Rule-based Graph Repair using Minimally Restricted Consistency-Improving Transformations

Alexander Lauer

December 20, 2022

Abstract

Contents

1 Preliminaries

Definition 1.1 (**subgraph**). *Let G_1 and G_2 be graphs. The graph G_2 is called a* subgraph *of G_1 if an injective morphism $p : G_2 \hookrightarrow G_1$ exists.*

Definition 1.2 (**intermediate-graph**). *Let G_1 and G_2 be graphs such that G_1 is a subgraph of G_2. A graph C is called an* intermediate-graph *of G_1 and G_2, if injective morphisms $p_1 : G_1 \hookrightarrow C$ and $p' : C \hookrightarrow G_2$ exist. The set of intermediate-graphs of G_1 and G_2 is denoted by $\mathrm{IG}(G_1, G_2)$.*

Definition 1.3 (**overlap**). *Let G_1 and G_2 be graphs. A graph H is called an* overlap of G_1 and G_2 *if morphisms $p : G_1 \hookrightarrow H$ and $p' : G_2 \hookrightarrow H$ exist such that p and p' are jointly surjective and $p(G_1) \cap p'(G_2) \neq \emptyset$. The morphisms p and p' are called* overlap morphisms.

The set of all overlaps of G_1 and G_2 is denoted by $\mathrm{ol}(G_1, G_2)$. Given an overlap H of G_1 and G_2, the overlap morphisms are denoted by $i^H_{G_1}$ and $i^H_{G_2}$, respectively.

Definition 1.4 (**partial morphism**). *Let graphs G_1, G_2, G_3, G_4 and morphisms $p : G_1 \hookrightarrow G_2$ and $p' : G_3 \hookrightarrow G_4$ be given. Then p' is called a* partial morphism *of p if G_3 is a subgraph of G_1, G_4 is a subgraph of G_2 and $p(v) = p'(v)$ for all $v \in G_3$. Given a morphism a, we use the notation a^p to denote that a^p is a partial morphism of a.*

Definition 1.5 (**nested graph condition**). *A* graph condition *over a graph C_0 is inductively defined as follows:*

- *true is a graph condition over every graph.*
- *$\exists(a : C_0 \hookrightarrow C_1, d)$ is a graph condition over C_0 if a is a injective graph morphism and d is a graph condition over C_1.*
- *$\neg d$ is a graph condition over C_0 if d is a graph condition over C_0.*
- *$d_1 \wedge d_2$ and $d_1 \vee d_2$ are graph conditions over C_0 if d_1 and d_2 are graph conditions over C_0.*

Conditions over the empty graph $\emptyset$ are called constraints. *Every injective morphism $p : C_0 \hookrightarrow G$ satisfies true. An injective morphism p satisfies $\exists(a : C_0 \hookrightarrow C_1, d)$ if there exists an injective morphism $q : C_1 \hookrightarrow G$ such that $q \circ a = p$ and q satisfies c. An injective morphism satisfies $\neg d$ if it does not satisfy d, it satisfies $d_1 \wedge d_2$ if it satisfies d_1 and d_2 and it satisfies $d_1 \wedge d_2$ if it satisfies d_1 or d_2. A graph G satisfies a constraint c, $G \models c$, if $p : \emptyset \hookrightarrow G$ satisfies c. We use the abbreviation $\forall(a : C_0 \hookrightarrow C_1, d) := \neg\exists(a : C_0 \hookrightarrow C_1, \neg d)$.*

The nesting level nl *of a condition is defined as* $\mathrm{nl}(\mathit{true} = 0$ *and* $\mathrm{nl}(\exists(a : P \to Q, d)) := \mathrm{nl}(d) + 1$.

Definition 1.6 (**alternating quantifier normal form (ANF)**[3]). *A graph condition c is in* alternating normal form *(ANF) if it is of the form*

$$c = Q(a_1 : C_0 \hookrightarrow C_1, \overline{Q}(a_2 : C_1 \hookrightarrow C_2, Q(a_3 : C_2 \hookrightarrow C_3, \overline{Q}(a_4 : C_3 \hookrightarrow C_4, \ldots))))$$

with $Q \in \{\exists, \forall\}$ and $\overline{Q} = \exists$ if $Q = \forall$, $\overline{Q} = \forall$ if $Q = \exists$.

2 consistency increasing

In this section, we introduce the notion of *consistency increasing* transformations and rules, which allows to increase the consistency of a constraint layer by layer.

2.1 Universally quantified ANF

To prevent the need of case discrimination, we will only consider a subset of the set of conditions in ANF, namely the set of universally quantified conditions in ANF, called *universally quantified alternating quantifier normal form* (UANF). Additionally we will show that these sets are expressively equivalent by showing that each condition in ANF can be transformed into an equivalent condition in UANF.

Definition 2.1 (**Universally quantified alternating quantifier normal form**). *A conditions c in ANF is in* universally quantified ANF *(UANF) if it is universally bound.*

Note that, given a condition c in UANF, every subcondition of c at layer $0 \leq k \leq \mathrm{nl}(c)$ is universally bound, if k is an even number and existentially bound, if k is an odd number.

Lemma 2.2. *Any condition in ANF can be transformed into an equivalent condition in UANF.*

Proof. Let a graph G and a constraint c in ANF be given. If c is universally bound, c is already in UANF.

If $c = \exists(a_0 : C_0 \hookrightarrow C_1, d)$, we show that c is equivalent to $c' := \forall(\mathrm{id}_{C_0} : C_0 \hookrightarrow C_0, c)$.

1. Let $p : C_0 \hookrightarrow G$ be a morphism, such that $q \models c$. Therefore a morphism $q : C_0 \to G$ with $q \models e$ and $p = q \circ a_1$ exists. Then, $p \models d$, since p is the only morphism from C_0 to G with $p = p \circ \mathrm{id}_{C_0}$ and $p \models c$.

2. Let $p : C_0 \hookrightarrow G$ be a morphism with $p \models c'$, therefore all morphisms $q : C_0 \hookrightarrow G$ with $p = q \circ \mathrm{id}_{C_0}$ satisfy c. Since $p = p \circ \mathrm{id}_{C_0}$, $p \models c$ follows immediately.

□

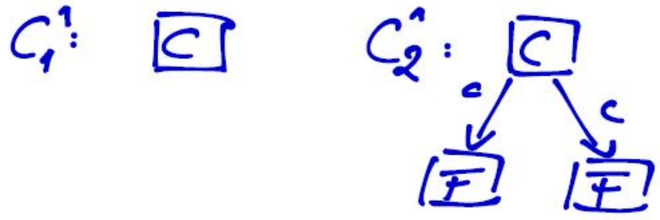
For we rest of this thesis, given a condition $c = \forall(a : C_0 \hookrightarrow C_1, d)$ in UANF, we assume that no morphism in c, except a, is an isomorphism since it can be shown that each condition in ANF can be transformed into an equivalent condition in ANF fulfilling this property.

2.2 Conditions up to layer

Definition 2.3 (**Layer of a subcondition**). *Let c be a condition and d a subcondition of c. The layer of d is defined as* $\mathrm{lay}(d) := \mathrm{nl}(c) - \mathrm{nl}(d)$.

Definition 2.4 (**Subcondition at layer**). *Let c be a condition. The* subcondition at layer k*, denoted by* $\mathrm{sub}_k(c)$ *, is the subcondition d of c with* $\mathrm{lay}(d) = k$.

$$C_1 = \forall C_1^1 \exists C_2^1$$



$$C_2 = \forall C_1^1 \exists C_2^2 \forall C_3^2 \exists C_4^2$$

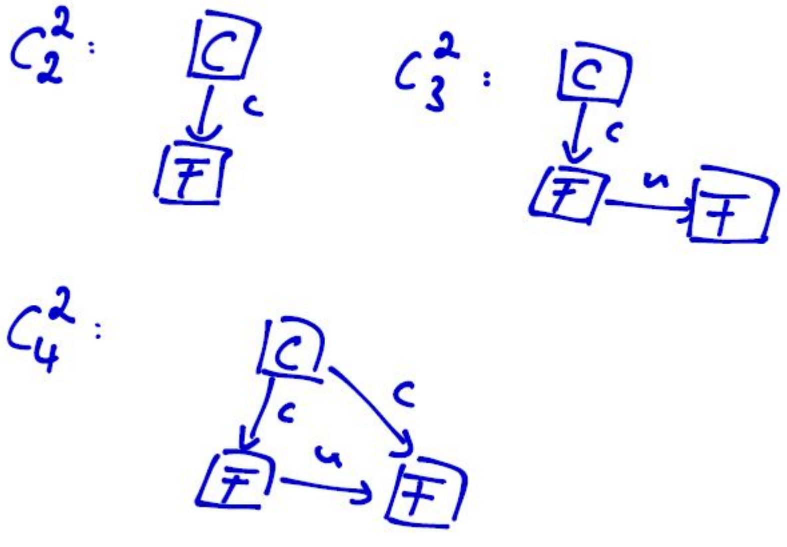


Figure 1: constraints

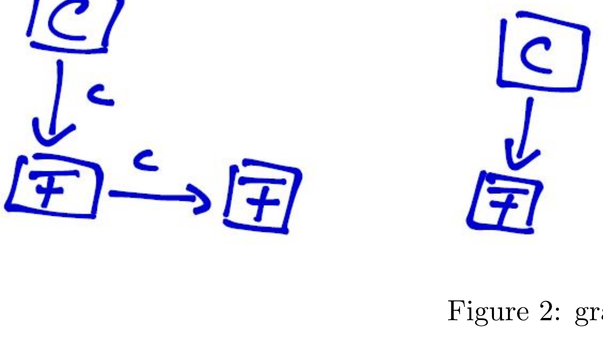


Figure 2: graph

For the rest of this paper, given a constraint c in UANF, $\mathrm{sub}_k(c)$ is always a condition over the graph C_k, for all $0 \leq k \leq \mathrm{nl}(c)$.

We define a notion of partial consistency, called *satisfaction at layer*, which will be used for the definition of consistency increasing. First, two operators are introduced to modify given constraints on a certain layer.

Definition 2.5 (**Replacement at layer**). *Let a condition $c = Q(a : C_0 \hookrightarrow C_1, d)$, with $Q \in \{\forall, \exists\}$ in ANF and a condition e over C_k in ANF be given. The* replacement in c at layer k with e, $\mathrm{rep}_k(c, e)$*, is recursively defined as:*

1. *If $k = 0$:*
$$\mathrm{rep}_0(c, e) := e$$
2. *If $k > 0$:*
$$\mathrm{rep}_k(c, e) := Q\big(a : C_0 \hookrightarrow C_1, \mathrm{rep}_{k-1}(d, e)\big)$$

Example 2.1. *Let the conditions $c := \forall(a_0 : C_0 \hookrightarrow C_1, \exists(a_1 : C_1 \hookrightarrow C_2, \mathit{true}))$ and $d = \exists(a_1' : C_1 \hookrightarrow C_3, e)$ be given. Then,*

$$\mathrm{rep}_1(c, d) = \forall(a_0 : C_0 \hookrightarrow C_1, \exists(a_1' : C_1 \hookrightarrow C_3, e)).$$

Through this, we define *truncated conditions.* Intuitively, a condition is cut off at a certain layer, by replacing the subcondition at this layer by true or false, depending on the quantifier, the replaced subcondition is bound by.

Definition 2.6 (**Truncated condition**). *Let c be a condition in UANF and $d = \mathrm{sub}_k(c)$ with $0 \leq k \leq \mathrm{nl}(c)$. The* truncated condition at layer k *of c, $\mathrm{cut}_k(c)$, is defined as*

$$\mathrm{cut}_k(c) := \begin{cases} \mathrm{rep}_{k+1}(c, \mathit{true}) & \text{if } d \text{ is existentially bound} \\ \mathrm{rep}_{k+1}(c, \mathit{false}) & \text{if } d \text{ is universally bound.} \end{cases}$$

Example 2.2. *Consider constraint c_2 given in figure 1. Then, $\mathrm{cut}_1(c_2) = \forall C_1^1 \exists C_2^2$*

Now, we are ready to define satisfaction at layer, which is a key ingredient for our notion of consistency increasement.

Definition 2.7 (**Satisfaction at layer**). *Let a graph G and a condition c in UANF be given. A morphism $p : C_0 \hookrightarrow G$* satisfies c at layer k, *$p \models_k c$, if*

$$p \models \mathrm{cut}_k(c).$$

A graph G satisfies a constraint c at layer k, $G \models_k c$, if $q : \emptyset \hookrightarrow G$ satisfies $\mathrm{cut}_k(c)$. The biggest $0 \leq k \leq \mathrm{nl}(c)$ such that $G \models_k c$ and no $k < j \leq \mathrm{nl}(c)$ with $G \models_j c$ exists is denoted by $\mathrm{k_{max}}(c, G)$. We use the abbreviation $\mathrm{k_{max}}$ when c and G are clear from the context.

Note that if $p \models_{\mathrm{nl}(c)-1} c$ it follows immediately that $p \models c$.

Example 2.3. *Consider the graph G given in figure 2 and the constraint c_2 given in figure 1. This graph does not satisfy c_2, since both occurrences of* `Class` *do not satisfy $\exists C_2^2 \forall C_3^2 \exists C_4^2$, but is does satisfy $\mathrm{cut}_1(c_2)$ and therefore*

$$G \models_1 c_2 \text{ and } \mathrm{k}_{\max} = 1$$

The following lemmas arise as a direct consequence of the definition of satisfaction at layer. If a graph satisfies a constraint up to a certain layer, let c be this condition up to this layer, that ends with $\forall(a : C \hookrightarrow C', \mathsf{false})$, the graph satisfies all constraints starting with c.

