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 w.r.t 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:

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

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 \textit{(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:
\[
P \vdash r(t, u) \iff \mathcal{F} \models O_r(t, u), \text{ for every ground term } u \text{ of } \mathcal{Y}.
\]

If we replace ‘iff’ by ‘implies’ in the condition above, then \( P \) is said to be \textit{partially correct} wrt \( S_r \), and if we replace ‘iff’ by ‘if’, then \( P \) is said to be \textit{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 \textit{steadfast} wrt its specification \( S_r \) in \( \{S_1, \ldots, S_m\} \) if and only if 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 \( \{S_1, \ldots, S_m\} \) if and only if 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 \textit{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 \textit{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 ellipsis, and $\bigwedge_{i=1}^n r(TX_i, TY_i)$ is a conjunct ellipsis.

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

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

**Definition 9** A transformation schema encoding a transformation technique is a 5-tuple $(S_1, S_2, A, O_{12}, O_{21})$, 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 $T_2$, where $T_1$ and $T_2$ 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 $T_i$ 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.

$TG_1 : \{ DCLR, TG, A_{11}, O_{1112}, O_{1121} \}$ where

$A_{11} : \{ (1) compose is associative

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

(3) $\forall X : A. \ I r(X) \land \operatorname{minimal}(X) \Rightarrow O r(X, e)$

(4) $\forall X : A. \ I r(X) \Rightarrow \neg \operatorname{minimal}(X) \Rightarrow \nonMinimal(X) \}$

$O_{1112} : \text{partial evaluation of the conjunction}$

$\operatorname{process}(H X, H Y), \operatorname{compose}(H Y, T Y, Y)$

results in the introduction of a non-recursive relation

$O_{1121} : \text{partial evaluation of the conjunction}$

$\operatorname{process}(H X, H Y), \operatorname{compose}(I_p, H Y, I_p)$

results in the introduction of a non-recursive relation

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

**Logic Program Template 1**

\[
\begin{align*}
&\begin{cases}
  r(X, Y) :- \\
  \text{minimal}(X), \\
  \text{solve}(X, Y)
  \\
  r(X, Y) :- \\
  \text{nonMinimal}(X), \\
  \text{decompose}(X, H X, TX_1, ..., TX_t), \\
  r(TX_1, TY_1), ..., r(TX_t, TY_t), \\
  I_0 = e, \\
  \text{compose}(I_0, TY_1, I_1), ..., \text{compose}(I_p-2, TY_{p-1}, I_{p-1}), \\
  \text{process}(H X, H Y), \text{compose}(I_{p-1}, H Y, I_p), \\
  \text{compose}(I_p, TY_p, I_{p+1}), ..., \text{compose}(I_t, TY_t, I_{t+1}), \\
  Y = I_{t+1}
  \\
  \end{cases}
\end{align*}
\]

**Logic Program Template 2**

\[
\begin{align*}
&\begin{cases}
  r(X, Y) :- \\
  r_\text{tupling}(X, Y) \\
  r_\text{tupling}(Xs, Y) :- \\
  Xs = [], \\
  Y = e \\
  r_\text{tupling}(Xs, Y) :- \\
  Xs = [X|TXs], \\
  \text{minimal}(X), \\
  r_\text{tupling}(TXs, TY), \\
  \text{solve}(X, H Y), \\
  \text{compose}(H Y, TY, Y) \\
  r_\text{tupling}(Xs, Y) :- \\
  Xs = [X|TXs], \\
  \text{nonMinimal}(X), \\
  \text{decompose}(X, H X, TX_1, ..., TX_t), \\
  \text{minimal}(TX_1), ..., \text{minimal}(TX_t), \\
  r_\text{tupling}(TXs, TY), \\
  \text{process}(H Xs, TY),
  \\
  \end{cases}
\end{align*}
\]
compose(HY, TY, Y)
\r_rupleing(Xs, Y) =
Xs = [X|Txs],
nonMinimal(X),
decompose(X, HX, TX₁, ..., TXₜ),
minimal(TX₁), ..., minimal(TXₚ₋₁),
(nonMinimal(TXₚ); ...; nonMinimal(TXₜ)),
r_rupleing([TXₚ, ..., TXₜ, Txs], TY),
process(HX, HY),
compose(HY, TY, Y)

r_rupleing(Xs, Y) =
Xs = [X|Txs],
nonMinimal(X),
decompose(X, HX, TX₁, ..., TXₜ),
(nonMinimal(TX₁); ...; nonMinimal(TXₚ₋₁)),
minimal(TXₚ), ..., minimal(TXₜ),
minimal(U₁), ..., minimal(Uₚ₋₁),
decompose(N, HX, U₁, ..., Uₚ₋₁, TXₚ, ..., TXₜ),
r_rupleing([TX₁, ..., TXₚ₋₁, N, Txs], Y)

r_rupleing(Xs, Y) =
Xs = [X|Txs],
nonMinimal(X),
decompose(X, HX, TX₁, ..., TXₜ),
(nonMinimal(TX₁); ...; nonMinimal(TXₚ₋₁)),
(nonMinimal(TXₚ); ...; nonMinimal(TXₜ)),
minimal(U₁), ..., minimal(Uₚ₋₁),
decompose(N, HX, U₁, ..., Uₚ₋₁),
r_rupleing([TX₁, ..., TXₚ₋₁, N, TXₚ, ..., TXₜ|Txs], Y)

and the specification \( S_r \) of relation \( r \) is:
\[
\forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad \text{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \text{O}_r(X, Y)]
\]
and the specification \( S_r \) of relation \( r_rupleing \) is:
\[
\forall X : \mathcal{X} \text{ list of } X, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in X \Rightarrow \text{I}_r(X)) \Rightarrow [r_rupleing(Xs, Y) \Leftrightarrow \langle Xs = \emptyset \rangle \land Y = \emptyset \land \bigwedge_{i=1}^{q} \text{O}_r(X, Y_i) \land \text{I}_1 = Y_1 \land \bigwedge_{i=2}^{q} \text{O}_r(X_{i-1}, Y_i, Y_i) \land Y = Y_q]
\]
where \( \text{O}_r \) is the output condition of \( \text{compose} \) and \( q \geq 1 \).

**Proof 1** To prove the correctness of the generalization schema \( TG_1 \), by Definition 10, we have to prove that templates \( DCLR \) and \( TG \) are equivalent wrt \( S_r \) under the applicability conditions \( A_{11} \). 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 \( S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solver}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \),

where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solver}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \) 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) \( TG \) is also steadfast wrt \( S_r \) in \( S \).
Note that the sets \( \{ S_1, \ldots, S_m \} \) and \( \{ S'_1, \ldots, S'_{\ell} \} \) in Definition 4 are equal to \( \mathcal{S} \) when \( Q \) is obtained by tupling generalization of \( P \).

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

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

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

If we separate the cases of \( q \geq 1 \) by \( q = 1 \lor q \geq 2 \), then \( S^\text{upling} \) becomes:

\[
\forall Xs : \text{list} \ of \ X, \forall Y : \mathcal{Y} . \ (\forall X : \mathcal{X} . \ X \in Xs \Rightarrow T_r(X)) \Rightarrow [r^\text{upling}(Xs,Y) \iff (Xs = [] \land Y = \epsilon) \\
\lor (Xs = [X1] \land \mathcal{O}_r(X1,Y1) \land Y1 = I1 \land Y = I1) \\
\lor (Xs = [X1,X2,\ldots,X_q] \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i) \land Y1 = Y_i \land \bigwedge_{i=2}^{q} \mathcal{O}_r(I_{i-1},Y_i,I_i) \land Y = I_q)]
\]

where \( q \geq 2 \).

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

\[
\forall Xs : \text{list} \ of \ X, \forall Y : \mathcal{Y} . \ (\forall X : \mathcal{X} . \ X \in Xs \Rightarrow T_r(X)) \Rightarrow [r^\text{upling}(Xs,Y) \iff (Xs = [] \land Y = \epsilon) \\
\lor (Xs = [X1] \land \mathcal{O}_r(X1,Y1) \land Y1 = I1 \land TY = \epsilon \land \mathcal{O}_r(I1,TY,Y)) \\
\lor (Xs = [X1,X2,\ldots,X_q] \land \bigwedge_{i=1}^{q} \mathcal{O}_r(X_i,Y_i) \land Y1 = I1 \land Y2 = I2 \land \bigwedge_{i=2}^{q} \mathcal{O}_r(I_{i-1},Y_i,I_i) \land TY = I_q \land \mathcal{O}_r(I1,TY,Y)])
\]

where \( q \geq 2 \).

By folding using \( S^\text{upling} \), and renaming:

\[
\forall Xs : \text{list} \ of \ X, \forall Y : \mathcal{Y} . \ (\forall X : \mathcal{X} . \ X \in Xs \Rightarrow T_r(X)) \Rightarrow [r^\text{upling}(Xs,Y) \iff (Xs = [] \land Y = \epsilon) \\
\lor (Xs = [X] \land \mathcal{O}_r(X,HY) \land r^\text{upling}(TXs,TY) \land \mathcal{O}_r(HY,TY,Y))]
\]

By taking the ‘decomposition’:

\[
\text{clause 1 : } \quad r^\text{upling}(Xs,Y) \leftarrow \\
\quad \text{Xs} = [] \land Y = \epsilon
\]

\[
\text{clause 2 : } \quad r^\text{upling}(Xs,Y) \leftarrow \\
\quad \text{Xs} = [X] \land \mathcal{O}_r(X,HY) \land r^\text{upling}(TXs,TY) \land \mathcal{O}_r(HY,TY,Y)
\]

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

\[
\text{clause 3 : } \quad r^\text{upling}(Xs,Y) \leftarrow \\
\quad \text{Xs} = [X] \land \mathcal{O}_r(X,HX) \land r^\text{upling}(TXs,TY) \land \mathcal{O}_r(HY,TY,Y)
\]

\[
\text{clause 4 : } \quad r^\text{upling}(Xs,Y) \leftarrow \\
\quad \text{Xs} = [X] \land \mathcal{O}_r(X,HX) \land \mathcal{O}_r(TXs,TY) \land \mathcal{O}_r(HY,TY,Y)
\]

By introducing
\[(\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_t)) \lor \ldots \lor (\text{nonMinimal}(TX_1) \lor \ldots \lor \text{nonMinimal}(TX_t))\]

clause 5: \( r\text{uplimg}(Xs, Y) \leftarrow \)

\( Xs = [X' \mid Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( l_0 = \varepsilon, \)
\( \text{compose}(l_h, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_{t-1}, TY_t, l_{t+1}), \)
\( HY = l_{t+1}, r\text{uplimg}(TXs, TY), \text{compose}(HY, TY, Y) \)

clause 6: \( r\text{uplimg}(Xs, Y) \leftarrow \)

\( Xs = [X' \mid Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \)
\( (\text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_t)), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( l_0 = \varepsilon, \)
\( \text{compose}(l_h, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_{t-1}, TY_t, l_{t+1}), \)
\( HY = l_{t+1}, r\text{uplimg}(TXs, TY), \text{compose}(HY, TY, Y) \)

clause 7: \( r\text{uplimg}(Xs, Y) \leftarrow \)

\( Xs = [X' \mid Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( (\text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1})), \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( l_0 = \varepsilon, \)
\( \text{compose}(l_h, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_{t-1}, TY_t, l_{t+1}), \)
\( HY = l_{t+1}, r\text{uplimg}(TXs, TY), \text{compose}(HY, TY, Y) \)

clause 8: \( r\text{uplimg}(Xs, Y) \leftarrow \)

\( Xs = [X' \mid Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( (\text{nonMinimal}(TX_1), \ldots, \text{nonMinimal}(TX_{p-1})), \)
\( (\text{nonMinimal}(TX_p), \ldots, \text{nonMinimal}(TX_t)), \)
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \)
\( l_0 = \varepsilon, \)
\( \text{compose}(l_h, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_{t-1}, TY_t, l_{t+1}), \)
\( HY = l_{t+1}, r\text{uplimg}(TXs, TY), \text{compose}(HY, TY, Y) \)

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):
\textbf{clause 9:} \( r\text{-}\text{tupling}(Xs, Y) = \\
Xs = [X[T\ Xs], \\
non\ Minimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t), \\
solve(TX_1, TY_1), \ldots, solve(TX_t, TY_t), \\
l_0 = \epsilon, \\
\text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
process(HX, H HY),\text{compose}(l_{p-1}, H HY, l_p), \\
\text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \\
HY = l_{t+1}, r\text{-}\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y) \\
\)

By using applicability condition (3):

\textbf{clause 10:} \( r\text{-}\text{tupling}(Xs, Y) = \\
Xs = [X[T\ Xs], \\
non\ Minimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t), \\
solve(TX_1, e), \ldots, solve(TX_t, e), \\
l_0 = \epsilon, \\
\text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \\
process(HX, H HY),\text{compose}(l_{p-1}, H HY, l_p), \\
\text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}), \\
HY = l_{t+1}, r\text{-}\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y) \\
\)

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

\textbf{clause 11:} \( r\text{-}\text{tupling}(Xs, Y) = \\
Xs = [X[T\ Xs], \\
non\ Minimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t), \\
solve(TX_1, e), \ldots, solve(TX_t, e), \\
l_0 = \epsilon, \\
\text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \\
process(HX, H HY),\text{compose}(l_{p-1}, H HY, l_p), \\
\text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}), \\
HY = l_{t+1}, r\text{-}\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y) \\
\)

By using applicability condition (2):

\textbf{clause 12:} \( r\text{-}\text{tupling}(Xs, Y) = \\
Xs = [X[T\ Xs], \\
non\ Minimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t), \\
solve(TX_1, e), \ldots, solve(TX_t, e), \\
l_0 = \epsilon, \\
l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \\
process(HX, H HY),\text{compose}(l_{p-1}, H HY, l_p), \\
l_{p+1} = l_p, \ldots, l_{t+1} = l_t, \\
HY = l_{t+1}, r\text{-}\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y) \\
\)

By simplification:

\textbf{clause 13:} \( r\text{-}\text{tupling}(Xs, Y) = \\
Xs = [X[T\ Xs], \\
non\ Minimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_t), \\
r\text{-}\text{tupling}(TXs, TY), \\
process(HX, H HY), \text{compose}(HY, TY, Y) \\
\)

By \( p-1 \) times unfolding clause 6 wrt \( r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}) \) using DCLR, and simplifying using condition (4):
clause 14: \ \texttt{r\_tupling}(X_s,Y) \\
\hspace{0.5em} X_s = [X[T X_s], \nonMinimal(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \minimal(T X_1), \ldots, \minimal(T X_{p-1}), (\nonMinimal(T X_p); \ldots; \nonMinimal(T X_t)), \minimal(T X_1), \ldots, \minimal(T X_{p-1}), \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \text{solve}(T X_{p}, T Y_{p}), \ldots, r(T X_t, T Y_t) \]\n\hspace{0.5em} l_p = \epsilon, \completeness{(l_0, T Y_1, l_1), \ldots, \completeness{(l_{p-2}, T Y_{p-1}, l_{p-1})}, \text{process}(H X, H Y), \completeness{(l_{p-1}, H Y, l_p)}, \completeness{(l_p, T Y_1, l_{p-1})}, \ldots, \completeness{(l_t, T Y_t, l_{t+1})}, H Y = l_{t+1}, \text{r\_tupling}(T X_s, T Y), \text{compose}(H Y, T Y, Y)\]

By deleting one of the \minimal(T X_1), \ldots, \minimal(T X_{p-1}) atoms in clause 14:

clause 15: \ \texttt{r\_tupling}(X_s,Y) \\
\hspace{0.5em} X_s = [X[T X_s], \nonMinimal(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \minimal(T X_1), \ldots, \minimal(T X_{p-1}), (\nonMinimal(T X_p); \ldots; \nonMinimal(T X_t)), \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \text{solve}(T X_{p}, T Y_{p}), \ldots, r(T X_t, T Y_t) \]\n\hspace{0.5em} l_p = \epsilon, \completeness{(l_0, T Y_1, l_1), \ldots, \completeness{(l_{p-2}, T Y_{p-1}, l_{p-1})}, \text{process}(H X, H Y), \completeness{(l_{p-1}, H Y, l_p)}, \completeness{(l_p, T Y_1, l_{p-1})}, \ldots, \completeness{(l_t, T Y_t, l_{t+1})}, H Y = l_{t+1}, \text{r\_tupling}(T X_s, T Y), \text{compose}(H Y, T Y, Y)\]

By rewriting clause 15 using applicability condition (1):

clause 16: \ \texttt{r\_tupling}(X_s,Y) \\
\hspace{0.5em} X_s = [X[T X_s], \nonMinimal(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \minimal(T X_1), \ldots, \minimal(T X_{p-1}), (\nonMinimal(T X_p); \ldots; \nonMinimal(T X_t)), \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \text{solve}(T X_{p}, T Y_{p}), \ldots, r(T X_t, T Y_t) \]\n\hspace{0.5em} l_p = \epsilon, \completeness{(l_0, T Y_1, l_1), \ldots, \completeness{(l_{p-2}, T Y_{p-1}, l_{p-1})}, \text{process}(H X, H Y), \completeness{(l_{p-1}, H Y, l_p)}, H Y = l_p, \completeness{(T Y_1, T Y_{p-1}, l_{p-1})}, \completeness{(l_{p-1}, T Y_1, l_{p-1})}, \ldots, \completeness{(l_{t-1}, T Y_1, l_t), \text{r\_tupling}(T X_s, T Y), \text{compose}(l_t, T Y, T Y), \text{compose}(H Y, T Y, Y)\]}

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

clause 17: \ \texttt{r\_tupling}(X_s,Y) \\
\hspace{0.5em} X_s = [X[T X_s], \nonMinimal(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \minimal(T X_1), \ldots, \minimal(T X_{p-1}), (\nonMinimal(T X_p); \ldots; \nonMinimal(T X_t)), \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \text{solve}(T X_{p}, T Y_{p}), \ldots, \text{r\_tupling}(T X_p, \ldots, T X_t[T X_s], T Y) \]\n\hspace{0.5em} l_p = \epsilon, \completeness{(l_0, T Y_1, l_1), \ldots, \completeness{(l_{p-2}, T Y_{p-1}, l_{p-1})}, \text{process}(H X, H Y), \completeness{(l_{p-1}, H Y, l_p)}, H Y = l_p, \completeness{(H Y, T Y, Y)\]}

By using applicability condition (3):
\text{clause 18: } r_{\text{tupling}}(X, S, Y) = \\
X = [X[t X s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, 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, \epsilon), \ldots, \text{solve}(TX_{p-1}, \epsilon), \\
r_{\text{tupling}}([TX_p, \ldots, TX_t[t X s], TY), \\
l_0 = \epsilon, \\
\text{compose}(I_0, \epsilon, I_1), \ldots, \text{compose}(I_{p-2}, \epsilon, I_{p-1}), \\
\text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \text{HY} = I_p, \\
\text{compose}(HY, TY, Y) \\

By using applicability condition (2):

\text{clause 19: } r_{\text{tupling}}(X, S, Y) = \\
X = [X[t X s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, 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, \epsilon), \ldots, \text{solve}(TX_{p-1}, \epsilon), \\
r_{\text{tupling}}([TX_p, \ldots, TX_t[t X s], TY), \\
l_0 = \epsilon, \\
l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \\
\text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \text{HY} = I_p, \\
\text{compose}(HY, TY, Y) \\

By simplification:

\text{clause 20: } r_{\text{tupling}}(X, S, Y) = \\
X = [X[t X s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, 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{tupling}}([TX_p, \ldots, TX_t[t X s], TY), \\
\text{process}(H X, H H Y), \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 \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) in clause 7:

\text{clause 21: } r_{\text{tupling}}(X, S, Y) = \\
X = [X[t X s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, 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), \\
l_0 = \epsilon, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{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, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
\text{HY} = I_{t+1}, r_{\text{tupling}}(TX_s, TY), \text{compose}(HY, TY, Y) \\

By using applicability condition (3):
**Clause 22:**
\[ \text{rupleing}(Xs, Y) \]  
\[ Xs = [X[T Xs], \]  
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]  
\[ \text{nonMinimal}(TX_1) ; \ldots \text{nonMinimal}(TX_{p-1}), \]  
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]  
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]  
\[ r(U_1, \epsilon), \ldots, r(U_{p-1}, \epsilon), \]  
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]  
\[ l_0 = \epsilon, \]  
\[ \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \]  
\[ \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \]  
\[ \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]  
\[ HY = l_{t+1}, \text{rupleing}(TXs, TY), \text{compose}(HY, TY, Y) \]

By using applicability condition (2):

**Clause 23:**
\[ \text{rupleing}(Xs, Y) \]  
\[ Xs = [X[T Xs], \]  
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]  
\[ \text{nonMinimal}(TX_1) ; \ldots \text{nonMinimal}(TX_{p-1}), \]  
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]  
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]  
\[ r(U_1, \epsilon), \ldots, r(U_{p-1}, \epsilon), \]  
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]  
\[ l_0 = \epsilon, \]  
\[ \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \]  
\[ \text{compose}(l_{p-1}, \epsilon, K_1), \text{compose}(K_1, \epsilon, K_2), \ldots, \text{compose}(K_{p-2}, \epsilon, K_{p-1}), \]  
\[ \text{process}(HX, HHY), \text{compose}(K_{p-1}, HHY, l_p), \]  
\[ \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]  
\[ HY = l_{t+1}, \text{rupleing}(TXs, TY), \text{compose}(HY, TY, Y) \]

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

**Clause 24:**
\[ \text{rupleing}(Xs, Y) \]  
\[ Xs = [X[T Xs], \]  
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]  
\[ \text{nonMinimal}(TX_1) ; \ldots \text{nonMinimal}(TX_{p-1}), \]  
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]  
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]  
\[ r(U_1, \epsilon), \ldots, r(U_{p-1}, \epsilon), \]  
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]  
\[ l_0 = \epsilon, \]  
\[ \text{compose}(l_0, \epsilon, l_1), \ldots, \text{compose}(l_{p-2}, \epsilon, l_{p-1}), \]  
\[ \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \]  
\[ \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]  
\[ HY = l_{t+1}, \text{rupleing}(TXs, TY), \text{compose}(HY, TY, TI), \]  
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \]  
\[ \text{compose}(K_{p-2}, TI, Y) \]

By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_{p-1} \) in place of some occurrences of \( \epsilon \):
clause 25: \( r_{\text{tupling}}(X, Y) \) —
\[
X = [X \cup \{X\}]
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}), \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_p-1), \)
\( r(U_1,YU_1), \ldots, r(U_{p-1},YU_{p-1}), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( l_0 = \epsilon, \)
\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \)
\( HY = I_{t+1}, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, TI), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(K_{p-2}, TI, Y) \)

By introducing \( \text{nonMinimal}(N) \) 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: \( r_{\text{tupling}}(X, Y) \) —
\[
X = [X \cup \{X\}]
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}), \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_p-1), \)
\( r(U_1,YU_1), \ldots, r(U_{p-1},YU_{p-1}), \)
\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( l_0 = \epsilon, \)
\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \)
\( HY = I_{t+1}, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, TI), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(K_{p-2}, TI, Y) \)

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

clause 27: \( r_{\text{tupling}}(X, Y) \) —
\[
X = [X \cup \{X\}]
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}), \)
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_p-1), \)
\( r(U_1,YU_1), \ldots, r(U_{p-1},YU_{p-1}), \)
\( \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \)
\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \)
\( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( l_0 = \epsilon, \)
\( \text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \)
\( HY = I_{t+1}, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, TI), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( \text{compose}(K_{p-2}, TI, Y) \)

By folding clause 27 using DC/IR.
\textbf{clause 28:} \textit{r\textsc{tupling}}(\textit{Xs}, \textit{Y}) =
\begin{align*}
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), \\
\text{r\textsc{tupling}}(\textit{TXs}, \textit{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:

\textbf{clause 29:} \textit{r\textsc{tupling}}(\textit{Xs}, \textit{Y}) =
\begin{align*}
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}), \\
\text{r\textsc{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 \textit{p} - 1 times folding clause 29 using clauses 1 and 2:

\textbf{clause 30:} \textit{r\textsc{tupling}}(\textit{Xs}, \textit{Y}) =
\begin{align*}
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), \\
\text{r\textsc{tupling}}([TX_1, \ldots, TX_{p-1}, N[TXs], Y)
\end{align*}

By introducing atoms \textit{minimal}(U_1), \ldots, \textit{minimal}(U_t) (with new, i.e. existentially quantified, variables \textit{U}_1, \ldots, \textit{U}_t) in clause 8:

\textbf{clause 31:} \textit{r\textsc{tupling}}(\textit{Xs}, \textit{Y}) =
\begin{align*}
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 &= \epsilon, \\
\text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
\text{process}(HX, HLY), \text{compose}(l_{p-1}, HLY, l_p), \\
\text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \\
HY &= l_{t+1}, \text{r\textsc{tupling}}(\textit{TXs}, \textit{TY}), \text{compose}(HY, TY, Y)
\end{align*}

By using applicability condition (3):

13
\textbf{clause 32:} \( \text{r_Jumping}(X, Y) \) —
\[
X_s = [X | TX_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), \)
\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \)
\( \text{(nonMinimal}(TX_{t}); \ldots; \text{nonMinimal}(TX_{t})), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \)
\( r(U_1,e), \ldots, r(U_t,e), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( l_0 = e, \)
\( \text{compose}(I_0,TY_1, I_1), \ldots, \text{compose}(I_p-2, TY_{p-1}, I_p-1), \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \)
\( HY = l_{t+1}, \text{r_Jumping}(TX_s, TY), \text{compose}(HY, TY, Y) \)

By using applicability condition (2):

\textbf{clause 33:} \( \text{r_Jumping}(X, Y) \) —
\[
X_s = [X | TX_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), \)
\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \)
\( \text{(nonMinimal}(TX_{t}); \ldots; \text{nonMinimal}(TX_{t})), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \)
\( r(U_1,e), \ldots, r(U_t,e), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( l_0 = e, \)
\( \text{compose}(I_0,TY_1, I_1), \ldots, \text{compose}(I_p-2, TY_{p-1}, I_p-1), \)
\( \text{compose}(I_{p-1}, I_{p+1}), \ldots, \text{compose}(I_t, I_{t+1}), \)
\( HY = l_{t+1}, \text{r_Jumping}(TX_s, TY), \text{compose}(HY, TY, Y) \)

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

\textbf{clause 34:} \( \text{r_Jumping}(X, Y) \) —
\[
X_s = [X | TX_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), \)
\( \text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \)
\( \text{(nonMinimal}(TX_{t}); \ldots; \text{nonMinimal}(TX_{t})), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \)
\( r(U_1,e), \ldots, r(U_t,e), \)
\( r(TX_1,TY_1), \ldots, r(TX_t,TY_t), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-3}), \)
\( l_0 = e, \)
\( \text{compose}(I_0, I_1), \ldots, \text{compose}(I_{p-2}, I_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, I_p), \)
\( \text{compose}(I_p, I_{p+1}), \ldots, \text{compose}(I_t, I_{t+1}), \)
\( HY = l_{t+1}, \text{r_Jumping}(TX_s, TY), \text{compose}(HY, TY, Y) \)

By introducing new, i.e. existentially quantified, variables \( YU_1, \ldots, YU_t \) in place of some occurrences of \( e \):
By introducing $\text{nonMinimal}(N)$ and $\text{decompose}(N, HX, U_1, \ldots, U_1)$, since

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

always holds (because $N$ is existentially quantified):

clause 36: $\text{r Tupling}(X, Y) =$

$$Xs = [X[Ts],$$

$\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),$ 
$\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t+1})$, 
$\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_{t+1})$, 
minimal($U_1$), \ldots, minimal($U_t$), 
r($U_1, YU_1$), \ldots, r($U_t, YU_t$), 
r($TX_1, TY_1$), \ldots, r($TX_t, TY_t$), 
$\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{t-3}, TY_{t-1}, K_{t-3})$, 
$l_0 = \epsilon$, 
$\text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{t-2}, YU_{t-1}, l_{t-1})$, 
$\text{process}(HX, HHY), \text{compose}(l_{t-1}, HHY, l_p)$, 
$\text{compose}(l_p, YU_p, l_{p+1}), \ldots, \text{compose}(l_t, YU_t, l_{t+1})$, 
$NY = l_{t+1}$, 
$\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_{t-1}, K_{t-1})$, 
$\text{r Tupling}(TXs, TY), \text{compose}(HY, TY, Y)$

By duplicating goal $\text{decompose}(N, HX, U_1, \ldots, U_1)$:
By folding clause 37 using \textsc{DCLR}:

clause 38:  \texttt{r.tupling}(Xs, Y) —
\begin{align*}
Xs &= [X[T 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), \\
rc(U_1, U_1), \ldots, r(U_1, U_1), \\
\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}), \\
l_0 &= e, \\
\text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{p-2}, YU_{p-1}, l_{p-1}), \\
\text{process}(HX, HHY), \text{compose}(I_{p-1}, HHY, l_p), \\
\text{compose}(I_p, YU_p, l_{p+1}), \ldots, \text{compose}(l_1, YU_1, l_{1+1}), \\
NY &= l_{1+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_{p-2}, NHY, T1), \text{compose}(T1, K_{t-1}, HY), \\
r.t.tupling(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}

By using applicability condition (1):

clause 39:  \texttt{r.tupling}(Xs, Y) —
\begin{align*}
Xs &= [X[T 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), \\
\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_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, T1), \text{compose}(T1, K_{t-1}, HY), \\
r.t.tupling(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}

By \( t = p + 1 \) times folding clause 39 using clauses 1 and 2:
Theorem 2 The generalization schema $TG_2$, which is given below, is correct.
\[ T_G : \{ DCRL, TG, A_{i2}, O_{i212}, O_{i221} \} \]

\( A_{i2} : \)
1. compose is associative
2. compose has \( \varepsilon \) as the left and right identity element, where \( \varepsilon \) appears in DCRL
3. \( \forall X : X. I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, \varepsilon) \)
4. \( \forall X : X. I_r(X) \Rightarrow [\text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)] \)

\( O_{i212} : \) partial evaluation of the conjunction
\[ \text{process}(HX, HY), \text{compose}(HY, TY, Y) \]
results in the introduction of a non-recursive relation

\( O_{i221} : \) 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 \( TG \) is Logic Program Template 2 in Theorem 1 and the template \( DCRL \) is Logic Program Template 3 below:

**Logic Program Template 3**

\[
\begin{align*}
\tau(X, Y) & = \\
\text{minimal}(X), \\
\text{solve}(X, Y)
\end{align*}
\]

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

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

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

and the specification \( S_{r,\text{upling}} \) of relation \( r_{\text{upling}} \) is:

\[
\forall X : X, \forall Y : Y. (\forall X : X. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{upling}}(Xs, Y) \Leftrightarrow \]
\[
(Xs = [] \land Y = \varepsilon) \\
\lor (Xs = [X_1, \ldots, X_n] \land \bigwedge_{i=1}^{n} O_r(X_i, Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{n} O_r(I_{i-1}, Y_i, I_i) \land Y = I_n)]
\]

**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 \( A_{i2} \). By Definition 5, the templates DCRL and TG are equivalent wrt \( S_r \) under the applicability conditions \( A_{i2} \) iff DCRL is equivalent to TG wrt the specification \( S_r \) provided that the conditions in \( A_{i2} \) 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 \( S = \{ S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solute}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \} \),

where \( S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solute}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \) are the specifications of \text{minimal}, \text{nonMinimal}, \text{solute}, \text{decompose}, \text{process}, \text{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 \( \{ S'_1, \ldots, S'_l \} \) 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 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: \( TG \) is steadfast wrt \( S_t \) in \( S \) if \( Pr_{\text{tupling}} \) is steadfast wrt \( S_{\text{tupling}} \) in \( S \), where \( Pr_{\text{tupling}} \) is the procedure for \( r \) and \( Pr_r \) is the procedure for \( r \).

