftarrow \)
\[
\text{Xs = } [X[TXs],
\text{\ nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\text{\ minimal}(TX_1), \ldots, \text{\ minimal}(TX_{p-1}),
\{ \text{\ nonMinimal}(TX_p) \}; \ldots; \text{\ nonMinimal}(TX_t)].
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
I_{t+1} = I_t
\]
\[
\text{\ compose}(TY_1, I_{t+1}, I_t), \ldots, \text{\ compose}(TY_p, I_{p+1}, I_p),
\text{\ process}(HX, HHY), \text{\ compose}(HHY, I_p, I_{p-1}),
\text{\ compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{\ compose}(TY_1, I_1, I_0).
\]
\[
HY = I_0.
\]
\[
\text{\ compose}(A, H, Y, NA), \mathcal{R} \mathcal{D}_1(TXs, Y, NA)
\]

\textbf{clause 7 :} \( \mathcal{R} \mathcal{D}_1(X, Y, A) \leftarrow \)
\[
\text{Xs = } [X[TXs],
\text{\ nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\{ \text{\ nonMinimal}(TX_1) \}; \ldots; \text{\ nonMinimal}(TX_{p-1}),
\text{\ minimal}(TX_p), \ldots, \text{\ minimal}(TX_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
I_{t+1} = I_t
\]
\[
\text{\ compose}(TY_1, I_{t+1}, I_t), \ldots, \text{\ compose}(TY_p, I_{p+1}, I_p),
\text{\ process}(HX, HHY), \text{\ compose}(HHY, I_p, I_{p-1}),
\text{\ compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{\ compose}(TY_1, I_1, I_0).
\]
\[
HY = I_0.
\]
\[
\text{\ compose}(A, H, Y, NA), \mathcal{R} \mathcal{D}_1(TXs, Y, NA)
\]

\textbf{clause 8 :} \( \mathcal{R} \mathcal{D}_1(X, Y, A) \leftarrow \)
\[
\text{Xs = } [X[TXs],
\text{\ nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\{ \text{\ nonMinimal}(TX_1) \}; \ldots; \text{\ nonMinimal}(TX_{p-1}),
\{ \text{\ nonMinimal}(TX_p) \}; \ldots; \text{\ nonMinimal}(TX_t)).
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
I_{t+1} = I_t
\]
\[
\text{\ compose}(TY_1, I_{t+1}, I_t), \ldots, \text{\ compose}(TY_p, I_{p+1}, I_p),
\text{\ process}(HX, HHY), \text{\ compose}(HHY, I_p, I_{p-1}),
\text{\ compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{\ compose}(TY_1, I_1, I_0).
\]
\[
HY = I_0.
\]
\[
\text{\ compose}(A, H, Y, NA), \mathcal{R} \mathcal{D}_1(TXs, Y, NA)
\]

By \( t \) times unfolding clause 5 \( \mathcal{R} \mathcal{D}_1(X, TX_1, \ldots, TX_t) \) using \( DCRL \), and simplifying using condition (4):

\textbf{clause 9 :} \( \mathcal{R} \mathcal{D}_1(X, Y, A) \leftarrow \)
\[
\text{Xs = } [X[TXs],
\text{\ nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\text{\ minimal}(TX_1), \ldots, \text{\ minimal}(TX_t),
\text{\ minimal}(TX_p), \ldots, \text{\ minimal}(TX_t),
\]
\[
\text{\ solve}(TX_1, TY_1), \ldots, \text{\ solve}(TX_t, TY_t),
I_{t+1} = I_t
\]
\[
\text{\ compose}(TY_1, I_{t+1}, I_t), \ldots, \text{\ compose}(TY_p, I_{p+1}, I_p),
\text{\ process}(HX, HHY), \text{\ compose}(HHY, I_p, I_{p-1}),
\text{\ compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{\ compose}(TY_1, I_1, I_0).
\]
\[
HY = I_0.
\]
\[
\text{\ compose}(A, H, Y, NA), \mathcal{R} \mathcal{D}_1(TXs, Y, NA)
\]

By using applicability condition (3):
clause 10: $\tau d_1(Xs, Y, A) \leftarrow$

\[ Xs = [X [TXs]]. \]

nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_t$).

minimal($TX_1$), \ldots, minimal($TX_t$).

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($A, HY, NA$), $\tau d_1(TXs, Y, NA)$

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

clause 11: $\tau d_1(Xs, Y, A) \leftarrow$

\[ Xs = [X [TXs]]. \]

nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_t$).

minimal($TX_1$), \ldots, minimal($TX_t$).

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($A, HY, NA$), $\tau d_1(TXs, Y, NA)$

By using applicability condition (2):

clause 12: $\tau d_1(Xs, Y, A) \leftarrow$

\[ Xs = [X [TXs]]. \]

nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_t$).

minimal($TX_1$), \ldots, minimal($TX_t$).

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($A, HY, NA$), $\tau d_1(TXs, Y, NA)$

By simplification:

clause 13: $\tau d_1(Xs, Y, A) \leftarrow$

\[ Xs = [X [TXs]]. \]

nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_t$).

minimal($TX_1$), \ldots, minimal($TX_t$).

process($HX, HHY$), compose($A, HY, NA$).

$\tau d_1(TXs, Y, NA)$

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):
\[ \text{clause 14: } \text{r.td}_1(X_s, Y, A) \rightarrow \]
\[ X_s = [X|TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HXY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_p-I_p, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HY, NA), \text{r.td}_1(TX_s, Y, NA) \]

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

\[ \text{clause 15: } \text{r.td}_1(X_s, Y, A) \rightarrow \]
\[ X_s = [X|TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HXY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_p-I_p, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HY, NA), \text{r.td}_1(TX_s, Y, NA) \]

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

\[ \text{clause 16: } \text{r.td}_1(X_s, Y, A) \rightarrow \]
\[ X_s = [X|TX_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t), \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HXY), \text{compose}(I_{p-1}, HHY, I_p), HY = I_p, \]
\[ \text{compose}(A, HY, NA), \]
\[ \text{compose}(TY_p, TY_{p+1}, I_{p+1}), \]
\[ \text{compose}(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, \text{compose}(I_{t-1}, TY_1, I_t), \]
\[ \text{compose}(NA, I_t, NNA), \]
\[ \text{r.td}_1(TX_s, Y, NNA) \]

By \( t - p \) times folding clause 16 using clauses 1 and 2:
clause 17:  \[ \text{r.td}_1(X, Y, A) \rightarrow \]
\[ X_s = [X[TX_s]]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}). \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \]
\[ \text{solute}(TX_1, TY_1), \ldots, \text{solute}(TX_{p-1}, TY_{p-1}), \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p), HY = I_p. \]
\[ \text{compose}(A, HY, NA), \]
\[ \text{r.td}_1([TX_p, \ldots, TX_t][TX_s], Y, NA) \]

By using applicability condition (3):

clause 18:  \[ \text{r.td}_1(X, Y, A) \rightarrow \]
\[ X_s = [X[TX_s]]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}). \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \]
\[ \text{solute}(TX_1, e), \ldots, \text{solute}(TX_{p-1}, e), \]
\[ I_0 = e, \]
\[ \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}). \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p), HY = I_p. \]
\[ \text{compose}(A, HY, NA), \]
\[ \text{r.td}_1([TX_p, \ldots, TX_t][TX_s], Y, NA) \]