Lemma 2.8. *Let a graph G, a condition c in UANF and a morphism $p : C_0 \hookrightarrow G$ with $p \models_k c$ be given. If the subcondition $d = \mathrm{sub}_k(c)$ is universally bound, then for any condition f over C_{k+1} it holds that*

$$p \models \mathrm{rep}_{k+1}(c, f).$$

Proof. Let $0 \leq k \leq \mathrm{nl}(c)$ be the smallest number with $\mathrm{sub}_k(c) = \forall(a_k : C_k \hookrightarrow C_{k+1}, d)$ being universally bound and $p \models_k c$. Let $q : C_k \hookrightarrow G$ be a morphism such that $q \models \forall(a_k : C_k \hookrightarrow C_{k+1}, \mathsf{false})$. This must exist, since $p \models_k c$ and k is the smallest even number such that $p \models_k c$. Therefore, there does not exist a morphism $q' : C_{k+1} \hookrightarrow G$ with $q = q' \circ a_k$. Hence, for every condition f over C_{k+1} a morphism $q' : C_{k+1} \hookrightarrow G$ with $q \not\models f$ and $q = q' \circ a_k$ cannot exist. It follows immediately that $q \models \forall(a_k : C_{k-1} \hookrightarrow C_k, f)$ and with that $p \models \mathrm{rep}_{k+1}(c, f)$.

It follows that for every even $k < j \leq \mathrm{nl}(c)$, such that $p \models_j c$, and every condition d over C_{j+1} it holds that $p \models \mathrm{rep}_{k+1}(c, f)$ with $f = \mathrm{rep}_{j-k+1}(\mathrm{sub}_{k+1}(c), d)$. Since $\mathrm{rep}_{k+1}(c, f) = \mathrm{rep}_{j+1}(c, d)$ it follows that $p \models \mathrm{rep}_{j+1}(c, d)$. □

As a direct consequence of the previous lemma, a graph satisfying a condition up to layer ending with $\forall(a : C \hookrightarrow C', \mathsf{false})$ also satisfies the whole constraint.

Lemma 2.9. *Let G be a graph, $p : C_0 \hookrightarrow G$ a morphism and c a condition over C_0 in UANF with $p \models_k c$. If $\mathrm{sub}_k(c)$ is universally bound,*

$$p \models_k c \implies p \models c.$$

Proof. Follows immediately by using lemma 2.8 and setting f to $\mathrm{sub}_{k+1}(c)$. □

Additionally, with lemma 2.8 it can be shown that given an graph G, a condition in UANF over C_0 and a morphism $p : C_0 \hookrightarrow G$ such that $p \models_k c$ and $\mathrm{sub}_k(c)$ is universally bound, $p \models_j c$ for all $k \leq j \leq \mathrm{nl}(c)$.

Lemma 2.10. *Let a graph G, a morphism $p : C_0 \hookrightarrow G$ and a constraint c in UANF be given. Then, $p \models_j c$ if $j < \mathrm{k}_{\max}$ and $\mathrm{sub}_j(c)$ is existentially bound.*

Proof. If a $j < \mathrm{k}_{\max}$ with $p \models_j c$ exists such that $\mathrm{sub}_j(c)$ is universally bound, let j_1 be the smallest of these. With lemma 2.8 follows that $p \models_{j_2} c$ for all $j_1 \leq j_2 < \mathrm{nl}(c)$.

Let $\ell < j_1$, such that $\mathrm{sub}_\ell(c)$ is existentially bound and let $d = \mathrm{cut}_{j_1-\ell}(\mathrm{sub}_\ell(c)) = \exists(a_\ell : C_\ell \hookrightarrow C_{\ell+1}, e)$ be the condition up to layer $j_1 - \ell$ of $\mathrm{sub}_\ell(c)$. Since $\ell < j_1$, a morphism $q : C_\ell \hookrightarrow G$ with $q \models d$ must exists and therefore a morphism $q' : C_{\ell+1} \hookrightarrow G$ with $q = q' \circ a_k$ must exists. It follows that $q \models \exists(a_\ell : C_\ell \hookrightarrow C_{\ell+1}, \mathsf{true})$ and with that $p \models_\ell c$. □

Through satisfaction at layer an increase of consistency can be detected in the following way. Let $G \Longrightarrow H$ be a transformation. If $\mathrm{k}_{\max}(c, G) < \mathrm{k}_{\max}(c, H)$, the transformation can be considered as consistency increasing, since H satisfies more layers of the constraint than G. The notion of consistency increasement should also be able to detect the smallest transformations that lead to an increase of consistency, namely the inserting of a single edge or node of an existentially bound graph. To remedy this issue, we introduce *intermediate conditions*. Intuitively, given a constraint c in UANF ending with $\mathrm{sub}_{\mathrm{nl}(c)-1}(c) = \exists(a_k : C_k \hookrightarrow C_{k+1}, d)$, the condition $\mathrm{sub}_{\mathrm{nl}(c)-1}(c)$ is replaced by $\exists(a_k^p : C_k \hookrightarrow C', \mathsf{true})$ with $C' \in \mathrm{IG}(C_k, C_{k+1})$.

The construction of intermediate conditions is designed to only replace graphs in existentially bound layers, since the replacement in an universally bound layer would lead to a more restrictive constraint than the original condition up to layer. That means, given the condition $c = \forall(a_0 : C_0 \hookrightarrow C_1, \mathsf{false})$ and let $C' \in \mathrm{IG}(C_0, C_1)$. If the condition $c' = \forall(a_0^p : C_0 \hookrightarrow C_1, \mathsf{false})$ is satisfied this implies that c is also satisfied but the backwards implication does not hold.

Definition 2.11 (**Intermediate condition**). *Let a constraint c in UANF be given. Let $0 \leq k < \mathrm{nl}(c)$ such that either $\mathrm{sub}_k(c) = \exists(a_k : C_k \hookrightarrow C_{k+1}, d)$ is existentially bound or $k = \mathrm{nl}(c) - 1$. The* intermediate condition, $\mathrm{IC}_k(c, C')$, of c at layer k with

$$C' \in \begin{cases} \mathrm{IG}(C_k, C_{k+1}) & \textit{if } \mathrm{sub}_k(c) \textit{ is existentially bound} \\ \{C_{\mathrm{nl}(c)}\} & \textit{otherwise} \end{cases}$$

is defined as:

$$\mathrm{IC}_k(c, C') := \begin{cases} \mathrm{rep}_k(c, \exists(a_k^r : C_k \hookrightarrow C', \textit{true})) & \textit{if } \mathrm{sub}_k(c) \textit{ is existentially bound} \\ c & \textit{otherwise} \end{cases}$$

Example 2.4. *Consider constraint c_1 given in figure 1. Since $C_2^2 \in \mathrm{IG}(C_1^1, C_1^2)$, we can construct a intermediate condition of c_1 at layer 1 with C_2^2 as $\mathrm{IC}_1(c_1, C_2^2) = \forall C_1^1 \exists C_2^2$. Whereas c_1 checks whether each node of the type* `Class` *is connected to at least two nodes of the type* `Feature`*, the partial condition checks whether each nodes of the type* `Class` *is connected to at least one nodes of the type* `Feature` *which is trivially satisfied if c_1 is satisfied.*

If a graph G does not satisfy the condition up to layer $k := \mathrm{k}_{\max} + 2$, $k < \mathrm{nl}(c)$, of a given constraint c, with $\mathrm{sub}_k(c)$ being existentially bound, there does exists a graph $C' \in \mathrm{IG}(C_k, C_{k+1})$, such that G satisfies $\mathrm{IC}_k(c, C')$ since G always satisfies $\mathrm{IC}_k(c, C_k)$.

2.3 Consistency increasing transformations and rules

With the results above, we are now ready to define the notions of *consistency increasement* and *direct consistency increasement*, with direct increasing being a more restrictive version of increasing, yielding the advantage that second-order logic formulas can be used in order to determine whether a transformation is (direct) consistency increasing or not.

These notions are designed to only detect transformations that increase the consistency of the first two unsatisfied layer of a constraint c. That means, given a graph G and a constraint c, let $k = \mathrm{k}_{\max} + 1$ and $\mathrm{sub}_k(c) := \forall(a_k : C_k \hookrightarrow C_{k+1}, \exists(a_{k+1} : C_{k+1} \hookrightarrow C_{k+2}, d))$. A transformation $t : G \Longrightarrow H$ is considered as (direct) consistency increasing if $\mathrm{k}_{\max}(c, G) \leq \mathrm{k}_{\max}(c, H)$, i.e. the satisfaction up to layer is not decreased, and more increasing insertions or deletions have been performed than decreasing ones. An increasing deletion is the deletion of an occurrence of C_{k+1} that does not satisfy $\exists(a_{k+1} : C_{k+1} \hookrightarrow C_{k+2}, \mathsf{true})$, an increasing insertion is the insertion of elements of C_{k+2}, such that for at least one occurrence p of C_{k+1} it holds that $p \not\models \exists(a^r_{k+1} : C_{k+1} \hookrightarrow C', \mathsf{true})$ and $\mathrm{tr}_t \circ p \models \exists(a^r_{k+1} : C_{k+1} \hookrightarrow C', \mathsf{true})$ for a graph $C' \in \mathrm{IG}(C_{k+1}, C_{k+2})$. A decreasing insertion is the creation of an occurrence of C_{k+1} not satisfying $\exists(a_{k+1} : C_{k+1} \hookrightarrow C_{k+2}, \mathsf{true})$ and a decreasing deletion is the deletion of elements of C_{k+2} such that for an occurrence p of C_{k+1} with $p \models \exists(a^r_{k+1} : C_{k+1} \hookrightarrow C', \mathsf{true})$ it holds that $\mathrm{tr}_t \circ p \not\models \exists(a^r_{k+1} : C_{k+1} \hookrightarrow C', \mathsf{true})$ for a graph $C' \in \mathrm{IG}(C_{k+1}, C_{k+2})$.

To evaluate this, we define the *number of violations.* Intuitively, for all occurrences p of C_{k+1} the number of graphs $C' \in \mathrm{IG}(C_{k+1}, C_{k+2})$ with $p \not\models \exists C'$ is add up and it can be determined whether more increasing or decreasing actions have been performed by a transformation.

Note, that the number of violations is defined for each layer of the constraint, but only for the first unsatisfied layer the sum is calculated as described above. For all layer k with $k \leq \mathrm{k}_{\max}$ it is set to 0 and for all layer k with $k > \mathrm{k}_{\max} + 1$ it is set to ∞. Through this, a transformation $tG \Longrightarrow H$ that increases the satisfaction up to layer can easily detected since the number of violations in H at layer $c^G_{\max} + 1$ will be set to 0.

Definition 2.12 (number of violations). *Let a graph G and a constraint c in UANF be given. The* number of violations $\mathrm{nvc}_j(G)$ at layer $0 \leq j < \mathrm{nl}(c)$ in G *is defined as:*

1. *If $j < \mathrm{k}_{\max} + 1$:*
$$\mathrm{nvc}_j(G) := 0$$

2. *If $j = \mathrm{k}_{\max} + 1$, let $d = \forall(a_k : C_{j+1} \hookrightarrow C_{j+2}, e)$ be the subcondition of c at layer $j + 1$.*

$$\mathrm{nvc}_j(G) := \begin{cases} \sum_{C' \in \mathrm{IG}(C_{j+2}, C_{j+3})} |\{q \mid q : C_{j+1} \hookrightarrow G \wedge q \not\models \mathrm{IC}_0(e, C')\}| & \text{if } e \neq \mathsf{false} \\ |\{q \mid q : C_{j+1} \hookrightarrow G\}| & \text{if } e = \mathsf{false} \end{cases}$$

3. *If $j > \mathrm{k}_{\max} + 1$:*
$$\mathrm{nvc}_j(G) := \infty$$

Via the number of violations, we now define *consistency increasing* transformations and rules, by checking whether the number of violations has decreased for any layer of the constraint.

Definition 2.13 (**consistency increasing transformations and rules**). *Let a graph G, a rule ρ and a constraint c in UANF be given.*

A transformation $G \Longrightarrow_{\rho,m} H$ is called consistency increasing w.r.t c*, if*

$$\mathrm{nvc}_k(H) < \mathrm{nvc}_k(G)$$

for any $0 \leq k < \mathrm{nl}(c)$. A rule r is called consistency increasing w.r.t c*, if all of its transformations are.*

Note that if $G \models c$ there does not exist a consistency increasing transformation w.r.t c, since $\mathrm{nvc}_j(G) = 0$ for all $0 \leq j < \mathrm{nl}(c)$. Also, no plain rule ρ is consistency increasing w.r.t c, since a graph G satisfying c, such that a transformation $t : G \Longrightarrow_{\rho,m} H$ exists can always be constructed. Therefore, each consistency increasing rule has to be equipped with at least one application condition.

As mentioned above, a transformation should be detected as consistency increasing if it increases the satisfaction up to layer, which is shown by the following theorem.

Theorem 2.1. *Let a graph G, a rule ρ and a constraint c in UANF be given. A transformation $t : G \Longrightarrow_{\rho,m} H$ is consistency increasing w.r.t c if $\mathrm{k}_{\max}(c, G) < \mathrm{k}_{\max}(c, H)$.*

Proof. No $\ell > \mathrm{k}_{\max}(c, G)$ with $G \models_\ell c$ exists. Hence, $\mathrm{nvc}_{\mathrm{k}_{\max}(c,G)+1}(G) > 0$ and $\mathrm{nvc}_{\mathrm{k}_{\max}(c,G)+1}(G) \neq \infty$. Since $\mathrm{k}_{\max}(c, H) > \mathrm{k}_{\max}(c, G)$, $\mathrm{nvc}_{\mathrm{k}_{\max}(c,H)+1}(H) = 0$ and it follows immediately that t is consistency increasing w.r.t c. □

Since no consistency increasing transformation originating in consistent graphs exist, there do not exist infinite long sequences of consistency increasing transformations. Additionally, if a set of rules $\mathcal{R}$ is given, and a sequence of consistency increasing transformations with rules of $\mathcal{R}$ ends with a graph G, then either G satisfies the constraint or there do not exist any consistency increasing transformations $G \Longrightarrow_{\rho,m} H$ with $\rho \in \mathcal{R}$.