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

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

\[
\forall Xs: \text{list of } X. \forall Y: \mathcal{Y}. \ (\forall X: \mathcal{X}. \ X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{tupling}}(Xs, Y) \Leftrightarrow (Xs = [\mathcal{E}] \land Y = e) \\
\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, Y_1) \land Y_1 = I_1 \land Y = I_1) \\
\lor (Xs = [X_1, X_2, \ldots , X_q] \land \bigwedge_{i=1}^q \mathcal{O}_r(X_i, Y_i) \land Y_1 = I_1 \land Y = I_q)]
\]

where \( q \geq 2 \).

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

\[
\forall Xs: \text{list of } X, \forall Y: \mathcal{Y}. \ (\forall X: \mathcal{X}. \ X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{tupling}}(Xs, Y) \Leftrightarrow (Xs = [\mathcal{E}] \land Y = e) \\
\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, Y_1) \land Y_1 = I_1 \land Y_1 = e \land \mathcal{O}_r(I_1, TY)) \\
\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, Y_1) \land Y_1 = I_1 \land Y = l_2 \land \\
\bigwedge_{i=1}^q \mathcal{O}_r(I_{i-1}, Y_i, I_i) \land TY = I_q \land \mathcal{O}_r(I_1, TY, Y))]\]

where \( q \geq 2 \).

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

\[
\forall Xs: \text{list of } X, \forall Y: \mathcal{Y}. \ (\forall X: \mathcal{X}. \ X \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{tupling}}(Xs, Y) \Leftrightarrow (Xs = [\mathcal{E}] \land Y = e) \\
\lor (Xs = [X_1] \land \mathcal{O}_r(X_1, HY) \land r_{\text{tupling}}(TXs, TY) \land \mathcal{O}_r(HY, TY, Y))]\]

By taking the ‘decomposition’:

\[
\text{clause 1: } r_{\text{tupling}}(Xs, Y) \leftarrow \\
Xs = [\mathcal{E}], Y = e
\]

\[
\text{clause 2: } r_{\text{tupling}}(Xs, Y) \leftarrow \\
Xs = [X_1] \land \mathcal{O}_r(X_1, HY), \\
r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, Y)
\]

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

\[
\text{clause 3: } r_{\text{tupling}}(Xs, Y) \leftarrow \\
Xs = [X_1], \text{minimal}(X), \\
r_{\text{tupling}}(TXs, TY), \\
\text{solve}(X, HY), \text{compose}(HY, TY, Y)
\]

\[
\text{clause 4: } r_{\text{tupling}}(Xs, Y) \leftarrow \\
Xs = [X_1], \text{nonMinimal}(X), \text{compose}(X, HY, TX_1, \ldots , TX_t), \\
r(TX_1, TY_1), \ldots , r(TX_t, TY_t), \\
l_{t+1} = e, \\
\text{compose}(TY_1, l_{t+1}, I_t), \ldots , \text{compose}(TY_p, l_{p+1}, I_p), \\
\text{process}(HX, HY), \text{compose}(HY, I_p, l_{p-1}), \\
\text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), \ldots , \text{compose}(TY_1, I_1, I_0), \\
HY = I_0, r_{\text{tupling}}(TXs, TY), \text{compose}(HY, TY, Y)
\]

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):
clause 5: \( r_{\text{Dupling}}(Xs, Y) \) = 
\[ Xs = [X \mid Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX, ..., TX), \]
\[ \text{minimal}(TX), ..., \text{minimal}(TX), \]
\[ r(TX, TY), ..., r(TX, TY), \]
\[ l_{i+1} = \epsilon, \]
\[ \text{compose}(TY, l_{i+1}, l_i), ..., \text{compose}(TY, l_{i+1}, l_i), \]
\[ \text{process}(HX, HY), \text{compose}(HY, HX, l_i, l_{i-1}), \]
\[ \text{compose}(TY, l_{i-1}, l_{i-2}), ..., \text{compose}(TY, l_{i-1}, l_{i-2}), \]
\[ HY = l_0, r_{\text{Dupling}}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 6: \( r_{\text{Dupling}}(Xs, Y) \) = 
\[ Xs = [X \mid Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX, ..., TX), \]
\[ \text{minimal}(TX), ..., \text{minimal}(TX), \]
\[ (\text{nonMinimal}(TX), ..., (\text{nonMinimal}(TX)), \]
\[ r(TX, TY), ..., r(TX, TY), \]
\[ l_{i+1} = \epsilon, \]
\[ \text{compose}(TY, l_{i+1}, l_i), ..., \text{compose}(TY, l_{i+1}, l_i), \]
\[ \text{process}(HX, HY), \text{compose}(HY, HX, l_i, l_{i-1}), \]
\[ \text{compose}(TY, l_{i-1}, l_{i-2}), ..., \text{compose}(TY, l_{i-1}, l_{i-2}), \]
\[ HY = l_0, r_{\text{Dupling}}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 7: \( r_{\text{Dupling}}(Xs, Y) \) = 
\[ Xs = [X \mid Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX, ..., TX), \]
\[ (\text{nonMinimal}(TX), ..., (\text{nonMinimal}(TX)), \]
\[ r(TX, TY), ..., r(TX, TY), \]
\[ l_{i+1} = \epsilon, \]
\[ \text{compose}(TY, l_{i+1}, l_i), ..., \text{compose}(TY, l_{i+1}, l_i), \]
\[ \text{process}(HX, HY), \text{compose}(HY, HX, l_i, l_{i-1}), \]
\[ \text{compose}(TY, l_{i-1}, l_{i-2}), ..., \text{compose}(TY, l_{i-1}, l_{i-2}), \]
\[ HY = l_0, r_{\text{Dupling}}(TXs, TY), \text{compose}(HY, TY, Y) \]

clause 8: \( r_{\text{Dupling}}(Xs, Y) \) = 
\[ Xs = [X \mid Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX, ..., TX), \]
\[ (\text{nonMinimal}(TX), ..., (\text{nonMinimal}(TX)), \]
\[ r(TX, TY), ..., r(TX, TY), \]
\[ l_{i+1} = \epsilon, \]
\[ \text{compose}(TY, l_{i+1}, l_i), ..., \text{compose}(TY, l_{i+1}, l_i), \]
\[ \text{process}(HX, HY), \text{compose}(HY, HX, l_i, l_{i-1}), \]
\[ \text{compose}(TY, l_{i-1}, l_{i-2}), ..., \text{compose}(TY, l_{i-1}, l_{i-2}), \]
\[ HY = l_0, r_{\text{Dupling}}(TXs, TY), \text{compose}(HY, TY, Y) \]

By \( t \) times unfolding clause 5 wrt. \( r(TX, TY), ..., r(TX, TY) \) using DCRL, and simplifying using condition (4):

clause 9: \( r_{\text{Dupling}}(Xs, Y) \) = 
\[ Xs = [X \mid Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX, ..., TX), \]
\[ \text{minimal}(TX), ..., \text{minimal}(TX), \]
\[ \text{solve}(TX, TY), ..., \text{solve}(TX, TY), \]
\[ l_{i+1} = \epsilon, \]
\[ \text{compose}(TY, l_{i+1}, l_i), ..., \text{compose}(TY, l_{i+1}, l_i), \]
\[ \text{process}(HX, HY), \text{compose}(HY, HX, l_i, l_{i-1}), \]
\[ \text{compose}(TY, l_{i-1}, l_{i-2}), ..., \text{compose}(TY, l_{i-1}, l_{i-2}), \]
\[ HY = l_0, r_{\text{Dupling}}(TXs, TY), \text{compose}(HY, TY, Y) \]
By using applicability condition (3):

\textbf{clause 10:}\quad r\text{Dupling}(Xs,Y) =
\begin{align*}
Xs &= \{X \mid X \in Xs\}, \\
non\text{Minimal}(X), &\text{decompose}(X,HX,TX_1, \ldots, TX_t), \\
minimal(TX_1), &\ldots, minimal(TX_t), \\
solve(TX_1, e), &\ldots, solve(TX_t, e), \\
l_{t+1} &= e, \\
\text{compose}(e,l_{t+1},l_t), &\ldots, \text{compose}(e,l_{p+1},l_p), \\
\text{process}(HX, HHY), &\text{compose}(HY, l_p, l_{p-1}), \\
\text{compose}(e,l_{p-1},l_{p-2}), &\ldots, \text{compose}(e,l_1,l_0), \\
HY &= l_0, r\text{Dupling}(TXs,TX), \text{compose}(HY,TY,Y)
\end{align*}

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

\textbf{clause 11:}\quad r\text{Dupling}(Xs,Y) =
\begin{align*}
Xs &= \{X \mid X \in Xs\}, \\
non\text{Minimal}(X), &\text{decompose}(X,HX,TX_1, \ldots, TX_t), \\
minimal(TX_1), &\ldots, minimal(TX_t), \\
solve(TX_1, e), &\ldots, solve(TX_t, e), \\
l_{t+1} &= e, \\
\text{compose}(e,l_{t+1},l_t), &\ldots, \text{compose}(e,l_{p+1},l_p), \\
\text{process}(HX, HHY), &\text{compose}(HY, l_p, l_{p-1}), \\
\text{compose}(e,l_{p-1},l_{p-2}), &\ldots, \text{compose}(e,l_1,l_0), \\
HY &= l_0, r\text{Dupling}(TXs,TX), \text{compose}(HY,TY,Y)
\end{align*}

By using applicability condition (2):

\textbf{clause 12:}\quad r\text{Dupling}(Xs,Y) =
\begin{align*}
Xs &= \{X \mid X \in Xs\}, \\
non\text{Minimal}(X), &\text{decompose}(X,HX,TX_1, \ldots, TX_t), \\
minimal(TX_1), &\ldots, minimal(TX_t), \\
solve(TX_1, e), &\ldots, solve(TX_t, e), \\
l_{t+1} &= e, \\
l_t &= l_{t+1}, \ldots, l_p = l_{p+1}, \\
\text{process}(HX, HHY), &\text{compose}(HY, l_p, l_{p-1}), \\
l_{p-2} &= l_{p-1}, \ldots, l_0 = l_1, \\
HY &= l_0, r\text{Dupling}(TXs,TX), \text{compose}(HY,TY,Y)
\end{align*}

By simplification:

\textbf{clause 13:}\quad r\text{Dupling}(Xs,Y) =
\begin{align*}
Xs &= \{X \mid X \in Xs\}, \\
non\text{Minimal}(X), &\text{decompose}(X,HX,TX_1, \ldots, TX_t), \\
minimal(TX_1), &\ldots, minimal(TX_t), \\
r\text{Dupling}(TXs,TX), &\text{process}(HX, HHY), \text{compose}(HY,TY,Y)
\end{align*}

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):
By rewriting clause /1/6 using applicability conditions /1/ and /2/:

\[ r_{\text{tuple}}(X, Y) = \]
\[ X = [X[X], X] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = l_0, r_{\text{tuple}}(TXs, TY), \text{compose}(HY, TY, Y) \]

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

\[ r_{\text{tuple}}(X, Y) = \]
\[ X = [X[X], X] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(TY_t, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = l_0, r_{\text{tuple}}(TXs, TY), \text{compose}(HY, TY, Y) \]

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

\[ r_{\text{tuple}}(X, Y) = \]
\[ X = [X[X], X] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p), HY = I_p, \]
\[ \text{compose}(TY_p, TY_{p+1}, I_{p+1}), \]
\[ \text{compose}(I_{p+1}, TY_{p+2}, I_{p+2}), \ldots, \text{compose}(I_{t+1}, TY_1, I_t), \]
\[ r_{\text{tuple}}(TXs, TY), \text{compose}(I_{t+1}, TTY, TY), \]
\[ \text{compose}(HY, TY, Y) \]

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

\[ r_{\text{tuple}}(X, Y) = \]
\[ X = [X[X], X] \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r_{\text{tuple}}(TXs, TXs[X], TY), \]
\[ I_{t+1} = \epsilon, \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p), HY = I_p, \]
\[ \text{compose}(HY, TY, Y) \]

By using applicability condition (3):
clause 18: \( r_tupling(X, Y) \) =
\[
\begin{align*}
X &= [X] P X, \\
non\text{Minimal}(X), &\text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, &\text{minimal}(TX_{\eta-1}), \\
(non\text{Minimal}(TX_\eta); \ldots; non\text{Minimal}(TX_t)), &\text{solve}(TX_1, e), \ldots, solve(TX_{\eta-1}, e), \\
r_tupling([TX_\eta, \ldots, TX_t] P X, TY), &l_0 = e, \\
&\text{compose}(l_0, e, l_1), \ldots, compose(l_{\eta-2}, e, l_{\eta-1}), \\
&\text{process}(HX, H HY), \text{compose}(l_{\eta-1}, H HY, l_\eta), H Y = l_\eta, \\
&\text{compose}(HY, TY, Y)
\end{align*}
\]
By using applicability condition (2):

clause 19: \( r_tupling(X, Y) \) =
\[
\begin{align*}
X &= [X] P X, \\
non\text{Minimal}(X), &\text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, &\text{minimal}(TX_{\eta-1}), \\
(non\text{Minimal}(TX_\eta); \ldots; non\text{Minimal}(TX_t)), &\text{solve}(TX_1, e), \ldots, solve(TX_{\eta-1}, e), \\
r_tupling([TX_\eta, \ldots, TX_t] P X, TY), &l_0 = e, \\
&l_1 = l_0, \ldots, l_{\eta-1} = l_{\eta-2}, \\
&\text{process}(HX, H HY), \text{compose}(l_{\eta-1}, H HY, l_\eta), H Y = l_\eta, \\
&\text{compose}(HY, TY, Y)
\end{align*}
\]
By simplification:

clause 20: \( r_tupling(X, Y) \) =
\[
\begin{align*}
X &= [X] P X, \\
non\text{Minimal}(X), &\text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, &\text{minimal}(TX_{\eta-1}), \\
(non\text{Minimal}(TX_\eta); \ldots; non\text{Minimal}(TX_t)), &\text{solve}(TX_1, e), \ldots, solve(TX_{\eta-1}, e), \\
r_tupling([TX_\eta, \ldots, TX_t] P X, TY), &\text{process}(HX, H HY), \text{compose}(HY, TY, Y)
\end{align*}
\]
By introducing atoms \( minimal(U_1), \ldots, minimal(U_{\eta-1}) \) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_{\eta-1} \)) in clause 7:

clause 21: \( r_tupling(X, Y) \) =
\[
\begin{align*}
X &= [X] P X, \\
non\text{Minimal}(X), &\text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, &\text{minimal}(TX_{\eta-1}), \\
minimal(U_1), \ldots, &\text{minimal}(U_{\eta-1}), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), &l_{t+1} = e, \\
&\text{compose}(TY_t, l_{t+1}, l_t), \ldots, compose(TY_\eta, l_{\eta+1}, l_{\eta}), \\
&\text{process}(HX, HY), \text{compose}(HY, l_\eta, l_{\eta-1}), \\
&\text{compose}(TY_\eta, l_{\eta-1}, l_{\eta-2}), \ldots, compose(TY_1, l_1, l_0), \\
&HY = l_0, r_tupling([TX_\eta, TY_\eta], compose(HY, TY, Y)
\end{align*}
\]
By using applicability condition (3):

23
clause 22: \( \text{r...} \) =
\[
X_1 = [X[TX_s],
\nonMinimal(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})),
\minimal(TX_p),\ldots,\minimal(TX_t),
\minimal(U_1),\ldots,\minimal(U_{p-1}),
\begin{align*}
&\begin{array}{c}
U_1,\ldots,U_{p-1},
\end{array}
&
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
X_{s/TX_s},
\end{array}
\end{array}
\end{array}
\end{align*}
\]
By using applicability condition (2):

clause 23: \( \text{r...} \) =
\[
X_1 = [X[TX_s],
\nonMinimal(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})),
\minimal(TX_p),\ldots,\minimal(TX_t),
\minimal(U_1),\ldots,\minimal(U_{p-1}),
\begin{align*}
&\begin{array}{c}
U_1,\ldots,U_{p-1},
\end{array}
&
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
X_{s/TX_s},
\end{array}
\end{array}
\end{array}
\end{align*}
\]
By using applicability conditions (1) and (2):

clause 24: \( \text{r...} \) =
\[
X_1 = [X[TX_s],
\nonMinimal(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(nonMinimal(TX_1);\ldots;nonMinimal(TX_{p-1})),
\minimal(TX_p),\ldots,\minimal(TX_t),
\minimal(U_1),\ldots,\minimal(U_{p-1}),
\begin{align*}
&\begin{array}{c}
U_1,\ldots,U_{p-1},
\end{array}
&
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
X_{s/TX_s},
\end{array}
\end{array}
\end{array}
\end{align*}
\]
By introducing new, i.e. existentially quantified, variables \( YU_1,\ldots,YU_{p-1} \) in place of some occurrences of \( \epsilon \):
clause 25: \( r_{\text{tupling}}(X, Y) \) —

\[
X_s = [X[T X_s],
\]
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}),
r(U_1, Y U_1), \ldots, r(U_p-1, Y U_{p-1}),
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
l_{t+1} = e,
\]
compos \( (TY_1, l_{t+1}, l_t), \ldots, compos \( (TY_p, l_{p+1}, l_p),
\]
compose \( (HX, HY), compos \( (HY, l_p, l_{p-1}),
\]
compos \( (HY, l_1, l_0),
\]
compos \( (HY, TY, TI),
\]
compos \( (TY_1, TY_2, K_1), compos \( (K_1, TY_3, K_2), \ldots, compos \( (K_{p-3}, TY_{p-1}, K_{p-2}),
\]
compos \( (K_{p-2}, TI, Y)\]

By introducing nonMinimal(N) and 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: \( r_{\text{tupling}}(X, Y) \) —

\[
X_s = [X[T X_s],
\]
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}),
r(U_1, Y U_1), \ldots, r(U_p-1, Y U_{p-1}),
\]
nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
l_{t+1} = e,
\]
compos \( (TY_1, l_{t+1}, l_t), \ldots, compos \( (TY_p, l_{p+1}, l_p),
\]
-compose \( (HX, HY), compos \( (HY, l_p, l_{p-1}),
\]
compos \( (HY, l_1, l_0),
\]
-compose \( (HY, TY, TI),
\]
compos \( (TY_1, TY_2, K_1), compos \( (K_1, TY_3, K_2), \ldots, compos \( (K_{p-3}, TY_{p-1}, K_{p-2}),
\]
compos \( (K_{p-2}, TI, Y)\]

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

clause 27: \( r_{\text{tupling}}(X, Y) \) —

\[
X_s = [X[T X_s],
\]
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}),
r(U_1, Y U_1), \ldots, r(U_p-1, Y U_{p-1}),
\]
nonMinimal(N), decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),
decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t),
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
l_{t+1} = e,
\]
compos \( (TY_1, l_{t+1}, l_t), \ldots, compos \( (TY_p, l_{p+1}, l_p),
\]
-compose \( (HX, HY), compos \( (HY, l_p, l_{p-1}),
\]
compos \( (HY, l_1, l_0),
\]
-compose \( (HY, TY, TI),
\]
compos \( (TY_1, TY_2, K_1), compos \( (K_1, TY_3, K_2), \ldots, compos \( (K_{p-3}, TY_{p-1}, K_{p-2}),
\]
compos \( (K_{p-2}, TI, Y)\]

By folding clause 27 using DCRL:
Clause 28: \( \text{ tupleing}(X,Y) \) →

\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, \text{HX}, \text{TX}_1, \ldots, \text{TX}_t), \]
\[ (\text{nonMinimal}(\text{TX}_1); \ldots; \text{nonMinimal}(\text{TX}_{p-1})), \]
\[ \text{minimal}(\text{TX}_p), \ldots, \text{minimal}(\text{TX}_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ \text{decompose}(N, \text{HX}, U_1, \ldots, U_{p-1}, \text{TX}_p, \ldots, \text{TX}_t), \]
\[ r(\text{TX}_1, \text{TY}_1), \ldots, r(\text{TX}_{p-1}, \text{TY}_{p-1}), r(N, HY), \]
\[ \text{tupleing}(\text{TXs}, \text{TY}), \text{compose}(\text{HY}, \text{TY}, \text{TI}), \]
\[ \text{compose}(\text{TY}_1, \text{TY}_2, K_1), \text{compose}(K_1, \text{TY}_3, K_2), \ldots, \text{compose}(K_{p-3}, \text{TY}_{p-1}, K_{p-2}), \]
\[ \text{compose}(K_{p-2}, T I, Y) \]

By folding clause 28 using clauses 1 and 2:

Clause 29: \( \text{ tupleing}(X,Y) \) →

\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, \text{HX}, \text{TX}_1, \ldots, \text{TX}_t), \]
\[ (\text{nonMinimal}(\text{TX}_1); \ldots; \text{nonMinimal}(\text{TX}_{p-1})), \]
\[ \text{minimal}(\text{TX}_p), \ldots, \text{minimal}(\text{TX}_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ \text{decompose}(N, \text{HX}, U_1, \ldots, U_{p-1}, \text{TX}_p, \ldots, \text{TX}_t), \]
\[ r(\text{TX}_1, \text{TY}_1), \ldots, r(\text{TX}_{p-1}, \text{TY}_{p-1}), \]
\[ \text{tupleing}(\text{TXs}, \text{TY}), \text{compose}(\text{TY}_1, \text{TY}_2, K_1), \text{compose}(K_1, \text{TY}_3, K_2), \ldots, \text{compose}(K_{p-3}, \text{TY}_{p-1}, K_{p-2}), \]
\[ \text{compose}(K_{p-2}, T I, Y) \]

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

Clause 30: \( \text{ tupleing}(X,Y) \) →

\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, \text{HX}, \text{TX}_1, \ldots, \text{TX}_t), \]
\[ (\text{nonMinimal}(\text{TX}_1); \ldots; \text{nonMinimal}(\text{TX}_{p-1})), \]
\[ \text{minimal}(\text{TX}_p), \ldots, \text{minimal}(\text{TX}_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ \text{decompose}(N, \text{HX}, U_1, \ldots, U_{p-1}, \text{TX}_p, \ldots, \text{TX}_t), \]
\[ r(\text{TX}_1, \text{TY}_1), \ldots, r(\text{TX}_{p-1}, \text{TY}_{p-1}), \]
\[ \text{tupleing}(\text{TXs}, \text{TY}), \text{compose}(\text{TX}_1, \ldots, \text{TX}_{p-1}, N[TXs], Y) \]

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: \( \text{ tupleing}(X,Y) \) →

\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, \text{HX}, \text{TX}_1, \text{TX}_2), \]
\[ (\text{nonMinimal}(\text{TX}_1); \ldots; \text{nonMinimal}(\text{TX}_{p-1})), \]
\[ (\text{non Minimal}(\text{TX}_p); \ldots; \text{non Minimal}(\text{TX}_t)), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{t}), \]
\[ r(\text{TX}_1, \text{TY}_1), \ldots, r(\text{TX}_t, \text{TY}_t), \]
\[ t_{t+1} = e, \]
\[ \text{compose}(\text{TY}_1, l_{t+1}, l_t), \ldots, \text{compose}(\text{TY}_p, l_{p+1}, l_p), \]
\[ \text{process}(\text{HX}, \text{HHY}), \text{compose}(\text{HHY}, l_{p+1}, l_p), \]
\[ \text{compose}(\text{TY}_{p-1}, l_{p-1}, l_{p-1}), \ldots, \text{compose}(\text{TY}_1, l_1, l_0), \]
\[ \text{HY} = l_0, \text{tupleing}(\text{TXs}, \text{TY}), \text{compose}(\text{HY}, \text{TY}, Y) \]

By using applicability condition (3):
\text{clause 32: \(r\text{\texttt{tupling}}(Xs,Y)\) =}
\begin{align*}
Xs &= [X[TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, TX_2), \\
(non\text{Minimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(non\text{Minimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, \epsilon), \ldots, r(U_t, \epsilon), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_{t+1} = \epsilon, \\
\text{compose}(TY_t, l_{t+1}, l_t), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{compose}(e, l_p, K_t), \\
\text{compose}(e, K_{t+1}, K_t), \ldots, \text{compose}(e, K_{p+1}, K_p), \\
\text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \\
\text{compose}(TY_{p+1}, l_{p+1}, l_{p-2}), \ldots, \text{compose}(TY_1, l_0, l_0), \\
HY = l_0, r\text{\texttt{tupling}}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}

By using applicability condition (2):

\text{clause 33: \(r\text{\texttt{tupling}}(Xs,Y)\) =}
\begin{align*}
Xs &= [X[TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(non\text{Minimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(non\text{Minimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, \epsilon), \ldots, r(U_t, \epsilon), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_{t+1} = \epsilon, \\
\text{compose}(TY_t, l_{t+1}, l_t), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{compose}(e, l_p, K_t), \\
\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, l_{p-1}), \\
\text{compose}(TY_{p+1}, l_{p+1}, l_{p-2}), \ldots, \text{compose}(TY_1, l_0, l_0), \\
HY = l_0, r\text{\texttt{tupling}}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}

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

\text{clause 34: \(r\text{\texttt{tupling}}(Xs,Y)\) =}
\begin{align*}
Xs &= [X[TXs], \\
non\text{Minimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(non\text{Minimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(non\text{Minimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, \epsilon), \ldots, r(U_t, \epsilon), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_{t+1} = \epsilon, \\
\text{compose}(e, l_{t+1}, l_t), \ldots, \text{compose}(e, l_{p+1}, l_p), \\
\text{compose}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \\
\text{compose}(e, l_{p+1}, l_{p-2}), \ldots, \text{compose}(e, l_1, l_0), \\
HY = l_0, \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p+3}, TY_{p+1}, K_{p+2}), \\
\text{compose}(TY_{p+1}, TY_{p+2}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{p+2}, TY_t, K_{t-1}), \\
\text{compose}(K_{p+2}, NHY, T1), \text{compose}(T1, K_{t-1}, HY), \\
r\text{\texttt{tupling}}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}

By introducing new, i.e. existentially quantified, variables \(YU_1, \ldots, YU_t\) in place of some occurrences of \(\epsilon\):
\[
\text{clause 35: } r\text{tupling}(X, Y) =
\]
\[
X = [X[T Xs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{T-1})),
\]
\[
(\text{nonMinimal}(TX_T); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_t),
\]
\[
r(U_1, YU_1), \ldots, r(U_t, YU_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_{t+1} = e,
\]
\[
\text{compose}(YU_1, l_{t+1}, l_t), \ldots, \text{compose}(YU_p, l_{p+1}, l_p),
\]
\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}),
\]
\[
\text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0),
\]
\[
\text{NHY} = I_0,
\]
\[
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{T-3}, TY_{T-1}, K_{T-2}),
\]
\[
\text{compose}(TY_T, TY_{T+1}, K_T), \text{compose}(K_T, TY_{T+1}, K_{T+1}), \ldots, \text{compose}(K_{T-2}, TY_{T-1}, K_{T-1}),
\]
\[
\text{compose}(K_{T-2}, \text{NHY}, T1), \text{compose}(T1, K_{T-1}, HY),
\]
\[
r\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\]

By introducing nonMinimal(N) and 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):

\[
\text{clause 36: } r\text{tupling}(X, Y) =
\]
\[
X = [X[T Xs],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{T-1})),
\]
\[
(\text{nonMinimal}(TX_T); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_t),
\]
\[
r(U_1, YU_1), \ldots, r(U_t, YU_t),
\]
\[
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_{t+1} = e,
\]
\[
\text{compose}(YU_1, l_{t+1}, l_t), \ldots, \text{compose}(YU_p, l_{p+1}, l_p),
\]
\[
\text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}),
\]
\[
\text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0),
\]
\[
\text{NHY} = I_0,
\]
\[
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{T-3}, TY_{T-1}, K_{T-2}),
\]
\[
\text{compose}(TY_T, TY_{T+1}, K_T), \text{compose}(K_T, TY_{T+1}, K_{T+1}), \ldots, \text{compose}(K_{T-2}, TY_{T-1}, K_{T-1}),
\]
\[
\text{compose}(K_{T-2}, \text{NHY}, T1), \text{compose}(T1, K_{T-1}, HY),
\]
\[
r\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\]

By duplicating goal decompose(N, HX, U_1, \ldots, U_t):
\[\text{tupling}(X, Y) = \]
\[\begin{align*}
X_0 & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\text{decompose}(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
I_{t+1} & = \epsilon, \\
\text{compose}(Y U_1, I_{t+1}, l_1), \ldots, \text{compose}(Y U_{p-1}, l_{p-1}, l_p), \\
\text{process}(HX, HY l_p, l_{p-1}), \\
\text{compose}(Y U_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(Y U_1, I_1, I_0), \\
N HY & = I_0, \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(K_{t-2}, N HY, T1), \text{compose}(T1, K_{t-1}, HY), \\
\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}\]

By folding clause 37 using DCRIL:

\[\text{tupling}(X, Y) = \]
\[\begin{align*}
X_0 & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\text{decompose}(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, N HY), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(K_{t-2}, N HY, T1), \text{compose}(T1, K_{t-1}, HY), \\
\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}\]

By using applicability condition (1):

\[\text{tupling}(X, Y) = \]
\[\begin{align*}
X_0 & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\text{decompose}(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, N HY), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(K_{t-2}, N HY, T1), \text{compose}(T1, K_{t-1}, HY), \\
\text{tupling}(TXs, TY), \text{compose}(HY, TY, Y)
\end{align*}\]

By \(t = p + 1\) times folding clause 39 using clauses 1 and 2:
Theorem 3 The generalization schema $DG_1$, which is given below, is correct.
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**

\[
\begin{align*}
r(X,Y) & \leftarrow \\
r_{\text{descending}}_1(X,Y,e) & \\
r_{\text{descending}}_1(X,Y,A) & \leftarrow \\
  \text{minimal}(X), \\
  \text{solve}(X,S), \text{compose}(A,S,Y) \\
r_{\text{descending}}_1(X,Y,A) & \leftarrow \\
  \text{nonMinimal}(X), \\
  \text{decompose}(X,HX,TX_1,\ldots,TX_T), \\
  \text{compose}(A,e,A_0), \\
  r_{\text{descending}}_1(TX_1,A_1,A_0), \ldots, r_{\text{descending}}_1(TX_{p-1},A_{p-1},A_{p-2}), \\
  \text{process}(HX,HY), \text{compose}(A_{p-1},HY,A_p), \\
  r_{\text{descending}}_1(TX_p,A_{p+1},A_p), \ldots, r_{\text{descending}}_1(TX_T,A_{t+1},A_t), \\
  Y = A_{t+1}
\end{align*}
\]

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

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

and the specification \( S_{r_{\text{descending}}_1} \) of relation \( r_{\text{descending}}_1 \) is:

\[
\forall X : X. \forall Y, A : Y. \; \text{I}_r(X) \Rightarrow [r_{\text{descending}}_1(X,Y,A) \Leftrightarrow \exists S : Y. \; O_r(X,S) \land O_r(A,S,Y)]
\]

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

(a) DCLR is steadfast wrt \( S_r \) in \( S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \), 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 DCLR.

