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

Alexander Lauer

December 4, 2022

Abstract

1

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 $i_{G_1}^H$ and $i_{G_2}^H$, 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:*

- *$\textbf{true}$ 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 $\textbf{true}$. 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 $\mathrm{nl}(\textbf{true} = 0$ 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.

$\square$

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 $\mathrm{sco}_k(c)$, is the subcondition d of c with $\mathrm{lay}(d) = k$.*

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

$C_1^1:$ ▭ C $C_2^1:$
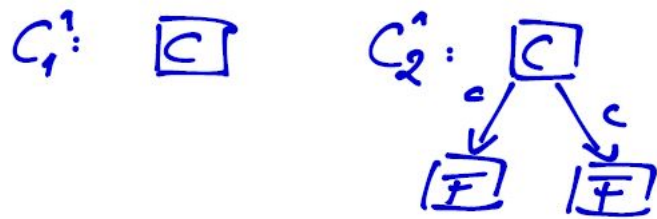
$$C_2 = \forall C_1^1 \, \exists \, C_2^2 \, \forall C_3^2 \, \exists \, C_4^2$$
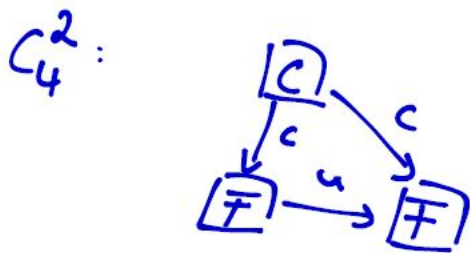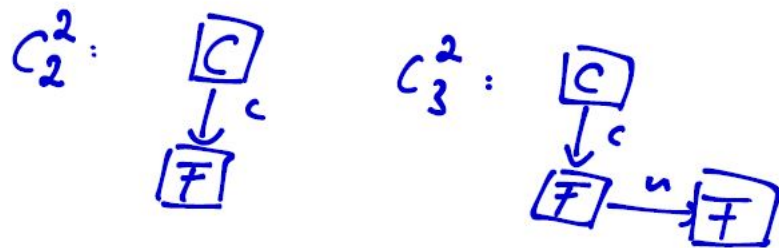
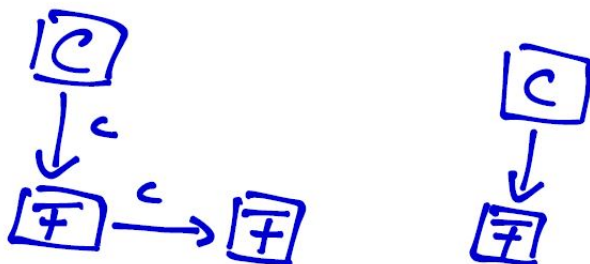$C_2^2:$ $C_3^2:$

$C_4^2:$

Figure 1: constraints

Figure 2: graph

5

For the rest of this paper, given a constraint c in UANF, $\mathrm{sco}_k(c)$ 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* $c := \forall(a_0 : C_0 \hookrightarrow C_1, \exists(a_1 : C_1 \hookrightarrow C_2, \mathsf{true}))$ *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* $d = \mathrm{sco}_k(c)$ *with* $0 \leq k \leq \mathrm{nl}(c)$. *The* truncated condition up to layer k of c, $\mathrm{cut}_k(c)$, *is defined as*

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

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.*

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 $d = \mathrm{sco}_k(c)$ 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 $\mathrm{sco}_k(c) = \forall(a_k : C_k \hookrightarrow C_{k+1}, d)$ 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 $f = \mathrm{rep}_{j-k+1}(\mathrm{sco}_{k+1}(c), d)$. Since $\mathrm{rep}_{k+1}(c, f) = \mathrm{rep}_{j+1}(c, d)$ it follows that $p \models \mathrm{rep}_{j+1}(c, d)$. $\square$

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 $\mathrm{sco}_k(c)$ is universally bound,*

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

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

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 $\mathrm{sco}_j(c)$ is existentially bound.*

Proof. 1. The subcondition of c at layer $c_{\max}^G$ is existentially bound: If a $j < c_{\max}^G$ with $p \models_j c$ exists such that $\mathrm{sco}_c(j)$ is universally bound, let j_1 be the smallest of these. With lemma 2.8 follows that $p \models_{j_2} c$ for all $j_1 < j_2$. Let $\ell < j_1$, such that $\mathrm{sco}_c(\ell)$ is existentially bound and let $d = \exists(a_\ell : C_\ell \hookrightarrow C_{\ell+1}, e)$ be the condition up to layer $j_1 - \ell$ of $\mathrm{sco}_c(\ell)$. 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 $p \models_\ell c$.

2. The subcondition of c at layer $c^G_{\max}$ is universally bound: With lemma 2.8 follows that $p \models_{k+1} c$. Since c is in EANF, 1. can be applied to $k+1$.

$\square$

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

The construction of partial 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 $C' \in \mathcal{U}(C_0, C_1)$. If the condition $c' = \forall(a^p_0 : C_0 \hookrightarrow C_1, \mathsf{false})$ is satisfied this implies that c is also satisfied but the backwards implication does not hold.