By using applicability condition (2):

clause 19:  \[ \text{r.td}_1(X, Y, A) \rightarrow \]
\[ X_s = [X[TX_s]]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}). \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \]
\[ \text{solute}(TX_1, e), \ldots, \text{solute}(TX_{p-1}, e), \]
\[ I_0 = e, \]
\[ I_1 = I_0, \ldots, I_{p-1} = I_{p-2}. \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p), HY = I_p. \]
\[ \text{compose}(A, HY, NA), \]
\[ \text{r.td}_1([TX_p, \ldots, TX_t][TX_s], Y, NA) \]

By simplification:

clause 20:  \[ \text{r.td}_1(X, Y, A) \rightarrow \]
\[ X_s = [X[TX_s]]. \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}). \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)). \]
\[ \text{process}(HX, HY), \text{compose}(A, HY, NA). \]
\[ \text{r.td}_1([TX_p, \ldots, TX_t][TX_s], Y, NA) \]

By introducing atoms \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_{p-1} \)) in clause 7:
\textbf{clause 21: } \textit{r.td}_1(Xs, Y, A) \leftarrow \\
\quad Xs = [X|TXs]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}). \\
\quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \\
\quad I_{t+1} = e, \\
\quad \text{compose}(TY_{t}, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\quad \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \\
\quad \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_t, I_t, I_0). \\
\quad HY = I_0, \\
\quad \text{compose}(A, HY, NA), \textit{r.td}_1(Xs, Y, NA)

\text{By using applicability condition (3):}

\textbf{clause 22: } \textit{r.td}_1(Xs, Y, A) \leftarrow \\
\quad Xs = [X|TXs]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \ldots, \text{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 \text{compose}(TY_{t}, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\quad \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \\
\quad \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_t, I_t, I_0). \\
\quad HY = I_0, \\
\quad \text{compose}(A, HY, NA), \textit{r.td}_1(Xs, Y, NA)

\text{By using applicability condition (2):}

\textbf{clause 23: } \textit{r.td}_1(Xs, Y, A) \leftarrow \\
\quad Xs = [X|TXs]. \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \\
\quad (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})). \\
\quad \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t). \\
\quad \text{minimal}(U_1), \ldots, \text{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 \text{compose}(TY_{t}, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p). \\
\quad \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \\
\quad \text{compose}(e, I_{p-1}, K_1), \text{compose}(e, K_1, K_2), \ldots, \text{compose}(e, K_{p-2}, K_{p-1}), \\
\quad \text{compose}(TY_{p-1}, K_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_t, I_t, I_0). \\
\quad HY = I_0, \\
\quad \text{compose}(A, HY, NA), \textit{r.td}_1(Xs, Y, NA)

\text{By using applicability conditions (1) and (2):}
clause 24: \[ rTd_1(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(e, I_p, I_{p-1}), \ldots, compose(e, I_1, I_0). \]
\[ HY = I_0, \]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \]
\[ rTd_1(TXs, Y, NNA). \]

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

clause 25: \[ rTd_1(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0). \]
\[ HY = I_0, \]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \]
\[ rTd_1(TXs, Y, NNA). \]

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

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

always holds (because \( N \) is existentially quantified)

clause 26: \[ rTd_1(Xs, Y, A) \rightarrow \]
\[ Xs = [X|TXs]. \]
\[ nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t). \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})). \]
\[ minimal(TX_p), \ldots, minimal(TX_t). \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ nonMinimal(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t). \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t). \]
\[ I_{t+1} = e, \]
\[ compose(TY_t, I_{t+1}, I_t), \ldots, compose(TY_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_{p-1}), \]
\[ compose(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, compose(YU_1, I_1, I_0). \]
\[ HY = I_0, \]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(A, K_{p-2}, NA), compose(NA, HY, NNA), \]
\[ rTd_1(TXs, Y, NNA). \]

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t) \):
clause 27: \[ r_1 d_1 (X, Y, A) \]
\[ X = [X|T X s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \]
\[ \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \]
\[ \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \]
\[ r(U_1, Y U_1), \ldots, r(U_{p-1}, Y U_{p-1}) \]
\[ \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t) \]
\[ \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t) \]
\[ r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t) \]
\[ I_{t+1} = e \]
\[ \text{compose}(T Y_1, I_{t+1}, I_t), \ldots, \text{compose}(T Y_p, I_{p+1}, I_p) \]
\[ \text{process}(H X, H H Y), \text{compose}(H H Y, I_p, I_{p-1}) \]
\[ \text{compose}(Y U_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(Y U_1, I_1, I_0) \]
\[ H Y = I_0 \]
\[ \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_p, T Y_{p-1}, K_{p-2}) \]
\[ \text{compose}(A, K_{p-2}, N A), \text{compose}(N A, H Y, N N A) \]
\[ r_1 d_1 ([X|T X s, Y, N N A]) \]

By folding clause 27 using DCRL:

clause 28: \[ r_1 d_1 (X, Y, A) \]
\[ X = [X|T X s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \]
\[ \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \]
\[ \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \]
\[ \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t) \]
\[ r(T X_1, T Y_1), \ldots, r(T X_{p-1}, T Y_{p-1}), r(N, H Y) \]
\[ \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}) \]
\[ \text{compose}(A, K_{p-2}, N A), \text{compose}(N A, H Y, N N A) \]
\[ r_1 d_1 ([X|T X s, Y, N N A]) \]

By folding clause 28 using clauses 1 and 2:

clause 29: \[ r_1 d_1 (X, Y, A) \]
\[ X = [X|T X s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \]
\[ \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \]
\[ \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \]
\[ \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t) \]
\[ r(T X_1, T Y_1), \ldots, r(T X_{p-1}, T Y_{p-1}), r(N, H Y) \]
\[ \text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_{p-1}, K_{p-2}) \]
\[ \text{compose}(A, K_{p-2}, N A), \text{compose}(N A, H Y, N N A) \]
\[ r_1 d_1 ([N|T X s, Y, N A]) \]

By \( p - 1 \) times folding clause 29 using clauses 1 and 2:

clause 30: \[ r_1 d_1 (X, Y, A) \]
\[ X = [X|T X s] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t) \]
\[ \text{(nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})) \]
\[ \text{minimal}(T X_p), \ldots, \text{minimal}(T X_t) \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \]
\[ \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t) \]
\[ r_1 d_1 ([T X_1, \ldots, T X_{p-1}, N|T X s, Y, A]) \]

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: \( r \vdash_1 (Xs, Y, A) \leftarrow \)
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HY, NA), r \vdash_1 (TXs, Y, NA) \]

By using applicability condition (3):

clause 32: \( r \vdash_1 (Xs, Y, A) \leftarrow \)
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \]
\[ r(U_1, e), \ldots, r(U_t, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HY, NA), r \vdash_1 (TXs, Y, NA) \]