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

Note that the sets \( \{S_1, \ldots, S_m\} \) and \( \{S'_1, \ldots, S'_t\} \) 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 steadiness: \(DCLR\) is steadfast wrt \(S_r\) in \(S\) if \(P_r\) is steadfast wrt \(S_r\) in \(S\), where \(P_r\) is the procedure for \(r\), and \(P\) is steadfast wrt \(S_r\) in \(\{S_r\}\), where \(P\) is the procedure for \(r\).

To prove that \(P_r\) 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 \(P_r\).

By taking the ‘decomposition’ of \(S_r\):

\[\text{clauses 1: } r \cdot \text{decomposition}_1(X,Y,A) \iff r(X,S) \land \text{compose}(A,S,Y)\]

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

\[\text{clauses 2: } r \cdot \text{decomposition}_2(X,Y,A) \iff \text{minimal}(X), \text{solve}(X,S) \land \text{compose}(A,S,Y)\]

\[\text{clauses 3: } r \cdot \text{decomposition}_3(X,Y,A) \iff \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), r(TX_1,TS_1), \ldots, r(TX_t,TS_t), I_0 = e, \text{compose}(I_0,TS_1,I_1), \ldots, \text{compose}(I_{p-2},TS_{p-1},I_{p-1}), \text{process}(HX,HX), \text{compose}(I_{p-1},HS,I_p), \text{compose}(I_p,TS_p,I_{p+1}), \ldots, \text{compose}(I_{t-1},TS_t,I_t), S = I_{t+1}, \text{compose}(A,S,Y)\]

By using applicability condition (1) on clause 3:

\[\text{clauses 4: } r \cdot \text{decomposition}_4(X,Y,A) \iff \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), r(TX_1,TS_1), \ldots, r(TX_t,TS_t), \text{compose}(A,e,A_0), \text{compose}(A_0,TS_1,A_1), \ldots, \text{compose}(A_{p-2},TS_{p-1},A_{p-1}), \text{process}(HX,HX), \text{compose}(A_{p-1},HS,A_p), \text{compose}(A_p,TS_p,A_{p+1}), \ldots, \text{compose}(A_{t-1},TS_t,A_{t+1}), Y = A_{t+1}\]

By \(t\) times folding clause 4 using clause 1:

\[\text{clauses 5: } r \cdot \text{decomposition}_5(X,Y,A) \iff \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t), \text{compose}(A,e,A_0), r \cdot \text{decomposition}_1(TX_1,A_1,A_0), \ldots, r \cdot \text{decomposition}_1(TX_{p-1},A_{p-1},A_{p-2}), \text{process}(HX,HX), \text{compose}(A_{p-1},HY,A_p), r \cdot \text{decomposition}_1(TX_p,A_{p+1},A_p), \ldots, r \cdot \text{decomposition}_1(TX_t,A_{t+1},A_t), Y = A_{t+1}\]

Clauses 2 and 5 are the clauses of the \(P_r\) that in \(S_r\). Therefore \(P_r\) is steadfast wrt \(S_r\) in \(S\).

To prove that \(P_r\) is steadfast wrt \(S_r\) in \(\{S_r\}\), 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:

\[r(X,Y) \iff r \cdot \text{decomposition}_1(X,Y,e)\]

By taking the ‘completion’:

\[\forall X:X, \forall Y:Y. \ T_r(X) \iff [r(X,Y) \iff r \cdot \text{decomposition}_1(X,Y,e)]\]

By unfolding the ‘completion’ above wrt \(r \cdot \text{decomposition}_1(X,Y,e)\) using \(S_r\):

\[\forall X:X, \forall Y:Y. \ T_r(X) \iff [r(X,Y) \iff \exists S:Y. \ \text{O}_r(X,S) \land S = Y]\]

By using applicability condition (2):

\[\forall X:X, \forall Y:Y. \ T_r(X) \iff [r(X,Y) \iff \exists S:Y. \ \text{O}_r(X,S) \land S = Y]\]
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 \(\{S_{\text{descending}}\}\).
Therefore, \(DGLR\) is also steadfast wrt \(S_r\) in \(S\).

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

\[
DG_2 : \{\ DCLR,\ DGLRL,\ A_{dg2},\ O_{dg212},\ O_{dg221}\ \}\text{ where} \\
A_{dg2} : (1) \text{ compose is associative} \\
(2) \text{ compose has } e \text{ as the left and right identity element,} \\
\text{ where } e \text{ appears in } DCLR \text{ and } DGLRL \\
O_{dg212} : - I_r(X) \land \text{ minimal}(X) \Rightarrow O_r(X,e) \\
- \text{ partial evaluation of the conjunction} \\
\text{ process}(HX, HY), \text{ compose}(HY, A_p, A_{p-1}) \\
\text{ results in the introduction of a non-recursive relation} \\
O_{dg221} : - \text{ partial evaluation of the conjunction} \\
\text{ process}(HX, HY), \text{ compose}(I_{p-1}, HY, I_p) \\
\text{ results in the introduction of a non-recursive relation} \\
\]

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

**Logic Program Template 5**

\[
\begin{align*}
\text{r}(X,Y) & \leftarrow \\
r_{\text{descending}}(X,Y,e) & \leftarrow \\
r_{\text{descending}}(X,Y,A) & \leftarrow \text{ minimal}(X), \\
& \text{ solve}(X,S), \text{ compose}(S,A,Y) \\
r_{\text{descending}}(X,Y,A) & \leftarrow \text{ nonMinimal}(X), \\
& \text{ decompose}(X,HX,TX_1, \ldots, TX_t), \\
& \text{ compose}(e,A,A_{t+1}), \\
& r_{\text{descending}}(TX_t, A_t, A_{t+1}), \ldots, r_{\text{descending}}(TX_p, A_p, A_{p+1}), \\
& \text{ process}(HX, HY), \text{ compose}(HY, A_p, A_{p-1}), \\
& r_{\text{descending}}(TX_{p-1}, A_{p-2}, A_{p-1}), \ldots, r_{\text{descending}}(TX_1, A_0, A_1), \\
Y & = A_0
\end{align*}
\]

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

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

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

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

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

(a) \(DCLR\) is steadfast wrt \(S_r\) in \(S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}\), 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) DGRL is also steadfast wrt $S$ in $\mathcal{S}$.

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

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

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

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

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

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

\begin{align*}
\text{minimal}(X), \\
\text{solve}(X, S), \text{compose}(S, A, Y)
\end{align*}

clause 3: $r_{\text{descending}}(X, Y, A) \leftarrow$

\begin{align*}
\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 = \epsilon, \\
\text{compose}(l_0, TS_1, l_1), \ldots, \text{compose}(l_{p-2}, TS_{p-1}, l_{p-1}), \\
\text{process}(HX, HS), \text{compose}(l_{p+1}, HS, l_p), \\
\text{compose}(l_p, TS_p, l_{p+1}), \ldots, \text{compose}(l_t, TS_t, l_{t+1}), \\
S = l_{t+1}, \text{compose}(S, A, Y)
\end{align*}

By using applicability condition (1) on clause 3:

clause 4: $r_{\text{descending}}(X, Y, A) \leftarrow$

\begin{align*}
\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_t, A, A_t), \ldots, \text{compose}(TS_p, S_{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}(\epsilon, A_0, Y)
\end{align*}

By using applicability condition (2) on clause 4:

clause 5: $r_{\text{descending}}(X, Y, A) \leftarrow$

\begin{align*}
\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_t, A, A_t), \ldots, \text{compose}(TS_p, S_{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
\end{align*}

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

clause 6: $r_{\text{descending}}(X, Y, A) \leftarrow$

\begin{align*}
\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}(\epsilon, A, A_{t+1}), \\
\text{compose}(TS_t, A_{t+1}, A_t), \ldots, \text{compose}(TS_p, S_{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
\end{align*}
By $t$ times folding clause 6 using clause 1:

\[ \text{clause 7: } r_{\text{descending}_{2}}(X,Y,A) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X,HX,TX_{1},\ldots,TX_{t}), \]
\[ \text{compose}(e,A,A_{t+1}), \]
\[ r_{\text{descending}_{2}}(TX_{1},A_{t},A_{t+1}), \ldots, r_{\text{descending}_{2}}(TX_{p},A_{p},A_{p+1}), \]
\[ \text{process}(HX,HY), \text{compose}(HY,A_{p},A_{p-1}), \]
\[ r_{\text{descending}_{2}}(TX_{p-1},A_{p-2},A_{p-1}), \ldots, r_{\text{descending}_{2}}(TX_{1},A_{0},A_{1}), \]
\[ Y = A_{0} \]

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, which is given below, is correct.

\[ \text{DG} : \{ \text{DCRL, DGRL, A}_{dg}, O_{dg312}, O_{dg321} \} \]

where

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

- \( O_{dg312} : \) - \text{compose} has $e$ as the left 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 $S$, under the applicability conditions $A_{dg3}$. By Definition 5, the templates DCRL and DGRL are equivalent wrt $S$, under the applicability conditions $A_{dg3}$ iff DCRL is equivalent to DGRL wrt the specification $S$, provided that the conditions in $A_{dg3}$ hold.

By Definition 4, DCRL is equivalent to DGRL wrt the specification $S$, iff the following two conditions hold:

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

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

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

In program transformation, we assume that the input program, here template DCRL, is steadfast wrt $S$, 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 $S$ in $\mathcal{S}$ if $P_{r,descending_2}$ is steadfast wrt $S_{r,descending_2}$ in $\mathcal{S}$, where $P_{r,descending_2}$ is the procedure for $r_{descending_2}$, and $P_r$ is steadfast wrt $S_r$ in $\{S_{r,descending_2}\}$, where $P_r$ is the procedure for $r$.

To prove that $P_{r,descending_2}$ is steadfast wrt $S_{r,descending_2}$ in $\mathcal{S}$, we do a constructive forward proof that we begin with $S_{r,descending_2}$ and from which we try to obtain $P_{r,descending_2}$.

By taking the ‘decompletion’ of $S_{r,descending_2}$:

\text{clause 1 : } r_{descending_2}(X, Y, A) \leftarrow r(X, S), compose(S, A, Y)

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

\text{clause 2 : } r_{descending_2}(X, Y, A) \leftarrow
\begin{align*}
\text{minimal}(X), \\
\text{solve}(X, S), \text{compose}(S, A, Y)
\end{align*}

\text{clause 3 : } r_{descending_2}(X, Y, A) \leftarrow
\begin{align*}
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
r(TX_1, TS_1), \ldots, r(TX_t, TS_t), \\
l_{t+1} = e, \\
\text{compose}(TS_t, l_{t+1}, l_t), \ldots, \text{compose}(TS_p, l_{p+1}, l_p), \\
\text{process}(HX, HS), \text{compose}(HS, l_p, l_{p-1}), \\
\text{compose}(TS_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(TS_1, l_1, l_0), \\
S = l_0, \text{compose}(S, A, Y)
\end{align*}

By using applicability condition (1) on clause 3:

\text{clause 4 : } r_{descending_2}(X, Y, A) \leftarrow
\begin{align*}
\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_t, A_{t+1}, A_t), \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
\end{align*}

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

\text{clause 5 : } r_{descending_2}(X, Y, A) \leftarrow
\begin{align*}
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{compose}(e, A, A_{t+1}), \\
r_{descending_2}(TX_1, A_1, A_{t+1}), \ldots, r_{descending_2}(TX_p, A_p, A_{p+1}), \\
\text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \\
r_{descending_2}(TX_{p-1}, A_{p-1}, A_{p-2}), \ldots, r_{descending_2}(TX_1, A_1, A_0), \\
Y = A_0
\end{align*}
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’:

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

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

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

By using applicability condition (2):

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

By simplification:

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

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 $DG_4$, which is given below, is correct.

$$DG_4 : \{ DCRL, DGLR, A_{dp4}, O_{dp421}, O_{dp412} \} \text{ where}$$

- $A_{dp4}$: $\text{- compose is associative}$
  - $\text{- compose has } e \text{ as the left and right identity element}$
  - where $e$ appears in DCRL and DGLR

- $O_{dp412}$: $\text{- } I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X,e)$
  - partial evaluation of the conjunction

- $O_{dp421}$: $\text{- } I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X,e)$
  - partial evaluation of the conjunction

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

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

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

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

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

**Proof 6** To prove the correctness of the generalization schema $DG_4$, by Definition 10, we have to prove that templates DCRL and DGLR are equivalent wrt $S_r$ under the applicability conditions $A_{dp4}$. By Definition 5, the templates DCRL and DGLR are equivalent wrt $S_r$ under the applicability conditions $A_{dp4}$ iff DCRL is equivalent to DGLR wrt the specification $S_r$ provided that the conditions in $A_{dp4}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{compose}}, S_{\text{process}}, S_{\text{decompose}}\}$, 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 $\mathcal{S}$.

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

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

To prove equivalence, we have to prove condition (b). We will use the following property of steadfastness: $DGLR$ is steadfast wrt $S_r$ in $\mathcal{S}$ if $P_{r\text{-descending}_1}$ is steadfast wrt $S_{r\text{-descending}_1}$ in $\mathcal{S}$, 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 $\mathcal{S}$, 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 ‘decompletion’ of $S_{r\text{-descending}_1}$:

clause 1: $r_{\text{descending}_1}(X, Y, A) \leftarrow r(X, S), \text{compose}(A, S, Y)$

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

clause 2: $r_{\text{descending}_1}(X, Y, A) \leftarrow$
minimal(X),
solve(X, S), \text{compose}(A, S, Y)$

clause 3: $r_{\text{descending}_1}(X, Y, A) \leftarrow$
nonMinimal(X), decompose(X, $HX, TX_1, \ldots, TX_l$),
$r(TX_1, TS_1), \ldots, r(TX_l, TS_l)$,
$I_{l+1} = e$,\ncompose(TS, I_{l+1}, I_l, \ldots, compose(TS, I_{l+1}, I_{l-1}),$
process(HX, HS), compose(HS, I_l, I_{l-1}),$
compose(TS_{p-1}, I_{p-2}, I_{p-1}), \ldots, compose(TS_{l}, I_{l-1}, I_l),$
$S = I_0, \text{compose}(A, S, Y)$

By using applicability condition (1) on clause 3:

clause 4: $r_{\text{descending}_1}(X, Y, A) \leftarrow$
nonMinimal(X), decompose(X, $HX, TX_1, \ldots, TX_l$),
$r(TX_1, TS_1), \ldots, r(TX_l, TS_l)$,
compose(A, TS, A_1), \ldots, compose(A_{p-2}, TS_{p-1}, A_{p-1}),$
process(HX, HS), compose(A_{p-1}, HS, A_p),$
compose(A_p, TS_{p}, A_{p+1}), \ldots, compose(A_t, TS_{t}, A_{t+1}),$
$Y = A_{t+1}$

By using applicability condition (2) on clause 4:

clause 5: $r_{\text{descending}_1}(X, Y, A) \leftarrow$
nonMinimal(X), decompose(X, $HX, TX_1, \ldots, TX_l$),
$r(TX_1, TS_1), \ldots, r(TX_l, TS_l)$,
compose(A, TS, A_1), \ldots, compose(A_{p-2}, TS_{p-1}, A_{p-1}),$
process(HX, HS), compose(A_{p-1}, HS, A_p),$
compose(A_p, TS_{p}, A_{p+1}), \ldots, 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 $A_{g}$:

clause 6: $r_{\text{descending}_1}(X, Y, A) \leftarrow$
nonMinimal(X), decompose(X, $HX, TX_1, \ldots, TX_l$),
$r(TX_1, TS_1), \ldots, r(TX_l, TS_l)$,
compose(A, e, A_{g}),$
compose(A_{g}, TS, A_1), \ldots, compose(A_{p-2}, TS_{p-1}, A_{p-1}),$
process(HX, HS), compose(A_{p-1}, HS, A_p),$
compose(A_p, TS_{p}, A_{p+1}), \ldots, compose(A_t, TS_{t}, A_{t+1}),$
$Y = A_{t+1}$
By \( t \) times folding clause 6 using clause 1:

\begin{align*}
\text{clause 7: } & \ r_{\text{descending}}(X, Y, A) \\
& \text{nonMinimal}(X), \ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \text{compose}(A, \epsilon, A_0), \\
& r_{\text{descending}}(TX_1, A_1, A_0), \ldots, r_{\text{descending}}(TX_p-1, A_p-1, A_{p-1}), \\
& \text{process}(HX, HY), \ \text{compose}(A_{p-1}, HY, A_p), \\
& r_{\text{descending}}(TX_p, A_p+1, A_p), \ldots, r_{\text{descending}}(TX_t, A_t+1, A_t), \\
& Y = A_{t+1}
\end{align*}

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

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

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

By taking the ‘completion’:

\[ \forall X : X, \forall Y : Y. \ I_r(X) \supseteq [r(X, Y) \Leftrightarrow r_{\text{descending}}(X, Y, \epsilon)] \]

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

\[ \forall X : X, \forall Y : Y. \ I_r(X) \supseteq [r(X, Y) \Leftrightarrow \exists S : Y. O_r(X, S) \land O_r(\epsilon, S, Y)] \]

By using applicability condition (2):

\[ \forall X : X, \forall Y : Y. \ I_r(X) \supseteq [r(X, Y) \Leftrightarrow \exists S : Y. O_r(X, S) \land S = Y] \]

By simplification:

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

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

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

\[ \square \]

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

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

\[ TDG_1 : \{ DCLR, T DGLR, A_{td1}, O_{td12}, O_{td21} \} \]

where

\[ A_{td1} : (1) \ \text{compose} \ \text{is associative} \]

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

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

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

\[ O_{td12} : \text{partial evaluation of the conjunction} \]

\[ \text{process}(HX, HY), \ \text{compose}(A, HY, \text{New} A) \]

results in the introduction of a non-recursive relation

\[ O_{td21} : \text{partial evaluation of the conjunction} \]

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

results in the introduction of a non-recursive relation

where the template \( DCLR \) is Logic Program Template 1 in Section 2 and the template \( T DGLR \) is Logic Program Template 6 below:
\[ r(X, Y) \]
\[ r_{td_1}(X, Y, e) \]
\[ r_{td_1}(Xs, Y, A) \]
\[ Xs = [], \]
\[ Y = A \]
\[ r_{td_1}(Xs, Y, A) \]
\[ Xs = [X[TXs], \]
\[ \text{minimal}(X), \]
\[ \text{solve}(X, HY), \]
\[ \text{compose}(A, HY, NewA), \]
\[ r_{td_1}(TXs, Y, NewA) \]
\[ r_{td_1}(Xs, Y, A) \]
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{process}(HX, HY), \text{compose}(A, HY, NewA), \]
\[ r_{td_1}(TXp, \ldots, TX_t[TXs], Y, NewA) \]
\[ r_{td_1}(Xs, Y, A) \]
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ \text{decompose}(N, TXp, \ldots, TX_t), \]
\[ r_{td_1}(TX_1, \ldots, TX_{p-1}, N[TXs], Y, A) \]
\[ r_{td_1}(Xs, Y, A) \]
\[ Xs = [X[TXs], \]
\[ \text{nonMinimal}(X), \]
\[ \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \]
\[ \text{compose}(N, TXp, \ldots, U_{t}), \]
\[ r_{td_1}(TX_1, \ldots, TX_{p-1}, N, TXp, \ldots, TX_t[TXs], Y, A) \]
and the specification $S_r$ of relation $r$ is:

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

and the specification $S_{r \downarrow d_i}$:

$$\forall X : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow \{r_{d_i}(Xs,Y,A) \iff (Xs = [] \land Y = A)$$

$$\land \bigwedge_{i=1}^{q} O_r(X_i,Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{q} O_c(I_{i-1},Y_i,I_i) \land O_c(A_i,I_i,I_{i+1}) \land Y = I_{q+1}]]$$

**Proof 7** To prove the correctness of the generalization schema $TDG_1$, by Definition 10, we have to prove that templates $DCLR$ and $TDGLR$ are equivalent wrt $S_r$ under the applicability conditions $A_{tdi}$. By Definition 5, the templates $DCLR$ and $TDGLR$ are equivalent wrt $S_r$ under the applicability conditions $A_{tdi}$ if $DCLR$ is equivalent to $TDGLR$ wrt the specification $S_r$ provided that the conditions in $A_{tdi}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solver}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\}$, where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solver}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}$ are the specifications of minimal, nonMinimal, solver, decompose, process, 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 tupleing-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: $TDGLR$ is steadfast wrt $S_r$ in $S$ if $P_{r \downarrow d_i}$ is steadfast wrt $S_{r \downarrow d_i}$ in $S$, where $P_{r \downarrow d_i}$ is the procedure for $r_{d_i}$, and $P_r$ is steadfast wrt $S_c$ in $\{S_{r \downarrow d_i}\}$, where $P_r$ is the procedure for $r$.

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

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

$$\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow \{r_{d_i}(Xs,Y,A) \iff$$

$$\begin{align*}
(Xs &= [] \land Y = A) \\
\land &\bigwedge_{i=1}^{q} O_r(X_i,Y_i) \land I_1 = Y_1 \land \bigwedge_{i=2}^{q} O_c(I_{i-1},Y_i,I_i) \land O_c(A_i,I_i,I_{i+1}) \land Y = I_{q+1}]]
\end{align*}$$

where $q \geq 2$.

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

$$\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow \{r_{d_i}(Xs,Y,A) \iff$$

$$\begin{align*}
(Xs &= [] \land Y = A) \\
\land &\bigwedge_{i=1}^{q} O_r(X_i,Y_i) \land Y_i = I_1 \land \bigwedge_{i=2}^{q} O_c(X_i,Y_i) \land Y_i = Y_1 \land \bigwedge_{i=2}^{q} O_c(I_{i-1},Y_i,I_i) \land \bigwedge_{i=2}^{q} O_c(I_{i-1},Y_i,I_i) \land TY = I_{q+1} \land O_c(A_i,I_i,I_{i+1}) \land O_c(TY,Y)]
\end{align*}$$

where $q \geq 2$.