Definition 2.11 (partial condition). *Let a constraint c in ANF be given. Let $k \leq \mathrm{nl}(c) + 1$ such that either $\mathrm{sco}_c(k) = \exists(a_k : C_{k-1} \hookrightarrow C_k, d)$ is existentially bound or $k = \mathrm{nl}(c) + 1$. The partial condition, $\mathrm{part}(k, c, C')$, of c at layer k with*

$$C' \in \begin{cases} \mathcal{U}(C_k, C_{k+1}) & \text{if } k \leq \mathrm{nl}(c) \\ C_{\mathrm{nl}(c)} & \text{otherwise} \end{cases}$$

is defined as:

$$\mathrm{part}(k, c, C') := \begin{cases} c & \text{if } k = \mathrm{nl}(c) + 1 \\ \mathrm{sub}\left(k, c, \exists(a^p_k : C_{k-1} \hookrightarrow C', \mathsf{true})\right) & \text{otherwise} \end{cases}$$

Example 2.4. *Consider constraint c_1 given in figure 1. Since $C^2_2 \in \mathcal{U}(C^1_1, C^2_1)$, we can construct a partial condition of c_1 at layer 2 with C^2_2 as $\mathrm{part}(2, c_1, C^2_2) = \forall C^1_1 \exists C^2_2$. Whereas c_2 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`.*

If a graph G does not satisfy the condition up to layer $k := c^G_{\max} + 2$, $k < \mathrm{nl}(c) + 1$, of a given constraint c, with $\mathrm{sco}_c(k)$ being existentially bound, there does exists a graph $C' \in \mathcal{U}(C_{k-1}, C_k)$, such that G satisfies $\mathrm{part}(k, c, C')$, note that G always satisfies $\mathrm{part}(k, c, C_{k-1})$. For the biggest of these graphs, we define the notions of *biggest partially satisfying conditions and graphs*.

Definition 2.12 (**biggest partially satisfying condition**). *Let G be a graph, c a condition in EANF over C_0 and $p : C_0 \hookrightarrow G$ a morphism with $p \models_k c$ and $p \not\models c$.*

A partial condition $c = \mathrm{part}(k + 2, c, C')$ with $p \models c$ is a biggest partially satisfying condition *w.r.t c of p if there does not exist a graph C'', such that C' is a subgraph of C'', with $p \models \mathrm{part}(c_{max}, c, C'')$. The graph C' is then called a* biggest partially satisfying graph *w.r.t c of p.*

Given a constraint c, the set of biggest partially satisfying conditions w.r.t c of p : $\emptyset \inf G$ is denoted by $\mathcal{P}_c^G$.

The set of all biggest partially satisfying graphs w.r.t c of p is denoted by $\mathcal{G}_c^G$.

2.3 consistency increasing transformations and rules

With the results above, we are now ready to define the notions of *consistency increasing* and *direct consistency increasing*, 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 $k = c_{\max}^G$ and $\forall(a_{k+1} : C_k \hookrightarrow C_{k+1}, \exists(a_{k+2} : C_{k+1} \hookrightarrow C_{k+2}, d)) =:$ $\mathrm{sco}_c(k+1)$. A transformation $t : G \Longrightarrow H$ is considered as (direct) consistency increasing if $c_{\max}^G \leq c_{\max}^H$, 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 $\exists(a_{k+2} : C_{k+1} \hookrightarrow C_{k+2}, \mathsf{true})$, 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 $p \not\models \exists(a_{k+2}^p : C_{k+1} \hookrightarrow C', \mathsf{true})$ and $\mathrm{tr}_t \circ p \models \exists(a_{k+2}^p : C_{k+1} \hookrightarrow C', \mathsf{true})$ for an $C' \in \mathcal{U}(C_{k+1}, C_{k+2})$. A decreasing insertion is the creation of an occurrence of C_{k+1} not satisfying $\exists(a_{k+2} : C_{k+1} \hookrightarrow C_{k+2}, \mathsf{true})$ and a decreasing deletion is the deletion of elements of C_{k+2} such that for an occurrence p of C_{k+1} with $p \models \exists(a_{k+2}^p : C_{k+1} \hookrightarrow C', \mathsf{true})$ it holds that $\mathrm{tr}_t \circ p \not\models \exists(a_{k+2}^p : C_{k+1} \hookrightarrow C', \mathsf{true})$.

To evaluate this, we define the *number of violations*. Intuitively, for all occurrences p of C_{k+1} the number of graphs $C' \in \mathcal{U}(C_{k+1}, C_{k+2})$ 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 $k \leq c_{\max}^G$ it is set to 0 and for all layer k with $k > c_{\max}^G + 1$ it is set to ∞. Through this, a transformation $tG \Longrightarrow H$ that increases the partial consistency can easily detected since the number of violations in H at layer $c_{\max}^G + 1$ will be set to 0.