By using applicability condition (2):

clause 33: \( r \vdash_1 (Xs, Y, A) \leftarrow \)
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \]
\[ r(U_1, e), \ldots, r(U_t, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ I_{t+1} = e, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{compose}(e, I_p, K_{t+1}), \]
\[ \text{compose}(e, K_{t+1}, K_t), \ldots, \text{compose}(e, K_{p+1}, K_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, K_p, K_{p-1}), \]
\[ \text{compose}(e, K_{p-1}, K_{p-2}), \ldots, \text{compose}(e, K_1, K_0), \]
\[ \text{compose}(e, K_0, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HY, NA), r \vdash_1 (TXs, Y, NA) \]

By using applicability conditions (1) and (2):
clause 34: \[ r \cdot \text{ld}_1(Xs, Y, A) = Xs = [X[TXs]]. \]

nonMinimal(X), decompose(X, HX, TX1, ..., TX1).
nonMinimal(TX1); ..., nonMinimal(TXp-1)).
nonMinimal(TXp); ..., nonMinimal(TX1)),
minimal(U1), ..., minimal(U1).
\[ r(U_1, e), ..., r(U_i, e), \]
\[ r(TX1, TY1), ..., r(TX1, TY1). \]
compose(e, I_{i+1}, I_i), ..., 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}), ..., compose(e, I_1, I_0).

\[ NHY = I_0, \]
\[ compose(TY1, TY2, K_1), compose(K_1, TY3, K_2), ..., compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), ..., compose(K_{t-2}, TY_t, K_{t-1}). \]
\[ compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \]
\[ compose(NA_2, K_{t-1}, NA), r \cdot \text{ld}_1(TXs, Y, NA) \]

By introducing new, i.e. existentially quantified, variables YU1, ..., YUt in place of some occurrences of e:

clause 35: \[ r \cdot \text{ld}_1(Xs, Y, A) = Xs = [X[TXs]]. \]

nonMinimal(X), decompose(X, HX, TX1, ..., TX1).
nonMinimal(TX1); ..., nonMinimal(TXp-1)).
nonMinimal(TXp); ..., nonMinimal(TX1)),
minimal(U1), ..., minimal(U1).
\[ r(U_1, YU_1), ..., r(U_i, YU_i), \]
\[ r(TX1, TY1), ..., r(TX1, TY1). \]
\[ I_{i+1} = e, \]
\[ compose(YU_1, I_{i+1}, I_i), ..., compose(YU_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_p-1), \]
\[ compose(YU_{p-1}, I_{p-1}, I_{p-2}), ..., compose(YU_1, I_1, I_0) \]
\[ NHY = I_0. \]
\[ compose(TY1, TY2, K_1), compose(K_1, TY3, K_2), ..., compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), ..., compose(K_{t-2}, TY_t, K_{t-1}). \]
\[ compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \]
\[ compose(NA_2, K_{t-1}, NA), r \cdot \text{ld}_1(TXs, Y, NA) \]

By introducing nonMinimal(N) and decompose(N, HX, U1, ..., U1), since
\[ \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U1, ..., U1) \]
always holds (because N is existentially quantified)

clause 36: \[ r \cdot \text{ld}_1(Xs, Y, A) = Xs = [X[TXs]]. \]

nonMinimal(X), decompose(X, HX, TX1, ..., TX1).
nonMinimal(TX1); ..., nonMinimal(TXp-1)).
nonMinimal(TXp); ..., nonMinimal(TX1)),
minimal(U1), ..., minimal(U1).
\[ r(U_1, YU_1), ..., r(U_i, YU_i), \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U1, ..., U1). \]
\[ r(TX1, TY1), ..., r(TX1, TY1). \]
\[ I_{i+1} = e, \]
\[ compose(YU_1, I_{i+1}, I_i), ..., compose(YU_p, I_{p+1}, I_p). \]
\[ process(HX, HHY), compose(HHY, I_p, I_p-1), \]
\[ compose(YU_{p-1}, I_{p-1}, I_{p-2}), ..., compose(YU_1, I_1, I_0) \]
\[ NHY = I_0, \]
\[ compose(TY1, TY2, K_1), compose(K_1, TY3, K_2), ..., compose(K_{p-3}, TY_{p-1}, K_{p-2}). \]
\[ compose(TY_p, TY_{p+1}, K_p), compose(K_p, TY_{p+2}, K_{p+1}), ..., compose(K_{t-2}, TY_t, K_{t-1}). \]
\[ compose(A, K_{p-2}, NA_1), compose(NA_1, NHY, NA_2), \]
\[ compose(NA_2, K_{t-1}, NA), r \cdot \text{ld}_1(TXs, Y, NA) \]
By duplicating goal decompose(\(N, HX, U_1, \ldots, U_t\)):

\[\text{clause } 37: \quad r.td_1(Xs, Y, A) \leftarrow\]
\[Xs = X[TXs], \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \quad \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}), \quad \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t), \quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \quad r(U_1, YU_1), \ldots, r(U_t, YU_t), \quad \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t), \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \quad I_{t+1} = \epsilon, \quad \text{compose}(YU_1, I_{t+1}, I_t), \ldots, \text{compose}(YU_p, I_{p+1}, I_p), \quad \text{compose}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \quad \text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0), \quad NHY = I_0, \quad \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \quad \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \quad \text{compose}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \quad \text{compose}(NA_2, K_{t-1}, NA), r.td_1(TXS, Y, NA)\]

By folding clause 37 using DCHR:

\[\text{clause } 38: \quad r.td_1(Xs, Y, A) \leftarrow\]
\[Xs = X[TXs], \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \quad \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}), \quad \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t), \quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \quad \text{decompose}(N, HX, U_1, \ldots, U_t), \quad r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \quad r(N, NHY), \quad \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \quad \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \quad \text{compose}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \quad \text{compose}(NA_2, K_{t-1}, NA), r.td_1(TXS, Y, NA)\]

By \(t - p + 1\) times folding clause 38 using clauses 1 and 2:

\[\text{clause } 39: \quad r.td_1(Xs, Y, A) \leftarrow\]
\[Xs = X[TXs], \quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \quad \text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1}), \quad \text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t), \quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \quad \text{decompose}(N, HX, U_1, \ldots, U_t), \quad r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), \quad r(N, NHY), \quad \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \quad \text{compose}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \quad r.td_1([TX_p, \ldots, TX_t]TXs), Y, NA_2\]

By folding clause 39 using clauses 1 and 2:
clause 40: $r.d_1(X,Y,A) = 
X_s = [X[TXs]],
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\text{minimal}(U_1), \ldots, \text{minimal}(U_t)
\text{decompose}(N, HX, U_1, \ldots, U_t),
r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}),
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}),
\text{compose}(A, K_{p-2}, NA_1),
r.d_1([N, TX_p, \ldots, TX_t TXs], Y, NA_1)$. 

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

clause 41: $r.d_1(X,Y,A) = 
X_s = [X[TXs]],
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\text{minimal}(U_1), \ldots, \text{minimal}(U_t)
\text{decompose}(N, HX, U_1, \ldots, U_t),
r.d_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t TXs], Y, A)$.  