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

$$\forall Xs : \text{list of } \mathcal{X}, \forall Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow \{r_{d_i}(Xs,Y,A) \iff$$

$$\begin{align*}
(Xs &= [] \land Y = A) \\
\land &\{Xs = [T]Xs] \land O_r(X,Y) \land O_c(A,H_Y,Y) \land r_{d_i}(TXs,Y,A)\}
\end{align*}$$

By taking the ‘decomposition’:

- **clause 1:** $r_{d_i}(Xs,Y,A) \Rightarrow Xs = [] \land Y = A$
- **clause 2:** $r_{d_i}(Xs,Y,A) \Rightarrow Xs = [T]Xs, r(X,Y),$ compose$(A,H_Y,Y), r_{d_i}(TXs,Y,A)$

By taking the ‘completeness’:

- **clause 1:** $r_{d_i}(Xs,Y,A) \Rightarrow Xs = [] \land Y = A$
- **clause 2:** $r_{d_i}(Xs,Y,A) \Rightarrow Xs = [T]Xs, r(X,Y),$ compose$(A,H_Y,Y), r_{d_i}(TXs,Y,A)$

41
By unfolding clause 2 wrt $r(X, HY)$ using $DCLR$, and using the assumption that $DCLR$ is steadfast wrt $S$, in $S$:

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

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

By introducing

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

in clause 4, using applicability condition (4):

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

clause 6: \[ r \Delta d_1(Xs, Y, A) \leftarrow \]
\[ Xs = [X \uparrow Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p) \land \ldots \land \text{nonMinimal}(TX_t)), \]
\[ \text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ \text{process}(HX, HY), \text{compose}(I_{p-1}, HHY, I_p), \]
\[ \text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \]
\[ HY = h_{t+1}, \text{compose}(A, HY, NA), \]
\[ r \Delta d_1(TXs, Y, NA) \]
clause 7: \[ \mathcal{d} d_1(X, Y, A) \leftarrow X s = [X[T X s],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{t-1})),
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_t),
r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t),
l_0 = e,
\text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{t-2}, T Y_{t-1}, l_{t-1}),
\text{process}(H X, H Y), \text{compose}(l_{t-1}, H Y, l_t),
\text{compose}(l_t, T Y_t, l_{t+1}), \ldots, \text{compose}(l_t, T Y_t, l_{t+1}),
HY = l_{t+1}, \text{compose}(A, HY, NA),
\mathcal{r} \mathcal{d} d_1(T X s, Y, NA)\]

By \( t \) times unfolding clause 5 wrt \( r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t) \) using DC LR, and simplifying using condition (4):

clause 8: \[ \mathcal{d} d_1(X, Y, A) \leftarrow X s = [X[T X s],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{t-1})),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_t)),
r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t),
l_0 = e,
\text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{t-2}, T Y_{t-1}, l_{t-1}),
\text{process}(H X, H Y), \text{compose}(l_{t-1}, H Y, l_t),
\text{compose}(l_t, T Y_t, l_{t+1}), \ldots, \text{compose}(l_t, T Y_t, l_{t+1}),
HY = l_{t+1}, \text{compose}(A, HY, NA),
\mathcal{r} \mathcal{d} d_1(T X s, Y, NA)\]

By using applicability condition (3):

clause 10: \[ \mathcal{d} d_1(X, Y, A) \leftarrow X s = [X[T X s],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_t),
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_t),
solve(T X_1, e), \ldots, \text{solve}(T X_t, e),
l_0 = e,
\text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{t-2}, e, l_{t-1}),
\text{process}(H X, H Y), \text{compose}(l_{t-1}, H Y, l_t),
\text{compose}(l_t, e, l_{t+1}), \ldots, \text{compose}(l_t, e, l_{t+1}),
HY = l_{t+1}, \text{compose}(A, HY, NA),
\mathcal{r} \mathcal{d} d_1(T X s, Y, NA)\]

By deleting one of the \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t) atoms in clause 10:
clause 11: \( r_{\mathcal{D}_1}(X_s, Y, A) = \)
\[ X_s = \left[ X \cup T X_s \right], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \]
\[ \text{solve}(TX_1, e), \ldots, \text{solve}(TX_t, e), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(l_{p-1}, H HY, l_p), \]
\[ \text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}), \]
\[ HY = l_{t+1}, \text{compose}(A, HY, NA), \]
\( r_{\mathcal{D}_1}(TX_s, Y, NA) \)

By using applicability condition (2):

clause 12: \( r_{\mathcal{D}_1}(X_s, Y, A) = \)
\[ X_s = \left[ X \cup T X_s \right], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \]
\[ \text{solve}(TX_1, e), \ldots, \text{solve}(TX_t, e), \]
\[ l_0 = e, \]
\[ l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \]
\[ \text{process}(H X, H HY), \text{compose}(l_{p-1}, H HY, l_p), \]
\[ l_{p+1} = l_p, \ldots, l_{t+1} = l_t, \]
\[ HY = l_{t+1}, \text{compose}(A, HY, NA), \]
\( r_{\mathcal{D}_1}(TX_s, Y, NA) \)

By simplification:

clause 13: \( r_{\mathcal{D}_1}(X_s, Y, A) = \)
\[ X_s = \left[ X \cup T X_s \right], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \]
\[ \text{process}(H X, H HY), \text{compose}(A, HY, NA), \]
\( r_{\mathcal{D}_1}(TX_s, Y, NA) \)

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

clause 14: \( r_{\mathcal{D}_1}(X_s, Y, A) = \)
\[ X_s = \left[ X \cup T X_s \right], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_{p-1}, TY_{p-1}), \]
\[ r(TX_p, TY_p), \ldots, r(TX_t, TY_t) \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(l_{p-1}, H HY, l_p), \]
\[ \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]
\[ HY = l_{t+1}, \text{compose}(A, HY, NA), \]
\( r_{\mathcal{D}_1}(TX_s, Y, NA) \)

By deleting one of the \( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}) \) atoms in clause 14:
clause 15: \[ r_{\mathcal{D}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ non\text{Minimal}(X), \]
\[ \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (non\text{Minimal}(T X_p); \ldots; non\text{Minimal}(T X_t)), \]
\[ \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \]
\[ r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t) \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-2}, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \]
\[ HY = I_{p+1}, \text{compose}(A, H Y, NA), \]
\[ r_{\mathcal{D}_1}(T X_s, Y, NA) \]

By rewriting clause 15 using applicability condition (1):

clause 16: \[ r_{\mathcal{D}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ non\text{Minimal}(X), \]
\[ \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (non\text{Minimal}(T X_p); \ldots; non\text{Minimal}(T X_t)), \]
\[ \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \]
\[ r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t) \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-2}, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \]
\[ HY = I_p, \text{compose}(A, H Y, NA), \]
\[ \text{compose}(TY_p, T Y_{p+1}, I_{p+1}), \]
\[ \text{compose}(I_{p+1}, T Y_{p+2}, I_{p+2}), \ldots, \text{compose}(I_{t-1}, T Y_t, I_t), \]
\[ \text{compose}(NA, I_t, NA), \]
\[ r_{\mathcal{D}_1}(T X_s, Y, NA) \]

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

clause 17: \[ r_{\mathcal{D}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ non\text{Minimal}(X), \]
\[ \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (non\text{Minimal}(T X_p); \ldots; non\text{Minimal}(T X_t)), \]
\[ \text{solve}(T X_1, T Y_1), \ldots, \text{solve}(T X_{p-1}, T Y_{p-1}), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-2}, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \]
\[ HY = I_p, \text{compose}(A, H Y, NA), \]
\[ r_{\mathcal{D}_1}([T X_p, \ldots, T X_t[T X_s]], Y, NA) \]

By using applicability condition (3):

clause 18: \[ r_{\mathcal{D}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ non\text{Minimal}(X), \]
\[ \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (non\text{Minimal}(T X_p); \ldots; non\text{Minimal}(T X_t)), \]
\[ \text{solve}(T X_1, e), \ldots, \text{solve}(T X_{p-1}, e), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \]
\[ \text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \]
\[ HY = I_p, \text{compose}(A, H Y, NA), \]
\[ r_{\mathcal{D}_1}([T X_p, \ldots, T X_t[T X_s]], Y, NA) \]

By using applicability condition (2):
clause 19:  \( r \mathcal{J}d_1(X_s, Y, A) = \\
X_s = [X[T Xs]], \\
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{\eta-1}), \\
(nonMinimal(TX_p); \ldots, nonMinimal(TX_t)), \\
solve(TX_1, \epsilon), \ldots, solve(TX_{\eta-1}, \epsilon), \\
l_0 = \epsilon, \\
l_1 = l_0, \ldots, l_{\eta-1} = l_{\eta-2}, \\
process(HX, HY), compose(I_{\eta-1}, HY, I_p), HY = I_p, \\
\mathcal{J}d_1([TX_p, \ldots, TX_t[T Xs], Y, NA)

By simplification:

clause 20:  \( r \mathcal{J}d_1(X_s, Y, A) = \\
X_s = [X[T Xs]], \\
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{\eta-1}), \\
(nonMinimal(TX_p); \ldots, nonMinimal(TX_t)), \\
process(HX, HY), compose(A, HY, NA), \\
\mathcal{J}d_1([TX_p, \ldots, TX_t[T Xs], Y, NA)

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

clause 21:  \( r \mathcal{J}d_1(X_s, Y, A) = \\
X_s = [X[T Xs]], \\
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{\eta-1}), \\
minimal(U_1), \ldots, minimal(U_{\eta-1}), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_0 = \epsilon, \\
\mathcal{J}d_1([TX_p, \ldots, TX_t[T Xs], Y, NA)

By using applicability condition (3):

clause 22:  \( r \mathcal{J}d_1(X_s, Y, A) = \\
X_s = [X[T Xs]], \\
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
minimal(TX_1), \ldots, minimal(TX_{\eta-1}), \\
minimal(U_1), \ldots, minimal(U_{\eta-1}), \\
r(U_1, \epsilon), \ldots, r(U_{\eta-1}, \epsilon), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_0 = \epsilon, \\
\mathcal{J}d_1([TX_p, \ldots, TX_t[T Xs], Y, NA)

By using applicability condition (2):
\[ r \mathcal{J} d_1(X,Y,A) = \]
\[ X_1 = X \]
\[ nonMinimal(X), \ decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}), \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ \lambda_0 = e, \]
\[ compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ compose(I_{p-1}, e, K_1), compose(K_1, e, K_2), \ldots, compose(K_{p-2}, e, K_{p-1}), \]
\[ process(HX, HHY), compose(K_{p-1}, HHY, I_0), \]
\[ compose(I_{p-2}, TY_{p-1}, I_{p-1}), compose(I_{p-1}, TY_{p-1}, I_{p+1}), \]
\[ HY = \lambda_{p+1}, compose(NA, HHY, NNA), \]
\[ r \mathcal{J} d_1(TX_s, Y, NNA) \]

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

\[ r \mathcal{J} d_1(X,Y,A) = \]
\[ X_1 = X \]
\[ nonMinimal(X), \ decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}), \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ compose(A, K_{p-2}, NNA), \]
\[ \lambda_0 = e, \]
\[ compose(I_0, e, I_1), \ldots, compose(I_{p-2}, e, I_{p-1}), \]
\[ process(HX, HHY), compose(I_{p-1}, HHY, I_0), \]
\[ compose(I_{p-2}, TY_{p-1}, I_{p-1}), compose(I_{p-1}, TY_{p-1}, I_{p+1}), \]
\[ HY = \lambda_{p+1}, compose(NA, HHY, NNA), \]
\[ r \mathcal{J} d_1(TX_s, Y, NNA) \]

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

\[ r \mathcal{J} d_1(X,Y,A) = \]
\[ X_1 = X \]
\[ nonMinimal(X), \ decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \]
\[ minimal(TX_p), \ldots, minimal(TX_t), \]
\[ minimal(U_1), \ldots, minimal(U_{p-1}), \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ compose(A, K_{p-2}, NNA), \]
\[ \lambda_0 = e, \]
\[ compose(I_0, YU_1, I_1), \ldots, compose(I_{p-2}, YU_{p-1}, I_{p-1}), \]
\[ process(HX, HHY), compose(I_{p-1}, HHY, I_0), \]
\[ compose(I_{p-2}, TY_{p-1}, I_{p-1}), compose(I_{p-1}, TY_{p-1}, I_{p+1}), \]
\[ HY = \lambda_{p+1}, compose(NA, HHY, NNA), \]
\[ r \mathcal{J} d_1(TX_s, Y, NNA) \]

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

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

always holds (because \( N \) is existentially quantified):
clause 26: \( r_{d1}(X_s, Y, A) = \\
\text{Xs} = [X[T Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
r(U_1,YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(A, K_{p-2}, NA), \\
I_0 = \epsilon, \\
\text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = l_{t+1}, \text{compose}(NA, HY, NNA), \\
r_{d1}(TXs, Y, NNA) \\
\) 

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

clause 27: \( r_{d1}(X_s, Y, A) = \\
\text{Xs} = [X[T Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
r(U_1,YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(A, K_{p-2}, NA), \\
I_0 = \epsilon, \\
\text{compose}(I_0, YU_1, I_1), \ldots, \text{compose}(I_{p-2}, YU_{p-1}, I_{p-1}), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = l_{t+1}, \text{compose}(NA, HY, NNA), \\
r_{d1}(TXs, Y, NNA) \\
\) 

By folding clause 27 using DCLR:

clause 28: \( r_{d1}(X_s, Y, A) = \\
\text{Xs} = [X[T Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{t-1})), \\
\text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \\
decompose(N, HX, U_1, \ldots, U_{p-1}, TX_p, \ldots, TX_t), \\
r(TX_1, TY_1), \ldots, r(TX_p, TY_p), r(N, HY), \\
\text{compose}(TY_1, TY_3, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(A, K_{p-2}, NA), \text{compose}(NA, HY, NNA), \\
r_{d1}(TXs, Y, NNA) \\
\) 

By folding clause 28 using clauses 1 and 2:
clause 29:  
\[ r \mathcal{J} d_1(X, Y, A) = \]
\[ X = [X[T X s], \]
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}),
\]
\[ compose(TY_1, TY_2, K_1), compose(K_1, TY_3, K_2), \ldots, compose(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ compose(A, K_{p-2}, NA), \]
\[ r \mathcal{J} d_1([TX_1, \ldots, TX_{p-1}, N[T X s], Y, A]) \]

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

clause 30:  
\[ r \mathcal{J} d_1(X, Y, A) = \]
\[ X = [X[T X s], \]
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 \mathcal{J} d_1([TX_1, \ldots, TX_{p-1}, N[T X s], Y, A]) \]

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 8:

clause 31:  
\[ r \mathcal{J} d_1(X, Y, A) = \]
\[ X = [X[T X s], \]
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_{p-1}),
r(X, TY_1), \ldots, r(TX_t, TY_{p-1}),
I_0 = \epsilon, \]
\[ compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ process(HX, HY), compose(I_{p-1}, HY, I_p), \]
\[ compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_{p-1}, TY_t, I_{t+1}), \]
\[ HY = I_{t+1}, compose(A, HY, NA), \]
\[ r \mathcal{J} d_1(TX s, Y, NA) \]

By using applicability condition (3):

clause 32:  
\[ r \mathcal{J} d_1(X, Y, A) = \]
\[ X = [X[T X s], \]
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_{p-1}),
r(U_1, \epsilon), \ldots, r(U_t, \epsilon),
r(TX_1, TY_1), \ldots, r(TX_t, TY_{p-1}),
I_0 = \epsilon, \]
\[ compose(I_0, TY_1, I_1), \ldots, compose(I_{p-2}, TY_{p-1}, I_{p-1}), \]
\[ process(HX, HY), compose(I_{p-1}, HY, I_p), \]
\[ compose(I_p, TY_p, I_{p+1}), \ldots, compose(I_{p-1}, TY_t, I_{t+1}), \]
\[ HY = I_{t+1}, compose(A, HY, NA), \]
\[ r \mathcal{J} d_1(TX s, Y, NA) \]

By using applicability condition (2):
\[ \text{clause 33:} \quad \forall X, Y, A \quad r \mathcal{J}d_1(X, Y, A) = \\
X = [X \| Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, e), \ldots, r(U_t, e), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
l_0 = e, \\
\text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \\
\text{compose}(l_{p-1}, e, l_1), \text{compose}(K_1, e, K_3), \ldots, \text{compose}(K_{p-2}, e, K_{p-1}), \\
\text{process}(HX, H HY), \text{compose}(K_{p-1}, H HY, K_p), \\
\text{compose}(K_{p-1}, e, K_{p+1}), \ldots, \text{compose}(K_{t-1}, e, I_t), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}, \text{compose}(A, HY, NA), \\
r \mathcal{J}d_1(TXs, Y, NA) \]

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

\[ \text{clause 34:} \quad r \mathcal{J}d_1(X, Y, A) = \\
X = [X \| Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, e), \ldots, r(U_t, e), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
l_0 = e, \\
\text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \\
\text{process}(HX, H HY), \text{compose}(l_{p-1}, H HY, l_p), \\
\text{compose}(l_p, e, I_{p+1}), \ldots, \text{compose}(l_t, e, I_{t+1}), \\
HY = I_{t+1}, \text{compose}(K_0, TY_{p+1}, K_p), \ldots, \text{compose}(K_{t-2}, TY_{t-2}, K_{t-1}), \\
\text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NA, NA), \\
r \mathcal{J}d_1(TXs, Y, NA) \]

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

\[ \text{clause 35:} \quad r \mathcal{J}d_1(X, Y, A) = \\
X = [X \| Xs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1) \ldots \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p) \ldots \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
l_0 = e, \\
\text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{p-2}, YU_{p-1}, l_{p-1}), \\
\text{process}(HX, H HY), \text{compose}(l_{p-1}, H HY, l_p), \\
\text{compose}(l_p, YU_p, l_{p+1}), \ldots, \text{compose}(l_t, YU_t, l_{t+1}), \\
HY = I_{t+1}, \text{compose}(K_0, TY_{p+1}, K_p), \ldots, \text{compose}(K_{t-2}, TY_{t-2}, K_{t-1}), \\
\text{compose}(A, K_{p-2}, NA), \text{compose}(NA, NA, NA), \\
r \mathcal{J}d_1(TXs, Y, NA) \]

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: \[ r_d(Xs, Y, A) =
\begin{align*}
Xs & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
l_0 = e, \\
\text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{p-2}, YU_{p-1}, l_{p-1}), \\
\text{process}(HX, HNY), \text{compose}(l_{p-1}, HNY, l_p), \\
\text{compose}(l_p, YU_p, l_{p+1}), \ldots, \text{compose}(l_t, YU_t, l_{t+1}), \\
NNY = l_{t+1}, \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(A, K_{t-2}, NA), \text{compose}(NA, NNY, NA_1), \\
\text{compose}(NA_1, K_{t-1}, N A_2), r_d(TXs, Y, NA_2)
\end{align*}
\]

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

clause 37: \[ r_d(Xs, Y, A) =
\begin{align*}
Xs & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t), \\
decompose(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
l_0 = e, \\
\text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{p-2}, YU_{p-1}, l_{p-1}), \\
\text{process}(HX, HNY), \text{compose}(l_{p-1}, HNY, l_p), \\
\text{compose}(l_p, YU_p, l_{p+1}), \ldots, \text{compose}(l_t, YU_t, l_{t+1}), \\
NNY = l_{t+1}, \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(A, K_{t-2}, NA), \text{compose}(NA, NNY, NA_1), \\
\text{compose}(NA_1, K_{t-1}, N A_2), r_d(TXs, Y, NA_2)
\end{align*}
\]

By folding clause 37 using DCBR:

clause 38: \[ r_d(Xs, Y, A) =
\begin{align*}
Xs & = [X[TXs], \\
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \\
\text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
decompose(N, HX, U_1, \ldots, U_t), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, NNY), \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
\text{compose}(A, K_{t-2}, NA), \text{compose}(NA, NNY, NA_1), \\
\text{compose}(NA_1, K_{t-1}, N A_2), r_d(TXs, Y, NA_2)
\end{align*}
\]

By \( t = p + 1 \) times folding clause 38 using clauses 1 and 2:
The procedure
By using applicability condition (2):
We obtain $S_r$, so $P_r$ is steadfast wrt $S_r$ in $\{S_r, r\}$. Therefore, $TDGLR$ is also steadfast wrt $S_r$ in $S$.  

52
Theorem 8 The generalization schema $TDG_2$, which is given below, is correct.

$TDG_2 : \{ DCLR, TDGRL, A_{td}, O_{td12}, O_{td21} \}$ where

\[ A_{td} : \]
(1) $compose$ is associative
(2) $compose$ has $\epsilon$ as the left and right identity element
(3) $\forall X : I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, \epsilon)$
(4) $\forall X : I_r(X) \Rightarrow [\neg \text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)]$

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

\[ O_{td21} : \]
\[
\text{process}(HX, HY), \text{compose}(I_{p-1}, HY, I_p)
\]
results in the introduction of a non-recursive relation

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

Logic Program Template 7

\[
\begin{align*}
r(X, Y) & = \\
r_{td2}(X, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
r_{td2}(Xs, Y, A) & = \\
\end{align*}
\]
decompose(N, HX, U₁,..., Uₚ₋₁, T_Xp,..., T_Xt),
\( r \cdot \mathcal{d}_2([T_X₁,..., T_Xp₋₁, N[T_Xs], Y, A]) \)
\( r \cdot \mathcal{d}_2(Xs, Y, A) = \)
\( Xs = [T_Xs], \)
nonMinimal(X),
decompose(X, HX, T_X₁,..., T_Xt),
(\nonMinimal(T_X₁);...;\nonMinimal(T_Xp₋₁)),
(\nonMinimal(T_Xp);...;\nonMinimal(T_Xt)),
mimimal(U₁),...;minimal(Uₚ),
decompose(N, HX, U₁,..., Uₚ),
\( r \cdot \mathcal{d}_2([T_X₁,..., T_Xp₋₁, N[T_Xs], Y, A]) \)

and the specification \( S_r \) of relation \( r \) is:
\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \mathcal{O}_r(X, Y)] \]
and the specification of \( r \mathcal{d}_2 \), namely \( S_{r \mathcal{d}_2} \), is:
\[ \forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r \mathcal{d}_2(Xs, Y, A) \Rightarrow (Xs = [] \land Y = A)] \]
\( \lor \forall Xs = [X₁, X₂,..., X_q] \land \mathcal{O}_r(X_i, Y_i) \land I \lor \land \mathcal{O}_r(I_{-1}, Y_i, I_i) \land \mathcal{O}_r(I_q, A, I_{q+1}) \land Y = I_{q+1}] \)

**Proof 8** To prove the correctness of the generalization schema TDG₁, by Definition 10, we have to prove that templates DCLR and TDGRL are equivalent wrt \( S_r \), under the applicability conditions \( A_{d₂} \). By Definition 5, the templates DCLR and TDGRL are equivalent wrt \( S_r \), under the applicability conditions \( A_{d₂} \) iff DCLR is equivalent to TDGRL wrt the specification \( S_r \), provided that the conditions in \( A_{d₂} \) 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 minimal, nonMinimal, solve, decompose, process, compose, which are all the undefined relation names appearing in DCLR.

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

Note that the sets \( \{S₁,..., S_m\} \) and \( \{S'₁,..., S'_m\} \) in Definition 4 are equal to \( S \) when \( Q \) is obtained by simultaneous tupling-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 \( P_{r \cdot \mathcal{d}_2} \) is steadfast wrt \( S_{r \cdot \mathcal{d}_2} \) in \( S \), where \( P_{r \cdot \mathcal{d}_2} \) is the procedure for \( r \mathcal{d}_2 \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r \cdot \mathcal{d}_2}\} \), where \( P_r \) is the procedure for \( r \).

To prove that \( P_{r \cdot \mathcal{d}_2} \) is steadfast wrt \( S_{r \cdot \mathcal{d}_2} \) in \( S \), we do a constructive forward proof that we begin with \( S_{r \cdot \mathcal{d}_2} \) and from which we try to obtain \( P_{r \cdot \mathcal{d}_2} \).

If we separate the cases of \( q \geq 1 \) by \( q = 1 \lor q \geq 2 \), then \( S_{r \cdot \mathcal{d}_2} \) becomes:
\( \forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r \mathcal{d}_2(Xs, Y, A) \Rightarrow (Xs = [] \land Y = A)] \)
\( \lor \forall Xs = [X₁, X₂,..., X_q] \land \mathcal{O}_r(X_i, Y_i) \land I \lor \land \mathcal{O}_r(I_{-1}, Y_i, I_i) \land \mathcal{O}_r(I_q, A, I_{q+1}) \land Y = I_{q+1}] \)

where \( q \geq 2 \).

By using applicability conditions (1) and (2):
\( \forall Xs : \text{list of } \mathcal{X}, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in Xs \Rightarrow I_r(X)) \Rightarrow [r \mathcal{d}_2(Xs, Y, A) \Rightarrow (Xs = [] \land Y = A)] \)
\( \lor \forall Xs = [X₁[T_Xs] \land TXs = [] \land \mathcal{O}_r(X₁Y₁) \land Y₁ = I₁ \land TY = A \land \mathcal{O}_r(TY, A, N, A) \land \mathcal{O}_r(I₁, N, A, Y)] \)
\( \lor \forall Xs = [X₁[T_Xs] \land TXs = [X₂,..., X_q] \land \mathcal{O}_r(X₁Y₁) \land Y₁ = I₁ \land Y₂ = I₂ \land \mathcal{O}_r(I_{-1}, Y_i, I_i) \land TY = I_q \land \mathcal{O}_r(TY, A, N, A) \land \mathcal{O}_r(I₁, N, A, Y)] \)
where $q \geq 2$.

By folding using $S_r \cup d_2$, and renaming:

$$\forall X s : \text{list of } \mathcal{X}, Y Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. \ X \in X s \Rightarrow I_r(X)) \Rightarrow [r_{d_2}(X s, Y, A) \Leftrightarrow$$

$$(X s = []) \lor Y = A)$$

$$\forall (X s = [X | TXs] \land \mathcal{O}_r(X, HY) \land r_{d_2}(TXs, NA, A) \land \mathcal{O}_r(HY, NA, Y))$$

By taking the ‘decompletion’:

clause 1: $r_{d_2}(X s, Y, A) \Leftarrow X s = [], Y = A$

clause 2: $r_{d_2}(X s, Y, A) \Leftarrow$

$$X s = [X | TXs], r(X, HY), r_{d_2}(TXs, NA, A), \text{compose}(HY, NA, Y)$$

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

clause 3: $r_{d_2}(X s, Y, A) \Leftarrow$

$$X s = [X | TXs], \text{minimal}(X), \text{solve}(X, HY), r_{d_2}(TXs, NA, A), \text{compose}(HY, NA, Y)$$

clause 4: $r_{d_2}(X s, Y, A) \Leftarrow$

$$X s = [X | TXs], \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), r(TX_1, TY_1), \ldots, r(TX_t, TY_t), l_0 = \epsilon, \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-2}, l_{p-1}), \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), HY = l_{t+1}, r_{d_2}(TXs, NA, A), \text{compose}(HY, NA, Y)$$

By introducing

$$(\text{minimal}(TX_1) \land \ldots \land \text{minimal}(TX_t)) \lor$$

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

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

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

in clause 4, using applicability condition (4):

clause 5: $r_{d_2}(X s, Y, A) \Leftarrow$

$$X s = [X | TXs], \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), r(TX_1, TY_1), \ldots, r(TX_t, TY_t), l_0 = \epsilon, \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-2}, l_{p-1}), \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \text{compose}(l_p, TY_p, l_{p+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), HY = l_{t+1}, r_{d_2}(TXs, NA, A), \text{compose}(HY, NA, Y)$$
clause 6: \( r\Delta d_2(X_s,Y,A) \leftarrow \)
\[ X_s = \{X[X]X_s], \nonMinimal(X), \decompose(X,HX,TX_1,\ldots,TX_t), \minimal(TX_1), \ldots, \minimal(TX_{r-1}), \nonMinimal(TX_r); \ldots; \nonMinimal(TX_t)\}, \]
\[ r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \]
\[ I_0 = \epsilon, \]
\[ \compose(I_0,TY_1,I_1),\ldots,\compose(I_{r-2},TY_{r-1},I_{r-1}), \]
\[ \process(HX,HHY),\compose(I_{r-1},HHY,I_p), \]
\[ \compose(I_p,TY_p,I_{p+1}),\ldots,\compose(I_t,TY_t,I_{t+1}), \]
\[ HY = I_{t+1}, \]
\[ r\Delta d_2(TX_s,NA,A),\compose(HY,NA,Y) \]

clause 7: \( r\Delta d_2(X_s,Y,A) \leftarrow \)
\[ X_s = \{X[X]X_s], \nonMinimal(X), \decompose(X,HX,TX_1,\ldots,TX_t), \nonMinimal(TX_1); \ldots; \nonMinimal(TX_t)\}, \]
\[ r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \]
\[ I_0 = \epsilon, \]
\[ \compose(I_0,TY_1,I_1),\ldots,\compose(I_{r-2},TY_{r-1},I_{r-1}), \]
\[ \process(HX,HHY),\compose(I_{r-1},HHY,I_p), \]
\[ \compose(I_p,TY_p,I_{p+1}),\ldots,\compose(I_t,TY_t,I_{t+1}), \]
\[ HY = I_{t+1}, \]
\[ r\Delta d_2(TX_s,NA,A),\compose(HY,NA,Y) \]

clause 8: \( r\Delta d_2(X_s,Y,A) \leftarrow \)
\[ X_s = \{X[X]X_s], \nonMinimal(X), \decompose(X,HX,TX_1,\ldots,TX_t), \nonMinimal(TX_1); \ldots; \nonMinimal(TX_t)\}, \]
\[ r(TX_1,TY_1),\ldots,r(TX_t,TY_t), \]
\[ I_0 = \epsilon, \]
\[ \compose(I_0,TY_1,I_1),\ldots,\compose(I_{r-2},TY_{r-1},I_{r-1}), \]
\[ \process(HX,HHY),\compose(I_{r-1},HHY,I_p), \]
\[ \compose(I_p,TY_p,I_{p+1}),\ldots,\compose(I_t,TY_t,I_{t+1}), \]
\[ HY = I_{t+1}, \]
\[ r\Delta d_2(TX_s,NA,A),\compose(HY,NA,Y) \]

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: \( r\Delta d_2(X_s,Y,A) \leftarrow \)
\[ X_s = \{X[X]X_s], \nonMinimal(X), \decompose(X,HX,TX_1,\ldots,TX_t), \minimal(TX_1), \ldots, \minimal(TX_t), \]
\[ \solve(TX_1,TY_1),\ldots,\solve(TX_t,TY_t), \]
\[ I_0 = \epsilon, \]
\[ \compose(I_0,TY_1,I_1),\ldots,\compose(I_{r-2},TY_{r-1},I_{r-1}), \]
\[ \process(HX,HHY),\compose(I_{r-1},HHY,I_p), \]
\[ \compose(I_p,TY_p,I_{p+1}),\ldots,\compose(I_t,TY_t,I_{t+1}), \]
\[ HY = I_{t+1}, \]
\[ r\Delta d_2(TX_s,NA,A),\compose(HY,NA,Y) \]

By using applicability condition (3):
\[ \text{clause 10: } r_{d_3}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t), \]
\[ \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \]
\[ \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), \]
\[ \text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}), \]
\[ H Y = l_{t+1}, \]
\[ r_{d_3}(TX_s, NA, A), \text{compose}(HY, NA, Y) \]

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

\[ \text{clause 11: } r_{d_3}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t), \]
\[ \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \]
\[ \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), \]
\[ \text{compose}(l_p, e, l_{p+1}), \ldots, \text{compose}(l_t, e, l_{t+1}), \]
\[ H Y = l_{t+1}, \]
\[ r_{d_3}(TX_s, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (2):

\[ \text{clause 12: } r_{d_3}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t), \]
\[ \text{solve}(T X_1, e), \ldots, \text{solve}(T X_t, e), \]
\[ l_0 = e, \]
\[ l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \]
\[ \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), \]
\[ l_{p+1} = l_p, \ldots, l_{t+1} = l_t, \]
\[ H Y = l_{t+1}, \]
\[ r_{d_3}(TX_s, NA, A), \text{compose}(HY, NA, Y) \]

By simplification:

\[ \text{clause 13: } r_{d_3}(X_s, Y, A) = \]
\[ X_s = [X[T X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_t), \]
\[ r_{d_3}(TX_s, NA, A), \]
\[ \text{process}(H X, H H Y), \text{compose}(HY, NA, Y) \]

By \(p-1\) times unfolding clause 6 wrt \(r(T X_1, TY_1), \ldots, r(T X_{p-1}, TY_{p-1})\) using DCLR, and simplifying using condition (4):
clause 14: \( r_{d_2}(X_s, Y, A) = \)

\[ X_s = [X[TX_s], \]

\[
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \\
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \\
(\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\text{solve}(T X_1, TY_1), \ldots, \text{solve}(T X_{p-1}, TY_{p-1}), \\
r(T X_p, TY_p), \ldots, r(T X_t, TY_t), \\
I_0 = \epsilon, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}, \\
r_{d_2}(T X_s, NA, A), \text{compose}(HY, NA, Y) \]