Definition 2.13 (**number of violations**). *Let G be a graph and c a constraint in EANF. The* number of violations $\mathrm{nvc}(j, G)$ *at layer j in G is defined as:*

1. If $j < c_{\max}^G + 1$:

$$\mathrm{nvc}(j, G) := 0$$

9

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

$$\text{nvc}(j, G) := \begin{cases} \sum_{C' \in \mathcal{U}(C_j, C_{j+1})} |\{q \mid q : C_{j+1} \hookrightarrow G \land q \not\models \text{part}(1, e, C')\}| & \text{if } e \neq \textit{false} \\ |\{q \mid q : C_{j+1} \hookrightarrow G\}| & \text{if } e = \textit{false} \end{cases}$$

3. *If $j > c_{\max}^G + 1$:*

$$\text{nvc}(j, G) := \infty$$

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 2.14 (consistency increasing). *Let a graph G, a rule ρ and a constraint c in EANF be given.*

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

$$\text{nvc}(k, H) < \text{nvc}(k, G)$$

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

Note that if $G \models c$ there does not exist a consistency increasing transformation w.r.t c, since $\text{nvc}(j, G) = 0$ for all $1 \leq j \leq \text{nl}(c)$. 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 partial consistency, which is shown by the following theorem.

Theorem 2.1. *Let a graph G, a rule ρ and a constraint c in EANF be given. A transformation $t : G \Longrightarrow_{\rho,m} H$ is consistency increasing if $c_{\max}^G < c_{\max}^H$.*

Proof. No $\ell > c_{\max}^G$ with $G \models_\ell c$ exists. Hence, $\text{nvc}(c_{\max}^G + 1, G) > 0$ and $\text{nvc}(c_{\max}^G + 1, G) \neq \infty$. Since $c_{\max}^H > c_{\max}^G$, $\text{nvc}(k + 1, H) = 0$ and it follows immediately that t is consistency increasing. $\qquad\square$

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 EANF and a set of rules $\mathcal{R}$ be given. Every sequence of minimal consistency improving transformation w.r.t c with rules in $\mathcal{R}$ is finite.*

Proof. Let G_0 be a graph and

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

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

We show that after at most $j := \mathrm{nvc}(c_{\max G_0} + 1, G_0)$ transformations $G_j \models_{c_{\max}^{G_0} + 2} c$ 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 $\mathrm{nvc}(c_{\max}^{G_{i+1}} + 1, G_{i+1}) \leq \mathrm{nvc}(c_{\max}^{G_i} + 1, G_i) - 1$. Therefore, after j transformations, $\mathrm{nvc}(c_{\max}^{G_0} + 1, G_j) \leq \mathrm{nvc}(c_{\max}^{G_0} + 1, G_0) - j = 0$ holds and with that $G_j \models_{c_{\max}^{G_0} + 2} c$. 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 $G_{k'}$ there may not exist a consistency increasing transformation $G_{k'} \Longrightarrow_{\rho} H$ with $\rho \in \mathcal{R}$ and therefore the sequence is also finite, but does not end with a consistent graph. $\qquad \square$

The replacement of a graph not satisfying a constraint c by a c satisfying graph via a transformation is a consistency increasing transformation. Therefore, a consistency increasing transformation can perform insertions or deletions that are unnecessary in order to increase the consistency. That is, the deletion of occurrences of existentially bound graphs, the deletions of occurrences of 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 a transformation to not being direct 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 be checked via an additional 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 secure that the partial consistency is not decreased.

Definition 2.15 (**direct consistency increasing**). *Let G be a graph, ρ a plain rule, c a constraint in EANF and $\mathrm{sco}_c(c_{\max}^G + 1) = \forall(a_k : C_{k-1} \hookrightarrow C_k, e)$. A transformation $t : G \Longrightarrow_{\rho, m} H$ is called* direct consistency increasing *if the following equations hold. Let*

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

Every occurrence of C_k in G that satisfies $\mathrm{part}(1, e, C')$ for any $C' \in \mathbf{G}$ still satisfies

11

$\mathrm{part}(1, e, C')$ *in H.*

$$\forall p : C_k \hookrightarrow G\Big(\bigwedge_{C' \in \mathbf{G}} \big(p \models \mathrm{part}(1, e, C') \wedge \mathrm{tr}_t \circ p \text{ is total}\big)$$
$$\implies \mathrm{tr}_t \circ p \models \mathrm{part}(1, e, C')\Big) \tag{2.1}$$

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

$$\forall p' : C_k \hookrightarrow H\big(\neg \exists p : C_k \hookrightarrow G(p' = \mathrm{tr}_t \circ p) \implies p' \models \mathrm{part}(1, e, C')\big) \tag{2.2}$$

At least one occurrence of C_k in G that does not satisfy $\mathrm{part}(1, e, C')$, for any $C' \in \mathbf{G}$, either has been removed or satisfies $\mathrm{part}(1, e, C')$ in H.

$$\exists p : C_k \hookrightarrow G\Big(\bigvee_{C' \in \mathbf{G}} \big(p \not\models \mathrm{part}(1, e, C') \wedge (\mathrm{tr}_t \circ p \text{ is not total}$$
$$\vee (\mathrm{tr}_t \circ p \text{ is total } \wedge \mathrm{tr}_t \circ p \models \mathrm{part}(1, e, C'))))\Big) \tag{2.3}$$

No occurrence of a universally bound graph C_j with $j < k$ gets inserted.