Theorem 2.2. *Let a constraint c in UANF and a set of rules $\mathcal{R}$ be given. Every sequence of consistency increasing transformations w.r.t c with rules in $\mathcal{R}$ is finite.*

Proof. Let G_0 be a graph and

$$G_0 \Longrightarrow_{\rho_1} G_1 \Longrightarrow_{\rho_2} G_2 \Longrightarrow_{\rho_3} \dots$$

be a sequence of consistency increasing transformations w.r.t c with $\rho_i \in \mathcal{R}$. We assume that $\mathrm{k}_{\max}(c, G_0) < \mathrm{nl}(c)$, otherwise $\mathrm{nvc}_j(G_0) = 0$ for all $0 \leq j < \mathrm{nl}(c)$ and no transformation $G_0 \Longrightarrow H$ is consistency increasing.

We show that after at most $j := \mathrm{nvc}_{\mathrm{k}_{\max}(c,G_0)+1}(G_0)$ transformations $G_j \models_{\mathrm{k}_{\max}(c,G_0)+2} c$ holds. Note that j has to be finite, since G_0 contains only a finite number of occurrences of C_{j+1}. After each transformation it holds that $\mathrm{nvc}_{\mathrm{k}_{\max}(c,G_i)+1}(G_{i+1}) \leq$

$\mathrm{nvc}_{\mathrm{k_{max}}(c,G_i)+1}(G_i)-1$. Therefore, after at most j transformations, $\mathrm{nvc}_{\mathrm{k_{max}}(c,G_0)+1}(G_j) \leq \mathrm{nvc}_{\mathrm{k_{max}}(c,G_0)+1}(G_0) - j = 0$ and $G_j \models_{\mathrm{k_{max}}(c,G_0)+2} c$. By iteratively applying this, it follows that after a finite number of transformations a graph G_k with $G_k \models c$ has to exist. Since no consistency increasing transformation $G_k \Longrightarrow G_{k+1}$ exists, the sequence has to be finite. Also, for some $G_{k'}$ with $k' < k$ a consistency increasing transformation $G_{k'} \Longrightarrow_\rho H$ with $\rho \in \mathcal{R}$ may not exists and the sequence up to $G_{k'}$ is also finite, but does not end with a graph satisfying c. □

The replacement of a graph not satisfying a constraint c by a c satisfying graph via a transformation is a consistency increasing transformation. Therefore, the notion of consistency increasement does allow insertions or deletions that are unnecessary in order to increase consistency. That is, the deletion of occurrences of existentially bound graphs, the deletions of occurrences of universally bound graphs C_k satisfying $\exists C_{k+1}$ or the insertion of occurrences of universally bound graphs and the insertion of existentially bound graphs C_k, such that the corresponding occurrence of C_{k-1} already satisfied $\exists C_k$.

Direct consistency increasing transformations are more restricted, in the sense, that these unnecessary deletions and insertions are leading to a transformation not being direct consistency increasing. The presence of these unnecessary actions can be checked via second-order logic formulas. Additionally, it is secured that no new violations are introduced since these can always be considered as a unnecessary insertion or deletion. With that, the removal of one violation is sufficient to state that the transformation is (direct) consistency increasing, which can also be checked via a second order logic formula. Intuitively, it is checked by the first two formulas, that no new violations are introduced. The third formula secures that at least one violation has been removed and the last formulas secures that the satisfaction up to layer is not decreased.

Definition 2.14 (**direct consistency increasing**). *Let G be a graph, ρ a plain rule, c a constraint in UANF and* $\mathrm{sub}_{\mathrm{k_{max}}+1}(c) = \forall(a_{k-1} : C_{k-1} \hookrightarrow C_k, e)$. *A transformation $t : G \Longrightarrow_{\rho,m} H$ is called* direct consistency increasing w.r.t c *if the following equations hold. Let*

$$\mathbf{G} = \begin{cases} \mathrm{IG}(C_k, C_{k+1}) & \textit{if } e \neq \textsf{false} \\ \{C_k\} & \textit{otherwise} \end{cases}$$

Every occurrence of C_k in G that satisfies $\mathrm{IC}_0(e, C')$ for any $C' \in \mathbf{G}$ still satisfies $\mathrm{IC}_0(e, C')$ in H.

$$\forall p : C_k \hookrightarrow G\Big(\bigwedge_{C' \in \mathbf{G}} \big(p \models \mathrm{IC}_0(e, C') \wedge \mathrm{tr}_t \circ p \textit{ is total}\big) \implies \mathrm{tr}_t \circ p \models \mathrm{IC}_0(e, C')\Big) \tag{2.1}$$

Let $C' = C_{k+1}$ if $e \neq \textsf{false}$ and $C' = C_k$ otherwise. Every new inserted occurrence of C_k by t satisfies $\mathrm{IC}_0(e, C')$.

$$\forall p' : C_k \hookrightarrow H\big(\neg\exists p : C_k \hookrightarrow G(p' = \mathrm{tr}_t \circ p) \implies p' \models \mathrm{IC}_0(e, C')\big) \tag{2.2}$$

At least one occurrence of C_k in G that does not satisfy $\mathrm{IC}_0(e, C')$*, for any* $C' \in \mathbf{G}$*, either has been removed or satisfies* $\mathrm{IC}_0(e, C')$ *in* H*.*

$$\exists p : C_k \hookrightarrow G\Big(\bigvee_{C' \in \mathbf{G}} (p \not\models \mathrm{IC}_0(e, C') \wedge (\mathrm{tr}_t \circ p \textit{ is not total } \vee \mathrm{tr}_t \circ p \models \mathrm{IC}_0(e, C')))\Big) \tag{2.3}$$

No occurrence of a universally bound graph C_j with $j < \mathrm{k}_{\max}$ gets inserted.

$$\bigwedge_{\substack{i < \mathrm{k}_{\max} \\ i \textit{ even}}} \forall p : C_i \hookrightarrow H(\exists p' : C_i \hookrightarrow G(p = \mathrm{tr}_t \circ p')) \tag{2.4}$$

No occurrence of an existentially bound graph C_j with $j \leq \mathrm{k}_{\max}$ gets deleted.

$$\bigwedge_{\substack{i \leq \mathrm{k}_{\max} \\ i \textit{ odd}}} \forall p : C_i \hookrightarrow G(\mathrm{tr}_t \circ p \textit{ is total}) \tag{2.5}$$

Note that (2.4) and (2.5) not only secure that the satisfaction up to layer does not decrease, as shown in the following lemma, but also prevent further unnecessary insertions and deletions, since the insertion of a universally and the deletion of a existentially bound graph will never lead to an increase of consistency.

Lemma 2.15. *Let a transformation $t : G \Longrightarrow H$ and a constraint c in UANF be given, such that (2.4) and (2.5) of definition 2.14 are satisfied. Then,*

$$H \models_{\mathrm{k}_{\max}(c,G)} c.$$

Proof. Assume that $H \not\models_{\mathrm{k}_{\max}(c,G)} c$. Then, either a new occurrence of an universally bound graph C_k with $k < \mathrm{k}_{\max}(c, G)$ has been inserted or an occurrence of an existentially bound graph C_k with $k \leq \mathrm{k}_{\max}(c, G)$ has been destroyed. Therefore, the following holds:

$$\exists p : C_i \hookrightarrow H(\neg \exists p' : C_i \hookrightarrow G(p = \mathrm{tr}_t \circ p')) \vee \exists p : C_j \hookrightarrow G(\mathrm{tr}_t \circ p \textit{ is not total})$$

with $i, j \leq \mathrm{k}_{\max}(c, G)$, i being even and j being odd. It follows immediately that either (2.4) or (2.5) is not satisfied. This is a contradiction. □

Now, we will show the already indicated relationship between direct consistency increasing and consistency increasing, namely that a direct consistency increasing transformation is also consistency increasing. Counterexamples, that the inversion of the implication does not hold can easily be constructed, showing that these notions are not identical, but related.

Lemma 2.16. *Let a graph G, a constraint c in UANF, a rule ρ and a direct consistency increasing transformation $t : G \Longrightarrow_{\rho,m} H$ w.r.t. c be given. Then, t is also a consistency increasing transformation.*

Proof. Let $k = \mathrm{k}_{\max}(c, G) + 2$ and $d = \mathrm{sub}_k(c)$ with $d \neq \mathsf{false}$. With lemma 2.15 follows that $\mathrm{k}_{\max}(c, G) \leq \mathrm{k}_{\max}(c, H)$ and with that $\mathrm{nvc}_k(H) \neq \infty$. Let

1. We show that equations (2.1) and (2.2) imply that $\mathrm{nvc}_k(H) \leq \mathrm{nvc}_k(G)$. Assume that $\mathrm{nvc}_k(H) > \mathrm{nvc}_k(G)$. Therefore, a morphism $p : C_k \hookrightarrow H$ with $p \not\models \mathrm{IC}_0(d, C')$ for any $C' \in \mathrm{IG}(C_k, C_{k+1})$ exists, such that either 1a or 1b is satisfied.

   (a) There does exist a morphism $q' : C_k \hookrightarrow G$ with $q' \models \mathrm{IC}_0(d, C')$ and $p = \mathrm{tr}_t \circ q'$.

   (b) There does not exist a morphism $q : C_k \hookrightarrow G$ with $p = \mathrm{tr}_t \circ q$.

   This is a contradiction, if 1a is satisfied, q' does not satisfy equation (2.1) and if 1b is satisfied q does not satisfy equation (2.2). It follows that

   $$|\{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{IC}_0(d, C')\}| \leq |\{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{IC}_0(d, C')\}|$$

   for all $C' \in \mathrm{IG}(C_k, C_{k+1})$.

2. Since (2.3) is satisfied, a morphism $p : C_k \hookrightarrow G$ with $p \not\models \mathrm{IC}_0(d, C')$, such that either $\mathrm{tr} \circ p$ is total and $\mathrm{tr}_t \circ p \models \mathrm{IC}_0(d, C')$ or $\mathrm{tr} \circ p$ is not total exists, for a graph $C' \in \mathrm{IG}(C_k, C_{k+1})$. In both cases the following holds:

   $$\begin{aligned} p &\in \{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{IC}_0(d, C')\} \wedge \\ \mathrm{tr} \circ p &\notin \{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{IC}_0(d, C')\} \end{aligned}$$

   With that and 1. it follows that

   $$|\{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{IC}_0(d, C')\}| < |\{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{IC}_0(d, C')\}|.$$

In total, $\mathrm{nvc}_k(G) < \mathrm{nvc}_k(G)$ and t is a consistency increasing transformation.

For the special case that $k = \mathrm{nl}(c)$ and c ends with $\forall(a_k : C_{k-1} \hookrightarrow C_k, \mathsf{false})$ it can be shown in a similar way that (2.1) and (2.2) imply that

$$|\{q \mid q : C_k \hookrightarrow G\}| \leq |\{q \mid q : C_k \hookrightarrow H\}|$$

and that (2.3) implies

$$|\{q \mid q : C_k \hookrightarrow G\}| < |\{q \mid q : C_k \hookrightarrow H\}|.$$

It follows that t is a consistency increasing transformation. □

2.4 Comparison with other consistency concepts

In this chapter, the notions of (direct) consistency increasing are compared to the already known concepts of consistency guaranteeing, consistency preserving [1], (direct) consistency increasing and sustaining [2] in order to reveal relationships between them and secure that (direct) consistency increasing is indeed a new notion of consistency. These relations are displayed in figure 3.

$$\begin{tikzcd}
 & c\text{-guaranteeing} \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & & \text{direct increasing w.r.t } c \arrow[dl, Rightarrow] \\
c\text{-preserving} & & \text{increasing w.r.t } c \arrow[ll, Rightarrow] &
\end{tikzcd}$$

Figure 3: General relations of consistency notions.

First, we compare (direct) consistency increasing with the notions of consistency-guaranteeing and -preserving. As already mentioned, given a constraint c and a graph G with $G \models c$, there does not exist a (direct) consistency increasing transformation $t : G \Longrightarrow H$ and it follows immediately that (direct) increasing w.r.t c implies c-preserving.

Obviously, guaranteeing implies consistency increasing, since this property is embeded in the definition of consistency increasing. The inversion of this implication does not holds, since increasing is a way stricter notion, in the sense, that the number of removed violations has to be greater than the number of introduced violations. For guaranteeing transformations this is not the case, an arbitrary number of violations can be inserted, long as the derived graph satisfies the constraint and therefore guaranteeing does not imply direct increasing, since a direct increasing transformations is not allowed to introduce any new violations.

Lemma 2.17. *Let a constraint c in UANF, graphs G and H with $G \not\models c$, and a transformation $t : G \Longrightarrow H$ be given. Then,*

$$\begin{array}{lll}
t \textit{ is c-guaranteeing} & \Longrightarrow & t \textit{ is increasing w.r.t c} \quad \wedge \\
t \textit{ is c-guaranteeing} & \not\Longrightarrow & t \textit{ is direct increasing w.r.t c} \wedge \\
t \textit{ is increasing w.r.t c} & \not\Longrightarrow & t \textit{ is c-guaranteeing.}
\end{array}$$

Proof.

1. Let t be a c-guaranteeing transformation, then $H \models c$. Since $G \not\models c$, $\text{nvc}_{\text{k}_{\max}(c,G)+1}(G) > 0$ and $\text{nvc}_{\text{k}_{\max}(c,G)+1}(H) = 0$. Therefore, t is consistency increasing.