By deleting one of the minimal\((T X_1), \ldots, \text{minimal}(T X_{p-1})\) atoms in clause 14:

clause 15: \( r_{d_2}(X_s, Y, A) = \)

\[ X_s = [X[TX_s], \]

\[
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \\
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \\
(\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\text{solve}(T X_1, TY_1), \ldots, \text{solve}(T X_{p-1}, TY_{p-1}), \\
r(T X_p, TY_p), \ldots, r(T X_t, TY_t), \\
I_0 = \epsilon, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \\
\text{compose}(I_p, TY_p, I_{p+1}), \ldots, \text{compose}(I_t, TY_t, I_{t+1}), \\
HY = I_{t+1}, \\
r_{d_2}(T X_s, NA, A), \text{compose}(HY, NA, Y) \]

By rewriting clause 15 using applicability condition (1):

clause 16: \( r_{d_2}(X_s, Y, A) = \)

\[ X_s = [X[TX_s], \]

\[
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \\
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \\
(\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\text{solve}(T X_1, TY_1), \ldots, \text{solve}(T X_{p-1}, TY_{p-1}), \\
r(T X_p, TY_p), \ldots, r(T X_t, TY_t), \\
I_0 = \epsilon, \\
\text{compose}(I_0, TY_1, I_1), \ldots, \text{compose}(I_{p-2}, TY_{p-1}, I_{p-1}), \\
\text{process}(H X, H HY), \text{compose}(I_{p-1}, H HY, I_p), \\
HY = I_p, \text{compose}(HY, NA, Y), \\
\text{compose}(TY_p, I_{p+1}, NA), \\
\text{compose}(TY_{p+1}, I_{p+2}, I_{p+1}), \ldots, \text{compose}(TY_{t-1}, I_t, I_{t-1}), \\
\text{compose}(TY_t, NA, I_t), \\
r_{d_2}(T X_s, NA, A) \]

By \( t - p \) times folding clause 16 using clauses 1 and 2:
clause 17: \( r.\mathcal{J}_d(Xs, Y, A) = \)
\( Xs = [X [T Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, \epsilon), \ldots, \text{solve}(TX_{p-1}, \epsilon), \)
\( l_0 = \epsilon, \)
\( \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HY = l_p, \)
\( r.\mathcal{J}_d([TX_p, \ldots, TX_t[X Xs]], NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):

clause 18: \( r.\mathcal{J}_d(Xs, Y, A) = \)
\( Xs = [X [T Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, \epsilon), \ldots, \text{solve}(TX_{p-1}, \epsilon), \)
\( l_0 = \epsilon, \)
\( \text{compose}(l_0, \epsilon, l_1), \ldots, \text{compose}(l_{p-2}, \epsilon, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HY = l_p, \)
\( r.\mathcal{J}_d([TX_p, \ldots, TX_t[X Xs]], NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (2):

clause 19: \( r.\mathcal{J}_d(Xs, Y, A) = \)
\( Xs = [X [T Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, \epsilon), \ldots, \text{solve}(TX_{p-1}, \epsilon), \)
\( l_0 = \epsilon, \)
\( l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HY = l_p, \)
\( r.\mathcal{J}_d([TX_p, \ldots, TX_t[X Xs]], NA, A), \text{compose}(HY, NA, Y) \)

By simplification:

clause 20: \( r.\mathcal{J}_d(Xs, Y, A) = \)
\( Xs = [X [T Xs], \)
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( r.\mathcal{J}_d([TX_p, \ldots, TX_t[X Xs]], NA, A), \)
\( \text{process}(HX, HHY), \text{compose}(HY, NA, Y) \)

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:
\textbf{clause 21:} \quad r_{d2}(X, Y, A) = \\
\text{X} = X \cdot T X, \\
\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(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \\
l_0 = \epsilon, \\
\text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-1}, l_{p-1}), \\
\text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), \\
\text{compose}(l_p, T Y_p, l_{p+1}), \ldots, \text{compose}(l_{t}, T Y_t, l_{t+1}), \\
H Y = l_{t+1}. \\
\text{By using applicability condition (3):} \\
\text{clause 22:} \quad r_{d2}(X, Y, A) = \\
\text{X} = X \cdot T X, \\
\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, \epsilon), \ldots, r(U_{p-1}, \epsilon), \\
r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \\
l_0 = \epsilon, \\
\text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-1}, l_{p-1}), \\
\text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), \\
\text{compose}(l_p, T Y_p, l_{p+1}), \ldots, \text{compose}(l_{t}, T Y_t, l_{t+1}), \\
H Y = l_{t+1}. \\
\text{By using applicability condition (2):} \\
\text{clause 23:} \quad r_{d2}(X, Y, A) = \\
\text{X} = X \cdot T X, \\
\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, \epsilon), \ldots, r(U_{p-1}, \epsilon), \\
r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \\
l_0 = \epsilon, \\
\text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-1}, l_{p-1}), \\
\text{compose}(l_{p-1}, \epsilon, K_1), \text{compose}(K_1, \epsilon, K_2), \ldots, \text{compose}(K_{p-2}, \epsilon, K_{p-1}), \\
\text{process}(H X, H H Y), \text{compose}(K_{p-1}, H H Y, l_p), \\
\text{compose}(l_p, T Y_p, l_{p+1}), \ldots, \text{compose}(l_{t}, T Y_t, l_{t+1}), \\
H Y = l_{t+1}. \\
\text{By using applicability conditions (1) and (2):}
clause 24: \( r \Delta_{d_2}(Xs, Y, A) = \)
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{r-1})), \]
\[ \text{minimal}(TX_r), \ldots, \text{minimal}(TX_{r-1}), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{r-1}), \]
\[ r(U_1, e), \ldots, r(U_{r-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_1), \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{r-3}, TY_{r-1}, K_{r-2}), \]
\[ \text{compose}(K_{r-2}, NA, Y), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{r-2}, e, l_{r-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(l_{r-1}, HHY, l_r), \]
\[ \text{compose}(l_r, TY_2, l_{r+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]
\[ H \equiv l_{t+1}, \text{compose}(HY, NNA, NNA), \]
\[ r \Delta_{d_5}(TXs, NNA, A), \]

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

clause 25: \( r \Delta_{d_2}(Xs, Y, A) = \)
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{r-1})), \]
\[ \text{minimal}(TX_r), \ldots, \text{minimal}(TX_{r-1}), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{r-1}), \]
\[ r(U_1, YU_1), \ldots, r(U_{r-1}, YU_{r-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_1), \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{r-3}, TY_{r-1}, K_{r-2}), \]
\[ \text{compose}(K_{r-2}, NA, Y), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{r-2}, YU_{r-1}, l_{r-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(l_{r-1}, HHY, l_r), \]
\[ \text{compose}(l_r, TY_2, l_{r+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]
\[ H \equiv l_{t+1}, \text{compose}(HY, NNA, NNA), \]
\[ r \Delta_{d_5}(TXs, NNA, A), \]

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

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

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

clause 26: \( r \Delta_{d_2}(Xs, Y, A) = \)
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{r-1})), \]
\[ \text{minimal}(TX_r), \ldots, \text{minimal}(TX_{r-1}), \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{r-1}, TX_p, \ldots, TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{r-1}), \]
\[ r(U_1, YU_1), \ldots, r(U_{r-1}, YU_{r-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_1), \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{r-3}, TY_{r-1}, K_{r-2}), \]
\[ \text{compose}(K_{r-2}, NA, Y), \]
\[ l_0 = e, \]
\[ \text{compose}(l_0, YU_1, l_1), \ldots, \text{compose}(l_{r-2}, YU_{r-1}, l_{r-1}), \]
\[ \text{process}(HX, HHY), \text{compose}(l_{r-1}, HHY, l_r), \]
\[ \text{compose}(l_r, TY_2, l_{r+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), \]
\[ H \equiv l_{t+1}, \text{compose}(HY, NNA, NNA), \]
\[ r \Delta_{d_5}(TXs, NNA, A), \]

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_{r-1}, TX_p, \ldots, TX_t) \):
**clause 27:** \( r \cdot d_2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})),
\text{minimal}(T X_p), \ldots, \text{minimal}(T X_t),
\text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t),
\text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t),
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}),
\text{r}(U_1, Y U_1, \ldots, \text{r}(U_{p-1}, Y U_{p-1}),
\text{r}(T X_1, T Y_1), \ldots, \text{r}(T X_t, T Y_t),
\text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_p-1, K_{p-2}),
\text{compose}(K_{p-2}, N A, Y),
\text{I}_0 = \epsilon,
\text{compose}(\text{I}_0, Y U_1, \text{I}_1), \ldots, \text{compose}(\text{I}_{p-2}, Y U_{p-1}, \text{I}_{p-1}),
\text{process}(H X, H H Y), \text{compose}(\text{I}_{p-1}, H H Y, \text{I}_p),
\text{compose}(I_p, T Y_p, I_{p+1}), \ldots, \text{compose}(I_t, T Y_t, I_{t+1}),
\text{HY} = I_{t+1}, \text{compose}(\text{HY}, N N A, N A),
 r \cdot d_2(T Xs, N N A, A)
\]

By folding clause 27 using DCLR:

**clause 28:** \( r \cdot d_2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})),
\text{minimal}(T X_p), \ldots, \text{minimal}(T X_t),
\text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t),
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}),
\text{r}(T X_1, T Y_1), \ldots, \text{r}(T X_p-1, T Y_p-1), \text{r}(N, \text{HY}),
\text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_p-1, K_{p-2}),
\text{compose}(K_{p-2}, N A, Y), \text{compose}(\text{HY}, N N A, N A),
 r \cdot d_2(T Xs, N N A, A)
\]

By folding clause 28 using clauses 1 and 2:

**clause 29:** \( r \cdot d_2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})),
\text{minimal}(T X_p), \ldots, \text{minimal}(T X_t),
\text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t),
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}),
\text{r}(T X_1, T Y_1), \ldots, \text{r}(T X_p-1, T Y_p-1), \text{r}(N, \text{HY}),
\text{compose}(T Y_1, T Y_2, K_1), \text{compose}(K_1, T Y_3, K_2), \ldots, \text{compose}(K_{p-3}, T Y_p-1, K_{p-2}),
\text{compose}(K_{p-2}, N A, Y),
 r \cdot d_2([N[T Xs], N N A, A)
\]

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

**clause 30:** \( r \cdot d_2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t),
(\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})),
\text{minimal}(T X_p), \ldots, \text{minimal}(T X_t),
\text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}),
\text{decompose}(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t),
 r \cdot d_2([T X_1, \ldots, T X_{p-1}, N[T Xs], Y, A)
\]

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:

62
clause 31: \[ r_d(X, Y, A) = \]
\[ X = [X[TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p+1})), \]
\[ process(HY), compose(K_1, HY, \ldots, HY, \ldots, HY, \ldots, HY), \]
\[ HY = I_{p+1}, \]
\[ r_d(Xs, NA, A), compose(HY, NA, Y) \]

By using applicability condition (3):

clause 32: \[ r_d(X, Y, A) = \]
\[ X = [X[TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p+1})), \]
\[ process(HY), compose(I_1, HY, \ldots, HY, \ldots, HY, \ldots, HY), \]
\[ HY = I_{p+1}, \]
\[ r_d(Xs, NA, A), compose(HY, NA, Y) \]

By using applicability condition (2):

clause 33: \[ r_d(X, Y, A) = \]
\[ X = [X[TXs], \]
\[ nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \]
\[ (nonMinimal(TX_1); \ldots; nonMinimal(TX_{p+1})), \]
\[ process(HY), compose(K_1, HY, \ldots, HY, \ldots, HY, \ldots, HY), \]
\[ HY = I_{p+1}, \]
\[ r_d(Xs, NA, A), compose(HY, NA, Y) \]

By using applicability conditions (1) and (2):
\textbf{clause 34:} \( r_{\mathcal{D}_2}(X_s, Y, A) \) —
\[
X_s = [X[T X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}), \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \)
\( r(U_1, e), \ldots, r(U_t, e), \)
\( r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( I_0 = e, \)
\( \text{compose}(I_0, e, I_1), \ldots, \text{compose}(I_{p-2}, e, I_{p-1}), \)
\( \text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \)
\( \text{compose}(I_p, e, I_{p+1}), \ldots, \text{compose}(I_t, e, I_{t+1}), \)
\( N H Y = I_{t+1}, \)
\( \text{compose}(TY_{p}, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \)
\( \text{compose}(K_{t-1}, NA, NA_1), \text{compose}(N H Y, NA_1, NA_2), \)
\( \text{compose}(K_{t-2}, NA_2, Y), r_{\mathcal{D}_2}(T X_s, NA, A) \)

By introducing new, i.e. existentially quantified, variables \( Y U_1, \ldots, Y U_t \) in place of some occurrences of \( e \):

\textbf{clause 35:} \( r_{\mathcal{D}_2}(X_s, Y, A) \) —
\[
X_s = [X[T X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \)
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}), \)
\( \text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t), \)
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \)
\( r(U_1, Y U_1), \ldots, r(U_t, Y U_t), \)
\( r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \)
\( \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \)
\( I_0 = e, \)
\( \text{compose}(I_0, Y U_1, I_1), \ldots, \text{compose}(I_{p-2}, Y U_{p-1}, I_{p-1}), \)
\( \text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \)
\( \text{compose}(I_p, Y U_p, I_{p+1}), \ldots, \text{compose}(I_t, Y U_t, I_{t+1}), \)
\( N H Y = I_{t+1}, \)
\( \text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+2}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \)
\( \text{compose}(K_{t-1}, NA, NA_1), \text{compose}(N H Y, NA_1, NA_2), \)
\( \text{compose}(K_{t-2}, NA_2, Y), r_{\mathcal{D}_2}(T X_s, NA, A) \)

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):
\text{clause 36:} \quad r_d^*(X_s, Y, A) = \\
\quad X_s = [X[T X_s], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \\
\quad (\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})), \\
\quad (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\quad \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_t), \\
\quad r(U_1, Y U_1), \ldots, r(U_t, Y U_t), \\
\quad r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \\
\quad \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}), \\
\quad I_0 = e, \\
\quad \text{compose}(I_0, Y U_1, I_1), \ldots, \text{compose}(I_{p-2}, Y U_{p-1}, I_{p-1}), \\
\quad \text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \\
\quad \text{compose}(I_p, Y U_p, I_{p+1}), \ldots, \text{compose}(I_t, Y U_t, I_{t+1}), \\
\quad N H Y = I_{t+1}, \\
\quad \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}), \\
\quad \text{compose}(K_{t-1}, N A, N A_1), \text{compose}(N H Y, N A_1, N A_2), \\
\quad \text{compose}(K_{p-2}, N A_2, Y), r_d^*(T X_s, N A, A)

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

\text{clause 37:} \quad r_d^*(X_s, Y, A) = \\
\quad X_s = [X[T X_s], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \\
\quad (\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})), \\
\quad (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\quad \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_t), \\
\quad r(U_1, Y U_1), \ldots, r(U_t, Y U_t), \\
\quad r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), \\
\quad \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}), \\
\quad I_0 = e, \\
\quad \text{compose}(I_0, Y U_1, I_1), \ldots, \text{compose}(I_{p-2}, Y U_{p-1}, I_{p-1}), \\
\quad \text{process}(H X, H H Y), \text{compose}(I_{p-1}, H H Y, I_p), \\
\quad \text{compose}(I_p, Y U_p, I_{p+1}), \ldots, \text{compose}(I_t, Y U_t, I_{t+1}), \\
\quad N H Y = I_{t+1}, \\
\quad \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}), \\
\quad \text{compose}(K_{t-1}, N A, N A_1), \text{compose}(N H Y, N A_1, N A_2), \\
\quad \text{compose}(K_{p-2}, N A_2, Y), r_d^*(T X_s, N A, A)

By folding clause 37 using DCLR:

\text{clause 38:} \quad r_d^*(X_s, Y, A) = \\
\quad X_s = [X[T X_s], \\
\quad \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \\
\quad (\text{nonMinimal}(T X_1); \ldots; \text{nonMinimal}(T X_{p-1})), \\
\quad (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \\
\quad \text{minimal}(U_1), \ldots, \text{minimal}(U_t), \\
\quad \text{nonMinimal}(N), \text{decompose}(N, H X, U_1, \ldots, U_t), \\
\quad r(T X_1, T Y_1), \ldots, r(T X_t, T Y_t), r(N, N H Y), \\
\quad \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}), \\
\quad \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}), \\
\quad \text{compose}(K_{t-1}, N A, N A_1), \text{compose}(N H Y, N A_1, N A_2), \\
\quad \text{compose}(K_{p-2}, N A_2, Y), r_d^*(T X_s, N A, A)

By \( t - p + 1 \) times folding clause 38 using clauses 1 and 2:
\textit{Theorem 9} The generalization schema $TDG_3$, which is given below, is correct.

\textbf{Proof:}

\textit{By folding clause 39 using clauses 1 and 2:}

\textbf{Clause 40:} \( r_4 d_{2}(Xs, Y, A) = \)

\begin{align*}
\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_4 d_{2}(TX_p, \ldots, TX_t[Xs], NA_1, A)
\end{align*}

\textit{By p - 1 times folding clause 40 using clauses 1 and 2:}

\textbf{Clause 41:} \( r_4 d_{2}(Xs, Y, A) = \)

\begin{align*}
\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_4 d_{2}(TX_1, \ldots, TX_p[TX_s], Y, A)
\end{align*}

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of \( P_r d_{2} \). Therefore \( P_r d_{2} \) is steadfast wrt \( S_r d_{2} \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r d_{2}\} \), 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:

\[ r(X, Y) = r_4 d_{2}([X], Y, e) \]

By taking the ‘completion’:

\[ \forall X : X, \forall Y : Y. \; I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r_4 d_{2}([X], Y, e)] \]

By unfolding the ‘completion’ above wrt \( r_4 d_{2}([X], Y, e) \) using \( S_r d_{2} \):

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

By using applicability condition (2):

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

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 \( \{S_r d_{2}\} \).

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

\( \square \)
TDG₃ : { DCRL, TDGRL, A₄₃, O₄₃₁₂, O₄₃₂₁ } where

A₄₃ : (1) compose is associative
(2) compose has ε as the left and right identity element
(3) \( I_\varepsilon(X) \land \text{minimal}(X) \Rightarrow O_{\varepsilon}(X, \varepsilon) \)
(4) \( I_r(X) \Rightarrow [\neg \text{minimal}(X) \Leftrightarrow \text{nonMinimal}(X)] \)

O₄₃₁₂ : partial evaluation of the conjunction
process(H,X, HY), compose(HY, A, NewA)
results in the introduction of a non-recursive relation

O₄₃₂₁ : partial evaluation of the conjunction
process(H,X, HY), compose(HY, I_p, I_p₋₁)
results in the introduction of a non-recursive relation

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

The specification \( S_r \) of relation \( r \) is:

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

The specification of \( r \bigcup d_2 \), namely \( S_{r \bigcup d_2} \), is:

\[
\forall X s : \text{list of } X, \forall Y, A : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in X s \Rightarrow I_r(X)) \Rightarrow [r \bigcup d_2(X s, Y, A) \Leftrightarrow (X s = [] \land Y = A) \\
\land O_s(I_q, A, I_q+1) \land Y = I_q+1)]
\]

**Proof 9** To prove the correctness of the generalization schema TDG₃, by Definition 10, we have to prove that templates DCRL and TDGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{4₃} \). By Definition 5, the templates DCRL and TDGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{4₃} \) iff DCRL is equivalent to TDGRL wrt the specification \( S_r \) provided that the conditions in \( A_{4₃} \) 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 \( \{S'_1, \ldots, S'_l\} \) 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 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 \( P_{r \bigcup d_2} \) is steadfast wrt \( S_{r \bigcup d_2} \) in \( S \), where \( P_r \) is the procedure for \( r \bigcup d_2 \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{S_{r \bigcup d_2}\} \), where \( P_r \) is the procedure for \( r \).

To prove that \( P_{r \bigcup d_2} \) is steadfast wrt \( S_{r \bigcup d_2} \) in \( S \), we do a constructive forward proof that we begin with \( S_{r \bigcup d_2} \) and from which we try to obtain \( P_{r \bigcup d_2} \).

We separate the cases of \( q \geq 1 \) by \( q = 1 \lor q \geq 2 \), then \( S_{r \bigcup d_2} \) becomes:

\[
\forall X s : \text{list of } X, \forall Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in X s \Rightarrow I_r(X)) \Rightarrow [r \bigcup d_2(X s, Y, A) \Leftrightarrow (X s = [] \land Y = A) \\
\land O_s(I_q, A, I_q+1) \land Y = I_q+1)]
\]

where \( q \geq 2 \).

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

\[
\forall X s : \text{list of } X, \forall Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in X s \Rightarrow I_r(X)) \Rightarrow [r \bigcup d_2(X s, Y, A) \Leftrightarrow (X s = [] \land Y = A) \\
\land O_s(I_q, A, I_q+1) \land Y = I_q+1)]
\]

\[
\forall X s : \text{list of } X, \forall Y : \mathcal{Y}. \ (\forall X : \mathcal{X}. X \in X s \Rightarrow I_r(X)) \Rightarrow [r \bigcup d_2(X s, Y, A) \Leftrightarrow (X s = [] \land Y = A) \\
\land O_s(I_q, A, I_q+1) \land Y = I_q+1)]
\]
where \( q \geq 2 \).

By folding using \( S_r \mathcal{U}_d \), and renaming:

\[
\forall X : \text{list of } \mathcal{X}, Y : \mathcal{Y}. \quad (\forall X : \mathcal{X}. \ X \in Xs \Rightarrow I_r(X)) \Rightarrow [r \mathcal{U}_d(Xs, Y, A) \Leftrightarrow \left( Xs = [] \land Y = A \right) \lor (Xs = [X]\mathcal{U}_d(Xs), r(X, HY) \land r \mathcal{U}_d(TXs, NA, A) \land \mathcal{O}_s(HY, NA, Y))]
\]

By taking the ‘decomposition’:

**clause 1:** \( r \mathcal{U}_d(Xs, Y, A) \leftarrow \)
\[
Xs = [], Y = A
\]

**clause 2:** \( r \mathcal{U}_d(Xs, Y, A) \leftarrow \)
\[
Xs = [X]\mathcal{U}_d(Xs), r(X, HY),
\]
\[
r \mathcal{U}_d(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

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

**clause 3:** \( r \mathcal{U}_d(Xs, Y, A) \leftarrow \)
\[
Xs = [X]\mathcal{U}_d(Xs),
\]
\[
\mathcal{O}_s(X), solve(X, HY),
\]
\[
r \mathcal{U}_d(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

**clause 4:** \( r \mathcal{U}_d(Xs, Y, A) \leftarrow \)
\[
Xs = [X]\mathcal{U}_d(Xs),
\]
\[
\mathcal{O}_s(X), \mathcal{O}_s(HX, TX_1, \ldots, TX_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_{t+1} = \epsilon,
\]
\[
\mathcal{O}_s(TY_1, l_{t+1}, l_t), \ldots, \mathcal{O}_s(TY_p, l_{p+1}, l_p),
\]
\[
\mathcal{O}_s(HX, HY), \mathcal{O}_s(HY, l_p, l_{p-1}),
\]
\[
\mathcal{O}_s(TY_{p+1}, l_{p-1}, l_{p-2}), \ldots, \mathcal{O}_s(TY_1, l_1, l_0),
\]
\[
HY = h,
\]
\[
r \mathcal{U}_d(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

By introducing

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

in clause 4, using applicability condition (4):

**clause 5:** \( r \mathcal{U}_d(Xs, Y, A) \leftarrow \)
\[
Xs = [X]\mathcal{U}_d(Xs),
\]
\[
\mathcal{O}_s(X), \mathcal{O}_s(HX, TX_1, \ldots, TX_t),
\]
\[
minimal(TX_1), \ldots, minimal(TX_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
l_{t+1} = \epsilon,
\]
\[
\mathcal{O}_s(TY_1, l_{t+1}, l_t), \ldots, \mathcal{O}_s(TY_p, l_{p+1}, l_p),
\]
\[
\mathcal{O}_s(HX, HY), \mathcal{O}_s(HY, l_p, l_{p-1}),
\]
\[
\mathcal{O}_s(TY_{p+1}, l_{p-1}, l_{p-2}), \ldots, \mathcal{O}_s(TY_1, l_1, l_0),
\]
\[
HY = h,
\]
\[
r \mathcal{U}_d(TXs, NA, A), \mathcal{O}_s(HY, NA, Y)
\]

68
\textbf{clause 6:} $r \downarrow d_2(Xs, Y, A) \leftarrow$

\begin{align*}
Xs & = [X[VXs], \\
non\text{Minimal}(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), \\
l_{t+1} & = e, \\
\text{compose}(TY_1, l_{t+1}, l_t), & \ldots , \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{process}(HX, HHY), & \text{compose}(HHY, l_p, l_{p-1}), \\
\text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), & \ldots , \text{compose}(TY_1, l_1, l_0), \\
y & = l_0, \\
r \downarrow d_2(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}

\textbf{clause 7:} $r \downarrow d_2(Xs, Y, A) \leftarrow$

\begin{align*}
Xs & = [X[VXs], \\
non\text{Minimal}(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_{t+1} & = e, \\
\text{compose}(TY_1, l_{t+1}, l_t), & \ldots , \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{process}(HX, HHY), & \text{compose}(HHY, l_p, l_{p-1}), \\
\text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), & \ldots , \text{compose}(TY_1, l_1, l_0), \\
y & = l_0, \\
r \downarrow d_2(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}

\textbf{clause 8:} $r \downarrow d_2(Xs, Y, A) \leftarrow$

\begin{align*}
Xs & = [X[VXs], \\
non\text{Minimal}(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_{t+1} & = e, \\
\text{compose}(TY_1, l_{t+1}, l_t), & \ldots , \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{process}(HX, HHY), & \text{compose}(HHY, l_p, l_{p-1}), \\
\text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), & \ldots , \text{compose}(TY_1, l_1, l_0), \\
y & = l_0, \\
r \downarrow d_2(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}

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):

\textbf{clause 9:} $r \downarrow d_2(Xs, Y, A) \leftarrow$

\begin{align*}
Xs & = [X[VXs], \\
non\text{Minimal}(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), \\
l_{t+1} & = e, \\
\text{compose}(TY_1, l_{t+1}, l_t), & \ldots , \text{compose}(TY_p, l_{p+1}, l_p), \\
\text{process}(HX, HHY), & \text{compose}(HHY, l_p, l_{p-1}), \\
\text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), & \ldots , \text{compose}(TY_1, l_1, l_0), \\
y & = l_0, \\
r \downarrow d_2(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}

By using applicability condition (3):
\textbf{clause 10: }\quad r_{\mathcal{J}d_5}(X_s,Y,A) = \\
X_s = [X[T X_s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_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), \\
h_{t+1} = e, \\
\text{compose}(e, h_{t+1}, h_t), \ldots, \text{compose}(e, l_p+1, l_p), \\
\text{process}(H X, H HY), \text{compose}(H HY, l_p, l_{p-1}), \\
\text{compose}(e, l_{p-1}, l_{p-2}), \ldots, \text{compose}(e, h_1, l_0), \\
HY = l_0, \\
r_{\mathcal{J}d_5}(TX_s, NA, A), \text{compose}(HY, NA, Y) \\
\text{By deleting one of the minimal}(TX_1), \ldots, \text{minimal}(TX_t) \text{ atoms in clause 10:} \\
\textbf{clause 11: }\quad r_{\mathcal{J}d_5}(X_s,Y,A) = \\
X_s = [X[T X_s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_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), \\
h_{t+1} = e, \\
\text{compose}(e, h_{t+1}, h_t), \ldots, \text{compose}(e, l_p+1, l_p), \\
\text{process}(H X, H HY), \text{compose}(H HY, l_p, l_{p-1}), \\
\text{compose}(e, l_{p-1}, l_{p-2}), \ldots, \text{compose}(e, h_1, l_0), \\
HY = l_0, \\
r_{\mathcal{J}d_5}(TX_s, NA, A), \text{compose}(HY, NA, Y) \\
\text{By using applicability condition (2):} \\
\textbf{clause 12: }\quad r_{\mathcal{J}d_5}(X_s,Y,A) = \\
X_s = [X[T X_s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_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), \\
h_{t+1} = e, \\
l_t = l_{t+1}, \ldots, l_p = l_{p+1}, \\
\text{process}(H X, H HY), \text{compose}(H HY, l_p, l_{p-1}), \\
l_{p-2} = l_{p-1}, \ldots, l_0 = l_1, \\
HY = l_0, \\
r_{\mathcal{J}d_5}(TX_s, NA, A), \text{compose}(HY, NA, Y) \\
\text{By simplification:} \\
\textbf{clause 13: }\quad r_{\mathcal{J}d_5}(X_s,Y,A) = \\
X_s = [X[T X_s], \\
\text{nonMinimal}(X), \text{decompose}(X, H X, TX_1, \ldots, TX_t), \\
\text{minimal}(T X_1), \ldots, \text{minimal}(T X_t), \\
r_{\mathcal{J}d_5}(TX_s, NA, A), \\
\text{process}(H X, H HY), \text{compose}(HY, NA, Y) \\
\text{By } p-1 \text{ times unfolding clause 6 wrt } r(T X_1, TY_1), \ldots, r(T X_{p-1}, TY_{p-1}) \text{ using DCRL, and simplifying using condition (4):}
clause 14: \( r_{\mathcal{D}_2}(X s, Y, A) \) 
\[ X s = [X[T X s], \]
nonMinimal(X), decompose(X, H X, T X_1, \ldots, T X_t), minimal(T X_1), \ldots, minimal(T X_{p-1}), nonMinimal(T X_p), \ldots; nonMinimal(T X_t)),
minimal(T X_1), \ldots, minimal(T X_{p-1}),
solve(T X_1, T Y_1), \ldots, solve(T X_{p-1}, T Y_{p-1}), \]
\( r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t), l_{t+1} = \epsilon, \)
\( \) compose(T Y_t, l_{t+1}, l_t), \ldots, compose(T Y_p, l_{p+1}, l_p), process(H X, HH Y), compose(H H Y, l_p, l_{p-1}),
\( r(T Y_{p-1}, l_{p-1}, l_{p-2}), \ldots, compose(T Y_1, l_1, l_0), H Y = l_0, r_{\mathcal{D}_2}(T X s, NA, A), compose(H Y, NA, Y) \)