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of $P_{r.d_4}$. Therefore $P_{r.d_4}$ is steadfast wrt $S_{r.d_4}$ in $S$.

To prove that $P_r$ is steadfast wrt $S_r$ in $\{S_{r.d_4}\}$, 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:

$r(X,Y) \leftarrow r.d_1([X], Y, e)$

By taking the ‘completion’:

$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow r.d_1([X], Y, e)]$

By unfolding the ‘completion’ above wrt $r.d_1([X], Y, e)$ using $S_{r.d_4}$:

$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow \exists Y_1, I_1 : \mathcal{Y}. \ O_r(X,Y_1) \land I_1 = Y_1 \land O_e(e, I_1, Y)]$

By using applicability condition (2):

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

By simplification:

$\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)]$.

We obtain $S_r$, so $P_r$ is steadfast wrt $S_r$ in $\{S_{r.d_4}\}$.

Therefore, $TDGLR$ is also steadfast wrt $S_r$ in $S$. \hfill $\square$

5 Proofs of the Duality Schemas

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

$D_{dc} : (DCLR, DCRL, A_{dc}, O_{dc12}, O_{dc21})$ where

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

$O_{dc12} : - \text{partial evaluation of the conjunction}$
$\text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1})$
$\text{results in the introduction of a non-recursive relation}$

$O_{dc21} : - \text{partial evaluation of the conjunction}$
$\text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p)$
$\text{results in the introduction of a non-recursive relation}$
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:

$$\forall X : X. \forall Y : Y. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)]$$

**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 $A_{dc}$. By Definition 5, the templates $DCLR$ and $DCRL$ are equivalent wrt $S_r$ under the applicability conditions $A_{dc}$ if $DCLR$ is equivalent to $DCRL$ wrt the specification $S_r$ provided that the conditions in $A_{dc}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{process}}, S_{\text{decompose}}, S_{\text{compose}}$ are the specifications of $\text{minimal}$, $\text{nonMinimal}$, $\text{solve}$, $\text{decompose}$, $\text{process}$, $\text{compose}$, which are all the undefined relation names appearing in $DCLR$.

(b) $DCRL$ is also steadfast wrt $S_r$ in $S$.

Note that the sets $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_l\}$ in Definition 4 are equal to $S$ 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 $S$, 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 $S$ 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 $S$ and $P_S$ contains no occurrences of the relation $r$.

To prove that $DCRL$ is steadfast wrt $S_r$ in $S$, 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 ‘decomposition’ of $S_r$:

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

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 $S$:

**clause 2**: $r(X,Y) \leftarrow$

\[
\text{minimal}(X), \\
\text{solve}(X,Y)
\]

**clause 3**: $r(X,Y) \leftarrow$

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_0 = e, \\
\text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
\text{process}(HX, HY), \text{compose}(l_{p-1}, HY, l_p), \\
\text{compose}(l_p, TY_r, l_{p+1}), \ldots, \text{compose}(l_{t_i}, TY_{t_i}, l_{t_i+1}), \\
Y = l_{t_i+1}
\]

By using applicability condition (1) on clause 3:

**clause 4**: $r(X,Y) \leftarrow$

\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_{t-1}, TY_t, A_{t-1}), \\
\text{compose}(TY_{t-2}, A_{t-1}, A_{t-2}), \ldots, \text{compose}(TY_p, A_{p+1}, A_p), \\
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \\
\text{compose}(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TY_1, A_1, A_0), \\
\text{compose}(e, A_0, Y)
\]

By using applicability conditions (1) and (2) on clause 4:
clause 5: \[ r(X,Y) \leftarrow \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_2, e, A_t) \\
\text{compose}(TY_{t-1}, A_t, A_{t-1}), \ldots, \text{compose}(TY_p, A_{p+1}, A_p). \\
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \\
\text{compose}(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TY_1, A_1, A_0), \\
Y = A_t \\
\]

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

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

Clauses 2 and 6 are the clauses of \( DCRL \).

Therefore \( DCRL \) is steadfast wrt \( S_r \) in \( S \).

**Theorem 12** The duality schema \( D_{dd} \), which is given below, is correct.

\[ D_{dd} : (\text{DGLR, DGRL, } A_{dd}, O_{dd1}, O_{dd2}) \text{ where} \\
A_{dd} : \begin{array}{l}
(1) \text{compose is associative} \\
(2) \text{compose has } e \text{ as the left and right identity element,}
\end{array} \\
O_{dd1} : \begin{array}{l}
- I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e) \\
- \text{partial evaluation of the conjunction} \\
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}) \\
\text{results in the introduction of a non-recursive relation}
\end{array} \\
O_{dd2} : \begin{array}{l}
- I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e) \\
- \text{partial evaluation of the conjunction} \\
\text{process}(HX, HY), \text{compose}(A_{p-1}, HY, A_p) \\
\text{results in the introduction of a non-recursive relation}
\end{array} \\
\]

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:

\[ \forall X : X. \forall Y : Y. \ I_r(X) \Rightarrow [r(X,Y) \Leftrightarrow O_r(X,Y)] \]

The specification \( S_{r,\text{descending}_1} \) of relation \( r_{\text{descending}_1} \) is:

\[ \forall X : X. \forall Y, A : Y. \ I_r(X) \Rightarrow [r_{\text{descending}_1}(X,Y,A) \Leftrightarrow \exists S : Y. O_r(X,S) \land O_r(A,S,Y)] \]

The specification \( S_{r,\text{descending}_2} \) of relation \( r_{\text{descending}_2} \) is:

\[ \forall X : X. \forall Y, A : Y. \ I_r(X) \Rightarrow [r_{\text{descending}_2}(X,Y,A) \Leftrightarrow \exists S : Y. O_r(X,S) \land O_r(S,A,Y)] \]

**Proof 12** To prove the correctness of the duality schema \( D_{dd} \), by Definition 10, we have to prove that templates \( DGLR \) and \( DGRL \) are equivalent wrt \( S_r \), under the applicability conditions \( A_{dd} \). By Definition 5, the templates \( DGLR \) and \( DGRL \) are equivalent wrt \( S_r \), under the applicability conditions \( A_{dd} \) iff \( DGLR \) is equivalent to \( DGRL \) wrt the specification \( S_r \) provided that the conditions in \( A_{dd} \) 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 \( S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \), where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{process}}, S_{\text{decompose}}, S_{\text{compose}} \) are the specifications of \( \text{minimal}, \text{nonMinimal}, \text{solve}, \text{decompose}, \text{process}, \text{compose} \), which are all the undefined relation names appearing in \( DGLR \).

(b) \( DGRL \) is also steadfast wrt \( S_r \) in \( S \).
Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S'_1, \ldots, S'_l\} \) in Definition 4 are equal to \( S \) 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 \( S \), 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 \( S \) if \( P_{r, \text{descending}^2} \) is steadfast wrt \( S_{r, \text{descending}^2} \) in \( S \), where \( P_{r, \text{descending}^2} \) is the procedure for \( r_{\text{descending}^2} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r, \text{descending}^2}\} \), where \( P_r \) is the procedure for \( r \).