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

No occurrence of an existentially bound graph C_j with $j < k$ gets deleted.

$$\bigwedge_{\substack{i < k \\ i \text{ even}}} \forall p : C_i \hookrightarrow G(\mathrm{tr}_t \circ p \text{ is total}) \tag{2.5}$$

Note that (2.4) and (2.5) not only secure that the partial consistency 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 2.16. *Let a transformation $t : G \implies H$ and a constraint c in EANF be given, such that (2.4) and (2.5) of definition 2.15 are satisfied. Then,*

$$H \models_{c_{\max}^G} c.$$

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

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

with $i, j \leq c_{\max}^G$, i being odd and j being even. It follows immediately that either (2.4) or (2.5) is not satisfied. This is a contradiction. $\qquad\square$

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

Lemma 2.17. *Let a graph G, a constraint c in EANF and a direct increasing transformation $t : G \Longrightarrow_{\rho,m} H$ w.r.t. c be given. Then, t is also a increasing transformation.*

Proof. Let G be a graph, $k = c_{\max}^G + 1 \leq \mathrm{nl}(c)$ and $d = \mathrm{sco}_c(k)$. With lemma 2.16 follows that $c_{\max}^H \geq c_{\max}^G$ and with that $\mathrm{nvc}(k, H) \neq \infty$.

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 $p \not\models \mathrm{part}(1, d, C')$ for any $C' \in \mathcal{U}(C, C_{k+1})$ exists, such that either 1a or 1b is satisfied.

   (a) There does exist a morphism $q' : C_k \hookrightarrow G$ with $q' \models \mathrm{part}(1, d, C')$ and $p = \mathrm{tr}_t \circ q'$.

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

   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

   $$|\{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{part}(1, e, C')\}| \leq |\{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{part}(1, e, C')\}|$$

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

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

   $$p \in \{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{part}(1, e, C')\} \wedge$$
   $$\mathrm{tr} \circ p \notin \{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{part}(1, e, C')\}$$

   With that and 1. it follows that

   $$|\{q \mid q : C_k \hookrightarrow G \wedge q \not\models \mathrm{part}(1, e, C')\}| < |\{q \mid q : C_k \hookrightarrow H \wedge q \not\models \mathrm{part}(1, e, C')\}|.$$

In total, $\mathrm{nvc}(k, G) < \mathrm{nvc}(k, G)$ and t is an increasing transformation.

For the special case that $k = \mathrm{nl}(c) - 1$ 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

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

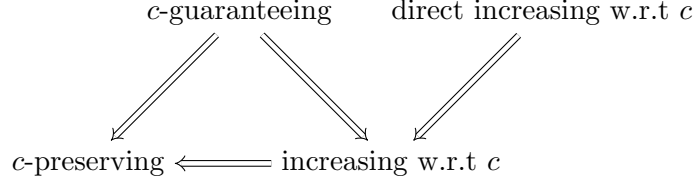It follows that t is an increasing transformation. $\qquad\square$

Figure 3: General relations of consistency notions.

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) increasing is indeed a new notion of consistency. These relations are displayed in figure 3.

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

Obviously, guaranteeing implies increasing, this property is embedded in the definition of 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 2.18. *Let a constraint c in ANF, graphs G and H with $G \not\models c$, and a transformation $t : G \implies H$ be given. Then,*

$$t \text{ is } c\text{-guaranteeing} \quad \implies \quad t \text{ is increasing w.r.t } c \qquad \wedge$$
$$t \text{ is } c\text{-guaranteeing} \quad \not\implies \quad t \text{ is direct increasing w.r.t } c \wedge$$
$$t \text{ is increasing w.r.t } c \not\implies t \text{ is } c\text{-guaranteeing.}$$

Proof. Let c' be the equivalent constraint in EANF.

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

2. Let a constraint $c = \exists(a_0 : \emptyset \hookrightarrow C_0, \forall(a_1 : C_0 \hookrightarrow C_1, \exists(a_2 : C_1 \hookrightarrow C_2, \text{true})))$ and a graph $G = C' \dot\cup C'$ with $C' = C_2 \setminus \{e\}$ for one edge $e \in E_{C_2 \setminus C_1}$ and the occurrences $p_1 : C_1 \hookrightarrow G$ and $p_2 : C_1 \hookrightarrow G$ be given. Let $t : G \implies H$ be a transformation with $H = C_2 \dot\cup C''$ and $C'' = C' \setminus \{e'\}$ for an $e' \in E_{C_2 \setminus C_1}$, such that $\text{tr}_t \circ p_1$ and

14

$\mathrm{tr}_t \circ p_2$ are total. Then, t is c-guaranteeing and not direct increasing, since $p_i \models \exists(a_2' : C_1 \hookrightarrow C', \mathsf{true})$ for $i = 1, 2$ and either $\mathrm{tr}_t \circ p_1 \not\models \exists(a_2' : C_1 \hookrightarrow C', \mathsf{true})$ or $\mathrm{tr}_t \circ p_2 \not\models \exists(a_2' : C_1 \hookrightarrow C', \mathsf{true})$.

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

$\square$

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 2.18, we can state the following.