By deleting one of the minimal(T X_1), \ldots, minimal(T X_{p-1}) atoms in clause 14:

clause 15: \( r_{\mathcal{D}_2}(X s, Y, A) \) 
\[ X s = [X[T X s], \]
nonMinimal(X), decompose(X, H X, T X_1, \ldots, T X_t), minimal(T X_1), \ldots, minimal(T X_{p-1}), nonMinimal(T X_p), \ldots; nonMinimal(T X_t)),
solve(T X_1, T Y_1), \ldots, solve(T X_{p-1}, T Y_{p-1}), \]
\( r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t), l_{t+1} = \epsilon, \)
\( \) compose(T Y_t, l_{t+1}, l_t), \ldots, compose(T Y_p, l_{p+1}, l_p), process(H X, HH Y), compose(H H Y, l_p, l_{p-1}),
\( r(T Y_{p-1}, l_{p-1}, l_{p-2}), \ldots, compose(T Y_1, l_1, l_0), H Y = l_0, r_{\mathcal{D}_2}(T X s, NA, A), compose(H Y, NA, Y) \)

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

clause 16: \( r_{\mathcal{D}_2}(X s, Y, A) \) 
\[ X s = [X[T X s], \]
nonMinimal(X), decompose(X, H X, T X_1, \ldots, T X_t), minimal(T X_1), \ldots, minimal(T X_{p-1}), nonMinimal(T X_p), \ldots; nonMinimal(T X_t)),
solve(T X_1, T Y_1), \ldots, solve(T X_{p-1}, T Y_{p-1}), \]
\( r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t), l_{t+1} = \epsilon, \)
\( \) compose(l_0, T Y_1, l_1), \ldots, compose(l_{p-2}, T Y_{p-1}, l_{p-1}), process(H X, H H Y), compose(l_{p-1}, H H Y, l_p),
\( H Y = l_p, compose(H Y, NA, Y), r(T Y_p, l_{p+1}, NA), r(T Y_{p+1}, l_{p+2}, l_{p+1}), \ldots, compose(T Y_{t-1}, l_t, l_{t-1}), Composer(T Y_t, NA, A, l_1), r_{\mathcal{D}_2}(T X s, NA, A) \)

By \( t = p \) times folding clause 16 using clauses 1 and 2:
\textit{clause 17: } \( \mathcal{J}_d(X, Y, A) = \)
\[
X_s = [X \, | \, X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( \text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e), \)
\( l_0 = e, \)
\( \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{p-2}, TY_{p-1}, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HX = l_p, \)
\( r \mathcal{J}_d([TX_p, \ldots, TX_t X_s], NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):

\textit{clause 18: } \( r \mathcal{J}_d(X, Y, A) = \)
\[
X_s = [X \, | \, X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( \text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e), \)
\( l_0 = e, \)
\( \text{compose}(l_0, e, l_1), \ldots, \text{compose}(l_{p-2}, e, l_{p-1}), \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HX = l_p, \)
\( r \mathcal{J}_d([TX_p, \ldots, TX_t X_s], NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (2):

\textit{clause 19: } \( r \mathcal{J}_d(X, Y, A) = \)
\[
X_s = [X \, | \, X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( \text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( \text{solve}(TX_1, e), \ldots, \text{solve}(TX_{p-1}, e), \)
\( l_0 = e, \)
\( l_1 = l_0, \ldots, l_{p-1} = l_{p-2}, \)
\( \text{process}(HX, HHY), \text{compose}(l_{p-1}, HHY, l_p), \)
\( HX = l_p, \)
\( r \mathcal{J}_d([TX_p, \ldots, TX_t X_s], NA, A), \text{compose}(HY, NA, Y) \)

By simplification:

\textit{clause 20: } \( r \mathcal{J}_d(X, Y, A) = \)
\[
X_s = [X \, | \, X_s],
\]
\( \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \)
\( \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{p-1}), \)
\( \text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \)
\( r \mathcal{J}_d([TX_p, \ldots, TX_t X_s], NA, A), \)
\( \text{process}(HX, HHY), \text{compose}(HY, NA, Y) \)

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:
clause 21: \( r_{\mathcal{J}}(X, Y, A) = \)
\[
X = [X[TXs]],
\]
nonMinimal(X), decompose(X, \( HX,TX_1, \ldots, TX_t \)),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \),
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \),
h_{t+1} = \epsilon,
\]
\( \text{compose}(TY_1, h_{t+1}, h_t), \ldots, \text{compose}(TY_p, l_{p+1}, l_p) \),
\( \text{process}(HX, HHY), \text{compose}(HY, I, l_p, l_{p-1}) \),
\( \text{compose}(TY_p, l_{p+1}, l_{p-1}), \ldots, \text{compose}(TY_1, I, I_0) \),
\( HY = I_0 \),
\( r_{\mathcal{J}}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (3):

clause 22: \( r_{\mathcal{J}}(X, Y, A) = \)
\[
X = [X[TXs]],
\]
nonMinimal(X), decompose(X, \( HX,TX_1, \ldots, TX_t \)),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \),
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \),
\( r(U_1, \epsilon), \ldots, r(U_{p-1}, \epsilon) \),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \),
h_{t+1} = \epsilon,
\]
\( \text{compose}(TY_1, h_{t+1}, h_t), \ldots, \text{compose}(TY_p, l_{p+1}, l_p) \),
\( \text{process}(HX, HHY), \text{compose}(HY, I, l_p, l_{p-1}) \),
\( \text{compose}(TY_p, l_{p+1}, l_{p-1}), \ldots, \text{compose}(TY_1, I, I_0) \),
\( HY = I_0 \),
\( r_{\mathcal{J}}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability condition (2):

clause 23: \( r_{\mathcal{J}}(X, Y, A) = \)
\[
X = [X[TXs]],
\]
nonMinimal(X), decompose(X, \( HX,TX_1, \ldots, TX_t \)),
\( \text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1}) \),
\( \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t) \),
\( \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}) \),
\( r(U_1, \epsilon), \ldots, r(U_{p-1}, \epsilon) \),
\( r(TX_1, TY_1), \ldots, r(TX_t, TY_t) \),
h_{t+1} = \epsilon,
\]
\( \text{compose}(TY_1, h_{t+1}, h_t), \ldots, \text{compose}(TY_p, l_{p+1}, l_p) \),
\( \text{process}(HX, HHY), \text{compose}(HY, I, l_p, l_{p-1}) \),
\( \text{compose}(\epsilon, l_p, K_1), \text{compose}(\epsilon, K_1, K_2), \ldots, \text{compose}(\epsilon, K_{p-2}, K_{p-1}) \),
\( \text{compose}(TY_p, l_{p+1}, l_{p-1}), \ldots, \text{compose}(TY_1, I, I_0) \),
\( HY = I_0 \),
\( r_{\mathcal{J}}(TXs, NA, A), \text{compose}(HY, NA, Y) \)

By using applicability conditions (1) and (2):
clause 24: \( r_d^2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(\text{nonMinimal}(TX_1);\ldots;\text{nonMinimal}(TX_{p-1})),
\text{minimal}(TX_p),\ldots,\text{minimal}(TX_t),
\text{minimal}(U_1),\ldots,\text{minimal}(U_{p-1}),
\]r(U_1,e),\ldots,r(U_{p-1},e),
\]r(TX_1,TY_1),\ldots,r(TX_t,TY_t),
\]h_{t+1} = e,
\]compose(TY_1,h_{t+1},h_t),\ldots,compose(TY_p,h_{p+1},h_p),
\]process(HX,H HY),compose(H HY,I_p,I_{p-1}),
\]compose(e,l_{p+1},l_{p-2}),\ldots,compose(e,h_1,h_0),
\]HY = l_0, r_d^2(T X, N A, A), compose(HY, N A, N N A),
\]compose(T Y, T Y_2, K_1), compose(K_1,T Y_3, K_2),\ldots,compose(K_{p-3},T Y_{p-1}, K_{p-2}),
\]compose(K_{p-2}, N N A, Y)
\]

By introducing new, i.e. existentially quantified, variables \( Y U_1,\ldots,Y U_{p-1} \) in place of some occurrences of \( e \):

clause 25: \( r_d^2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(\text{nonMinimal}(TX_1);\ldots;\text{nonMinimal}(TX_{p-1})),
\text{minimal}(TX_p),\ldots,\text{minimal}(TX_t),
\text{minimal}(U_1),\ldots,\text{minimal}(U_{p-1}),
\]r(U_1,Y U_1),\ldots,r(U_{p-1},Y U_{p-1}),
\]r(TX_1,TY_1),\ldots,r(TX_t,TY_t),
\]h_{t+1} = e,
\]compose(TY_1,h_{t+1},h_t),\ldots,compose(TY_p,h_{p+1},h_p),
\]process(HX,H HY),compose(H HY,I_p,I_{p-1}),
\]compose(Y U_{p-1},l_{p+1},l_{p-2}),\ldots,compose(Y U_1,l_1,l_0),
\]HY = l_0, r_d^2(T X, N A, A), compose(HY, N A, N N A),
\]compose(T Y, T Y_2, K_1), compose(K_1,T Y_3, K_2),\ldots,compose(K_{p-3},T Y_{p-1}, K_{p-2}),
\]compose(K_{p-2}, N N A, Y)
\]

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

clause 26: \( r_d^2(Xs, Y, A) = \)
\[
Xs = [X[T Xs],
\text{nonMinimal}(X), \text{decompose}(X,HX,TX_1,\ldots,TX_t),
(\text{nonMinimal}(TX_1);\ldots;\text{nonMinimal}(TX_{p-1})),
\text{minimal}(TX_p),\ldots,\text{minimal}(TX_t),
\text{minimal}(U_1),\ldots,\text{minimal}(U_{p-1}),
\]r(U_1,Y U_1),\ldots,r(U_{p-1},Y U_{p-1}),
\]nonMinimal(N), \text{decompose}(N, H X, U_1,\ldots,U_{p-1}, T X_p,\ldots,T X_t),
\]r(TX_1,TY_1),\ldots,r(TX_t,TY_t),
\]h_{t+1} = e,
\]compose(TY_1,h_{t+1},h_t),\ldots,compose(TY_p,h_{p+1},h_p),
\]process(HX,H HY),compose(H HY,I_p,I_{p-1}),
\]compose(Y U_{p-1},l_{p+1},l_{p-2}),\ldots,compose(Y U_1,l_1,l_0),
\]HY = l_0, r_d^2(T X, N A, A), compose(HY, N A, N N A),
\]compose(T Y, T Y_2, K_1), compose(K_1,T Y_3, K_2),\ldots,compose(K_{p-3},T Y_{p-1}, K_{p-2}),
\]compose(K_{p-2}, N N A, Y)
\]

By duplicating goal \( \text{decompose}(N, H X, U_1,\ldots,U_{p-1}, T X_p,\ldots,T X_t) \):
clause 27:  \( r \mathcal{A}_d(X, Y, A) \)
\[
\begin{align*}
X s &= [X \uparrow T X s], \\
nonMinimal(X), &
decompose(X, H X, T X_1, \ldots, T X_t), \\
(nonMinimal(T X_1); \ldots; &nonMinimal(T X_{p-1})), \\
minimal(T X_p), \ldots, &
minimal(T X_t), \\
minimal(U_1), \ldots, &
minimal(U_{p-1}), \\
r(U_1, Y U_1), \ldots, &
r(U_{p-1}, Y U_{p-1}), \\
nonMinimal(N), &
decompose(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t), \\
decompose(N, &H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t), \\
r(T X_1, T Y_1), \ldots, &r(T X_1, T Y_1), \\
l_{t+1} &= e, \\
compose(T Y_1, l_{t+1}, l_t), \ldots, &compose(T Y_p, l_{p+1}, l_p), \\
process(&H X, H H Y), &compose(H H Y, l_p, l_{p-1}), \\
compose(Y U_{p-1}, l_{p-1}, l_{p-2}), \ldots, &compose(Y U_1, l_1, l_0), \\
HY &= l_0, &r \mathcal{A}_d(T X s, N A, A), &compose(H Y, N A, N N A), \\
compose(T Y_1, T Y_2, K_1), &compose(K_1, T Y_3, K_2), \ldots, &compose(K_{p-3}, T Y_{p-1}, K_{p-2}), \\
compose(K_{p-2}, N N A, Y) &
\end{align*}
\]

By folding clause 27 using **DCRI:**

clause 28:  \( r \mathcal{A}_d(X, Y, A) \)
\[
\begin{align*}
X s &= [X \uparrow T X s], \\
nonMinimal(X), &
decompose(X, H X, T X_1, \ldots, T X_t), \\
(nonMinimal(T X_1); \ldots; &nonMinimal(T X_{p-1})), \\
minimal(T X_p), \ldots, &
minimal(T X_t), \\
minimal(U_1), \ldots, &
minimal(U_{p-1}), \\
decompose(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t), \\
r(T X_1, T Y_1), \ldots, &r(T X_{p-1}, T Y_{p-1}), r(N, H Y), \\
r \mathcal{A}_d(T X s, N A, A), &compose(H Y, N A, N N A), \\
compose(T Y_1, T Y_2, K_1), &compose(K_1, T Y_3, K_2), \ldots, &compose(K_{p-3}, T Y_{p-1}, K_{p-2}), \\
compose(K_{p-2}, N N A, Y) &
\end{align*}
\]

By folding clause 28 using clauses 1 and 2:

clause 29:  \( r \mathcal{A}_d(X, Y, A) \)
\[
\begin{align*}
X s &= [X \uparrow T X s], \\
nonMinimal(X), &
decompose(X, H X, T X_1, \ldots, T X_t), \\
(nonMinimal(T X_1); \ldots; &nonMinimal(T X_{p-1})), \\
minimal(T X_p), \ldots, &
minimal(T X_t), \\
minimal(U_1), \ldots, &
minimal(U_{p-1}), \\
decompose(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t), \\
r(T X_1, T Y_1), \ldots, &r(T X_{p-1}, T Y_{p-1}), \\
r \mathcal{A}_d(N[T X s], N N A, A), &compose(T Y_1, T Y_2, K_1), &compose(K_1, T Y_3, K_2), \ldots, &compose(K_{p-3}, T Y_{p-1}, K_{p-2}), \\
compose(K_{p-2}, N N A, Y) &
\end{align*}
\]

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

clause 30:  \( r \mathcal{A}_d(X, Y, A) \)
\[
\begin{align*}
X s &= [X \uparrow T X s], \\
nonMinimal(X), &
decompose(X, H X, T X_1, \ldots, T X_t), \\
(nonMinimal(T X_1); \ldots; &nonMinimal(T X_{p-1})), \\
minimal(T X_p), \ldots, &
minimal(T X_t), \\
minimal(U_1), \ldots, &
minimal(U_{p-1}), \\
decompose(N, H X, U_1, \ldots, U_{p-1}, T X_p, \ldots, T X_t), \\
r \mathcal{A}_d(T X_1, \ldots, T X_{p-1}, N[T X s], Y, A) &
\end{align*}
\]

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:}
By using applicability condition (3):

\[ \text{clause 31: } r_{\mathcal{D}_2}(Xs, Y, A) = \]

\[ Xs = [X[TXS], \]

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

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

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

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

\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i), \]

\[ h_{i+1} = \epsilon, \]

\[ \text{compose}(TY_1, h_{i+1}, h_1), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \]

\[ \text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \]

\[ \text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(TY_1, l_1, l_0), \]

\[ HY = l_0, \]

\[ r_{\mathcal{D}_2}(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability condition (2):

\[ \text{clause 32: } r_{\mathcal{D}_2}(Xs, Y, A) = \]

\[ Xs = [X[TXS], \]

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

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

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

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

\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i), \]

\[ r(U_1, \epsilon), \ldots, r(U_i, \epsilon), \]

\[ h_{i+1} = \epsilon, \]

\[ \text{compose}(TY_1, h_{i+1}, h_1), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \]

\[ \text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \]

\[ \text{compose}(TY_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(TY_1, l_1, l_0), \]

\[ HY = l_0, \]

\[ r_{\mathcal{D}_2}(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability conditions (1) and (2):
clause 34: \( rJd_2(X_s, Y, A) = \)
\[
X_s = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
minimal(U_1), \ldots, minimal(U_t), \\
r(U_1, e), \ldots, r(U_t, e), \\
h_{t+1} = e, \\
\text{compose}(e, I_{t+1}, I_t), \ldots, \text{compose}(e, I_{p+1}, I_p), \\
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\text{compose}(e, I_{p-1}, I_{p-2}), \ldots, \text{compose}(e, I_1, I_0), \\
NHY = I_0, \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+1}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
rJd_2(TX_s, NA, A), \text{compose}(K_{t-1}, NA, NA_1), \\
\text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y)
\]

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

\[
\text{clause 35: } rJd_2(X_s, Y, A) = \\
X_s = [X[TXs], \\
nonMinimal(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
(nonMinimal(TX_1); \ldots; nonMinimal(TX_{p-1})), \\
(nonMinimal(TX_p); \ldots; nonMinimal(TX_t)), \\
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \\
minimal(U_1), \ldots, minimal(U_t), \\
r(U_1, YU_1), \ldots, r(U_t, YU_t), \\
h_{t+1} = e, \\
\text{compose}(YU_1, I_{t+1}, I_t), \ldots, \text{compose}(YU_p, I_{p+1}, I_p), \\
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}), \\
\text{compose}(YU_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0), \\
NHY = I_0, \\
\text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_2, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
\text{compose}(TY_p, TY_{p+1}, K_p), \text{compose}(K_p, TY_{p+1}, K_{p+1}), \ldots, \text{compose}(K_{t-2}, TY_t, K_{t-1}), \\
rJd_2(TX_s, NA, A), \text{compose}(K_{t-1}, NA, NA_1), \\
\text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y)
\]

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

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

always holds (because \( N \) is existentially quantified):
clause 36: $r(\mathcal{D}_2(X_s, Y, A) =
\begin{align*}
X_s &= [X[T 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{r}(TX_1, TY_1), \ldots, \text{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),
\text{r}(U_1, Y U_1), \ldots, \text{r}(U_t, Y U_t),
\end{align*}
\begin{align*}
l_{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}(HX, HY Y), \text{compose}(HY 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),
\text{NHY} &= l_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{r}(\mathcal{D}_2(X_s, N, A), \text{compose}(K_{t-1}, NA, NA_1),
\text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y)
\end{align*}

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

clause 37: $r(\mathcal{D}_2(X_s, Y, A) =
\begin{align*}
X_s &= [X[T 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{r}(TX_1, TY_1), \ldots, \text{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),
\text{r}(U_1, Y U_1), \ldots, \text{r}(U_t, Y U_t),
\end{align*}
\begin{align*}
l_{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}(HX, HY Y), \text{compose}(HY 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),
\text{NHY} &= l_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{r}(\mathcal{D}_2(X_s, N, A), \text{compose}(K_{t-1}, NA, NA_1),
\text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y)
\end{align*}

By folding clause 37 using DCRL:

clause 38: $r(\mathcal{D}_2(X_s, Y, A) =
\begin{align*}
X_s &= [X[T 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{r}(TX_1, TY_1), \ldots, \text{r}(TX_t, TY_t), \text{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}),
\text{r}(\mathcal{D}_2(X_s, N, A), \text{compose}(K_{t-1}, NA, NA_1),
\text{compose}(NHY, NA_1, NA_2), \text{compose}(K_{p-2}, NA_2, Y)
\end{align*}

By $t - p + 1$ times folding clause 38 using clauses 1 and 2:
\( \text{clause 39: } r_d(Xs, Y, A) = 
\begin{align*}
Xs &= [X [TXs]], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_i), \\
(\text{nonMinimal}(TX_1); & \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); & \ldots; \text{nonMinimal}(TX_i)), \\
r(TX_1, TY_1), & \ldots, r(TX_{p-1}, TY_{p-1}), r(N, NHY), \\
\text{decompose}(N, HX, U_1, \ldots, U_i), \\
\text{minimal}(U_1), & \ldots, \text{minimal}(U_i), \\
\text{compose}(TY_1, TY_2, K_1), & \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
r_d([TX_p, \ldots, TX_i [TXs], N A_1, A]), \\
\text{compose}(NY, N A_1, N A_2), & \text{compose}(K_{p-2}, N A_2, Y) 
\end{align*} \)

By folding clause 39 using clauses 1 and 2:

\( \text{clause 40: } r_d(Xs, Y, A) = 
\begin{align*}
Xs &= [X [TXs]], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_i), \\
(\text{nonMinimal}(TX_1); & \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); & \ldots; \text{nonMinimal}(TX_i)), \\
r(TX_1, TY_1), & \ldots, r(TX_{p-1}, TY_{p-1}), \\
\text{decompose}(N, HX, U_1, \ldots, U_i), \\
\text{minimal}(U_1), & \ldots, \text{minimal}(U_i), \\
\text{compose}(TY_1, TY_2, K_1), & \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \\
r_d([N, TX_p, \ldots, TX_i [TXs], N A_1, A]), \\
\text{compose}(K_{p-2}, N A_2, Y) 
\end{align*} \)

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

\( \text{clause 41: } r_d(Xs, Y, A) = 
\begin{align*}
Xs &= [X [TXs]], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_i), \\
(\text{nonMinimal}(TX_1); & \ldots; \text{nonMinimal}(TX_{p-1})), \\
(\text{nonMinimal}(TX_p); & \ldots; \text{nonMinimal}(TX_i)), \\
\text{decompose}(N, HX, U_1, \ldots, U_i), \\
\text{minimal}(U_1), & \ldots, \text{minimal}(U_i), \\
r_d([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_i [TXs], Y, A) 
\end{align*} \)

Clauses 1, 3, 13, 20, 30 and 41 are the clauses of \( P_r \). Therefore \( P_r \) is steadfast wrt \( S_r \). Therefore, \( P_r \) is also steadfast wrt \( S_r \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r \} \), 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:

\[ r(X, Y) = r_d([X], Y, \epsilon) \]

By taking the ‘completion’:

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

By unfolding the ‘completion’ above wrt \( r_d([X], Y, \epsilon) \) using \( S_r \cdot d_3 \):

\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, l_1 : \mathcal{Y}. \quad \mathcal{O}_r(X, Y_1) \land l_1 = Y_1 \land \mathcal{O}_c(l_1, \epsilon, Y)] \]

By using applicability condition (2):

\[ \forall X : \mathcal{X}, \forall Y : \mathcal{Y}. \quad \mathcal{I}_r(X) \Rightarrow [r(X, Y) \Leftrightarrow \exists Y_1, l_1 : \mathcal{Y}. \quad \mathcal{O}_r(X, Y_1) \land l_1 = Y_1 \land Y = l_1] \]

By simplification:

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

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r \} \).

Therefore, \( TDGRL \) is also steadfast wrt \( S_r \) in \( S \).
Theorem 10  The generalization schema $TDG_4$, which is given below, is correct.

$TDG_4 : \{ \text{DCRL, TDGLR, } A_{d4}, O_{d412}, O_{d421} \}$ where

$A_{d4} : (1) \text{ compose is associative}
(2) \text{ compose has } e \text{ as the left and right identity element}
(3) \forall X : \text{ minimal}(X) \implies O_r(X, e)
(4) \forall X : \text{ minimal}(X) \implies \text{ nonMinimal}(X)$

$O_{d412} : \text{ partial evaluation of the conjunction }$

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

$O_{d421} : \text{ partial evaluation of the conjunction }$

$\text{ process}(HX, HY), \text{ compose}(HY, I_p, I_{p-1})$
results in the introduction of a non-recursive relation

where the template 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:

$\forall X : X, \forall Y : Y. \quad \forall r(X) \implies [r(X, Y) \iff O_r(X, Y)]$

The specification $S_{rd4}$:

$\forall X : \text{ list of } X, \forall Y : Y. \quad (\forall X : X. \quad \forall r(X) \implies [r(X, Y) \iff (X = [] \land Y = A)]$

$\land \forall(X = [X_1, X_2, \ldots, X_q]) \land [\land_{i=1} \text{ minimal}(X_i) \land \land_{i=1} \text{ nonMinimal}(X_i)]$

$\land [\land_{i=1} \text{ minimal}(Y_i) \land \land_{i=1} \text{ nonMinimal}(Y_i)]$

$\land [\land_{i=1} \text{ compose}(X_i, Y_i) \land \land_{i=1} \text{ compose}(I_i, I_{i+1})]$

$\land O_r(A, I_p, I_{p+1}) \land Y = I_{p+1}]$

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 $A_{d4}$. By Definition 5, the templates DCRL and TDGLR are equivalent wrt $S_r$ under the applicability conditions $A_{d4}$ iff DCRL is equivalent to TDGLR wrt the specification $S_r$ provided that the conditions in $A_{d4}$ 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 $S = \{S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{compose}}, S_{\text{process}}, S_{\text{compose}}\}$,

where $S_{\text{minimal}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{compose}}, S_{\text{process}}, S_{\text{compose}}$ are the specifications of minimal, nonMinimal, solve, compose, 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 $\{S_1, \ldots, S_m\}$ and $\{S'_1, \ldots, S'_l\}$ in Definition 4 are equal to $S$ when $Q$ is obtained by simultaneous tuple-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 $P_r d_4$ is steadfast wrt $S_{rd4}$ in $S$, where $P_r d_4$ is the procedure for $r d_4$, and $P_r$ is starfast wrt $S_r$ in $\{S_{rd4}\}$, where $P_r$ is the procedure for $r$.

To prove that $P_r d_4$ is steadfast wrt $S_{rd4}$ is $S$, we do a constructive forward proof that we begin with $S_{rd4}$, and from which we try to obtain $P_r d_4$.

If we separate the cases of $q \geq 1$ by $q = 1 \lor q \geq 2$, then $S_{rd4}$ becomes:

$\forall X : \text{ list of } X, \forall Y : Y. \quad (\forall X : X. \quad \forall r(X) \implies [r(X, Y) \iff O_r(X, Y)]$ 

$(X = []) \land Y = A)$

$\land (X = [X_1]) \land O_r(X_1, Y_1) \land I_1 = I_1 \land O_r(A, I_1, I_2) \land Y = I_2)$

$\land (X = [X_1, X_2, \ldots, X_q]) \land [\land_{i=1} \text{ minimal}(X_i) \land \land_{i=1} \text{ nonMinimal}(X_i)]$

$\land [\land_{i=1} \text{ minimal}(Y_i) \land \land_{i=1} \text{ nonMinimal}(Y_i)]$

$\land [\land_{i=1} \text{ compose}(X_i, Y_i) \land \land_{i=1} \text{ compose}(I_i, I_{i+1})]$

$\land O_r(A, I_p, I_{p+1}) \land Y = I_{p+1}]$

where $q \geq 2$.

By using applicability conditions (1) and (2);
\[\forall x : \text{list of } X, \forall y : Y. \quad (\forall x : X. X \in x \Rightarrow I_r(x)) \Rightarrow [\mathcal{r}d_1(x, y, a) \iff (x = [] \wedge y = a)]
\]
\[
\forall (x = [x_1, \ldots, x_i]) \wedge x_1 = I_1 \wedge T Y = A \wedge O_c(A, I_1, N) \wedge \bigwedge_{\nu = 1}^n O_c(I_{\nu - 1}, Y_i, I_i) \wedge T Y = I_q \wedge O_c(A, I_1, N) \wedge O_c(N, I, Y, T Y, Y)]
\]

where \( q \geq 2 \).