2. Let a constraint $c = \exists(a_0 : \emptyset \hookrightarrow C_1, \forall(a_1 : C_1 \hookrightarrow C_2, \exists(a_2 : C_2 \hookrightarrow C_3, \mathsf{true})))$ and a graph $G = C' \dot\cup C'$ with $C' = C_3 \setminus \{e\}$ for one edge $e \in E_{C_3 \setminus C_2}$ and the occurrences $p_1 : C_3 \hookrightarrow G$ and $p_2 : C_2 \hookrightarrow G$ be given. Let $t : G \Longrightarrow H$ be a transformation with $H = C_3 \dot\cup C''$ and $C'' = C' \setminus \{e'\}$ for an $e' \in E_{C_3 \setminus C_2}$, such that $\text{tr}_t \circ p_1$ and $\text{tr}_t \circ p_2$ are total. Then, t is c-guaranteeing and not direct increasing, since $p_i \models \exists(a'_2 : C_2 \hookrightarrow C', \mathsf{true})$ for $i = 1, 2$ and either $\text{tr}_t \circ p_1 \not\models \exists(a'_2 : C_2 \hookrightarrow C', \mathsf{true})$ or $\text{tr}_t \circ p_2 \not\models \exists(a'_2 : C_2 \hookrightarrow C', \mathsf{true})$.

3. Let t be a consistency increasing transformation w.r.t c, such that $\text{nvc}_{\text{k}_{\max}(c,G)+1}(G) > \text{nvc}_{\text{k}_{\max}(c,G)+1}(H) > 1$ and $H \not\models_{\text{k}_{\max}(c,G)+2} c$. Then, $H \not\models c$ and t is not a c-guaranteeing transformation.

□

The definition of consistency improving only differs from guaranteeing if the corresponding constraint is universally bound and these notions are identical for existentially bound constraint. Therefore, with lemma 2.17, we can state the following.

Corollary 2.18. *Let c be an existentially bound constraint in ANF. Then,*

$$\begin{aligned} &t \textit{ is consistency improving w.r.t } c &&\implies t \textit{ is consistency increasing w.r.t } c \wedge \\ &t \textit{ is consistency increasing w.r.t } c &&\not\Longrightarrow t \textit{ is consistency improving w.r.t } c \end{aligned}$$

The notions of increasing and improving are equivalent for universally bound constraints with nesting level 1. Note that, with corollary 2.18, this equivalence does not hold for existentially bound constraints with nesting level 1.

Lemma 2.19. *let a universally bound constraint c in UANF with $\mathrm{nl}(c) = 1$, a graph G with $G \not\models c$ and a transformation $t : G \Longrightarrow H$ be given. Then,*

$$t \textit{ is consistency improving w.r.t } c \iff t \textit{ is consistency increasing w.r.t } c$$

Proof. Let $c = \forall(a : \emptyset \hookrightarrow C, \mathsf{false})$. Since $\mathrm{sub}_1(c) = \mathsf{false}$, $\mathrm{nvc}_1(G)$ is the number of occurrences of C in G. This is exactly the definition of the number of violations for consistency improving transformations and the statement follows immediately. □

For universally bound constraints c with $\mathrm{nl}(c) \geq 2$, (direct) consistency increasing is not related to (direct) consistency improving and sustaining. As shown in [2], (direct) consistency improving implies (direct) consistency sustaining. Therefore, it is sufficient to show that direct improving does not imply increasing and that direct increasing does not imply consistency sustainment.

Lemma 2.20. *Let a universally bound constraint c in UANF with $\mathrm{nl}(c) \geq 2$, a graph G with $G \not\models c$ and a transformation $t : G \Longrightarrow H$ be given. Then,*

$$\begin{aligned} &t \textit{ is direct consistency improving w.r.t } c &&\not\Longrightarrow t \textit{ is consistency increasing w.r.t } c \wedge \\ &t \textit{ is direct increasing w.r.t } c &&\not\Longrightarrow t \textit{ is consistency sustaining w.r.t } c \end{aligned}$$

Proof.

1. Let $c = \forall(a_0 : \emptyset \hookrightarrow C_1, \exists(a_1 : C_1 \hookrightarrow C_2, \mathsf{true}))$ be a constraint. Let $V_{C_1} = V_{C_2}$ and $|E_{C_2}| - |E_{C_1}| = 2$. Let $G = C' \dot\cup C'$ with $C' = C_2 \backslash \{e\}$ with $e \in E_{C_2} \backslash E_{C_1}$ and the occurrences $p_1 : C_1 \hookrightarrow G$ and $p_2 : C_1 \hookrightarrow G$ be given. It follows that $\mathrm{nvc}_0(G) = 2$. Let $t : G \Longrightarrow H$ be a transformation, such that $H = C_1$. Then, t is a direct consistency improving transformation, since H contains only one occurrence of C_1 not satisfying $\exists(a_1 : C_1 \hookrightarrow C_2\, \mathsf{true})$. But, t is not consistency increasing, since $\mathrm{nvc}_0(H) = 3$.

2. Let $c := \forall(a_0 : \emptyset \hookrightarrow C_1, \exists(a_1 : C_1 \hookrightarrow C_2, \forall(a_2 : C_2 \hookrightarrow C_3, \exists(a_3 : C_3 \hookrightarrow C_4, \mathsf{true}))))$ be a constraint.

   Let a graph $G = C_1$ with the morphism $q : C_1 \hookrightarrow G$ and a transformation $t : G \Longrightarrow H$ with $H := C_3 \dot\cup C_3$ be given, such that $\mathrm{tr}_t \circ q$ is total. Then t is a

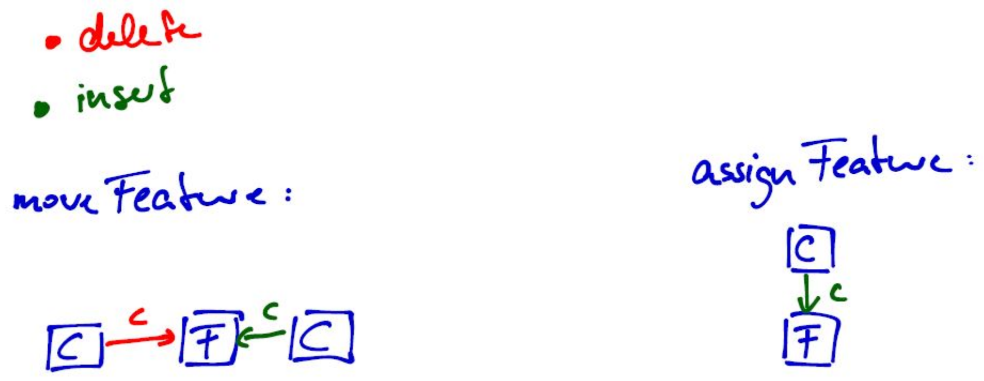


Figure 4: rules

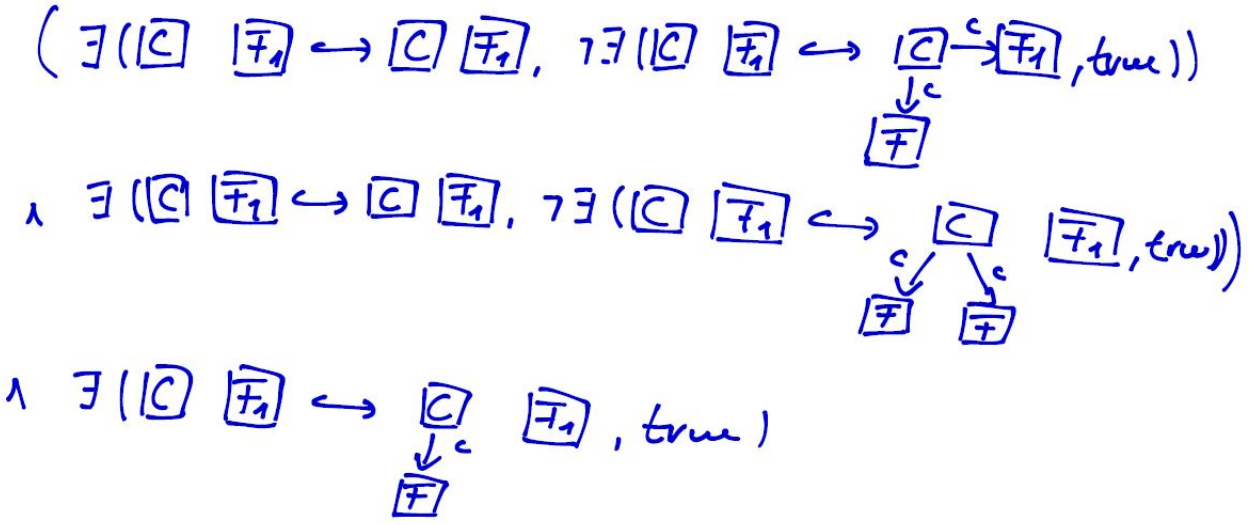


Figure 5: application condition

direct consistency increasing transformation but not a consistency sustaining one, since H contains more occurrences of C_1 not satisfying $\exists(a_1 : C_1 \hookrightarrow C_2, \forall(a_2 : C_2 \hookrightarrow C_3, \exists(a_3 : C_3 \hookrightarrow C_4, \mathsf{true})))$ than G.

□

3 general application conditions

To guarantee that each transformation $t : G \Longrightarrow_{\rho,m} H$, given a rule $\rho = L \hookleftarrow K \hookrightarrow R$, is (direct) consistency increasing w.r.t to a constraint c, we present applications conditions ensuring this property. It has to be ensured that at least one violation will be removed. For this, it is necessary (a) to check that an occurrence p of the universally bound graph $C_{\mathrm{k_{max}}(c,G)+1}$ exists, such that p and the match m do overlap, i.e $p(C_{\mathrm{k_{max}}(c,G)+1}) \cap m(L) \neq$

$\emptyset$, and (b) that p does not satisfy $c' = \exists C_{\mathrm{k_{max}}(c,G)+2}$. Only in this case, it is possible that the transformation does remove a violation. To ensure that p does not satisfy c', the non-existence of all possible overlaps of L and $C_{\mathrm{k_{max}}(c,G)+2}$ such that $p \models c'$ has to be checked. For this, we introduce *extended overlaps*. Intuitively, given an overlap C of L and $C_{\mathrm{k_{max}}(c,G)+1}$ with $p : C_{\mathrm{k_{max}}(c,G)+1} \hookrightarrow C$, the set of extended overlaps contains all overlaps C' of L and $C_{\mathrm{k_{max}}(c,G)+2}$, such that $p \models c'$.

Definition 3.1 (**extended overlaps**). *Let G, C_0 and C_1 with $i^{C_1}_{C_0} : C_0 \hookrightarrow C_1$ be graphs. Let C be an overlap of C_0 and G with the inclusion $i^{C}_{C_0} : C_0 \hookrightarrow C$. The set of* extended overlaps of C with $i^{C_1}_{C_0}$, $\mathrm{eol}(C, i^{C_1}_{C_0})$, *is the set of all overlaps C' of G and C_1, such that an injective morphism $i^{C'}_{C} : C \hookrightarrow C'$ with $i^{C'}_{C} \circ i^{C}_{C_0} \models \exists(i^{C_1}_{C_0} : C_0 \hookrightarrow C_1, \mathsf{true})$ exists.*

A direct consistency increasing application condition has to, as mentioned above, (a) ensure that at least one violation will be removed, (b) ensure that no new violations get inserted and (c) secure that the satisfaction up to layer is not decreased. In the definition below, exv() and remv() ensure that (a) is met, with exv() ensuring that a violation is present and remv() ensuring that this violation will be removed. Note, that the construction of these is divided in two cases. Firstly, either $k < \mathrm{nl}(c) - 2$ or the constraints ends with a condition of the form $\exists(a : C \hookrightarrow C', \mathsf{true})$ and secondly, $k = \mathrm{nl}(c) - 2$ and the constraint ends with a condition of the form $\forall(a : C_0 \hookrightarrow C_1, \mathsf{false})$.

For exv(), the first case will be checked as already described above. In the second case, it is sufficient to check whether an occurrence p of C_k with $m(L) \cap p(C_k) \neq \emptyset$ exists.

For remv(), in the first case, a violation can be removed by either deleting an occurrence p of C_k or inserting elements of C_{k+1}, such that $p \not\models \exists C'$ and $\mathrm{tr}_t \circ p \models \exists C'$ for a graph $C' \in \mathrm{IG}(C_k, C_{k+1})$. In the second case, a violation can only be removed by deleting an occurrence p of C_1. This will only occur if $m(L \setminus K) \cap p(C_1) \neq \emptyset$.

The conditions constructed by nui() and nds() ensure that (2) is met, with nui() checking that no new occurrence of C_k will be inserted and nds() ensuring that for an occurrence p of C_k if $p \models \exists C'$ it is secured that $\mathrm{tr}_t \circ p \models \exists C'$ for any $C' \in \mathrm{IG}(C_k, C_{k+1})$.

Additionally, the conditions constructed by ned() and nui() ensure (3), with ned() ensuring that no occurrence of existentially bound graphs C_j with $j \leq k$ will be deleted and nui() ensuring that no occurrences of universally bound graphs C_j with $j \leq k$ will be inserted. With this, it is guaranteed that the satisfaction up to layer will not be decreased.