Corollary 2.19. *Let c be an existentially bound constraint in ANF. Then,*

t is consistency improving w.r.t c $\implies$ t is consistency increasing w.r.t c $\wedge$

t is consistency increasing w.r.t c $\not\implies$ t is consistency improving w.r.t c

The notions of increasing and improving are equivalent for universally bound constraints with nesting level 1. Note that, with corollary 2.19, this equivalence does not hold for existentially bound constraints with nesting level 1.

Lemma 2.20. *let a universally bound constraint c with $\mathrm{nl}(c) = 1$ in ANF, a graph G with $G \not\models c$ and a transformation $t : G \implies H$ be given. Then,*

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

Proof. Let $c = \forall(a : \emptyset \hookrightarrow C, \mathsf{false})$. Since $\mathcal{U}(C, C) = \{C\}$, $\mathrm{nvc}(1, G)$ 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. $\square$

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

t is direct consistency improving w.r.t c $\not\implies$ t is consistency increasing w.r.t c $\wedge$

t is direct increasing w.r.t c $\not\implies$ t is consistency sustaining w.r.t c

Proof. 1. Let $c = \forall(a_0 : \emptyset \hookrightarrow C_0, \exists(a_1 : C_0 \hookrightarrow C_1, \mathsf{true}))$ be a constraint. Let $V_{C_0} = V_{C_1}$ and $|E_{C_1}| - |E_{C_0}| = 2$. Let $G = C' \dot\cup C'$ with $C' = C_1 \backslash \{e\}$ with $e \in E_{C_1} \backslash E_{C_0}$ and the occurrences $p_1 : C_0 \hookrightarrow G$ and $p_2 : C_0 \hookrightarrow G$ be given. It follows that $\mathrm{nvc}(1, G) = 2$.

Figure 4: rules

Let $t : G \implies H$ be a transformation, such that $H = C_0$. Then, t is a direct consistency improving transformation, since H contains only one occurrence of C_0 not satisfying $\exists(a_1 : C_0 \hookrightarrow C_1 \text{ true})$. But, t is not consistency increasing, since $\text{nvc}(1, H) = 3$.

2. Let $c := \forall(a_0 : \emptyset \hookrightarrow C_0, \exists(a_1 : C_0 \hookrightarrow C_1, \forall(a_2 : C_1 \hookrightarrow C_2, \exists(a_3 : C_2 \hookrightarrow C_3, \text{true}))))$ be a constraint.

Let a graph $G = C_0$ with the morphism $q : C_0 \hookrightarrow G$ and a transformation $t : G \implies H$ with $H := C_2 \dot\cup C_2$ be given, such that $\text{tr}_t \circ q$ is total. Then t is a direct consistency increasing transformation but not a consistency sustaining one, since H contains more occurrences of C_0 not satisfying $\exists(a_1 : C_0 \hookrightarrow C_1, \forall(a_2 : C_1 \hookrightarrow C_2, \exists(a_3 : C_2 \hookrightarrow C_3, \text{true})))$ than G.

$\square$

3 general application conditions

To guarantee that each transformation $t : G \implies_{\rho,m} H$, 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 the universally bound graph $C_{c_{\max}^G+1}$ exists, such that p and the match m do overlap, i.e $p(C_{c_{\max}^G+1}) \cap m(L) \neq \emptyset$, and (b) that p does not satisfy $c' = \exists C_{c_{\max}^G+2}$. 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 $C_{c_{\max}^G+2}$ such that $p \models c'$ has to be checked. For this, we introduce *extended overlaps*. Intuitively, given an overlap C of L and $C_{c_{\max}^G+1}$ with $p : C_{c_{\max}^G+1} \hookrightarrow C$, the set of extended overlaps contains all overlaps C' of L and $C_{c_{\max}^G+2}$, such that $p \models c'$ if one of these overlaps exists.

Definition 3.1 (extended overlaps). *Let G, C_0 and C_1 with $i : C_0 \hookrightarrow C_1$ be graphs. Let C be an overlap of C_0 and G with the inclusion $q : C_0 \hookrightarrow C$. The set of* extended
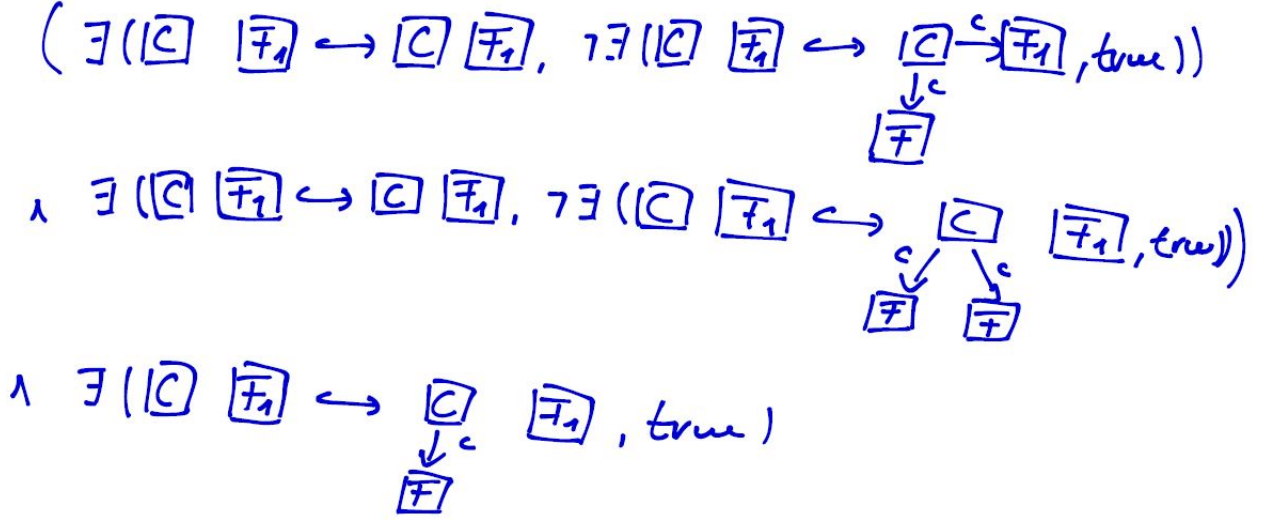
16