By folding using \( S_{r \downarrow d_1} \), and renaming:

\[
\forall x : \text{list of } X, \forall y : Y. \quad (\forall x : X. X \in x \Rightarrow I_r(x)) \Rightarrow [\mathcal{r}d_1(x, y, a) \iff (x = [] \wedge y = a)]
\]
\[
\forall (x = [x_1, \ldots, x_i]) \wedge x_1 = I_1 \wedge T Y = A \wedge O_c(A, I_1, N) \wedge r_d(T X, Y, N A)]
\]

By taking the 'decompletion':

\[\text{clause 1: } \mathcal{r}d_1(x, y, a) \iff X = [], Y = A\]

\[\text{clause 2: } \mathcal{r}d_1(x, y, a) \iff X = [X.T X, r(X, HY), \text{compose}(A, HY, N A), r_d(T X, y, N A)]\]

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

\[\text{clause 3: } \mathcal{r}d_1(x, y, a) \iff X = [X.T X, \text{minimal}(X), \text{solve}(X, HY), \text{compose}(A, HY, N A), r_d(T X, Y, N A)]\]

\[\text{clause 4: } \mathcal{r}d_1(x, y, a) \iff X = [X.T X, \text{nonMinimal}(X), \text{compose}(X, HX, TX_1, \ldots, TX_i), r(TX_1, TY_1), \ldots, r(TX_i, TY_i), I_{i+1} = \epsilon, \text{compose}(TY_i, I_{i+1}, I_i), \ldots, \text{compose}(TY_{i}, I_{p+1}, I_p), \text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1}), \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), HY = I_0, \text{compose}(A, HY, N A), r_d(T X, Y, N A)]\]

By introducing

\[((\text{minimal}(TX_1)) \wedge \ldots \wedge \text{minimal}(TX_i)) \wedge ((\text{nonMinimal}(TX_1)) \wedge \ldots \wedge (\text{nonMinimal}(TX_{p-1})) \wedge ((\text{minimal}(TX_1)) \wedge \ldots \wedge (\text{minimal}(TX_i)))]\]

in clause 4, using applicability condition (4):

\[\text{clause 5: } \mathcal{r}d_1(x, y, a) \iff X = [X.T X, \text{nonMinimal}(X), \text{compose}(X, HX, TX_1, \ldots, TX_i), \text{minimal}(TX_1), \ldots, \text{minimal}(TX_i), r(TX_1, TY_1), \ldots, r(TX_i, TY_i), I_{i+1} = \epsilon, \text{compose}(TY_i, I_{i+1}, I_i), \ldots, \text{compose}(TY_{i}, I_{p+1}, I_p), \text{process}(HX, HY), \text{compose}(HY, I_p, I_{p-1}), \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), HY = I_0, \text{compose}(A, HY, N A), r_d(T X, Y, N A)]\]
clause 6: \[ r\mathcal{d}_1(X_s, Y, A) \leftarrow \]
\[ X_s = [X \map X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{t-1}), \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_t, TY_t), \]
\[ I_{t+1} = \varepsilon, \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HHY, NA), r\mathcal{d}_1(TX_s, Y, NA) \]

clause 7: \[ r\mathcal{d}_1(X_s, Y, A) \leftarrow \]
\[ X_s = [X \map X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HHY, NA), r\mathcal{d}_1(TX_s, Y, NA) \]

clause 8: \[ r\mathcal{d}_1(X_s, Y, A) \leftarrow \]
\[ X_s = [X \map X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HHY, NA), r\mathcal{d}_1(TX_s, Y, NA) \]

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

clause 9: \[ r\mathcal{d}_1(X_s, Y, A) \leftarrow \]
\[ X_s = [X \map X_s], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{minimal}(TX_1), \ldots, \text{minimal}(TX_{t-1}), \]
\[ \text{solve}(TX_1, TY_1), \ldots, \text{solve}(TX_t, TY_t), \]
\[ I_{t+1} = \varepsilon, \]
\[ \text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HHY, I_p, I_{p-1}), \]
\[ \text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0), \]
\[ HY = I_0, \]
\[ \text{compose}(A, HHY, NA), r\mathcal{d}_1(TX_s, Y, NA) \]

By using applicability condition (3):
clause 10:  \( r_{d_1}(X_s, Y, A) = X_s = [X_{[T X_s]}], \)
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),
minimal(TX_1), minimal(TX_t),
solve(TX_1, \epsilon), \ldots, solve(TX_t, \epsilon),
\( l_{t+1} = \epsilon, \)
compose(\( \epsilon, l_{t+1}, l_t \)), \ldots, compose(\( \epsilon, l_{p+1}, l_p \)),
process(HX, H HY), compose(H HY, l_p, l_{p-1}),
compose(\( \epsilon, l_{p-1}, l_{p-2} \)), \ldots, compose(\( \epsilon, l_1, l_0 \)),
HY = l_0,
compose(A, HY, NA), r_{d_1}(TX_s, Y, NA)

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

clause 11:  \( r_{d_1}(X_s, Y, A) = X_s = [X_{[T X_s]}], \)
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),
minimal(TX_1), minimal(TX_t),
solve(TX_1, \epsilon), \ldots, solve(TX_t, \epsilon),
\( l_{t+1} = \epsilon, \)
compose(\( \epsilon, l_{t+1}, l_t \)), \ldots, compose(\( \epsilon, l_{p+1}, l_p \)),
process(HX, H HY), compose(H HY, l_p, l_{p-1}),
compose(\( \epsilon, l_{p-1}, l_{p-2} \)), \ldots, compose(\( \epsilon, l_1, l_0 \)),
HY = l_0,
compose(A, HY, NA), r_{d_1}(TX_s, Y, NA)

By using applicability condition (2):

clause 12:  \( r_{d_1}(X_s, Y, A) = X_s = [X_{[T X_s]}], \)
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),
minimal(TX_1), minimal(TX_t),
solve(TX_1, \epsilon), \ldots, solve(TX_t, \epsilon),
\( l_{t+1} = \epsilon, \)
\( l_t = l_{t+1}, \ldots, l_p = l_{p+1}, \)
process(HX, H HY), compose(H HY, l_p, l_{p-1}),
\( l_{p-2} = l_{p-1}, \ldots, l_0 = l_1, \)
HY = l_0,
compose(A, HY, NA), r_{d_1}(TX_s, Y, NA)

By simplification:

clause 13:  \( r_{d_1}(X_s, Y, A) = X_s = [X_{[T X_s]}], \)
nonMinimal(X), decompose(X, HX, TX_1, \ldots, TX_t),
minimal(TX_1), minimal(TX_t),
process(HX, HY), compose(A, HY, NA),
r_{d_1}(TX_s, 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):

83
By rewriting clause 14 using applicability conditions (1) and (2):

\[ \text{clause 14: } r_d(Xs, Y, A) = \]
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \]
\[ \text{solv}(T X_1, T Y_1), \ldots, \text{solv}(T X_{p-1}, T Y_{p-1}), \]
\[ r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t), \]
\[ l_{t+1} = \epsilon, \]
\[ \text{compose}(T Y_1, l_{t+1}, l_1), \ldots, \text{compose}(T Y_p, l_{p+1}, l_p), \]
\[ \text{process}(H X, H H Y), \text{compose}(H H Y, l_p, l_{p-1}), \]
\[ \text{compose}(T Y_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(T Y_1, l_1, l_0), \]
\[ H Y = l_0, \]
\[ \text{compose}(A, H Y, N A), r_d(T X s, Y, N A) \]

By deleting one of the \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}) atoms in clause 14:

\[ \text{clause 15: } r_d(Xs, Y, A) = \]
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \]
\[ \text{solv}(T X_1, T Y_1), \ldots, \text{solv}(T X_{p-1}, T Y_{p-1}), \]
\[ r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t), \]
\[ l_{t+1} = \epsilon, \]
\[ \text{compose}(T Y_1, l_{t+1}, l_1), \ldots, \text{compose}(T Y_p, l_{p+1}, l_p), \]
\[ \text{process}(H X, H H Y), \text{compose}(H H Y, l_p, l_{p-1}), \]
\[ \text{compose}(T Y_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(T Y_1, l_1, l_0), \]
\[ H Y = l_0, \]
\[ \text{compose}(A, H Y, N A), r_d(T X s, Y, N A) \]

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

\[ \text{clause 16: } r_d(Xs, Y, A) = \]
\[ Xs = [X[T Xs], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, H X, T X_1, \ldots, T X_t), \]
\[ \text{minimal}(T X_1), \ldots, \text{minimal}(T X_{p-1}), \]
\[ (\text{nonMinimal}(T X_p); \ldots; \text{nonMinimal}(T X_t)), \]
\[ \text{solv}(T X_1, T Y_1), \ldots, \text{solv}(T X_{p-1}, T Y_{p-1}), \]
\[ r(T X_p, T Y_p), \ldots, r(T X_t, T Y_t) \]
\[ l_0 = \epsilon, \]
\[ \text{compose}(l_0, T Y_1, l_1), \ldots, \text{compose}(l_{p-2}, T Y_{p-1}, l_{p-1}), \]
\[ \text{process}(H X, H H Y), \text{compose}(l_{p-1}, H H Y, l_p), H Y = l_p, \]
\[ \text{compose}(A, H Y, N A), \]
\[ \text{compose}(T Y_p, T Y_{p+1}, l_{p+1}), \]
\[ \text{compose}(l_{p+1}, T Y_{p+2}, l_{p+2}), \ldots, \text{compose}(l_{t+1}, T Y_t, l_t), \]
\[ \text{compose}(N A, l_t, N N A), \]
\[ r_d(T X s, Y, N N A) \]

By \( t - p \) times folding clause 16 using clauses 1 and 2:
clause 17: $r \mathcal{D}_1(X, Y, A) = $

$X = [X' \cup X],$
nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_τ$),
minimal($TX_1$), \ldots, minimal($TX_{p-1}$),
(nonMinimal($TX_p$); \ldots; nonMinimal($TX_τ$)),
solve($TX_1, TY_1$), \ldots, solve($TX_{p-1}, TY_{p-1}$),
I_0 = ε,
compose($I_0, TY_1, I_1$), \ldots, compose($I_{p-2}, TY_{p-1}, I_{p-1}$),
process($HX, HHY$), compose($I_{p-1, HHY, I_p}$), HY = I_p,
compose(A, HY, NA),
$r \mathcal{D}_1([TX_p, \ldots, TX_τ, TX], Y, NA)$

By using applicability condition (3):

clause 18: $r \mathcal{D}_1(X, Y, A) = $

$X = [X' \cup X],$
nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_τ$),
minimal($TX_1$), \ldots, minimal($TX_{p-1}$),
(nonMinimal($TX_p$); \ldots; nonMinimal($TX_τ$)),
solve($TX_1, I_0$), \ldots, solve($TX_{p-1}, I_0$),
I_0 = ε,
compose($I_0, I_1$), \ldots, compose($I_{p-2, I_0, I_{p-1}}$),
process($HX, HHY$), compose($I_{p-1, HHY, I_p}$), HY = I_p,
compose(A, HY, NA),
$r \mathcal{D}_1([TX_p, \ldots, TX_τ, TX], Y, NA)$

By using applicability condition (2):

clause 19: $r \mathcal{D}_1(X, Y, A) = $

$X = [X' \cup X],$
nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_τ$),
minimal($TX_1$), \ldots, minimal($TX_{p-1}$),
(nonMinimal($TX_p$); \ldots; nonMinimal($TX_τ$)),
solve($TX_1, I_0, I_1$), \ldots, solve($TX_{p-1, I_0, I_{p-1}}$),
I_0 = ε,
I_1 = I_0, \ldots, I_{p-1} = I_{p-2},
process($HX, HHY$), compose($I_{p-1, HHY, I_p}$), HY = I_p,
compose(A, HY, NA),
$r \mathcal{D}_1([TX_p, \ldots, TX_τ, TX], Y, NA)$

By simplification:

clause 20: $r \mathcal{D}_1(X, Y, A) = $

$X = [X' \cup X],$
nonMinimal($X$), decompose($X, HX, TX_1, \ldots, TX_τ$),
minimal($TX_1$), \ldots, minimal($TX_{p-1}$),
(nonMinimal($TX_p$); \ldots; nonMinimal($TX_τ$)),
process($HX, HY$), compose(A, HY, NA),
$r \mathcal{D}_1([TX_p, \ldots, TX_τ, TX], Y, NA)$

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: $r \mathcal{A} d_1 (X_s, Y, A) -$ 

$X_s = [X[TX_s],$
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ..., nonMinimal(TX_{p-1})),
minimal(TX_p), ..., minimal(TX_t),
minimal(U_1), ..., minimal(U_{p-1}),
r(TX_1, TY_1), ..., r(TX_t, TY_t),
$h_{t+1} = e,$
compose(TY_t, HX, HX, TX_1, ..., TX_t),
process(HX, HY, HY), compose(HHY, I_p, I_{p-1}),
compose(TY_{p-1}, I_p, I_{p-2}), ..., compose(TY_1, I_1, I_0),
HY = I_0,
compose(A, HY, NA), r \mathcal{A} d_1 (TX_s, Y, NA)$

By using applicability condition (3):

clause 22: $r \mathcal{A} d_1 (X_s, Y, A) -$ 

$X_s = [X[TX_s],$
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ..., nonMinimal(TX_{p-1})),
minimal(TX_p), ..., minimal(TX_t),
minimal(U_1), ..., minimal(U_{p-1}),
r(U_1, e), ..., r(U_{p-1}, e),
r(TX_1, TY_1), ..., r(TX_t, TY_t),
$h_{t+1} = e,$
compose(TY_t, HX, HX, TX_1, ..., TX_t),
process(HX, HY, HY), compose(HHY, I_p, I_{p-1}),
compose(TY_{p-1}, I_p, I_{p-2}), ..., compose(TY_1, I_1, I_0),
HY = I_0,
compose(A, HY, NA), r \mathcal{A} d_1 (TX_s, Y, NA)$

By using applicability condition (2):

clause 23: $r \mathcal{A} d_1 (X_s, Y, A) -$ 

$X_s = [X[TX_s],$
nonMinimal(X), decompose(X, HX, TX_1, ..., TX_t),
(nonMinimal(TX_1); ..., nonMinimal(TX_{p-1})),
minimal(TX_p), ..., minimal(TX_t),
minimal(U_1), ..., minimal(U_{p-1}),
r(U_1, e), ..., r(U_{p-1}, e),
r(TX_1, TY_1), ..., r(TX_t, TY_t),
$h_{t+1} = e,$
compose(TY_t, HX, HX, TX_1, ..., TX_t),
process(HX, HY, HY), compose(HHY, I_p, I_{p-1}),
compose(e, I_p, I_{p-2}), ..., compose(e, I_1, I_0),
HY = I_0,
compose(A, HY, NA), r \mathcal{A} d_1 (TX_s, Y, NA)$

By using applicability conditions (1) and (2):
clause 24: \[ r_{\text{jd}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s]], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ r(U_1, e), \ldots, r(U_{p-1}, e), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_{t+1} = e, \]
\[ \text{compose}(TY_1, l_{t+1}, l_1), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \]
\[ \text{compose}(e, l_{p-1}, l_{p-2}), \ldots, \text{compose}(e, l_1, l_0), \]
\[ HY = l_0, \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, HY, NNA), \]
\[ r_{\text{jd}_1}(TX_s, Y, Y, NNA) \]

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

clause 25: \[ r_{\text{jd}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s]], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_{t+1} = e, \]
\[ \text{compose}(TY_1, l_{t+1}, l_1), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \]
\[ \text{compose}(YU_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(YU_1, l_1, l_0), \]
\[ HY = l_0, \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, HY, NNA), \]
\[ r_{\text{jd}_1}(TX_s, Y, Y, NNA) \]

By introducing \( \text{nonMinimal}(N) \) and \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t) \), since

\[ \exists N : X. \text{nonMinimal}(N) \land \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t) \]

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

clause 26: \[ r_{\text{jd}_1}(X_s, Y, A) = \]
\[ X_s = [X[T X_s]], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_{p-1}), \]
\[ r(U_1, YU_1), \ldots, r(U_{p-1}, YU_{p-1}), \]
\[ \text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ l_{t+1} = e, \]
\[ \text{compose}(TY_1, l_{t+1}, l_1), \ldots, \text{compose}(TY_p, l_{p+1}, l_p), \]
\[ \text{process}(HX, HHY), \text{compose}(HY, l_p, l_{p-1}), \]
\[ \text{compose}(YU_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(YU_1, l_1, l_0), \]
\[ HY = l_0, \]
\[ \text{compose}(TY_1, TY_2, K_1), \text{compose}(K_1, TY_3, K_2), \ldots, \text{compose}(K_{p-3}, TY_{p-1}, K_{p-2}), \]
\[ \text{compose}(A, K_{p-2}, NA), \text{compose}(NA, HY, NNA), \]
\[ r_{\text{jd}_1}(TX_s, Y, Y, NNA) \]

By duplicating goal \( \text{decompose}(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t) \):
\textbf{clause 27:} \( r_{\mathcal{D}_1}(Xs,Y,A) = \)

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

nonMinimal(X), decompose(X, \( HX, TX_1, \ldots, TX_t \)),

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

minimal(TX_p), \ldots, minimal(TX_t),

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

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

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

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

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

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

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

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

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

\( \)HY = \( I_0 \),

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

\( \)compose(\( A, K_{p-2}, NA \)), \( \)compose(\( NA, HY, NNA \)),

\( r_{\mathcal{D}_1}(TXs,Y,NNA) \)

By folding clause 27 using \( DC/RI \):

\textbf{clause 28:} \( r_{\mathcal{D}_1}(Xs,Y,A) = \)

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

nonMinimal(X), decompose(X, \( HX, TX_1, \ldots, TX_t \)),

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

minimal(TX_p), \ldots, minimal(TX_t),

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

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

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

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

\( \)compose(\( A, K_{p-2}, NA \)), \( \)compose(\( NA, HY, NNA \)),

\( r_{\mathcal{D}_1}(TXs,Y,NNA) \)

By folding clause 28 using clauses 1 and 2:

\textbf{clause 29:} \( r_{\mathcal{D}_1}(Xs,Y,A) = \)

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

nonMinimal(X), decompose(X, \( HX, TX_1, \ldots, TX_t \)),

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

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(\( A, K_{p-2}, NA \)), \( \)compose(\( NA, HY, NNA \)),

\( r_{\mathcal{D}_1}([TXs],Y,NNA) \)

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

\textbf{clause 30:} \( r_{\mathcal{D}_1}(Xs,Y,A) = \)

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

nonMinimal(X), decompose(X, \( HX, TX_1, \ldots, TX_t \)),

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

minimal(TX_p), \ldots, minimal(TX_t),

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

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

\( r_{\mathcal{D}_1}([TX_1, \ldots, TX_{p-1}, N[TXs],Y,A] \)

By introducing atoms minimal(U_1), \ldots, minimal(U_t) (with new, i.e. existentially quantified, variables \( U_1, \ldots, U_t \)) in clause 8:
By using applicability condition (1):

\[
\text{clause 31: } r \text{d}_1(X_s, Y, A) =
\]
\[
X_s = [X[TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
\]
\[
\text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_t),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
h_{t+1} = e,
\]
\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]
\[
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}),
\]
\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0),
\]
\[
HY = l_0
\]
\[
\text{compose}(A, HY, NA), r \text{d}_1(TX_s, Y, NA)
\]

By using applicability condition (3):

\[
\text{clause 32: } r \text{d}_1(X_s, Y, A) =
\]
\[
X_s = [X[TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
\]
\[
\text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_t),
\]
\[
r(U_1, e), \ldots, r(U_t, e),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
h_{t+1} = e,
\]
\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]
\[
\text{process}(HX, HHY), \text{compose}(HY, I_p, I_{p-1}),
\]
\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0),
\]
\[
HY = l_0
\]
\[
\text{compose}(A, HY, NA), r \text{d}_1(TX_s, Y, NA)
\]

By using applicability condition (2):

\[
\text{clause 33: } r \text{d}_1(X_s, Y, A) =
\]
\[
X_s = [X[TX_s],
\]
\[
\text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t),
\]
\[
\text{(nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})),
\]
\[
\text{(nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)),
\]
\[
\text{minimal}(U_1), \ldots, \text{minimal}(U_t),
\]
\[
r(U_1, e), \ldots, r(U_t, e),
\]
\[
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),
\]
\[
h_{t+1} = e,
\]
\[
\text{compose}(TY_1, I_{t+1}, I_t), \ldots, \text{compose}(TY_p, I_{p+1}, I_p),
\]
\[
\text{compose}(e, I_p, K_{i+1}),
\]
\[
\text{compose}(e, K_{i+1}, K_t), \ldots, \text{compose}(e, K_{p+1}, K_p),
\]
\[
\text{process}(HX, HHY), \text{compose}(HY, K_p, K_{p-1}),
\]
\[
\text{compose}(e, K_{p-1}, K_{p-2}), \ldots, \text{compose}(e, K_1, K_0),
\]
\[
\text{compose}(e, K_0, I_{p-1}),
\]
\[
\text{compose}(TY_{p-1}, I_{p-1}, I_{p-2}), \ldots, \text{compose}(TY_1, I_1, I_0),
\]
\[
HY = l_0
\]
\[
\text{compose}(A, HY, NA), r \text{d}_1(TX_s, Y, NA)
\]

By using applicability conditions (1) and (2):
\begin{align*}
\text{clause 34:} \quad r_{\mathcal{D}_1}(X, Y, A) = \\
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)), \\
\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}(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), \\
NY = 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}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \\
\text{compose}(NA_2, K_{t-1}, NA), r_{\mathcal{D}_1}(TX_s, Y, NA)
\end{align*}

By introducing new, i.e. existentially quantified, variables \(Yu_1, \ldots, Yu_t\) in place of some occurrences of \(e\):

\begin{align*}
\text{clause 35:} \quad r_{\mathcal{D}_1}(X, Y, A) = \\
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)), \\
\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), \\
h_{t+1} = \epsilon, \\
\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}(Yu_0, I_{p-1}, I_{p-2}), \ldots, \text{compose}(Yu_1, I_1, I_0), \\
NY = 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}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \\
\text{compose}(NA_2, K_{t-1}, NA), r_{\mathcal{D}_1}(TX_s, Y, NA)
\end{align*}

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):

\begin{align*}
\text{clause 36:} \quad r_{\mathcal{D}_1}(X, Y, A) = \\
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)), \\
\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), \\
h_{t+1} = \epsilon, \\
\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}(YU_0, I_{p-1}, I_{p-2}), \ldots, \text{compose}(YU_1, I_1, I_0), \\
NY = 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}(A, K_{p-2}, NA_1), \text{compose}(NA_1, NHY, NA_2), \\
\text{compose}(NA_2, K_{t-1}, NA), r_{\mathcal{D}_1}(TX_s, Y, NA)
\end{align*}
By duplicating goal $\text{decompose}(N, HX, U_1, \ldots, U_t)$:

**clause 37**:  $r\mathcal{J}_d (Xs, Y, A) = $

\[ Xs = [X[T 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, Y U_1), \ldots, r(U_t, Y U_t),
\text{nonMinimal}(N), \text{decompose}(N, HX, U_1, \ldots, U_t),
r(TX_1, TY_1), \ldots, r(TX_t, TY_t),$

\[ l_{t+1} = e,
\text{compose}(Y U_1, l_{t+1}, l_t), \ldots, \text{compose}(Y U_p, l_{p+1}, l_p),
\text{process}(HX, H HY), \text{compose}(H HY, l_p, l_{p-1}),
\text{compose}(Y U_{p-1}, l_{p-1}, l_{p-2}), \ldots, \text{compose}(Y U_1, l_1, l_0),
N HY = l_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}(A, K_{p-2}, N A_1), \text{compose}(N A_1, N HY, N A_2),
\text{compose}(N A_2, K_{t-1}, N A), r\mathcal{J}_d (T Xs, Y, N A)$

By folding clause 37 using $DCRL$:

**clause 38**:  $r\mathcal{J}_d (Xs, Y, A) = $

\[ Xs = [X[T 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),
\text{decompose}(N, HX, U_1, \ldots, U_t),
r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, 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}(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}, N A_1), \text{compose}(N A_1, N HY, N A_2),
\text{compose}(N A_2, K_{t-1}, N A), r\mathcal{J}_d (T Xs, Y, N A)$

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

**clause 39**:  $r\mathcal{J}_d (Xs, Y, A) = $

\[ Xs = [X[T 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),
\text{decompose}(N, HX, U_1, \ldots, U_t),
r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, 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}, N A_1), \text{compose}(N A_1, N HY, N A_2),
\text{compose}(N A_2, N A_1, N A_2), r\mathcal{J}_d (T Xs, Y, N A_2)$

By folding clause 39 using clauses 1 and 2:
and from which we try to obtain T D GLR. 

Theorem 5: Proofs of the Duality Schemas

To prove that \( P_r \) is steadfast wrt \( S_r \), 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:

\[
r(X, Y) \equiv r_{d_1}(X, Y, \epsilon)
\]

By taking the ‘completion’:

\[
\forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \Leftrightarrow r_{d_1}(X, Y, \epsilon)]
\]

By unfolding the ‘completion’ above wrt \( r_{d_1}(X, Y, \epsilon) \) using \( S_{d_1} \):

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

By using applicability condition (2):

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

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 \( \{S_{d_1}\} \).

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

5 Proofs of the Duality Schemas

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

\[
D_{dc} : \{ DCLR, DCRL, A_{dce}, O_{dce1}, O_{dce2} \}
\]

where

\[
A_{dce} : (1) \text{ compose is associative}
\]

(2) compose has \( \epsilon \) as the left and right identity element, where \( \epsilon \) appears in \( DCLR \) and \( DCRL \).

\[
O_{dce1} : \text{ partial evaluation of the conjunction}
\]

\[
\text{process} (HX, HY), \text{compose}(HY, \ I_1, \ Y_{1-1})
\]

results in the introduction of a non-recursive relation

\[
O_{dce2} : \text{ partial evaluation of the conjunction}
\]

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

results in the introduction of a non-recursive relation
where the template \( \text{DCLR} \) is Logic Program Template 1 in Section 2 and the template \( \text{DCRL} \) is Logic Program Template 3 in Section 3.

The specification \( S_r \) of both a \( \text{DCLR} \) program and a \( \text{DCRL} \) program for relation \( r \) is:

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

**Proof 11** To prove the correctness of the duality schema \( \text{Ddc} \), by Definition 10, we have to prove that templates \( \text{DCLR} \) and \( \text{DCRL} \) are equivalent wrt \( S_r \) under the applicability conditions \( \text{Addc} \). By Definition 5, the templates \( \text{DCLR} \) and \( \text{DCRL} \) are equivalent wrt \( S_r \) under the applicability conditions \( \text{Addc} \) iff \( \text{DCLR} \) is equivalent to \( \text{DCRL} \) wrt the specification \( S_r \) provided that the conditions in \( \text{Addc} \) hold. By Definition 4, \( \text{DCLR} \) is equivalent to \( \text{DCRL} \) wrt the specification \( S_r \) if the following two conditions hold:

(a) \( \text{DCLR} \) is steadfast wrt \( S_r \) in \( S = \{S_{\text{minimum}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}}\} \), where \( S_{\text{minimum}}, S_{\text{nonMinimal}}, S_{\text{solve}}, S_{\text{decompose}}, S_{\text{process}}, S_{\text{compose}} \) 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 \( \text{DCLR} \).

(b) \( \text{DCRL} \) 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 duality transformation of \( P \).

In program transformation, we assume that the input program, here template \( \text{DCLR} \), 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: \( \text{DCLR} \) is steadfast wrt \( S_r \) in \( S \) iff \( \text{DCLR} \cup P_s \) 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 \( \text{DCLR} \) 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 \( \text{DCRL} \).

By taking the ‘decompletion’ of \( S_r \):

**clause 1:** \( r(X,Y) \iff r(X,Y) \)

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

**clause 2:** \( r(X,Y) \iff \text{minimal}(X), \text{solve}(X,Y) \)

**clause 3:** \( r(X,Y) \iff \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1, \ldots, TX_t), \text{compose}(l_0, TY_1, l_1), \ldots, \text{compose}(l_{t-2}, TY_{t-1}, l_{t-1}), \text{process}(HX, HY), \text{compose}(l_{t-1}, HY, l_t), \text{compose}(l_t, TY_t, l_{t+1}), \ldots, \text{compose}(l_t, TY_t, l_{t+1}), Y = l_{t+1} \)

By using applicability condition (1) on clause 3:

**clause 4:** \( r(X,Y) \iff \text{nonMinimal}(X), \text{decompose}(X,HX,TX_1, \ldots, TX_t), \text{compose}(TY_{t-1}, TY_1, A_{t-1}), \text{compose}(TY_{t-2}, A_{t-1}, A_{t-2}), \ldots, \text{compose}(TY_p, A_{p+1}, A_p), \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \text{compose}(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TY_1, A_1, A_0), \text{compose}(e, A_0, Y) \)

By using applicability conditions (1) and (2) on clause 4:
clause 5: \[ r(XY) \leftarrow \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), \]
\[ \text{compose}(TY_1, e, A_t), \]
\[ \text{compose}(TY_{t+1}, A_t, A_{t+1}), \ldots, \text{compose}(TY_p, A_{p+1}, A_p), \]
\[ \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \]
\[ \text{compose}(TY_{p-1}, A_{p-1}, A_{p-2}), \ldots, \text{compose}(TY_1, A_1, A_0), \]
\[ Y = A_0 \]

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

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

Clauses 2 and 6 are the clauses of DCRL.

Therefore DCRL is steadfast wrt \( S_r \) in \( S \).

\[ \square \]

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

\[ D_{dg} : \{ \text{DGLR, DGRL, A}_{ddg}, O_{ddg12}, O_{ddg21} \} \]

where

\[ A_{ddg} : (1) \text{ compose is associative} \]
\[ (2) \text{ compose has } e \text{ as the left and right identity element,} \]

\[ O_{ddg12} : I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e) \]
- 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

\[ O_{ddg21} : I_r(X) \land \text{minimal}(X) \Rightarrow O_r(X, e) \]
- partial evaluation of the conjunction
\[ \text{process}(HX, HY), \text{compose}(A_{p-1}, HY, A_p) \]
results in the introduction of a non-recursive relation

and the templates DGLR and DGRL are Logic Program Templates 4 and 5 in Section 3.

The specification \( S_r \) of both a DGLR program and a DGRL program for relation \( r \) is:

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

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 \( D_{dg} \), by Definition 10, we have to prove that templates DGLR and DGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{ddg} \). By Definition 5, the templates DGLR and DGRL are equivalent wrt \( S_r \) under the applicability conditions \( A_{ddg} \) if DGLR is equivalent to DGRL wrt the specification \( S_r \) provided that the conditions in \( A_{ddg} \) hold. By Definition 4, DGLR is equivalent to DGRL wrt the specification \( S_r \) if the following two conditions hold:

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

(b) DGRL is also steadfast wrt \( S_r \) in \( S \).
Note that the sets \( \{ S_1, \ldots, S_m \} \) and \( \{ S'_1, \ldots, S'_r \} \) 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 \( S_\tau \) 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 \( S_\tau \) in \( S \) if \( P_{\text{descending}_2} \) is steadfast wrt \( S_{\text{descending}_2} \) in \( S \), where \( P_{\text{descending}_2} \) is the procedure for \( r_{\text{descending}_2} \), and \( P_r \) is steadfast wrt \( S_r \) in \( \{ S_{\text{descending}_2} \} \), where \( P_r \) is the procedure for \( r \).