Definition 3.2 (**general application condition**). *Let a rule $\rho = L \hookleftarrow K \hookrightarrow R$ and a constraint c in UANF be given. Let $d = \mathrm{sub}_k(c) = \forall(a_k : C_k \hookrightarrow C_{k+1}, e)$ with $0 \leq k < \mathrm{nl}(c)$ and k being even.*

The application condition $\mathrm{ap}_k(C')$ of c at layer k with $C' = C_{k+1}$, if $e = \mathsf{false}$, and $C' \in \mathrm{IG}(C_{k+1}, C_{k+2})$ otherwise is defined as:

$$\mathrm{ap}_k(C') := \Big(\bigvee_{P \in \mathrm{ol}(L, C_{k+1})} \mathrm{exv}(P, C') \wedge \mathrm{remv}(P, C') \Big) \wedge \mathrm{nui}() \wedge \mathrm{ned}() \wedge \mathrm{nds}() \qquad (3.1)$$

with

1. *Let $a^r : C_{k+1} \hookrightarrow C'$ be a restricted morphism of a_{k+1} and i_L^P and i_P^Q the inclusions of L in P and P in Q, respectively.*

$$\mathrm{exv}(P, C') := \begin{cases} \exists(i_L^P : L \hookrightarrow P, \mathsf{true}) & \textit{if } e = \mathsf{false} \\ \bigwedge_{Q \in \mathrm{eol}(P,a^r)} \exists(i_L^P : L \hookrightarrow P, \neg\exists(i_P^Q : P \hookrightarrow Q, \mathsf{true})) & \textit{otherwise} \end{cases}$$

2. *If $i_L^P(L \setminus K) \cap i_{C_{k+1}}^P(C_{k+1}) \neq \emptyset$, we set*

$$\mathrm{remv}(P, C') := \mathsf{true}.$$

Otherwise, let P' be the graph derived by the transformation $P \Longrightarrow_{\rho,m} P'$. Then, P' is an overlap of R and C_k. If this transformation does not exist, we set $\mathrm{remv}(P, C') := \mathsf{false}$. *Let $a^r : C_{k+1} \hookrightarrow C'$ be a partial morphism of a_{k+1}, then*

$$\mathrm{remv}(P, C') := \begin{cases} \mathsf{false} & \textit{if } e = \mathsf{false} \\ \bigvee_{Q \in \mathrm{eol}(P',a^p)} \mathrm{Left}(\exists(i_R^Q : R \hookrightarrow Q, \mathsf{true}), \rho) & \textit{otherwise} \end{cases}$$

3. *Let E be the set of all existentially bound graphs graphs C_j with $j \leq k$ and let $\mathbf{P}_{C_j}$ be the set all overlaps P' of L and C_j with $i_L^{P'}(L \setminus K) \cap i_{C_j}^{P'}(C_j) \neq \emptyset$.*

$$\mathrm{ned}() := \bigwedge_{C \in E} \bigwedge_{P' \in \mathbf{P}_{C_j}} \neg\exists(i_L^{P'} : L \hookrightarrow P', \mathsf{true})$$

4. *Let U be the set of all universally bound graphs C_j with $j \leq k$, and let $\mathbf{P}_{C_j}$ be the set of all overlaps P' of R and C_j with $i_R^{P'}(R \setminus K) \cap i_{C_j}^{P'}(C_j) \neq \emptyset$.*

$$\mathrm{nui}() := \bigwedge_{C \in U} \bigwedge_{P' \in \mathbf{P}_{C_j}} \mathrm{Left}(\neg\exists(i_R^{P'} : R \hookrightarrow P', \mathsf{true}), \rho)$$

5. *Let E be the set of all overlaps of L and C_k, such that each $P' \in E$ is also an overlap of L and $C'' \in \mathrm{IG}(C_{k+1}, C_{k+2})$, $i_{C_k}^{P'} \models \exists(a_{k+1}^r : C_{k+1} \hookrightarrow C'', \mathsf{true})$ and $i_L^{P'}(L \setminus K) \cap i_{C''}^{P'}(C'' \setminus C_k) \neq \emptyset$.*

$$\mathrm{nds}() := \bigwedge_{P' \in E} \neg\exists(i_L^{P'} : L \hookrightarrow P', \mathsf{true})$$

Note that $\mathrm{ap}_k(C')$, for any $C' \in \mathrm{IG}(C_{k+1}, C_{k+2})$, will only be evaluated with true if an occurrence p of C_{k+1} with $p \not\models \exists(a_{k+1}^r : C_{k+1} \hookrightarrow C', \mathsf{true})$ and $\mathrm{tr}_t \circ p \models \exists(a_{k+1}^r : C_{k+1} \hookrightarrow C', \mathsf{true})$ exists. For any smaller improvements, i.e a similar improvement for a subgraph $C'' \in \mathrm{IG}(C_{k+1}, C_{k+2})$ of C', $\mathrm{ap}(k, C')$ would be evaluated with false. For any bigger improvements, i.e the same improvement for a supergraph $C'' \in \mathrm{IG}(C_{k+1}, C_{k+2})$ of C', $\mathrm{ap}_k(C')$ would also be evaluated with false, if $p \models \exists(a_{k+1}^r : C_{k+1} \hookrightarrow C', \mathsf{true})$. In both cases, the application condition would prohibit the transformation, even if it would be consistency increasing. To resolve this problem, multiple application conditions could be combined by

$$\bigvee_{C' \in \mathrm{IG}(C_{k+1}, C_{k+2})} \mathrm{ap}_k(C').$$

This application condition will be evaluated with `true` if the cases described above do appear, with the drawback that this leads to a huge condition, even if duplicate conditions are removed. At least all duplicates of ned(), nui() and nds() can be removed, since they are identical for each $\mathrm{ap}(k, C')$ and only need to be constructed once.

In general, these application conditions are a trade-off between conditions-size and restrictiveness. They are very restrictive, since they do not allow any deletions of occurrences of existentially bound and insertions of universally bound graphs. For example, any of these application conditions with the rule `moveFeature` and constraint c_1 will be equivalent to false; nds() will always be evaluated with false since `moveFeature` does remove elements of the existentially bound graph C_2^1. A change of the conditions constructed by nds() such that it is checked whether two nodes of the type `Feature` are connected to a node `Class` will yield application conditions that are satisfiable with `moveFeature`, but for a similar rule moving two nodes of type `Feature`, this newly constructed nds() would still be evaluated with false. Therefore, this only leads to a slight decrease of restrictiveness.

The conditions constructed by ned() and nui() could be changed in a similar fashion. For ned() and the universally bound graph C_j, by checking whether there does exist an additional occurrence p of C_{j+1} such that $p \models \mathrm{sub}_{j+2}(\mathrm{cut}_{\mathrm{k_{max}}}(c))$ and for nui(), by checking whether an introduced occurrence p of C_j does satisfy $\mathrm{sub}_{j+1}(\mathrm{cut}_{\mathrm{k_{max}}}(c))$. The construction of these is similar to the construction of consistency guaranteeing application conditions as introduced by Habel and Pennemann [1], which is known to construct huge application conditions. Also, they do get more and more restrictive for increasing k, since the number of conditions constructed by ned() and nui() also increases.

For constraints of the form $c := \forall C_0 \exists C_1$ it can be shown that $\mathrm{ap}_0(C_2)$ is not only a direct consistency increasing, but also a direct consistency improving application condition.

Example 3.1. *Consider the rule* `assignFeature` *of figure 4 and constraint c_1 of figure 1. The application condition of c_1 at layer* 1 *with C_2^1 for* `assignFeature` *constructed by definition 3.2 is shown in figure 5. The first two rows are conditions constructed by* $\mathrm{exv}(\cdot, \cdot)$ *and the third row is the condition constructed by* $\mathrm{remv}(\cdot, \cdot)$. *Note that* ned(), nui() *and* nwo() *did not construct any conditions since* `assignFeature` *does not create elements of C_1^1 and does not delete elements of C_2^1.*

Additionally, this application condition is also a consistency improving application condition w.r.t c_2.

Let us now show that the construction above generates consistency increasing application conditions.

Theorem 3.1. *Let a graph G, a constraint c in UANF with $G \not\models c$ and a plain rule ρ be given. Then, $\rho' = (\rho, \mathrm{ap}_{\mathrm{k_{max}}+1}(C))$ with $C \in \mathrm{IG}(C_{\mathrm{k_{max}}+1}, C_{\mathrm{k_{max}}+2})$ is a consistency increasing rule.*

Proof. Let $t : G \Longrightarrow_{\rho',m} H$ be a transformation and $k = \mathrm{k_{max}}+1$. We show, by contradiction, that this transformation is direct consistency increasing by showing that (2.1), (2.2), (2.3), (2.4) and (2.5) are satisfied. With that, ρ' is a consistency increasing rule. Let $\mathrm{sub}_k(c) = \forall(a_k : C_k \hookrightarrow C_{k+1}, e)$ be the subcondition of c at layer k. Let

$$\mathbf{G} := \begin{cases} \mathrm{IG}(C_{k+1}, C_{k+2}) & \text{if } e \neq \mathsf{false} \\ \{C_k\} & \text{otherwise.} \end{cases}$$

1. Assume that (2.1) does not hold. Then, a morphism $p : C_{k+1} \hookrightarrow G$ exists, such that $p \models \mathrm{IC}_0(e, C')$, $\mathrm{tr}_t \circ p$ is total and $\mathrm{tr}_t \circ p \not\models \mathrm{IC}_0(e, C')$ for a graph $C' \in \mathbf{G}$. Therefore, an overlap P of L and C' such that $i^P_{C_{k+1}} \models \exists(a^r_{k+1} : C_{k+1} \hookrightarrow C', \mathsf{true})$ with $i^P_L(L \setminus K) \cap i^P_{C'}(C' \setminus C_{k+1}) \neq \emptyset$ exists and $m \models \exists(i^P_L : L \hookrightarrow P, \mathsf{true})$ holds. Thus, nds() and consequently also $\mathrm{ap}_k(C)$ cannot be satisfied.

2. Assume that (2.2) does not hold and let

   $$d := \begin{cases} \mathrm{IC}_0(e, C_{k+2}) & \text{if } e \neq \mathsf{false} \\ \mathsf{false} & \text{otherwise.} \end{cases}$$

   Then, a morphism $p' : C_{k+1} \hookrightarrow H$ with $p' \not\models d$ exists, such that no morphism $p : C_{k+1} \hookrightarrow G$ with $\mathrm{tr}_t \circ p = p'$ exists. Therefore, an overlap P of R and C_{k+1} with $i^P_R(R \setminus K) \cap i^P_{C_{k+1}}(C_{k+1}) \neq \emptyset$ exists, such that $m \models \mathrm{Left}(\exists(i^P_R : R \hookrightarrow P, \mathsf{true}), \rho)$ and m does not satisfy nui().

3. Assume that (2.3) does not hold. Then, no morphism $p : C_{k+1} \hookrightarrow G$ with $p \not\models \mathrm{IC}_0(e, C')$, such that $\mathrm{tr}_t \circ p$ is not total or $\mathrm{tr}_t \circ p \models \mathrm{IC}_0(e, C')$ exists, for any $C' \in \mathbf{G}$. Then, no overlap P of L and C_{k+1} with $i^P_L(L \setminus K) \cap i^P_{C_{k+1}}(C_{k+1}) \neq \emptyset$ exists, such that $m \models \mathrm{exv}(P, C)$.

   Let P be an overlap of C_{k+1} and L with $i^P_L(L \setminus K) \cap i^P_{C_{k+1}}(C_{k+1}) = \emptyset$. If $e = \mathsf{false}$, then $\mathrm{remv}(P, C) = \mathsf{false}$. Otherwise, if $m \models \mathrm{exv}(P, C) \wedge \mathrm{remv}(P, C)$, it holds that $i^P_{C_{k+1}} \not\models \mathrm{IC}_0(e, C)$ and a graph $Q \in \mathrm{eol}(P', a^r)$ exists, such that $m \models \mathrm{Left}(\exists(i^Q_R : R \hookrightarrow Q, \mathsf{true}), \rho)$. In this case, $\mathrm{tr}_t \circ i^P_{C_{k+1}} \models \mathrm{IC}_0(e, C)$ follows and (2.3) is satisfied.

   In total, $m \not\models \mathrm{exv}(P, C) \wedge \mathrm{remv}(P, C)$ follows for all $P \in \mathrm{ol}(L, C_{k+1})$ and therefore $m \not\models \mathrm{ap}_k(C)$.

4. Assume that (2.4) does not hold. Then, a morphism $p : C_j \hookrightarrow H$ with C_j being universally bound and $j < k$ exists, such that no morphism $p' : C_j \hookrightarrow G$ with $\mathrm{tr}_t \circ p' = p$ exists. Then, an overlap P of C_j and R with $i^P_R(R \setminus K) \cap i^P_{C_j}(C_j) \neq \emptyset$ exists, such that $m \models \mathrm{Left}(\exists(i^P_R : R \hookrightarrow P, \mathsf{true}), \rho)$. Hence, $m \not\models \mathrm{nui}()$ and $m \not\models \mathrm{ap}_k(C)$.

5. Assume that (2.5) does not hold. Then, a morphism $p : C_j \hookrightarrow G$ with C_j being existentially bound and $j < k$ exists, such that $\mathrm{tr}_t \circ p$ is not total. Then, an overlap P of C_j and L with $i^P_L(L \setminus K) \cap i^P_{C_j}(C_j) \neq \emptyset$ exists, such that $m \models \exists(i^P_L : L \hookrightarrow P, \mathsf{true})$. Hence, $m \not\models \mathrm{ned}()$ and $m \not\models \mathrm{ap}_k(C)$.