$$\Big(\exists\big(\boxed{C}\ \boxed{F_1} \hookrightarrow \boxed{C}\ \boxed{F_1},\ \neg\exists\big(\boxed{C}\ \boxed{F_1} \hookrightarrow \boxed{C}\xrightarrow{c}\boxed{F_1},\ \mathsf{true}\big)\big)$$

$$\wedge\ \exists\big(\boxed{C}\ \boxed{F_1} \hookrightarrow \boxed{C}\ \boxed{F_1},\ \neg\exists\big(\boxed{C}\ \boxed{F_1} \hookrightarrow \boxed{C}\ \boxed{F_1},\ \mathsf{true}\big)\big)$$

$$\wedge\ \exists\big(\boxed{C}\ \boxed{F_1} \hookrightarrow \boxed{C}\ \boxed{F_1},\ \mathsf{true}\big)$$

Figure 5: application condition

overlaps of C with i, $\mathrm{eol}(C, i)$, *is the set of all overlaps C' of G and C_1, such that an injective morphism $i_C : C \hookrightarrow C'$ with $i_C \circ q \models \exists(i : C_0 \hookrightarrow C_1, \mathsf{true})$ 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 partial consistency is not decreased. In the definition below, nex() and rep() ensure that (a) is met, with nex() ensuring that a violation is present and rep() ensuring that this violation will be removed. Note, that the construction of these is divided in two cases. Firstly, either $k < \mathrm{nl}(c) - 1$ or the constraints ends with a condition of the form $\exists(a : C \hookrightarrow C', \mathsf{true})$ and secondly, $k = \mathrm{nl}(c) - 1$ and the constraint ends with a condition of the form $\forall(a : C_0 \hookrightarrow C_1, \mathsf{false})$.

For nex(), 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 rep(), 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 $C' \in \mathcal{U}(C_k, C_{k+1})$. 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 nin() and nwo() ensure that (2) is me, with nin() checking that no new occurrence of C_k will be inserted and nwo() 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 $C' \in \mathcal{U}(C_k, C_{k+1})$.

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

Definition 3.2 (general application condition). *Let $\rho = L \hookleftarrow K \hookrightarrow R$ be a plain*

rule and c a constraint in EANF. Let $d = \forall(a_k : C_{k-1} \hookrightarrow C_k, e)$ be the subcondition of c at layer k with k being an odd number.

The application condition ap_k of c at layer k with $C' = C_k$, if $e = \mathsf{false}$, and $C' \in \mathcal{U}(C_k, C_{k+1})$ otherwise is defined as:

$$\mathrm{ap}(k, C') := \Big(\bigvee_{P \in \mathrm{ol}(L, C_k)} \mathrm{nex}(P, C') \wedge \mathrm{rep}(P, C') \Big) \wedge \mathrm{nin}() \wedge \mathrm{rem}() \wedge \mathrm{nwo}() \qquad (3.1)$$

with

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

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

2. If $i_L^P(L \setminus K) \cap i_{C_k}^P(C_k) \neq \emptyset$, we set

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

Otherwise, let P' be the graph derived 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 $\mathrm{rep}(P, C') := \mathsf{false}$. Let $a^p : C_k \hookrightarrow C'$ be a partial morphism of a_{k+1}, then

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

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

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

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

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

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 $C'' \in \mathcal{U}(C_k, C_{k+1})$, $i_{C_k}^{P'} \models \exists(a_k^p : C_k \hookrightarrow C'', \mathsf{true})$ and $i_L^{P'}(L \setminus K) \cap i_{C''}^{P'}(C'' \setminus C_k) \neq \emptyset$.

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

18

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

$$\bigvee_{C' \in \mathcal{U}(C_k, C_{k+1})} \mathrm{ap}(k, C').$$

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 $\mathrm{rem}()$, $\mathrm{nin}()$ and $\mathrm{nwo}()$ 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; $\mathrm{nwo}()$ 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 $\mathrm{nwo}()$ 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 $\mathrm{nwo}()$ would still be evaluated with false. Therefore, this only leads to a slight decrease of restrictiveness.