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

By taking the ‘decompletion’ of \( S_{\text{descending}_2} \):

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

By unfolding clause 1 wrt \( r(X, S) \) using \( DGLR \), and using the assumption that \( DGLR \) is steadfast wrt \( S_\tau \) in \( S \):

\[
\text{clause 2 : } r_{\text{descending}_2}(X, Y, A) \rightarrow r_{\text{descending}_1}(X, S, \epsilon), \text{compose}(S, A, Y)
\]

By unfolding clause 2 wrt \( r_{\text{descending}_1}(X, S, \epsilon) \) using \( DGLR \), and using the assumption that \( P_{\text{descending}_1} \) is steadfast wrt \( S_{\text{descending}_1} \) in \( S \), since \( P_{\text{descending}_1} \) must be steadfast wrt \( S_{\text{descending}_1} \) in \( S \) for \( DGLR \) to be steadfast wrt \( S_\tau \) in \( S \):

\[
\text{clause 3 : } r_{\text{descending}_2}(X, Y, A) \rightarrow \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, S'), \text{compose}(\epsilon, S', S), \text{compose}(S, A, Y)
\]

\[
\text{clause 4 : } r_{\text{descending}_2}(X, Y, A) \rightarrow \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad \text{compose}(\epsilon, S, A_1), \\
\quad r_{\text{descending}_1}(TX_1, A_1, A_1), \ldots, r_{\text{descending}_1}(TX_{t-1}, A_{p-1}, A_{p-2}), \\
\quad \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \\
\quad r_{\text{descending}_1}(TX_p, A_{p+1}, A_p), \ldots, r_{\text{descending}_1}(TX_1, A_{t+1}, A_t), \\
\quad S = A_{t+1}, \text{compose}(S, A, Y)
\]

By using applicability condition (2) on clause 3:

\[
\text{clause 5 : } r_{\text{descending}_2}(X, Y, A) \rightarrow \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, S'), S = S, \text{compose}(S, A, Y)
\]

By simplification:

\[
\text{clause 6 : } r_{\text{descending}_2}(X, Y, A) \rightarrow \\
\quad \text{minimal}(X), \\
\quad \text{solve}(X, S), \text{compose}(S, A, Y)
\]

By \( t \) times unfolding clause 4 wrt the atoms \( r_{\text{descending}_1}(TX_1, A_1, A_1), \ldots, r_{\text{descending}_1}(TX_{t-1}, A_{p-1}, A_{p-2}) \), \( r_{\text{descending}_1}(TX_p, A_{p+1}, A_p), \ldots, r_{\text{descending}_1}(TX_1, A_{t+1}, A_t) \) using the ‘decompletion’ of \( S_{\text{descending}_1} \) (refer to Proofs 3 and 6):

\[
\text{clause 7 : } r_{\text{descending}_2}(X, Y, A) \rightarrow \\
\quad \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\quad \text{compose}(\epsilon, S, A_1), \\
\quad r(TX_1, TS_1), \ldots, r(TX_{p-1}, TS_{p-1}), \\
\quad \text{compose}(A_{p-1}, TS_{p-1}, A_{p-1}), \ldots, \text{compose}(A_{p-2}, TS_{p-1}, A_{p-2}), \\
\quad \text{process}(HX, HS), \text{compose}(A_{p-1}, HS, A_p), \\
\quad r(TX_p, TS_p), \ldots, r(TX_1, TS_1), \\
\quad \text{compose}(A_p, TS_p, A_{p+1}), \ldots, \text{compose}(A_t, TS_t, A_{t+1}), \\
\quad S = A_{t+1}, \text{compose}(S, A, Y)
\]

By using applicability condition (1) on clause 7:
\[ \text{clause 8: } r_{\text{descending}}(X, Y, A) = \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{compose}(e, I, Y), \text{compose}(e, A_0, I), \]
\[ r(TX_1, TS_1), \ldots, r(TX_{t-1}, TS_{t-1}), \]
\[ \text{compose}(TS_1, A_1, A_0), \ldots, \text{compose}(TS_{t-1}, A_{t-1}, A_{t-2}), \]
\[ \text{process}(HX, HS), \text{compose}(HS, A_p, A_{p-1}), \]
\[ r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \]
\[ \text{compose}(TS_p, A_{p+1}, A_p), \ldots, \text{compose}(TS_t, A_{t+1}, A_t), \]
\[ \text{compose}(e, A, A_{t+1}) \]

By using applicability condition (2):

\[ \text{clause 9: } r_{\text{descending}}(X, Y, A) = \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ Y = A_0, \]
\[ r(TX_1, TS_1), \ldots, r(TX_{t-1}, TS_{t-1}), \]
\[ \text{compose}(TS_1, A_1, A_0), \ldots, \text{compose}(TS_{t-1}, A_{t-1}, A_{t-2}), \]
\[ \text{process}(HX, HS), \text{compose}(HS, A_p, A_{p-1}), \]
\[ r(TX_p, TS_p), \ldots, r(TX_t, TS_t), \]
\[ \text{compose}(TS_p, A_{p+1}, A_p), \ldots, \text{compose}(TS_t, A_{t+1}, A_t), \]
\[ \text{compose}(e, A, A_{t+1}) \]

By \( t \) times folding clause 9 using clause 1:

\[ \text{clause 10: } r_{\text{descending}}(X, Y, A) = \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ \text{compose}(e, A, A_{t+1}), \]
\[ r_{\text{descending}}(TX_1, A_t, A_{t+1}), \ldots, r_{\text{descending}}(TX_p, A_p, A_{p+1}), \]
\[ \text{process}(HX, HY), \text{compose}(HY, A_p, A_{p-1}), \]
\[ r_{\text{descending}}(TX_{p+1}, A_{p+2}, A_{p+1}), \ldots, r_{\text{descending}}(TX_1, A_0, A_1), \]
\[ Y = A_0 \]

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:

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

By taking the ‘completion’:

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

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

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

By using applicability condition (2):

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

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 \( \{ S_{r_{\text{descending}}} \} \).

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

**Theorem 13** The duality schema \( D_{\text{dgr}} \), which is given below, is correct.

\[ D_{\text{dgr}} : \{ T \text{DGRL}, T \text{DGRL}, A_{\text{dgr}}, O_{\text{dgr}}, O_{\text{dgr}} \} \text{ where } \]
\[ A_{\text{dgr}} : (1) \text{ compose is associative} \]
\[ (2) \text{ compose has } e \text{ as the left and right identity element}. \]
where \( e \) appears in \( TDGLR \) and \( TDGRL \).

\[ O_{\text{atdg12}} \rightarrow \forall X : \mathcal{X} \cdot I_r(X) \wedge \text{minimal}(X) \Rightarrow O_r(X, e) \]
- partial evaluation of the conjunction

\[ \text{process}(I X, HY), \text{compose}(HY, \text{NewA}, F) \]
results in the introduction of a non-recursive relation

\[ O_{\text{atdg21}} \rightarrow \forall X : \mathcal{X} \cdot I_r(X) \wedge \text{minimal}(X) \Rightarrow O_r(X, e) \]
- partial evaluation of the conjunction

\[ \text{process}(I X, HY), \text{compose}(A, HY, \text{NewA}) \]
results in the introduction of a non-recursive relation

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

The specification \( S_r \) of relation \( r \) is:

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

The specification of \( r_{\text{atd1}} \), namely \( S_{r_{\text{atd1}}} \), is:

\[ \forall X, s : \text{list of} \mathcal{X}, Y, A : \mathcal{Y} \cdot (\forall X : \mathcal{X} \cdot \text{X} \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{atd1}}(Xs, Y, A) \Leftrightarrow (Xs = [] \wedge Y = A) \wedge O_r(A, I_0, I_{q+1}) \wedge Y = I_{q+1}] \]

The specification of \( r_{\text{atd2}} \), namely \( S_{r_{\text{atd2}}} \), is:

\[ \forall X, s : \text{list of} \mathcal{X}, Y, A : \mathcal{Y} \cdot (\forall X : \mathcal{X} \cdot \text{X} \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{atd2}}(Xs, Y, A) \Leftrightarrow (Xs = [] \wedge Y = A) \wedge O_r(A, I_0, I_{q+1}) \wedge Y = I_{q+1}] \]

**Proof 13** To prove the correctness of the duality schema \( D_{\text{atd1}} \), by Definition 10, we have to prove that templates \( TDGLR \) and \( TDGRL \) are equivalent wrt \( S_r \) under the applicability conditions \( \text{A_{atdg}} \).

By Definition 5, the templates \( TDGLR \) and \( TDGRL \) are equivalent wrt \( S_r \), under the applicability conditions \( \text{A_{atdg}} \).

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) \( TDGRL \) 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 duality transformation of \( P \).

In program transformation, we assume that the input program, here template \( TDGRL \), 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 \( P_{r_{\text{atd}}} \) is steadfast wrt \( S_{r_{\text{atd}}} \) in \( S \), where \( P_{r_{\text{atd}}} \) is the procedure for \( r_{\text{atd1}} \), and \( P_r \) is the procedure for \( r \).

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

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

\[ \forall X, s : \text{list of} \mathcal{X}, Y, A : \mathcal{Y} \cdot (\forall X : \mathcal{X} \cdot \text{X} \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{atd}}(Xs, Y, A) \Leftrightarrow (Xs = [] \wedge Y = A) \wedge O_r(A, I_0, I_{q+1}) \wedge Y = I_{q+1}] \]

where \( q \geq 2 \).

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

\[ \forall X, s : \text{list of} \mathcal{X}, Y, A : \mathcal{Y} \cdot (\forall X : \mathcal{X} \cdot \text{X} \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{atd}}(Xs, Y, A) \Leftrightarrow (Xs = [] \wedge Y = A) \wedge O_r(A, I_0, I_{q+1}) \wedge Y = I_{q+1}] \]

\[ \forall X, s : \text{list of} \mathcal{X}, Y, A : \mathcal{Y} \cdot (\forall X : \mathcal{X} \cdot \text{X} \in Xs \Rightarrow I_r(X)) \Rightarrow [r_{\text{atd}}(Xs, Y, A) \Leftrightarrow (Xs = [] \wedge Y = A) \wedge O_r(A, I_0, I_{q+1}) \wedge Y = I_{q+1}] \]

97
where $q \geq 2$.

By folding using $S_r \downarrow d_2$, and renaming:

\[ \forall X : \text{list of } \forall Y : \forall Y. \quad (\forall X : \forall X. \quad X \in X s \Rightarrow T_r(X)) \Rightarrow [r \downarrow d_2(X s, Y, A) \Leftrightarrow (X s = [[] \wedge Y = A) \wedge (X s = [X[T X s] \wedge O_r(X, HY) \wedge r \downarrow d_2(T X s, NA, A) \wedge O_r(HY, NA, Y)])]

By taking the ‘decompletion’:

\begin{align*}
\text{clause 1} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [[], Y = A \\
\text{clause 2} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [X[T X s], r(X, HY), \\
& r \downarrow d_2(T X s, NA, A), \text{compose}(HY, NA, Y)] \\
\text{clause 3} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [X[T X s], r \downarrow d l([X], HY, e), \\
& r \downarrow d_2(T X s, NA, A), \text{compose}(HY, NA, Y)] \\
\text{clause 4} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [X[T X s], \\
& X s' = [X[T X s'], TX s' = [], \\
& \text{minimal}(X), \text{solve}(X, HY'), \\
& \text{compose}(e, HY', NA'), r \downarrow d l(T X s', HY, NA'), \\
& r \downarrow d_2(T X s, NA, A), \text{compose}(HY, NA, Y)] \\
\text{clause 5} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [X[T X s], \\
& X s' = [X[T X s'], TX s' = [], \\
& \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
& \text{minimal}(TX_1), \ldots, \text{minimal}(TX_t), \\
& \text{process}(HX, HY'), \text{compose}(e, HY', NA'), \\
& r \downarrow d l(T X s', HY, NA'), \\
& r \downarrow d_2(T X s, NA, A), \text{compose}(HY, NA, Y)] \\
\text{clause 6} \quad & r \downarrow d_2(X s, Y, A) \leftarrow X s = [X[T X s], \\
& X s' = [X[T X s'], 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}(e, HY', NA'), \\
& r \downarrow d l(T X s, \ldots, TX t[T X s'], HY, NA'), \\
& r \downarrow d_2(T X s, NA, A), \text{compose}(HY, NA, Y)]
\end{align*}
\textbf{clause 7:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\ns' = \lfloor X \rfloor X s', \quad TX s' = [\], \\
\text{nonMinimal}(X), \quad \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), \\
\text{r} \downarrow d_1([T X_1, \ldots, T X_{p-1}, N, T X_p, \ldots, T X_t] X s', \quad \text{TY}, \quad \epsilon), \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By unfolding clause 4 wrt \( \text{r} \downarrow d_1(T X s', \quad \text{HY}, \quad \text{NA}') \) using \( \text{TDGLR} \), and using the assumption that \( P_{\text{r} \downarrow d_i} \) is steadfast wrt \( \lfloor S, d_i \rfloor \) in \( S \):

\textbf{clause 8:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\ns' = \lfloor X \rfloor X s', \quad TX s' = [\], \\
\text{nonMinimal}(X), \quad \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_{p-1}), \\
\text{decompose}(N, H X, U_1, \ldots, U_{p-1}), \\
\text{r} \downarrow d_1([T X_1, \ldots, T X_{p-1}, N, T X_p, \ldots, T X_t] X s', \quad \text{HY}, \quad \epsilon), \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By using applicability condition (2):

\textbf{clause 9:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\ns' = \lfloor X \rfloor X s', \quad TX s' = [\], \\
\text{minimal}(X), \quad \text{solve}(X, \quad \text{HY}'), \\
\text{compose}(\epsilon, \quad \text{HY}', \quad \text{NA}'), \\
\text{TX s'} = [\], \quad \text{HY} = \quad \text{NA}', \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By simplification:

\textbf{clause 10:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\ns' = \lfloor X \rfloor X s', \quad TX s' = [\], \\
\text{minimal}(X), \quad \text{solve}(X, \quad \text{HY}'), \\
\text{HY'} = \quad \text{NA}', \\
\text{TX s'} = [\], \quad \text{HY} = \quad \text{NA}', \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By unfolding clause 5 wrt \( \text{r} \downarrow d_1(T X s', \quad \text{HY}, \quad \text{NA}') \) using \( \text{TDGLR} \), and using the assumption that \( P_{\text{r} \downarrow d_i} \) is steadfast wrt \( \lfloor S, d_i \rfloor \) in \( S \):

\textbf{clause 11:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\n\text{minimal}(X), \quad \text{solve}(X, \quad \text{HY}'), \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By using applicability condition (2):

\textbf{clause 12:} \( r \downarrow d_2(Xs, Y, A) \leftarrow \\
s = \lfloor X \rfloor X s, \\\ns' = \lfloor X \rfloor X s', \quad TX s' = [\], \\
\text{nonMinimal}(X), \quad \text{decompose}(X, \quad \text{HX}, \quad \text{TX}_1, \ldots, \quad \text{TX}_t), \\
\text{minimal}(\text{TX}_1), \ldots, \quad \text{minimal}(\text{TX}_t), \\
\text{process}(\text{HX}, \quad \text{HY}'), \quad \text{compose}(\epsilon, \quad \text{HY}', \quad \text{NA}'), \\
\text{TX s'} = [\], \quad \text{HY} = \quad \text{NA}', \\
\text{r} \downarrow d_2(T X s, N A, A), \quad \text{compose}(\text{HY}, \quad \text{NA}, \quad \text{YA}) \)

By using applicability condition (2):
clause 13: \( r_{\text{j}d_2}(Xs, Y, A) = \)
\[
\begin{align*}
Xs & = [X[TXs], \\
Xs' & = [X'[TXs'], TXs' = [], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), & \ldots, \text{minimal}(TX_t), \\
\text{process}(HX, HY'), & HY' = NA', \\
TXs' & = [], HY = NA', \\
r_{\text{j}d_2}(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}
\]

By simplification:

clause 14: \( r_{\text{j}d_2}(Xs, Y, A) = \)
\[
\begin{align*}
Xs & = [X[TXs], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), & \ldots, \text{minimal}(TX_t), \\
r_{\text{j}d_2}(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}
\]

By \( t \) times unfolding clause 6 wrt
\[
r_{\text{j}d_1}([TX_p, \ldots, TX_t][TXs'], HX, NA'), \ldots, r_{\text{j}d_1}([TX_t[TXs'], HX, NA_{t-1})
\]

using the “decomposition” of \( S_{r_{\text{j}d_1}} \) in Section 4:

clause 15: \( r_{\text{j}d_2}(Xs, Y, A) = \)
\[
\begin{align*}
Xs & = [X[TXs], \\
Xs' & = [X'[TXs'], TXs' = [], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), & \ldots, \text{minimal}(TX_{t-1}), \\
\text{process}(HX, HY'), & \text{compose}(e, HY', NA'), \\
r(TX_p, TY_p), & \ldots, r(TX_t, TY_t), \\
\text{compose}(NA', TY_p, NA'_p), & \ldots, \text{compose}(NA_{t-1}, TY_t, NA'_t), \\
r_{\text{j}d_1}(TXs', HY, NA_t), \\
r_{\text{j}d_2}(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}
\]

By unfolding clause 15 wrt \( r_{\text{j}d_1}(TXs', HY, NA_t) \) using \( TDGLR \), and using the assumption that \( P_{r_{\text{j}d_1}} \) is steadfast wrt \( S_{r_{\text{j}d_1}} \) in \( S' \):

clause 16: \( r_{\text{j}d_2}(Xs, Y, A) = \)
\[
\begin{align*}
Xs & = [X[TXs], \\
Xs' & = [X'[TXs'], TXs' = [], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), & \ldots, \text{minimal}(TX_{t-1}), \\
\text{process}(HX, HY'), & \text{compose}(e, HY', NA'), \\
r(TX_p, TY_p), & \ldots, r(TX_t, TY_t), \\
\text{compose}(NA', TY_p, NA'_p), & \ldots, \text{compose}(NA_{t-1}, TY_t, NA'_t), \\
TXs' & = [], HY = NA_t, \\
r_{\text{j}d_2}(TXs, NA, A), & \text{compose}(HY, NA, Y)
\end{align*}
\]

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

clause 17: \( r_{\text{j}d_2}(Xs, Y, A) = \)
\[
\begin{align*}
Xs & = [X[TXs], \\
\text{nonMinimal}(X), & \text{decompose}(X, HX, TX_1, \ldots, TX_t), \\
\text{minimal}(TX_1), & \ldots, \text{minimal}(TX_{t-1}), \\
\text{process}(HX, HY), & \text{compose}(HY, I_p, Y), \\
r(TX_p, TY_p), & \ldots, r(TX_t, TY_t), \\
\text{compose}(TY_p, I_{p+1}, I_p), & \ldots, \text{compose}(TY_t, NA, I_t), \\
r_{\text{j}d_2}(TXs, NA, A)
\end{align*}
\]

By \( t \) times folding clause 17 using clauses 1 and 2:
By using applicability conditions (1) and (2), and simplification:

\[ r_{\mathcal{J}}d_5(X, Y, A) = \]

\[ Xs = [X[TXs], \]
\[ \begin{array}{l}
  non\text{-}Minimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
  minimal(TX_t), \ldots, minimal(TX_{p-1}), \\
  (non\text{-}Minimal(TX_1); \ldots; non\text{-}Minimal(TX_{p-1})), \\
  r_{\mathcal{J}}d_5([TX_1, \ldots, TX_{p-1}, [TXs], NA, A], \\
  process(HX, HY), compose(HY, NA, Y) \\
\end{array} \]

By unfolding clause 19 wrt

\[ r_{\mathcal{J}}d_1([TX_1, \ldots, TX_{p-1}, N[TXs'], HY, NA'], \ldots, r_{\mathcal{J}}d_1([N[TXs'], HY, NA_{p-1}] \]

using the “decomposition” of \( S_{r_{\mathcal{J}}d_1} \) in Section 4:

\[ r_{\mathcal{J}}d_5(X, Y, A) = \]

\[ Xs = [X[TXs], \]
\[ \begin{array}{l}
  Xs' = [X[TXs'], TXs' = []], \\
  non\text{-}Minimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
  (non\text{-}Minimal(TX_1); \ldots; non\text{-}Minimal(TX_{p-1})), \\
  minimal(TX_t), \ldots, minimal(TX_{p-1}), \\
  minimal(U_1), \ldots, minimal(U_{p-1}), \\
  decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \\
  r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, Y N), \\
  compose(e, TY_1, NA_1), \\
  compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
  compose(NA_{p-1}, Y N, NA_p), \\
  r_{\mathcal{J}}d_1(TXs', HY, NA_p), \\
  r_{\mathcal{J}}d_5(TXs, NA, A), compose(HY, NA, Y) \\
\end{array} \]

By unfolding clause 19 wrt \( r_{\mathcal{J}}d_1(TXs', HY, NA_p) \) using \( T_{DLR} \), and using the assumption that \( P_{r_{\mathcal{J}}d_1} \) is steadfast wrt \( S_{r_{\mathcal{J}}d_1} \) in \( S \):

\[ r_{\mathcal{J}}d_5(X, Y, A) = \]

\[ Xs = [X[TXs], \]
\[ \begin{array}{l}
  Xs' = [X[TXs'], TXs' = []], \\
  non\text{-}Minimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
  (non\text{-}Minimal(TX_1); \ldots; non\text{-}Minimal(TX_{p-1})), \\
  minimal(TX_t), \ldots, minimal(TX_{p-1}), \\
  minimal(U_1), \ldots, minimal(U_{p-1}), \\
  decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \\
  r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, Y N), \\
  compose(e, TY_1, NA_1), \\
  compose(NA_1, TY_2, NA_2), \ldots, compose(NA_{p-2}, TY_{p-1}, NA_{p-1}), \\
  compose(NA_{p-1}, Y N, NA_p), \\
  TXs' = [], HY = NA_p, \\
  r_{\mathcal{J}}d_1(TXs, NA, A), compose(HY, NA, Y) \\
\end{array} \]

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

\[ r_{\mathcal{J}}d_5(X, Y, A) = \]

\[ Xs = [X[TXs], \]
\[ \begin{array}{l}
  non\text{-}Minimal(X), decompose(X, HX, TX_1, \ldots, TX_t), \\
  (non\text{-}Minimal(TX_1); \ldots; non\text{-}Minimal(TX_{p-1})), \\
  minimal(TX_t), \ldots, minimal(TX_{p-1}), \\
  minimal(U_1), \ldots, minimal(U_{p-1}), \\
  decompose(N, HX, U_1, \ldots, U_{p-1}, TX_1, \ldots, TX_t), \\
  r(TX_1, TY_1), \ldots, r(TX_{p-1}, TY_{p-1}), r(N, Y N), \\
  compose(TY_1, I_1, Y), \\
  compose(TY_2, I_2, I_1), \ldots, compose(TY_{p-1}, I_p, I_{p-1}), \\
  compose(Y N, NA, I_p), \\
  r_{\mathcal{J}}d_5(TXs, NA, A) \\
\end{array} \]

By \( p \) times folding clause 21 using clauses 1 and 2:
\textbf{clause 22:}
\[ r \_d_2(Xs, Y, A) = \]
\[ Xs = [ X[ T Xs ], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ \text{minimal}(TX_p), \ldots, \text{minimal}(TX_t), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i), \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_{r-1}, TX_p, \ldots, TX_t), \]
\[ r \_d_2([TX_1, \ldots, TX_{p-1}, N[TXs], Y, A]) \]

By \( t + 1 \) times unfolding clause 8 wrt
\[ r \_d_1([TX_1, \ldots, TX_{p-1}, N, TX_p, \ldots, TX_t[TXs']], HY, NA'), \ldots, r \_d_1([TX_t[TXs'], HY, NA_t) \]

using the \textit{“decompletion”} of \( S_{r \_d_1} \) in Section 4:

\textbf{clause 23:}
\[ r \_d_2(Xs, Y, A) = \]
\[ Xs = [ X[ T Xs ], \]
\[ Xs' = [ X[ T Xs'], TXs' = []], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i), \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_i), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, YN), \]
\[ \text{compose}(e, TY_1, N, A_1), \]
\[ \text{compose}(NA_1, TY_2, NA_2), \ldots, \text{compose}(NA_{p-2}, TY_{p-1}, NA_{p-1}), \]
\[ \text{compose}(NA_{p-1}, YN, NA_p), \]
\[ \text{compose}(NA_p, TY_p, NA_{p+1}), \ldots, \text{compose}(NA_t, TY_t, NA_{t+1}), \]
\[ r \_d_1(TXs', HY, NA_{t+1}), \]
\[ r \_d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By unfolding clause 23 wrt \( r \_d_1(TXs', HY, NA_{t+1}) \) using TDGLR, and using the assumption that \( P_{r \_d_1} \) is steadfast wrt \( S_{r \_d_1} \) in \( S \):

\textbf{clause 24:}
\[ r \_d_2(Xs, Y, A) = \]
\[ Xs = [ X[ T Xs ], \]
\[ Xs' = [ X[ T Xs'], TXs' = []], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); \ldots; \text{nonMinimal}(TX_{p-1})), \]
\[ (\text{nonMinimal}(TX_p); \ldots; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(U_1), \ldots, \text{minimal}(U_i), \]
\[ \text{decompose}(N, HX, U_1, \ldots, U_i), \]
\[ r(TX_1, TY_1), \ldots, r(TX_t, TY_t), r(N, YN), \]
\[ \text{compose}(e, TY_1, N, A_1), \]
\[ \text{compose}(NA_1, TY_2, NA_2), \ldots, \text{compose}(NA_{p-2}, TY_{p-1}, NA_{p-1}), \]
\[ \text{compose}(NA_{p-1}, YN, NA_p), \]
\[ \text{compose}(NA_p, TY_p, NA_{p+1}), \ldots, \text{compose}(NA_t, TY_t, NA_{t+1}), \]
\[ TXs' = [], HY = NA_{t+1}, \]
\[ r \_d_2(TXs, NA, A), \text{compose}(HY, NA, Y) \]

By using applicability conditions (1) and (2), and simplification:
\[ \text{clause 25: } r_{\Delta d_2}(Xs, Y, A) = \]
\[ Xs = [X[TXs]], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); ; \ldots ; \text{nonMinimal}(TX_{p-1})), \]
\[ (\text{nonMinimal}(TX_p); ; \ldots ; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(U_1); ; \ldots ; \text{minimal}(U_t), \]
\[ \text{decompose}(N, HX, U_1; ; \ldots ; U_t), \]
\[ r(TX_1, TY_1); ; \ldots ; r(TX_t, TY_t), r(N, Y, N), \]
\[ \text{compose}(TY_1, I_1, Y), \]
\[ \text{compose}(TY_2, I_2, I_1), \ldots ; \text{compose}(TY_{p-1}, I_p, I_{p-1}), \]
\[ \text{compose}(YN, I_{p+1}, I_t), \]
\[ \text{compose}(TY_p, I_{p+2}, I_{p+1}), \ldots ; \text{compose}(TY_t, I_{t+1}, I_t), \]
\[ \text{compose}(TY_t, NA, I_{t+1}), \]
\[ r_{\Delta d_2}(TXs, NA, A) \]

By \( t+1 \) times folding clause 25 using clauses 1 and 2:

\[ \text{clause 26: } r_{\Delta d_2}(Xs, Y, A) = \]
\[ Xs = [X[TXs]], \]
\[ \text{nonMinimal}(X), \text{decompose}(X, HX, TX_1, \ldots, TX_t), \]
\[ (\text{nonMinimal}(TX_1); ; \ldots ; \text{nonMinimal}(TX_{p-1})), \]
\[ (\text{nonMinimal}(TX_p); ; \ldots ; \text{nonMinimal}(TX_t)), \]
\[ \text{minimal}(U_1); ; \ldots ; \text{minimal}(U_t), \]
\[ \text{decompose}(N, HX, U_1; ; \ldots ; U_t), \]
\[ r_{\Delta d_2}([TXs], Y, A) \]

Clauses 1, 11, 14, 18, 22 and 26 are the clauses of \( P_r \Delta d_2 \). Therefore \( P_r \Delta d_2 \) is steadfast wrt \( S_r \Delta d_2 \) in \( S \).

To prove that \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r \Delta d_2\} \), 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:

\[ r(X, Y) \leftarrow r_{\Delta d_2}([X], Y, e) \]

By taking the ‘completion’:

\[ \forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \iff r_{\Delta d_2}([X], Y, e)] \]

By unfolding the ‘completion’ above wrt \( r_{\Delta d_2}([X], Y, e) \) using \( S_r \Delta d_2 \):

\[ \forall X : X, \forall Y : Y. \ I_r(X) \Rightarrow [r(X, Y) \iff \exists Y_1, I_1 : Y. \ O_r(X, Y_1) \land I_1 = Y_1 \land O_r(I_1, e, Y)] \]

By using applicability condition (2):

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

By simplification:

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

We obtain \( S_r \), so \( P_r \) is steadfast wrt \( S_r \) in \( \{S_r \Delta d_2\} \).

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

\[ \Box \]

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