In total follows, if $\models \mathrm{ap}_k(C)$, then t is a direct consistency increasing transformation. □

3.1 Conflicts within and between conditions

Definition 3.3 (Conflicts within condition). *Let a condition c in UANF be given. Then, an existentially bound graph C_k has a* conflict *with an universally bound graph C_j if a transformation $t : G \Longrightarrow_\rho H$ with $\rho = C_{k-1} \overset{\mathrm{id}}{\hookleftarrow} C_{k-1} \xhookrightarrow{a_{k-1}} C_k$ exists such that the following holds*

$$\exists p : C_j \hookrightarrow H(\neg\exists p' : C_j \hookrightarrow G(\mathrm{tr}_t \circ p' = p)).$$

An universally bound graph C_k is has a conflict *with an existentially bound graph C_j if a transformation $t : G \Longrightarrow_\rho H$ with $\rho = C_k \xhookleftarrow{i_{C'}} C' \overset{\mathrm{id}}{\hookrightarrow} C'$ and $C' \in \mathrm{IG}(C_{k-1}, C_k)$ exists such that the following holds.*

$$\exists p : C_j \hookrightarrow G(\mathrm{tr}_t \circ p \textit{ is not total}).$$

A condition c in UANF is called conflict free *if no graph C_k in c has a conflict with a graph C_j with $0 \leq j < k \leq \mathrm{nl}(c)$. A graph C_k has a* transitive conflict *with $C_{j'}$ a graph C_j exists such that C_k has a conflict with C_j and C_j has a (transitive) conflict with $C_{j'}$. A condition c does contain a* circular conflict *if a graph C_k exists such that C_k has a transitive conflict with itself. Otherwise, c is called* circular conflict free*.*

Lemma 3.4. *Let a constraint c in UANF be given.*

1. *Let C_k be an existentially bound graph of c. Then, C_k has a conflict with C_j if and only if an overlap P of C_k and C_j exists such that*

$$i^P_{C_k}(C_k \setminus C_{k-1}) \cap i^P_{C_j}(C_j) \neq \emptyset$$

and the rule $\rho = C_k \overset{l}{\hookleftarrow} C_{k-1} \overset{r}{\hookrightarrow} C_{k-1}$ is applicable at match $i^P_{C_k}$.

2. *Let C_k be an universally bound graph of c. Then C_k has a conflict with C_j if and only if an overlap P of C_k and C_j exists such that*

$$i^P_{C_k}(C_k \setminus C_{k-1}) \cap i^P_{C_j}(C_j \setminus C_{j-1}) \neq \emptyset$$

and each rule $\rho = C_k \xhookleftarrow{i_{C'}} C' \overset{\mathrm{id}}{\hookrightarrow} C'$ with $C' \in \mathrm{IG}(C_{k-1}, C_k)$ is applicable at match $i^P_{C_k}$.

Proof. Let a condition c in UANF be given.

1. "$\Longrightarrow$": Let C_k be an existentially bound graph that has a conflict with an universally bound graph C_j. Then, there does exists a transformation $t : G \Longrightarrow_\rho H$ with $\rho = C_{k-1} \overset{l}{\hookleftarrow} C_{k-1} \overset{r}{\hookrightarrow} C_k$ such that a new occurrence p of C_j has been inserted. Since only elements of $C_k \backslash C_{k-1}$ have been inserted it holds that $p(C_j) \cap n(C_k \backslash C_{k-1})$ with n being the comatch of t. The graph $p(C_j) \cup n(C_k)$ is the searched for overlap and the rule $\rho^{-1} = C_k \overset{l}{\hookleftarrow} C_k \overset{r}{\hookrightarrow} C_{k-1}$ has to be applicable at n.
   "$\Longleftarrow$": Let C_k be an existentially and C_j an universally bound graph such that an overlap P of C_k and C_j with $i^P_{C_k}(C_k \setminus C_{k-1}) \cap i^P_{C_j}(C_j) \neq \emptyset$ exists such that the rule $\rho = C_k \overset{l}{\hookleftarrow} C_k \overset{r}{\hookrightarrow} C_{k-1}$ is applicable at match $i^P_{C_k}$. Then, the inverse transformation of $t : P \Longrightarrow_{\rho, i^P_{C_k}} H$ is the searched for transformation and C_k has a conflict with C_j.

2. "$\Longrightarrow$": Let C_k be an universally bound graph that has a conflict with an existentially bound graph C_j. Then, a transformation $t : G \Longrightarrow_\rho H$ with $\rho = C_k \overset{l}{\hookleftarrow} C' \overset{r}{\hookrightarrow} C'$ and C' being a subgraph of C_k exists such that a occurrence p of C_j has been deleted. Then, the graph $p(C_j) \cup m(C_k)$ is the searched for overlap and $i^P_{C_k}(C_k \setminus C_{k-1}) \cap i^P_{C_j}(C_j \setminus C_{j-1}) \neq \emptyset$ has to hold since ρ only deletes elements of $C_k \setminus C_{k-1}$.
   "$\Longleftarrow$": Let C_k be universally and C_j existentially bound such that an overlap P of C_k and C_j with $i^P_{C_k}(C_k \setminus C_{k-1}) \cap i^P_{C_j}(C_j \setminus C_{j-1}) \neq \emptyset$ exists such that each rule $\rho = C_k \overset{i_{C'}}{\hookleftarrow} C' \overset{\text{id}}{\hookrightarrow} C'$ with $C' \in \text{IG}(C_{k-1}, C_k)$ is applicable at match $i^P_{C_k}$. Then, the inverse transformation of $t : P \Longrightarrow_{\rho, i^P_{C_k}} H$ is the searched for transformations and C_k has a conflict with C_j.

$\square$

Definition 3.5 (**Conflicts between conditions**). *Let conditions c_1 and c_2 be given. Then, c_1 has a conflict with c_2 if graphs C_k and C_j of c_1 and $c_2$exist such that C_k has a conflict with C_j or vice versa. A set of conditions is called* conflict free *if no conditions c_1 and c_2 exist such that c_1 has a conflict with c_2 or vice versa. A condition c_1 has a* transitive conflict with *c_2 if a condition c' exists such that c_1 has a conflict with c' and c' has a (transitive) conflict with c_2. A set of conditions does contain a* circular conflict *if a condition c has a transitive conflict with itself. Otherwise, the set is called* circular conflict free.

Definition 3.6 (**conflict free graphs**). *Let a graph G and a constraint c in UANF be given. Then, G is called* conflict free *w.r.t c if for each transformation $t : G \Longrightarrow_{\rho,m} H$ with*

$$\rho = C_{\text{k}_{\max}} \overset{\text{id}}{\hookleftarrow} C_{\text{k}_{\max}} \overset{i_{C_{\text{k}_{\max}}}}{\hookrightarrow} C'$$

and $C' \in \text{IG}(C_{\text{k}_{\max}+1}, C_{\text{k}_{\max}+2})$ *or*

$$\rho = C_{\text{k}_{\max}+1} \overset{1_{C'}}{\hookleftarrow} C' \overset{\text{id}}{\hookrightarrow} C'$$

and $C' \in \mathrm{IG}(C_{\mathrm{k_{max}}}, C_{\mathrm{k_{max}}+1})$ *and* (2.4) *and* (2.5) *hold.*

Lemma 3.7. *Let a graph G and a constraint c in UANF be given. Let P be the set of all Then, G is conflict free w.r.t c if and only if either c is conflict free or*

$$G \models \bigwedge_{P \in P} \forall(a : \emptyset \hookrightarrow P, \mathit{false})$$

Proof. □

3.2 Basic increasing rules

As described above, the application conditions for general rules are very restrictive. This is because the application condition does not allow a transformation to delete occurrences of existentially bound or insert occurrences of universally bound graphs even if this would not lead to a decrease of satisfaction up to layer. Now, we will consider a special set of rules called *basic increasing rules*. The main idea is that these rules are not able to delete occurrences of existentially or insert occurrences of universally bound graphs and additionally are able to increase consistency. That means, given a basic increasing rule ρ, there does exist a transformation $t : G \Longrightarrow_\rho H$ such that t is a consistency increasing transformation w.r.t to a constraint c in UANF. For *basic increasing rules*, consistency increasing application conditions can be constructed that are less restrictive and smaller in size compared to application conditions for general rules.

First, we define *non-consistency decreasing rules at layer* and *non-consistency decreasing rules up to layer* where a non-consistency decreasing rule at layer k is not able to delete occurrences of C_{k+1}, if $\mathrm{sub}_k(c)$ is existentially bound and not able to create occurrences of C_{k+1} if $\mathrm{sub}_k(c)$ is universally bound. This can be checked via second order formulas by, in the first case, checking whether all occurrences of C_{k+1} still exist in H and, in the second case, checking whether all occurrences of C_{k+1} in H already existed in G. The notion of non-consistency decreasing rules up to layer k is a abbreviation, meaning that a rule ρ is non-consistency decreasing up to layer k if it is non-consistency decreasing at layer j for all $0 \leq j \leq k$.

Definition 3.8 (**non-consistency decreasing rule**). *Let a plain rule $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ and a constraint c in UANF be given. Then, ρ is called a* non-consistency decreasing rule w.r.t c at layer k *if*

1. *If $\mathrm{sub}_k(c)$ is universally bound and for all transformations $t : G \Longrightarrow_\rho H$ it holds that*
$$\forall p : C_{k+1} \hookrightarrow H (\exists p' : C_{k+1} \hookrightarrow G (\mathrm{tr}_t \circ p' = p)). \tag{3.2}$$
2. *If $\mathrm{sub}_k(c)$ is existentially bound for all transformations $t : G \Longrightarrow_\rho H$ it holds that*
$$\forall p : C_{k+1} \hookrightarrow G (\mathrm{tr}_t \circ p \textit{ is total}). \tag{3.3}$$

The rule ρ is called non-consistency decreasing w.r.t c up to layer k *if ρ is non-consistency decreasing w.r.t c at layer j for all $0 \leq j \leq k$ and it holds that*

$$\bigwedge_{C' \in \mathrm{IG}(C_k, C_{k+1})} \forall p : C' \hookrightarrow G(\mathrm{tr}_t \circ p \textit{ is total}) \tag{3.4}$$

if scondkc is existentially bound.

Given a graph G and a constraint c, we will show that non-decreasing rules w.r.t c up to layer $\mathrm{k_{max}}$ cannot decrease the satisfaction up to layer.

Lemma 3.9. *Let a graph G, a constraint c in UANF and a non-consistency decreasing rule w.r.t c up to layer $\mathrm{k_{max}}$ be given. Then, for each transformation $t : G \Longrightarrow_\rho H$ it holds that*

$$\mathrm{k_{max}}(c, G) \leq \mathrm{k_{max}}(c, H).$$

Proof. Assume that $\mathrm{k_{max}}(c, G) > \mathrm{k_{max}}(c, H)$. Then, either 1. or 2. applies.

1. There does exists an occurrence $p : C_j \hookrightarrow H$ with $0 \leq j \leq \mathrm{k_{max}}(c, G)$ of a universally bound graph C_j such that there does not exist a occurrence $q : C_j \hookrightarrow G$ of C_j with $p = \mathrm{tr}_t \circ q$. Then ρ is not a non-consistency decreasing rule up to layer $\mathrm{k_{max}}(c, G)$ since (3.2) is not satisfied.

2. There does exists an occurrence $p : C_j \hookrightarrow G$ with $0 \leq j \leq \mathrm{k_{max}}(c, G)$ of an existentially bound graph C_j such that $\mathrm{tr} \circ p$ is not total. Then, ρ is not a non-consistency decreasing rule up to layer $\mathrm{k_{max}}(c, G)$ since (3.3) is not satisfied.

□

Since it is very time consuming to check whether a rule ρ is non-consistency decreasing based on the definition above, we present a characterisation for non-consistency decreasing rules which only relies on ρ itself. First, let us assume that ρ is able to create occurrences of a universally bound graph C_j. This is possible if (a) ρ does insert an edge of $C_j \setminus C_{j-1}$ which connects already existing nodes of C_j, since it is unclear whether this would create a new occurrence of C_j. Or (b) if ρ does insert a node v of C_j such that all edges $e \in E_{C_j}$ with $\mathrm{src}(e) = v$ or $\mathrm{tar}(e) = v$ are also inserted. If at least one of these edges is not inserted, it is guaranteed that this insertion does not create an occurrence of C_j since v is only connected to edges that have also been inserted by ρ.

Second, let us assume that ρ is able to delete occurrences of a existentially bound graph C_j. This is possible if (a) ρ does delete an edge of $C_j \setminus C_{j-1}$ or (b) ρ does delete a node v of $C_j \setminus C_{j-1}$ such that all edges $e \in E_{C_j}$ with $\mathrm{src}(e) = v$ or $\mathrm{tar}(e) = v$ are also deleted. If ρ deletes a node c of $C_j \setminus C_{j-1}$ without all its connected edges in C_j there does not exist a transformation such that ρ deletes an occurrence of C_j by deleting this node since the dangling edge condition is not satisfied.

Then, we are able to determine whether a rule ρ is non-consistency decreasing by checking whether ρ does not satisfy the just mentioned properties.

Lemma 3.10. *Let a plain rule $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ and a constraint c in EANF be given. Then, ρ is a non-consistency decreasing rule at layer k w.r.t c if either 1. or 2. applies.*

1. *If C_k is universally bound, let I be the set of all isolated nodes of C_k, P be the set of all partial morphisms $p : C_k \hookrightarrow R$,*

$$E = \bigcup_{p \in P} p(E_{C_k}) \cap (E_R \setminus E_K) \text{ and } V = \bigcup_{p \in P} p(V_{C_k}) \cap (V_R \setminus V_K).$$

   Then, ρ is called a non-consistency decreasing rule at layer k w.r.t c *if $I \cap V = \emptyset$ and*

$$\begin{aligned}&\big(\neg \exists e \in E : \exists v, v' \in V_K : \operatorname{src}(e) = r(v) \wedge \operatorname{tar}(e) = r(v')\big) && \wedge\\ &\big(\forall v \in V : \exists e \in E_{C_k} : \exists p \in P : \operatorname{src}(e) = p^{-1}(v) \vee \operatorname{tar}(e) = p^{-1}(v)\\ &\wedge \neg \exists e' \in E : \exists p \in P : p^{-1}(e') = e\big)\end{aligned}$$

   holds.