To prove that \( P_{r, \text{descending}^2} \) is steadfast wrt \( S_{r, \text{descending}^2} \) in \( S \), we do a constructive forward proof that we begin with \( S_{r, \text{descending}^2} \) and from which we try to obtain \( P_{r, \text{descending}^2} \).

By taking the ‘decomposition’ of \( S_{r, \text{descending}^2} \):

\[
\text{clause 1: } r_{\text{descending}^2}(X, Y, A) \leftarrow r(X, S), \text{compose}(S, A, Y)
\]

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

\[
\text{clause 2: } r_{\text{descending}^2}(X, Y, A) \leftarrow r_{\text{descending}^1}(X, S, e), \text{compose}(S, A, Y)
\]

By unfolding clause 2 wrt \( r_{\text{descending}^1}(X, S, e) \) using \( DGLR \), and using the assumption that \( P_{r, \text{descending}^1} \) is steadfast wrt \( S_{r, \text{descending}^1} \) in \( S \), since \( P_{r, \text{descending}^1} \) must be steadfast wrt \( S_{r, \text{descending}^1} \) in \( S \) for \( DGLR \) to be steadfast wrt \( S_r \) in \( S \):

\[
\text{clause 3: } r_{\text{descending}^2}(X, Y, A) \leftarrow \\
\text{minimal}(X), \\
\text{solve}(X, S'), \text{compose}(e, S', S), \text{compose}(S, A, Y)
\]

\[
\text{clause 4: } r_{\text{descending}^2}(X, Y, A) \leftarrow \\
\text{nonMinimal}(X), \text{decompose}(X, hX, TX_1, \ldots, TX_t), \\
\text{compose}(e, e, A_0), \\
r_{\text{descending}^1}(TX_1, A_1, A_0), \ldots, r_{\text{descending}^1}(TX_{p-1}, A_{p-1}, A_{p-2}), \\
\text{process}(hX, hS), \text{compose}(A_{p-1}, hS, A_p), \\
r_{\text{descending}^1}(TX_p, A_{p+1}, A_p), \ldots, r_{\text{descending}^1}(TX_t, A_{t+1}, A_t), \\
S = A_{t+1}, \text{compose}(S, A, Y)
\]

By using applicability condition (2) on clause 3:

\[
\text{clause 5: } r_{\text{descending}^2}(X, Y, A) \leftarrow \\
\text{minimal}(X), \\
\text{solve}(X, S'), S = S, \text{compose}(S, A, Y)
\]

By simplification:

\[
\text{clause 6: } r_{\text{descending}^2}(X, Y, A) \leftarrow \\
\text{minimal}(X), \\
\text{solve}(X, S), \text{compose}(S, A, Y)
\]

By \( t \) times unfolding clause 4 wrt the atoms \( r_{\text{descending}^1}(TX_1, A_1, A_0), \ldots, r_{\text{descending}^1}(TX_{p-1}, A_{p-1}, A_{p-2}), \ldots, r_{\text{descending}^1}(TX_t, A_{t+1}, A_t) \) using the ‘decomposition’ of \( S_{r, \text{descending}^1} \) (refer to Proofs 3 and 6):

\[
\text{clause 7: } r_{\text{descending}^2}(X, Y, A) \leftarrow \\
\text{nonMinimal}(X), \text{decompose}(X, hX, TX_1, \ldots, TX_t), \\
\text{compose}(e, e, A_0), \\
r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \\
\text{compose}(A_0, TS_1, A_1), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-1}), \\
\text{process}(hX, hS), \text{compose}(A_{p-1}, hS, A_p), \\
r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \\
\text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \\
S = A_{t+1}, \text{compose}(S, A, Y)
\]

By using applicability condition (1) on clause 7:
clause 8: \( r_{\text{descending}}(X, Y, A) \leftarrow \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{compose}(e, I, Y), \text{compose}(e, A_0, I), \)
\( r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \)
\( \text{compose}(TS_1, A_1, A_0), \ldots, \text{compose}(TS_{p-1}, A_{p-1}, A_{p-2}), \)
\( \text{process}(HX, HS), \text{compose}(HS, A_p, A_{p-1}), \)
\( r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \)
\( \text{compose}(TS_p, A_{p+1}, A_p), \ldots, \text{compose}(TS_t, A_{t+1}, A_t), \)
\( \text{compose}(e, A, A_{t+1}) \)

By using applicability condition (2):

clause 9: \( r_{\text{descending}}(X, Y, A) \leftarrow \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( Y = A_0, \)
\( r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \)
\( \text{compose}(TS_1, A_1, A_0), \ldots, \text{compose}(TS_{p-1}, A_{p-1}, A_{p-2}), \)
\( \text{process}(HX, HS), \text{compose}(HS, A_p, A_{p-1}), \)
\( r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \)
\( \text{compose}(TS_p, A_{p+1}, A_p), \ldots, \text{compose}(TS_t, A_{t+1}, A_t), \)
\( \text{compose}(e, A, A_{t+1}) \)

By \( t \) times folding clause 9 using clause 1:

clause 10: \( r_{\text{descending}}(X, Y, A) \leftarrow \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{compose}(e, A, A_{t+1}), \)
\( r_{\text{descending}}(TX_1, A_1, A_{t+1}), \ldots, r_{\text{descending}}(TX_p, A_p, A_{p+1}), \)
\( \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \)
\( r_{\text{descending}}(TX_{p-1}, A_{p-2}, A_{p-1}), \ldots, r_{\text{descending}}(TX_1, A_0, A_1), \)
\( Y = A_0 \)

Clauses 2 and 10 are the clauses of \( P_r_{\text{descending}} \). Therefore \( P_r_{\text{descending}} \) is steadfast wrt \( S_r_{\text{descending}} \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \{S_r_{\text{descending}}\}, 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:
\[ r(X, Y) \leftarrow r_{\text{descending}}(X, Y, e) \]

By taking the ‘completion’:
\[ \forall X : \mathcal{A}, \forall Y : \mathcal{Y}. \hspace{1em} \tau_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r_{\text{descending}}(X, Y, e)] \]

By unfolding the ‘completion’ above wrt \( r_{\text{descending}}(X, Y, e) \) using \( S_r_{\text{descending}} \):
\[ \forall X : \mathcal{A}, \forall Y : \mathcal{Y}. \hspace{1em} \tau_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists S : \mathcal{Y}. \hspace{1em} \sigma_r(X, S) \land \sigma_e(S, e, Y)] \]

By using applicability condition (2):
\[ \forall X : \mathcal{A}, \forall Y : \mathcal{Y}. \hspace{1em} \tau_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists S : \mathcal{Y}. \hspace{1em} \sigma_r(X, S) \land S = Y] \]

By simplification:
\[ \forall X : \mathcal{A}, \forall Y : \mathcal{Y}. \hspace{1em} \tau_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \sigma_r(X, Y)] \]

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \{S_r_{\text{descending}}\}.