The conditions constructed by $\mathrm{rem}()$ and $\mathrm{nin}()$ could be changed in a similar fashion. For $\mathrm{rem}()$ and the universally bound graph C_k, by checking whether there does exist an additional occurrence p of C_{k+1} such that $p \models \mathrm{sco}_{\mathrm{cond}(c_{\max}^G, c)}(c_{\max}^G - k + 2)$ and for $\mathrm{nin}()$, by checking whether an introduced occurrence p of C_k does satisfy $\mathrm{sco}_{\mathrm{cond}(c_{\max}^G, c)}(c_{\max}^G - k + 1)$. 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 $\mathrm{rem}()$ and $\mathrm{nin}()$ also increases.

For constraints of the form $c := \forall C_1 \exists C_2$ it can be shown that $\mathrm{ap}(1, C_2)$ is not only a direct increasing, but also a direct 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* $\mathrm{nex}(\cdot, \cdot)$ *and the third row is the condition constructed by* $\mathrm{rep}(\cdot, \cdot)$. *Note that* $\mathrm{rem}()$, $\mathrm{nin}()$ *and* $\mathrm{nwo}()$ *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 EANF with $G \not\models c$ and a plain rule ρ be given. Then, $\rho' = (\rho, \mathrm{ap}(c_{\max}^G + 1, C))$ with $C \in \mathcal{U}(C_{c_{\max}^G + 1}, C_{c_{\max}^G + 2})$ is a consistency increasing rule.*

Proof. Let $t : G \Longrightarrow_{\rho', m} H$ be a transformation and $k = c_{\max}^G + 1$. 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 $\mathrm{sco}_c(k) = \forall(a_k : C_k \hookrightarrow C_{k+1}, e)$ be the subcondition of c at layer k. Let

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

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

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

$$d := \begin{cases} \mathrm{part}(1, e, C_{k+1}) & \text{if } e \neq \mathsf{false} \\ \mathsf{false} & \text{otherwise.} \end{cases}$$

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

3. Assume that (2.3) does not hold. Then, no morphism $p : C_k \hookrightarrow G$ with $p \not\models \mathrm{part}(1, e, C')$, such that $\mathrm{tr}_t \circ p$ is not total or $\mathrm{tr}_t \circ p \models \mathrm{part}(1, e, C')$ and $\mathrm{tr}_t \circ p$ is total exists, for any $C' \in \mathbf{G}$. Then, no overlap P of L and C_k with $i_L^P(L \setminus K) \cap i_{C_k}^P(C_k) \neq \emptyset$ exists, such that $m \models \mathrm{nex}(P, C)$.

   Let P be an overlap of C_k and L with $i_L^P(L \setminus K) \cap i_{C_k}^P(C_k) = \emptyset$. If $e = \mathsf{false}$, then $\mathrm{rep}(P, C) = \mathsf{false}$. Otherwise, if $m \models \mathrm{nex}(P, C) \wedge \mathrm{rep}(P, C)$, it holds that $i_{C_k}^P \not\models \mathrm{part}(1, e, C)$ and a graph $Q \in \mathrm{eol}(P', a^p)$ exists, such that $m \models \mathrm{Left}(\exists(i_R^Q : R \hookrightarrow Q, \mathsf{true}), \rho)$. In this case, $\mathrm{tr}_t \circ i_{C_k}^P \models \mathrm{part}(1, e, C)$ follows and (2.3) is satisfied.

   In total, $m \not\models \mathrm{nex}(P, C) \wedge \mathrm{rep}(P, C)$ follows for all $P \in \mathrm{ol}(L, C_K)$ 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 $j < k$ and j being odd exists, such that no morphism $p' : C_j \hookrightarrow G$ with $\text{tr}_t \circ p' = p$ exists. Then, an overlap P of C_j and R with $i_R^P(R \setminus K) \cap i_{C_j}^P(C_j) \neq \emptyset$ exists, such that $m \models \text{Left}(\exists(i_R : R \hookrightarrow P, \text{true}), \rho)$. Hence, $m \not\models \text{nin}()$ and $m \not\models \text{ap}(k, C)$.

5. Assume that (2.5) does not hold. Then, a morphism $p : C_j \hookrightarrow G$ with $j < k$ and j being even exists, such that $\text{tr}_t \circ p$ is not total. Then, an overlap P of C_j and L with $i_L^P(L \setminus K) \cap i_{C_j}^P(C_j) \neq \emptyset$ exists, such that $m \models \exists(i_L^P : L \hookrightarrow P, \text{true})$. Hence, $m \not\models \text{rem}()$ and $m \not\models \text{ap}(k, C)$.