2. *If C_k is existentially bound, let I be the set of all isolated nodes of C_k, P be the set of al morphisms $p : C_k \hookrightarrow L$,*

$$E = \bigcup_{p \in P} p(E_{C_k}) \cap (E_L \setminus E_K) \text{ and } V = \bigcup_{p \in P} p(V_{C_k}) \cap (V_L \setminus V_K).$$

Let I be the set of all isolated nodes of C_k. Then, ρ is called a non-consistency decreasing rule at layer k w.r.t c *if $I \cap V = \emptyset$ and*

$$\begin{aligned}&\big(\neg \exists e \in E_{C_k} \setminus E_{C_{k-1}} : \exists p \in P : \exists e' \in E : p(e) = e'\big) && \wedge\\ &\big(\forall v \in V : \exists e \in E_{C_k} : \exists p \in P : \operatorname{src}(e) = p^{-1}(v) \vee \operatorname{tar}(e) = p^{-1}(v)\\ &\wedge \neg \exists e' \in E : \exists p \in P : p^{-1}(e') = e\big)\end{aligned}$$

Proof. □

Now we are ready to define *basic increasing rules at layer k* with the set of basic increasing rules being a subset of the set of non-decreasing rules. With the property that each basic increasing rule ρ is also a non-decreasing rule up to layer k it is satisfied that the satisfaction up to layer is not decreased. Additionally, if $\text{sub}_k(c)$ is existentially bound it has to be checked that the consistency for an occurrence of C_k is not decreased and that ρ removes an violation. This can be done by either deleting an element of $C_{k+1} \setminus C_k$ if $\text{sub}_k(c)$ is universally bound or by inserting elements of C_{k+1} such that for one occurrence p of C_k the consistency in increased if $\text{sub}_k(c)$ is existentially bound. Additionally, the left-hand side of each basic increasing rule has to contain C_{k+1} if $\text{sub}_k(c)$ is universally bound such that either C_{k+1} gets deleted and the left-hand side has to contain C_k if $\text{sub}_k(c)$ is existentially bound such that the consistency of C_k is increased. This property yields the advantage that application conditions for basic increasing rules are less complex, smaller in size and during a repair process matches of these rules are easier to handle.

This, on first sight seems like a restriction of the set of basic increasing rules but the context of each rule ρ that does satisfy all properties of a basic increasing rule excluding that C_k or C_{k+1} is a subgraph of the left-hand side can be expanded such that ρ is a basic increasing rule and the semantic of ρ is not increased. Later on, a method to derive this rules will be presented.

Basic increasing rules at layer k are called *deleting basic increasing rules* if $\mathrm{sub}_k(c)$ is universally bound and *inserting basic increasing rules* if $\mathrm{sub}_k(c)$ is existentially bound. For deleting basic increasing rules we introduce the restriction that these rules are only allowed to delete edges but no nodes of C_{k+1} since otherwise it is not possible, given a rule set and a constraint to decide whether this rules set is able to repair an arbitrary graph based only on deleting basic increasing rules. For example, consider a rule that deletes a node of C_{k+1}. Then, it is unknown whether this node is connected by edges not belonging to C_{k+1} and it is unclear whether all occurrence of C_{k+1} could be destroyed by ρ since the dangling edge condition could be unsatisfied.

Definition 3.11 (**basic increasing rule**). *Let a constraint c in UANF and a plain rule $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ be given. Then, ρ is called* basic increasing w.r.t c at layer k *if*

1. *If $\mathrm{sub}_k(c)$ is universally bound: ρ is a non-consistency decreasing rule up to layer $k+1$ and there does exist a morphism $i : C_{k+1} \hookrightarrow L$ such that*
$$\exists e \in E_{C_{k+1}} \setminus E_{C_k}\big(\neg\exists e' \in E_K(i(e) = l(e'))\big)\wedge$$
$$\forall v \in V_{C_{k+1}} \setminus V_{C_k}\big(\exists v' \in V_K(i(c) = l(v')\big)$$
holds. Then, ρ is called a deleting basic increasing rule.

2. *If $\mathrm{sub}_k(c)$ is existentially bound: ρ is a non-consistency decreasing rule up to layer k and there does exist a morphism $i : C_k \hookrightarrow L$ such that*
$$i \not\models \exists(a_k^r : C_k \hookrightarrow C', \mathit{true}) \wedge r \circ l^{-1} \circ i \models \exists(a_k^r : C_k \hookrightarrow C', \mathit{true})$$
for any $C' \in \mathrm{IG}(C_k, C_{k+1})$. Then, ρ is called a inserting basic increasing rule with *C'.*

Again, 3.11 relies on every transformation of a rule ρ. Therefore, we present an alternative method to determine whether ρ satisfies 3.11 or not by checking that ρ does not delete any edges or nodes of $C_{k+1} \setminus C_k$.

Lemma 3.12. *let the sets E and V be constructed as introduced in lemma 3.10 it holds that*
$$\forall v \in V : \exists v' \in V_K : l(v') = v\wedge$$
$$\forall e \in E : \exists e' \in E_K : l(e') = e$$
Then, ρ is called an inserting basic increasing rule *and i is called the increasing morphism of ρ.*

Proof. □

Lemma 3.13. *Let a constraint c in UANF, a basic increasing rule ρ at layer* $\mathrm{k_{max}}+1$ *and a transformation $t : G \Longrightarrow_\rho H$ be given. Then,*

$$t \text{ is consistency increasing w.r.t } c \iff t \text{ is direct consistency increasing w.r.t } c$$

Proof. The "$\Longleftarrow$" side of the equivalence holds with lemma 2.16. We need to show that the "$\Longrightarrow$" side of the equivalence holds. Let $t : G \Longrightarrow_\rho H$ be a consistency increasing transformation and ρ a basic increasing rule. Then, t satisfies (3.11) and the satisfaction of (2.1) follows immediately. Since ρ is also a non-consistency decreasing rule up to layer $\mathrm{k_{max}}$, t satisfies (3.2) and (3.3), then the satisfaction (2.2), (2.4) and (2.5) also follows immediately.

1. If ρ is a deleting basic increasing rule. Since t is a consistency increasing rule there has to exist a morphism $p : C_{\mathrm{k_{max}}+2} \hookrightarrow G$ such that $\mathrm{tr}_t \circ p$ is not total and with that (2.3) is satisfied.
2. If ρ is a inserting basic increasing rule there exists an occurrence $p : C_{\mathrm{k_{max}}+2} \hookrightarrow G$ and a graph $C \in \mathrm{IG}(C_{\mathrm{k_{max}}+2}, C_{\mathrm{k_{max}}+3})$ such that $p \not\models \mathrm{IC}_{\mathrm{k_{max}}+2}(c, C)$ and $\mathrm{tr}_t \circ p \not\models \mathrm{IC}_{\mathrm{k_{max}}+2}(c, C)$. Therefore (2.3) is satisfied.

In total follows that t is a direct consistency increasing transformation w.r.t c. □

For basic increasing rules at layer $j < \mathrm{k_{max}}+1$ this equality does not hold since these rules are allowed to destroy occurrences of universally bound graphs $C_{j'}$ or to create occurrences of existentially bound graphs $C_{j'}$ with $j < j' < \mathrm{k_{max}}+1$. With this results follow that if $t : G \Longrightarrow_\rho H$ is a c-guaranteeing transformation and ρ is a basic increasing rule at layer $\mathrm{k_{max}}+1$, t is also a direct increasing rule w.r.t c.

Definition 3.14 (**application conditions for basic increasing rules**)**.** *Let a constraint c in UANF and a basic increasing rule $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ w.r.t c at layer k be given. We define the application condition for ρ as:*

1. *If ρ is a deleting increasing rule:*

$$\mathrm{basic}_j(\rho) := \begin{cases} \bigvee_{P \in \mathrm{ol}(L, C_k)} \bigwedge_{P' \in \mathrm{eol}(P, a_k)} \exists(i_L^P : L \hookrightarrow P, \neg\exists(i_P^{P'} : P \hookrightarrow P', \mathit{true})) & \text{if } j = k \\ \mathit{false} & \text{if } j \neq k. \end{cases}$$

2. *If ρ is an inserting increasing rule. Let $a^p : C_k \hookrightarrow C$ be a partial morphism of a_k , $e : C_k \hookrightarrow L$ be the increasing morphism of c and Q be set of all overlaps P of L and C such that $i_L^P \circ e \models \exists(a^p : C_k \hookrightarrow C, \mathit{true})$:*

$$\mathrm{basic}_j(\rho) := \begin{cases} \bigvee_{P \in Q} \exists(i_L^P : L \hookrightarrow P, \mathit{true}) & , \text{if } j = k \\ \mathit{false} & , \text{if } j \neq k \end{cases}$$

Theorem 3.2. *Let a graph G, a constraint c and a basic increasing rule ρ at layer $\mathrm{k_{max}} < k \leq \mathrm{k_{max}} + 2$ be given. Then, each transformation*

$$t : G \Longrightarrow_{\rho', m} H$$

with the rule $\rho' = (\rho, \mathrm{basic}_k(\rho))$ is a direct consistency increasing transformation.

Proof. Since ρ is a basic increasing rule it is also a non-consistency decreasing rule up to layer $\mathrm{k_{max}} + 2$. Therefore, (2.1) is trivially satisfied since ρ satisfies (3.4), also (2.2) is satisfied since ρ satisfies (3.2) also (2.4) and (2.5) are satisfied because ρ satisfies (3.2) and (3.3) for all $0 \leq j \leq k + 2$. It remains to show that (2.3) is satisfied.

1. Let ρ be a deleting basic increasing rule. Then, $k = \mathrm{k_{max}} + 1$ and $\mathrm{basic}_k(\rho) = \bigvee_{P \in \mathrm{ol}(L, C_k)} \bigwedge_{P' \in \mathrm{eol}(P, a_k)} \exists(i_L^P : L \hookrightarrow P, \neg\exists(i_P^{P'} : P \hookrightarrow P', \mathsf{true}))$. If $m \models \mathrm{basic}_j(\rho)$ there does exist a occurrence $p : C_k \hookrightarrow G$ with $p \not\models \exists(a_{k+1} : C_k \hookrightarrow C_{k+1}, \mathsf{true})$ and $p(C_k \setminus C_{k-1}) \cap m(L \setminus K) \neq \emptyset$. If follows that $\mathrm{tr}_t \circ p$ is not total and (2.3) is satisfied.
2. Let ρ be a inserting basic increasing rule with $C \in \mathrm{IG}(C_k, C_{k+1})$. Then, $k = \mathrm{k_{max}} + 1$ and $\mathrm{basic}_j(\rho) = \bigvee_{P \in Q} \exists(i_L^P : L \hookrightarrow P, \mathsf{true})$. If $m \models \mathrm{basic}_j(\rho)$, $m \circ e \not\models \exists(a_k^r : C_k \hookrightarrow C, \mathsf{true})$ with e being the increasing morphism of ρ. Therefore $\mathrm{tr}_t \circ m \circ i \models \exists(a_k^r : C_k \hookrightarrow C, \mathsf{true})$ and (2.3) is satisfied.

In total follows that t is a direct consistency increasing transformation. □

Theorem 3.3. *Let a constraint c in EANF and a set of rules $\mathcal{R}$ be given. Then, $\mathcal{R}$ is a repairing set of c at layer $k \leq \mathrm{nl}(c)$ if either 1 or 2 applies.*

1. *For any universally bound graph C_j at layer $j \leq k$ of c, $(\rho, \mathrm{basic}(j, C_{j+1})) \in \mathcal{R}$ and ρ is a deleting potentially minimal improving rule at layer j with C_{j+1}, such that ρ only deletes edges of C_j.*
2. *A decomposition $\mathbf{P}$ of C_k with C_{k-1} exists, such that for each $P \in \mathbf{P}$ a rule $(\rho, \mathrm{basic}(k, P)) \in \mathcal{R}$ exists, such that ρ is an inserting basic improving rule at layer k with P.*
3. *There does exist a sequence of graphs*

$$C_k = C'_0 \hookrightarrow C'_1 \hookrightarrow \ldots \hookrightarrow C'_n = C_{k+1}$$

such that a inserting basic increasing rule $\rho_i = L_i \overset{l_i}{\hookleftarrow} K_i \overset{r_i}{\hookrightarrow} R_i$ at layer k with C_i exists such that $L_i \hookrightarrow R_{i-1}$ and $e_i = i_{Ci-1}^{Ci} \circ e_{i-1}$ with e_i being the increasing morphism of ρ_i for all $i \in \{1, \ldots n\}$.

Proof. Let a constraint c in EANF, a rule set $\mathcal{R}$ and a graph G with $k = \mathrm{c_{max}}$ and $\mathrm{c_{max}} < \mathrm{nl}(c)$ be given. We show that a sequence $G = C'_0 \Rightarrow \ldots \Rightarrow C'_n = H$ with rules of $\mathcal{R}$ exists, such that $H \models_{k+2} c$ if 1. or 2. of theorem 3.3 is satisfied.

1. Assume that 1. of theorem 3.3 holds. Let $(\rho, \text{basic}(j, C_{j+1})) \in \mathcal{R}$, such that $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ is a deleting potentially minimal improving rule at layer $j \leq k$ with C_{j+1} and C_j is a universally bound graph of c. Then, $\text{basic}(j, C_{j+1}) = \exists\big(b : L \hookrightarrow C_j, \neg\exists(a_j : C_j \hookrightarrow C_{j+1}, \mathsf{true})\big)$. Let $q : C_j \hookrightarrow G$ be a morphism such that $q \not\models \exists(a_j : C_j \hookrightarrow C_{j+1}, \mathsf{true})$. Since $L \subseteq C_j$, we can construct a morphism $m_1 : L \hookrightarrow G$ with $m_1 = q \circ b$ and therefore $m_1 \models \text{basic}(j, C_{j+1})$. Since r only deletes edges, a transformation $t : G = G_0 \Rightarrow_{r,m_1} G_1$ exists and $\text{tr}_t \circ p$ is not total. Because r does not insert any elements of C_j:

   $$|\{q : C_k \hookrightarrow G_0 \mid q \not\models d\}| < |\{q : C_k \hookrightarrow G_1 \mid q \not\models d\}|$$

   with $d = \exists(a_j : C_j \hookrightarrow C_{j+1}, \mathsf{true})$. By iteratively applying this construction, we can generate a finite sequence of transformations

   $$G = G_0 \Rightarrow_{r,m_1} G_1 \Rightarrow_{r,m_2} \ldots \Rightarrow_{r,m_n} G_n = H$$

   such that $|\{q : C_k \hookrightarrow G_n \mid q \not\models d\}| = 0$ and therefore $H \models_j c$. With lemma 2.9, $H \models_{k+2} c$ and $H \models c$ follows.