Therefore, \( DGRL \) is also steadfast wrt \( S_r \) in \( S \). \( \square \)

**Theorem 13** The duality schema \( D_{\text{dtd}} \), which is given below, is correct.

\[ D_{\text{dtd}} : \{ TDGLR, TDGRL, A_{\text{dtd}}, O_{\text{dtd}12}, O_{\text{dtd}21} \} \] where
\[ A_{\text{dtd}} : (1) \text{ compose is associative} \]
\[ (2) \text{ compose has } e \text{ as the left and right identity element} \]
where \( e \) appears in TDGLR and TDGRL

\[
O_{\text{tdg12}}: \forall X : \mathcal{X}, \text{I}_r(X) \land \text{minimal}(X) \Rightarrow \mathcal{O}_r(X, e)
\]
- partial evaluation of the conjunction

\[
\text{process}(\text{HX, HY}), \text{compose}(\text{HY, NewA, F})
\]
results in the introduction of a non-recursive relation

\[
O_{\text{tdg21}}: \forall X : \mathcal{X}, \text{I}_r(X) \land \text{minimal}(X) \Rightarrow \mathcal{O}_r(X, e)
\]
- partial evaluation of the conjunction

\[
\text{process}(\text{HX, HY}), \text{compose}(\text{A, HY, NewA, F})
\]
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 of \( S_r \) of relation \( r \) is:

\[
\forall X : \mathcal{X}, \forall Y : \mathcal{Y}, \text{I}_r(X) \Rightarrow [r(X, Y) \iff \mathcal{O}_r(X, Y)]
\]

The specification of \( r_{\text{Id1}} \), namely \( S_{\text{Id1}} \), is:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow \text{I}_r(X)) \Rightarrow [r_{\text{Id1}}(Xs, Y, A) \iff (Xs = [] \land Y = A) \\
\lor (Xs = [X_1, X_2, \ldots, X_n] \land \bigwedge_{i=1}^{n} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{n} \mathcal{O}_e(I_{i-1}, Y_i, I_i) \land \mathcal{O}_e(I_1, A, I_1) \land Y = I_{n+1})]
\]

The specification of \( r_{\text{Id2}} \), namely \( S_{\text{Id2}} \), is:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow \text{I}_r(X)) \Rightarrow [r_{\text{Id2}}(Xs, Y, A) \iff (Xs = [] \land Y = A) \\
\lor (Xs = [X_1, X_2, \ldots, X_n] \land \bigwedge_{i=1}^{n} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{n} \mathcal{O}_e(I_{i-1}, Y_i, I_i) \land \mathcal{O}_e(I_1, A, I_1) \land Y = I_{n+1})]
\]

**Proof 13** To prove the correctness of the duality schema \( D_{\text{tdg}} \), by Definition 10, we have to prove that templates TDGLR and TDGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{\text{tdg}} \). By Definition 5, the templates TDGLR and TDGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{\text{tdg}} \) if TDGLR is equivalent to TDGRL wrt the specification \( S_r \) provided that the conditions in \( A_{\text{tdg}} \) 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 \( S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \),

where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{process}}, S_{\text{decompose}}, S_{\text{compose}} \) are the specifications of minimal, nonMinimal, solve, decompose, process, compose, which are all the undefined relation names appearing in TDGLR.

(b) TDGLR is also steadfast wrt \( S_r \) in \( S \).

Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S'_1, \ldots, S'_l\} \) in Definition 4 are equal to \( S \) 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 \( S \), 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 \( S \) if \( P_{\text{r}, \text{tdg}} \) is steadfast wrt \( S_{\text{r}, \text{tdg}} \) in \( S \), where \( P_{\text{r}, \text{tdg}} \) is the procedure for \( r_{\text{Id2}} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{\text{r}, \text{tdg}}\} \), where \( P_r \) is the procedure for \( r \).

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

If we separate the cases of \( q \geq 1 \) by \( q = 1 \lor q \geq 2 \), then \( S_{\text{r}, \text{tdg}} \) becomes:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow \text{I}_r(X)) \Rightarrow [r_{\text{Id2}}(Xs, Y, A) \iff (Xs = [] \land Y = A) \\
\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, Y_1) \land I_1 = Y_1 \land \mathcal{O}_e(I_1, A, I_1) \land Y = I_2) \\
\lor (Xs = [X_1, X_2, \ldots, X_n] \land \bigwedge_{i=1}^{n} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{n} \mathcal{O}_e(I_{i-1}, Y_i, I_i) \land \mathcal{O}_e(I_1, A, I_1) \land Y = I_{n+1})]
\]

where \( q \geq 2 \).

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

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow \text{I}_r(X)) \Rightarrow [r_{\text{Id2}}(Xs, Y, A) \iff (Xs = [] \land Y = A) \\
\lor (Xs = [X_1] \land \bigwedge_{i=1}^{n} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land TY = A \land \mathcal{O}_e(TY, A, NA) \land \mathcal{O}_e(I_1, NA, Y)) \\
\lor (Xs = [X_1, X_2, \ldots, X_n] \land \bigwedge_{i=1}^{n} \mathcal{O}_r(X_i, Y_i) \land I_1 = Y_1 \land Y_2 = I_2 \land \bigwedge_{i=2}^{n} \mathcal{O}_e(I_{i-1}, Y_i, I_i) \land TY = I_1 \land \bigwedge_{i=2}^{n} \mathcal{O}_e(TY, A, NA) \land \mathcal{O}_e(I_1, NA, Y))]
\]

\[
\mathcal{O}_e(I_{n+1}, Y_{n+1})
\]
where \( q \geq 2 \).

By folding using \( S_{r,td2} \), and renaming:

\[
\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \ (YX : \mathcal{X}. \ X \in Xs \Rightarrow I_s(X)) \Rightarrow [r_{td2}(Xs, Y, A) \Leftarrow \\
\left[ Xs = [] \land Y = A \right] \land (Xs = [X[TXs]] \land \mathcal{O}_s(X, HY) \land r_{td2}(TXs, NA, A) \land \mathcal{O}_s(HY, NA, Y))]
\]

By taking the ‘decompletion’:

\[
\text{clause 1: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [], Y = A
\]
\[
\text{clause 2: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [X[TXs]], r(X, HY).
\]
\[
\quad r_{td2}(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

By unfolding clause 2 wrt \( r(X, HY) \) using \( TDGLR \), and using the assumption that \( DCRLR \) is steadfast wrt \( S_r \) in \( S \):

\[
\text{clause 3: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [X[TXs]], r_{td1}([X], HY, e),
\]
\[
\quad r_{td2}(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

By unfolding clause 3 wrt \( r_{td1}([X], HY, e) \) using \( TDGLR \), and using the assumption that \( P_{r,td} \), is steadfast wrt \( S_{r,td} \), in \( S \), since \( P_{r,td} \) must be steadfast wrt \( S_{r,td} \), in \( S \) for \( TDGLR \) to be steadfast wrt \( S_r \) in \( S \):

\[
\text{clause 4: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [X[TXs]],
\]
\[
\quad Xs' = [X[TXs'], TXs' = []],
\]
\[
\quad \text{minimal}(X), \text{solve}(X, HY'),
\]
\[
\quad \text{compose}(e, HY', NA'), r_{td1}(TXs', HY, NA').
\]
\[
\quad r_{td2}(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

\[
\text{clause 5: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [X[TXs]],
\]
\[
\quad Xs' = [X[TXs'], TXs' = []],
\]
\[
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t),
\]
\[
\quad \text{process}(HX, HY'), \text{compose}(e, HY', NA').
\]
\[
\quad r_{td1}(TXs', HY, NA').
\]
\[
\quad r_{td2}(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

\[
\text{clause 6: } r_{td2}(Xs, Y, A) \Leftarrow Xs = [X[TXs]],
\]
\[
\quad Xs' = [X[TXs'], TXs' = []],
\]
\[
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\quad \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}),
\]
\[
\quad \{\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)\},
\]
\[
\quad \text{process}(HX, HY'), \text{compose}(e, HY', NA').
\]
\[
\quad r_{td1}([TX_1, \ldots, TX_t], HY, NA').
\]
\[
\quad r_{td2}(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]
clause 7: \( r_{td_2}(Xs, Y, A) \) —
\[
Xs = [X | TXs], \\
Xs' = [X | TXs'], \ TXs' = [\], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\}. \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \\
r_{td_1}([TX_1, \ldots, TX_{p-1}, N | TXs'], HY, e). \\
r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)
\]

By unfolding clause 4 wrt \( r_{td_1}(TXs', HY, NA') \) using TDGLR, and using the assumption that \( P_{r_{td_i}} \) is steadfast wrt \( S_{r_{td_i}} \) in \( S' \):

clause 8: \( r_{td_2}(Xs, Y, A) \) —
\[
Xs = [X | TXs], \\
Xs' = [X | TXs'], \ TXs' = [\], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\{\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})\}. \\
\text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
decompose(N, HX, U_1, \ldots, U_t), \\
r_{td_1}([TX_1, \ldots, TX_{p-1}, N, TX_1, \ldots, TX_t | TXs'], HY, e), \\
r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)
\]

By using applicability condition (2):

clause 9: \( r_{td_2}(Xs, Y, A) \) —
\[
Xs = [X | TXs], \\
Xs' = [X | TXs'], \ TXs' = [\], \\
\text{minimal}(X), \text{solve}(X, HY'), \\
\text{compose}(e, HY', NA'), \\
TXs' = [\], HY = NA', \\
r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)
\]

By using simplification:

clause 10: \( r_{td_2}(Xs, Y, A) \) —
\[
Xs = [X | TXs], \\
Xs' = [X | TXs'], \ TXs' = [\], \\
\text{minimal}(X), \text{solve}(X, HY'), \\
HY' = NA', \\
TXs' = [\], HY = NA', \\
r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)
\]

By unfolding clause 5 wrt \( r_{td_1}(TXs', HY, NA') \) using TDGLR, and using the assumption that \( P_{r_{td_i}} \) is steadfast wrt \( S_{r_{td_i}} \) in \( S' \):

clause 11: \( r_{td_2}(Xs, Y, A) \) —
\[
Xs = [X | TXs], \\
\text{minimal}(X), \text{solve}(X, HY), \\
r_{td_2}(TXs, NA, A), \text{compose}(HY, NA, Y)
\]

By using applicability condition (2):
clause 13: \( r.ID_3(X_s, Y, A) \) —

\[
X_s = [X\lbrack TXs\rbrack, \\
TXs' = [X\lbrack TXs'\rbrack, TXs' = []]. \\
nonMinimal(X), \text{ decompose}(X, PX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t). \\
process(HX, HY'), HY' = NA', \\
TXs' = [], HY = NA', \\
r.ID_2(TXs, NA, A), \text{ compose}(HY, NA, Y)
\]

By simplification:

clause 14: \( r.ID_2(X_s, Y, A) \) —

\[
X_s = [X\lbrack TXs\rbrack, \\
nonMinimal(X), \text{ decompose}(X, PX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t). \\
r.ID_2(TXs, NA, A), \\
process(HX, HY), \text{ compose}(HY, NA, Y)
\]

By \( t \) times unfolding clause 6 wrt

\[
r.ID_1([TX_p, \ldots, TX_t\lbrack TXs'\rbrack, HY, NA'], \ldots, r.ID_1([TX_t\lbrack TXs'\rbrack, HY, NA_{t-1}])
\]

using the “decompletion” of \( S_{r.ID} \) in Section 4:

clause 15: \( r.ID_2(X_s, Y, A) \) —

\[
X_s = [X\lbrack TXs\rbrack, \\
TXs' = [X\lbrack TXs'\rbrack, TXs' = []], \\
nonMinimal(X), \text{ decompose}(X, PX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{p-1}). \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
process(HX, HY'), \text{ compose}(e, HY', NA') \\
r(TX_p, TY_p), \ldots, r(TX_t, TY_t). \\
\text{ compose}(NA', TY_p, NA_p), \ldots, \text{ compose}(NA_{t-1}, TY_t, NA_t) \\
r.ID_1(TXs', HY, NA_t), \\
r.ID_2(TXs, NA, A), \text{ compose}(HY, NA, Y)
\]

By unfolding clause 15 wrt \( r.ID_1(TXs', HY, NA_t) \) using \( TDGLR \), and using the assumption that
\( P_{r.ID} \) is steadfast wrt \( S_{r.ID} \) in \( S \):

clause 16: \( r.ID_2(X_s, Y, A) \) —

\[
X_s = [X\lbrack TXs\rbrack, \\
TXs' = [X\lbrack TXs'\rbrack, TXs' = []], \\
nonMinimal(X), \text{ decompose}(X, PX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{p-1}). \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
process(HX, HY'), \text{ compose}(e, HY', NA') \\
r(TX_p, TY_p), \ldots, r(TX_t, TY_t). \\
\text{ compose}(NA', TY_p, NA_p), \ldots, \text{ compose}(NA_{t-1}, TY_t, NA_t) \\
TXs' = [], HY = NA_t, \\
r.ID_2(TXs, NA, A), \text{ compose}(HY, NA, Y)
\]

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

clause 17: \( r.ID_2(X_s, Y, A) \) —

\[
X_s = [X\lbrack TXs\rbrack, \\
nonMinimal(X), \text{ decompose}(X, PX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{p-1}). \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
process(HX, HY), \text{ compose}(HY, I_p, Y) \\
r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \\
\text{ compose}(TY_p, I_p, I_p), \ldots, \text{ compose}(TY_t, NA, I_t) \\
r.ID_2(TXs, NA, A)
\]

By \( t \) times folding clause 17 using clauses 1 and 2:
clause 18: \( r.l.d_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ minimal(TX_p), \ldots, minimal(TX_{p-1}). \]
\[ (nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \]
\[ r.l.d_2([TX_p, \ldots, TX_1|TXs], N, A, A), \]
\[ process(HX, HY), compose(HY, N, A, Y) \]

By \( p \) times unfolding clause 7 wrt
\[ r.l.d_1([TX_1, \ldots, TX_{p-1}, N|TXs'], HY, NA'), \ldots, r.l.d_1([N|TXs'], HY, NA_{p-1}) \]
using the “decompletion” of \( S_{r.l.d_i} \) in Section 4:

clause 19: \( r.l.d_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs], \]
\[ Xs' = [X|TXs'], TXs' = []. \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \]
\[ compose(e, TY_1, NA_1), \]
\[ compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \]
\[ compose(NA_{p-1}, YN, NA_p). \]
\[ r.l.d_1(TXs', HY, NA_p). \]
\[ r.l.d_2(TXs, N, A, A), compose(HY, N, A, Y). \]