2. Assume that 2. of theorem 3.3 holds. Let $\rho_0 = (\rho, \text{basic}(k, P)) \in \mathcal{R}$, such that ρ is an inserting basic improving rule of c at layer k with $P \in \mathbf{P}$. Then, $\text{basic}(k, P) = \neg\exists(b : L \hookrightarrow P, \mathsf{true})$. Let $q_0 : C_k \hookrightarrow G$ be a morphism, such that $q_0 \not\models \exists(a'_k : C_k \hookrightarrow P, \mathsf{true})$ with a'_k being a partial morphism of a_k. Since $L = C_k$, we set $m_0 : C_k \hookrightarrow G$ with $m_0 = q_0$. It follows that $m_0 \models \neg\exists(a'_k : C_k \hookrightarrow P, \mathsf{true}) = \text{basic}(k, P)$. Because r does not delete any elements, a transformation $t_0 : G \Longrightarrow_{r_0,m_0} G_1$ exists and $\text{tr}_t \circ q \models \exists(a'_k : C_k \hookrightarrow P, \mathsf{true})$. We set $q_1 = \text{tr}_{t_0} \circ q_0$ and apply the same method to q_1.

   By iteratively applying this, we can construct a finite sequence of transformations

   $$G \Longrightarrow_{r_0,m_0} G_0 \Longrightarrow_{r_1,m_1} \ldots \Longrightarrow_{r_n,m_n} G_n$$

   such that $m_i = \text{tr}_{t_{i-1}} \circ \ldots \circ \text{tr}_{t_0} \circ m_0$ and $q \models \exists(b_i : C_k \hookrightarrow P_i, \mathsf{true})$ for all $P_i \in \mathbf{P}$ with $q = \text{tr}_{t_n} \circ q_n$. Let $p_i : P_i \hookrightarrow G_n$ be the morphism, such that $q = p_i \circ b_i$.

   Now, we can construct a morphism $p : C_{k+1} \hookrightarrow G$ with

   $$p(e) := \begin{cases} p_1(e) & \text{,if } e \in P_1 \\ & \vdots \\ p_j(e) & \text{,if } e \in P_j. \end{cases}$$

   Let $e \in C_k$, because $q(e) = p_i \circ b_i(e)$ and $q(e) = p_\ell \circ b_\ell(e)$ and b_i and b_ℓ are both partial morphisms of a_k, it follows that $b_i(e) = b_\ell(e)$ and therfore $p_i(e) = p_\ell(e)$. Because $(P_i \cap P_\ell) \setminus C_k = \emptyset$ for all $i \neq \ell$, p is a morphism and by the definition of p it follows that $q = p \circ a_k$ and therefore $q \models \exists(a_k : C_k \hookrightarrow C_{k+1}, \mathsf{true})$.

   By iteratively applying this whole construction to all occurrences of C_k that do not satisfy $\exists(a_k : C_k \hookrightarrow C_{k+1}, \mathsf{true})$ the derived graph graph H does not contain any occurrences of C_k not satisfying $\exists(a_k : C_k \hookrightarrow C_{k+1}, \mathsf{true})$ and therefore $H \models_{k+2} c$.

□

Corollary 3.15. *If a set of rule $\mathcal{R}$ is a repairing set of c at layer $k \leq \mathrm{nl}(c)$ and 1. of theorem 3.3 applies, then $\mathcal{R}$ is a repairing set of c at layer j for all $k \leq j \leq \mathrm{nl}(c)$.*

Lemma 3.16. *Let a graph G_0, a constraint c in EANF and a repairing set $\mathcal{R}$ at layer $c^{G_0}_{\max}+1$ be given, such that each rule in $\mathcal{R}$ is a basic increasing. Then, for every sequence*

$$G_0 \Longrightarrow_{\rho_0,m_0} G_1 \Longrightarrow_{\rho,m_0} \ldots \Longrightarrow_{\rho_n,m_n} G_n$$

such that $\rho_i = (\rho, \mathrm{basic}(c^{G_0}_{\max} + 1, P)$ with $\rho \in \mathcal{R}$ being a consistency increasing rule at layer $c^{G_0}_{\max} + 1$ with P, it holds that

3.3 Repairing rule Sets

Definition 3.17 (repairing rule set). *Let a constraint c in UANF and a set of rules $\mathcal{R}$ be given. Then, $\mathcal{R}$ is called a* repairing rule set for c at layer k *if for all graphs G with $\mathrm{k}_{\max} = k$ a sequence*

$$G = G_0 \Longrightarrow_{\rho_0} \ldots \Longrightarrow_{\rho_{n-1}} G_n = H$$

exists, such that $\mathrm{k}_{\max}(c, H) \geq \mathrm{k}_{\max}(c, G) + 2$ and $\rho_j \in \mathcal{R}$ for all $0 \leq j \leq n - 1$.

Corollary 3.18. *Let a constraint c in UANF and a rule set $\mathcal{R}$ be given. If either (a) $\mathcal{R}$ is a repairing rule set at layer k for all odd $0 \leq k \leq \mathrm{nl}(c)$ or (b) $\mathcal{R}$ is a repairing rule set at layer k with k being even and $\mathcal{R}$ is also a repairing rule set for all odd $0 \leq j < k$ then for all graphs G a sequence*

$$G = G_0 \Longrightarrow_{\rho_0} \ldots \Longrightarrow_{\rho_{n-1}} G_n$$

with $G_n \models c$ and $\rho_j \in \mathcal{R}$ for all $0 \leq j \leq n - 1$ exists.

3.3.1 Construction of repairing sets

Definition 3.19 (decomposition of a graph). *Let graphs $G_0 \hookrightarrow G_1$ be given.*

A decomposition of G_1 with G_0 is a minimal set

$$\mathbf{D} \subseteq \{D_v \mid v \in V_{C_1 \setminus C_0}\}$$

of subgraphs of G_1, such that every element of G_1 is contained in at least one $D \in \mathbf{D}$ and every D_v is constructed in the following way: $G_0 \hookrightarrow D_v$, $v \in D_v$ and for all nodes $v' \in D \setminus C_0$ D contains all edges $e \in E_{G_1}$ and all nodes $u \in V_{G_1}$ such that either $\mathrm{tar}(e) = v' \wedge \mathrm{src}(e) = u$ or $\mathrm{src}(e) = u \wedge \mathrm{tar}(e) = v'$.

Lemma 3.20. *Let graphs $G_0 \hookrightarrow G_1$ and a decomposition $\mathbf{D}$ of G_1 with G_0 be given. Then, for each pair $D, D' \in \mathbf{D}$ with $D \neq D'$ the following holds:*

$$(D \setminus C_k) \cap (D' \setminus C_k) = \emptyset$$

Proof. Assume that $(D \setminus C_k) \cap (D' \setminus C_k) \neq \emptyset$, therefore a node $v \in G_1 \setminus G_0$ with $v \in D \cap D'$ exists. By the construction of D and D' it follows that $D = D_v$ and $D' = D_v$ and therefore $D = D'$. This is a contradiction. □

Lemma 3.21. *Let graphs $G_0 \hookrightarrow G_1$ and a decomposition $\mathbf{D}$ of G_1 with G_0 be given. Then,*

$$G_1 = \bigcup_{P \in \mathbf{P}} P.$$

Proof. Let $H := \bigcup_{P \in \mathbf{P}} P$. Firstly, we show that $H \subseteq G_1$. Since every $P \in \mathbf{P}$ is a subgraph of G_1 it follows that $V_H \subseteq V_{G_1}$ and $E_H \subseteq E_{G_1}$

Secondly, we show that $G_1 \subseteq H$. Let $u \in V_{G_1}$ be a node, if $u \in V_{G_0}$, then u is contained in every $P \in \mathbf{P}$ and therefore $u \in V_H$. Otherwise, if $u \notin V_{G_0}$, then u has to be, by the definition of $\mathbf{P}$, contained in at least one $P \in \mathbf{P}$ and $V_{G_1} \subseteq V_H$ follows. Let $e \in E_{G_1}$ be an edge. If $e \in E_{G_0}$, then e is contained in every $P \in \mathbf{P}$ and $e \in E_H$. Otherwise, if $e \notin E_{G_0}$, by the definition of $\mathbf{P}$, e has to be contained in at least one $P \in \mathbf{P}$. It follows that $e \in E_H$ and with that $E_{G_1} \subseteq E_H$. □

Now we are ready to provide the construction for

Construction 3.22. *Let a constraint c in UANF be given. For each existentially bound subcondition $\mathrm{sub}_k(c)$ we construct a graph decomposition $\mathbf{D_k}$ as described in definition 3.19. Then, for each element $D \in \mathbf{D_k}$ a rule*

$$\rho = C_k \xhookleftarrow{\mathrm{id}} C_k \xhookrightarrow{i_{C_k}^{D}} D$$

is derived.

Additionally, for each universally bound subcondition $\mathrm{sub}_k(c)$ and for each edge $e \in E_{C_{k+1} \setminus C_k}$ a rule

$$\rho = C_{k+1} \xhookleftarrow{i_D^{C_{k+1}}} D \xhookrightarrow{\mathrm{id}} D$$

with $D = C_{k+1} \setminus \{e\}$ is derived.

Example 3.2.

4 Rule-based Graph Repair

4.1 Rule based Graph repair for conflict free constraints

Theorem 4.1. *Let a rule set $\mathcal{R}$ and a conflict free constraint c in UANF be given. Then, $\mathcal{R}$ is a repairing set of c at layer $k \leq \mathrm{nl}(c)$ if a sequence*

$$C_k \Longrightarrow_{\rho_0, m_0} \ldots \Longrightarrow_{\rho_n, m_n} H$$

Algorithm 1: Repair for conflict free constraints and repairing set

```
Data: A graph $G$, a conflict free constraint in UANF and a repairing set $\mathcal{R}$ for
      each layer $\mathrm{k}_{\max} < k\,\mathrm{nl}(c)$
Result: A graph $H$ with $H \models c$
1 for $i = \mathrm{k}_{\max} + 1$ to $\mathrm{nl}(c)$ do
2     while $G \not\models_i c$ do
3         Choose one occurrence $p : C_i \hookrightarrow G$ uniformly at random ;
4         apply a repairing sequence $G \Longrightarrow \ldots \Longrightarrow H$;
5     end
6 end
```

of consistency increasing or non-consistency decreasing transformations $t_0, \ldots, t_n$ *with* $\rho_0, \ldots, \rho_n \in \mathcal{R}$ *exists such that*

$$\forall v \in V_{C_k}(\exists v' \in V_H(\mathrm{tr}_{t_n} \circ \ldots \circ \mathrm{tr}_{t_0} \circ \mathrm{id}_{C_k}(v) = v'))$$

and

1. *If* $\mathrm{sub}_k(c)$ *is universally bound:*

$$\mathrm{tr}_{t_n} \circ \ldots \circ \mathrm{tr}_{t_0} \textit{ is not total}$$

2. *If* $\mathrm{sub}_k(c)$ *is existentially bound:*

$$\mathrm{tr}\, t_n \circ \ldots \mathrm{tr}\, t_0 \circ \mathrm{id}_{C_k} \models \mathrm{sub}_k(c)\mathrm{cut}_{k+1}(c)$$

Proof. □

Lemma 4.1. *The set rule set constructed as described in construction 3.22 is a repairing set.*

Proof. □

Theorem 4.2. *Algorithm 1 is correct and does terminate.*

Proof. □

4.2 Rule based Graph repair for non-conflict free constraints

Theorem 4.3. *Algorithm 2 terminates and returns an consistent graph.*

Proof. □

4.3 Rule based repair for non-repairing sets

Algorithm 2: Repair for non conflict free constraints and repairing set

```
Data: A graph $G$, a non-conflict free constraint in UANF and a repairing set $\mathcal{R}$
      for each layer $\mathrm{k}_{\max} < k\,\mathrm{nl}(c)$
Result: A graph $H$ with $H \models c$
1  for $i = \mathrm{k}_{\max} + 1$ to $\mathrm{nl}(c)$ do
2      while $G \not\models_i c$ do
3          Choose one occurrence $p : C_i \hookrightarrow G$ uniformly at random ;
4          apply a repairing sequence $G \Longrightarrow \ldots \Longrightarrow H$;
5          while $G \not\models_{\mathrm{k}_{\max}}$ do
6              Choose an occurrence $p : C_j \hookrightarrow G$ uniformly at random;
7              apply the repairing sequence.
8          end
9      end
10 end
```

Algorithm 3: Repair for conflict free constraints and repairing set

```
Data: A graph $G$, a conflict free constraint in UANF and a repairing set $\mathcal{R}$ for
      each layer $\mathrm{k}_{\max} < k\,\mathrm{nl}(c)$
Result: A graph $H$ with $H \models c$
1  $\mathcal{G} \leftarrow \{G\}$ for $i = 0$ to $m$ do
2      foreach $G \in \mathcal{G}$ do
3          foreach $t : G \Longrightarrow H$ do
4              if $H \models c$ then
5                  return $H$
6              end
7              if $t$ is non-consistency decreasing or consistency increasing then
8                  $\mathcal{G} \leftarrow \mathcal{G} \cup \{H\}$;
9              end
10         end
11     end
12 end
```

References

[1] A. Habel and K.-H. Pennemann. Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science*, 19(2):245–296, 2009.

[2] J. Kosiol, D. Strüber, G. Taentzer, and S. Zschaler. Sustaining and improving graduated graph consistency: A static analysis of graph transformations. *Science of Computer Programming*, 214:102729, 2022.

[3] C. Sandmann and A. Habel. Rule-based graph repair. *arXiv preprint arXiv:1912.09610*, 2019.