By unfolding clause 19 wrt \( r.l.d_1(TXs', HY, NA_p) \) using TDGLR, and using the assumption that \( P_{r.l.d_i} \) is steadfast wrt \( S_{r.l.d_i} \) in \( S \):

clause 20: \( r.l.d_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs], \]
\[ Xs' = [X|TXs'], TXs' = []. \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \]
\[ compose(e, TY_1, NA_1), \]
\[ compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \]
\[ compose(NA_{p-1}, YN, NA_p). \]
\[ TXs' = [], HY = NA_p. \]
\[ r.l.d_2(TXs, N, A), compose(HY, N, A, Y). \]

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

clause 21: \( r.l.d_2(Xs, Y, A) \rightarrow \)
\[ Xs = [X|TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}). \]
\[ decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, YN), \]
\[ compose(TY_1, I_1, Y), \]
\[ compose(TY_2, I_2, I_1), \ldots, compose(TY_{p-1}, I_p, I_{p-1}), \]
\[ compose(YN, NA, I_p), \]
\[ r.l.d_2(TXs, N, A, A). \]

By \( p \) times folding clause 21 using clauses 1 and 2:
clause 22: \( r_{\text{td}_2}(Xs, Y, A) \) —
\[ Xs = [X[TXs]]. \\
non\text{Minimal}(X), decompose(X, HX, TX_1, \ldots, TX_t). \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
minimal(TX_p), \ldots, minimal(TX_t). \\
minimal(U_1), \ldots, minimal(U_{p-1}). \\
de decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
r_{\text{td}_2}(TX_1, \ldots, TX_{p-1}, N[TXs], Y, A) \]

By \( t + 1 \) times unfolding clause 8 wrt
\[ r_{\text{td}_1}(TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t[TXs'], HY, NA'), \ldots, r_{\text{td}_3}([TX|TXs'], HY, NA_t) \]
using the “decomposition” of \( S_{r_{\text{td}_4}} \) in Section 4:

clause 23: \( r_{\text{td}_2}(Xs, Y, A) \) —
\[ Xs = [X[TXs]]. \\
Xs' = [X[TXs'], TXs' = []. \\
non\text{Minimal}(X), decompose(X, HX, TX_1, \ldots, TX_t). \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
(non\text{Minimal}(TX_p); \ldots; non\text{Minimal}(TX_t)). \\
minimal(U_1), \ldots, minimal(U_{p-1}). \\
de decompose(N, HX, U_1, \ldots, U_t). \\
r(TX_1, TY_1), \ldots, r(TX_1, TY_t), r(N, YN), \\
com pos e(e, TY_1, NA_1), \\
com pos e(NA_1, TY_2, NA_2), \ldots, com pos e(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
com pos e(NA_{p-1}, YN, NA_p), \\
com pos e(NA_p, TY_p, NA_{p+1}), \ldots, com pos e(NA_t, TY_t, NA_{t+1}), \\
r_{\text{td}_4}(TXs', HY, NA_{t+1}), \\
r_{\text{td}_2}(TXs, NA, A), com pos e(HY, NA, Y) \]

By unfolding clause 23 wrt \( r_{\text{td}_3}(TXs', HY, NA_{t+1}) \) using TDGLR, and using the assumption that \( H_{r_{\text{td}_4}} \) is steadfast wrt \( S_{r_{\text{td}_4}} \) in \( S \):

clause 24: \( r_{\text{td}_2}(Xs, Y, A) \) —
\[ Xs = [X[TXs]]. \\
Xs' = [X[TXs'], TXs' = []. \\
non\text{Minimal}(X), decompose(X, HX, TX_1, \ldots, TX_t). \\
(non\text{Minimal}(TX_1); \ldots; non\text{Minimal}(TX_{p-1})). \\
(non\text{Minimal}(TX_p); \ldots; non\text{Minimal}(TX_t)). \\
minimal(U_1), \ldots, minimal(U_{p-1}). \\
de decompose(N, HX, U_1, \ldots, U_t). \\
r(TX_1, TY_1), \ldots, r(TX_1, TY_t), r(N, YN), \\
com pos e(e, TY_1, NA_1), \\
com pos e(NA_1, TY_2, NA_2), \ldots, com pos e(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
com pos e(NA_{p-1}, YN, NA_p), \\
com pos e(NA_p, TY_p, NA_{p+1}), \ldots, com pos e(NA_t, TY_t, NA_{t+1}), \\
TXs' = [], HY = NA_{t+1}. \\
r_{\text{td}_2}(TXs, NA, A), com pos e(HY, NA, Y) \]

By using applicability conditions (1) and (2), and simplification:
clause 25: \( r.td_2(Xs, Y, A) \rightarrow Xs = [X[TXs]] \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i) \\
\quad \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \\
\quad \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i) \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \\
\quad \text{decompose}(N, HX, U_1, \ldots, U_i) \\
\quad r(TX_1, TY_1), \ldots, r(TX_i, TY_i), r(N, Y, N) \\
\quad \text{compose}(TY_1, I_1, Y) \\
\quad \text{compose}(TY_2, I_2, I_1), \ldots, \text{compose}(TY_{p-1}, I_p, I_{p-1}) \\
\quad \text{compose}(Y, N, I_{p+1}, I_p) \\
\quad \text{compose}(TY_p, I_{p+2}, I_p, I_{p+1}), \ldots, \text{compose}(TY_{i-1}, I_{i+1}, I_i) \\
\quad \text{compose}(TY_i, N, A, I_{i+1}) \\
\quad r.td_2(TXs, N, A, A) \)

By \( t + 1 \) times folding clause 25 using clauses 1 and 2:

clause 26: \( r.td_2(Xs, Y, A) \rightarrow Xs = [X[TXs]] \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_i) \\
\quad \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \\
\quad \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_i) \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_i) \\
\quad \text{decompose}(N, HX, U_1, \ldots, U_i) \\
\quad r.td_2([TX_1, \ldots, TX_{p-1}, N, TX_1, \ldots, TX_i], TXs), Y, A) \)

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:

\[
r(X, Y) \rightarrow r.td_2([X], Y, e)\]

By taking the ‘completion’:

\[
\forall X : X, \forall Y : Y. \; I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r.td_2([X], Y, e)]
\]

By unfolding the ‘completion’ above wrt \( r.td_2([X], Y, e) \) using \( S_{r.td_2} \):

\[
\forall X : X, \forall Y : Y. \; I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : Y. \; S_r(X, Y_1) \land I_1 = Y \land S_r(I_1, e, Y)]
\]

By using applicability condition (2):

\[
\forall X : X, \forall Y : Y. \; I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, I_1 : Y. \; S_r(X, Y_1) \land I_1 = Y \land Y = I_1]
\]

By simplification:

\[
\forall X : X, \forall Y : Y. \; I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow S_r(X, Y)]
\]

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 \( \mathcal{S} \).

\[\square\]

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.
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