In total, if $\models \text{ap}(k, C)$, then t is a direct consistency increasing transformation. $\qquad\square$

3.1 potentially minimal improving rules

Definition 3.3 (non-consistency decreasing rule). *Let a plain rule $\rho = L \xleftarrow{l} K \xrightarrow{r} R$ and a constraint c in EANF be given. Then, ρ is called a* non-consistency decreasing *rule w.r.t c at layer k if*

1. *If C_k is universally bound and for all transformations $t : G \Longrightarrow_\rho H$ it holds that*
$$\forall p : C_k \hookrightarrow H : \exists p' : C_k \hookrightarrow G : \text{tr}_t \circ p' = p.$$

2. *If C_k is existentially bound for all transformations $t : G \Longrightarrow_\rho H$ it holds that*
$$\forall p : C_k \hookrightarrow G : \text{tr}_t \circ p \text{ is total.}$$

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 $1 \leq j \leq k$.*

Corollary 3.4. *Let a graph G, a constraint c in EANF and a non-consistency decreasing rule w.r.t c up to layer $c_{\max}^G$ be given. Then, for each transformation $t : G \Longrightarrow_\rho H$ it holds that*
$$c_{\max}^G \leq c_{\max}^H.$$

Lemma 3.5. *Let a plain rule $\rho = L \xleftarrow{l} K \xrightarrow{r} R$ 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*
$$\begin{aligned}
&\left(\neg \exists e \in E : \exists v, v' \in V_K : \text{src}(e) = r(v) \wedge \text{tar}(e) = r(v')\right) &\wedge\\
&\left(\forall v \in V : \exists e \in E_{C_k} : \exists p \in P : \text{src}(e) = p^{-1}(v) \vee \text{tar}(e) = p^{-1}(v)\right.\\
&\left. \wedge \neg \exists e' \in E : \exists p \in P : p^{-1}(e') = e\right)
\end{aligned}$$
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

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

Proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Definition 3.6 (**basic increasing rule**). *Let a constraint c and a plain rule $\rho = L \xleftarrow{l} K \xrightarrow{r} R$ be given. Then, ρ is called a basic increasing w.r.t c at layer k if ρ is a non-consistency decreasing w.r.t c up to layer k and*

1. *If C_k is universally bound: There does exist a morphism $i : L \hookrightarrow C_k$ and*

$$\exists e \in C_k \setminus C_{k-1} : \neg \exists e' \in K : i(e) = l(e).$$

   Then, ρ is called a deleting basic increasing rule.

2. *If C_k is existentially bound: There does exist a morphism $i : C_{k-1} \hookrightarrow L$ such that $i \not\models \exists(a_k^p : C_{k-1} \hookrightarrow C', \textsf{true})$ and $r \circ l^{-1} \circ i \models \exists(a_k^p : C_{k-1} \hookrightarrow C', \textsf{true})$ for any $C' \in \mathcal{U}(C_{k-1}, C_k)$, let the sets E and V be constructed as introduced in lemma 3.5 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.

Definition 3.7 (**application conditions for basic increasing rules**). *Let a constraint c in EANF and a basic increasing rule $\rho = L \xleftarrow{l} K \xrightarrow{r} R$ w.r.t c at layer k with $C \in \mathcal{U}(C_k, C_{k+1}$ be given. We define the application condition for ρ as:*

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

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

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 $i_L^P \circ e \models \exists(a^p : C_k \hookrightarrow C, \text{true})$:*

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

Theorem 3.2. *Let a graph G, a constraint c in EANF, with $G \models \mathrm{part}(c_{\max}^G, c, C)$ and $\mathrm{part}(c_{\max}^G, c, C) \in \mathcal{P}_c^G$, and a potentially minimal improving rule $\rho = L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R$ at layer $c_{\max}^G$ with $C_{c_{\max}+1} \subseteq P \subseteq C_{c_{\max}+2}$ be given. Then, $\rho' = (\rho, \mathrm{ap_{pi}}(c_{\max}^G + 1, P))$ is a direct minimal consistency improving rule.*

Proof. Let $t : G \Longrightarrow_{\rho', m} H$ be a transformation, $k = c_{\max}^G + 1$ and e be the subcondition of c at layer k. We show that t is a direct minimal consistency improving transformation. First, we show that equation (2.1) is satisfied. Let $p : C_k \hookrightarrow G$ be a morphism. If ρ is a deleting potentially minimal improving rule, either 1. or 2. applies, if ρ is an inserting and not a deleting potentially minimal improving rule, only 2. applies, because ρ cannot destroy any occurrences of C_k in G.

1. If $p(C_k) \cap m(L \setminus K) \neq \emptyset$, $\mathrm{tr}_t \circ p$ is not total, since at least one element of $p(C_k)$ has been deleted by t and p does satisfy $\bigwedge_{C' \in \mathcal{U}(C_k, C_{k+1})}(p \models \mathrm{part}(1, e, C') \wedge \mathrm{tr}_t \circ p$ is total) $\implies \mathrm{tr}_t \circ p \models \mathrm{part}(1, e, C'))$.

2. If $p(C_k) \cap m(L \setminus K) = \emptyset$, $\mathrm{tr}_t \circ p$ is total, since no element of $p(C_k)$ has been deleted by t. Because (**??**) holds, t does not delete any elements of $C_{k+1} \setminus C_k$ and $p \models \mathrm{part}(1, e, C') \implies \mathrm{tr}_t \circ p \models \mathrm{part}(1, e, C')$ for all $C' \in (U)(C_k, C_{k+1})$.

With 1. and 2. follows that (2.1) is satisfied.

Second, we show that equation (2.2) is satisfied. Let $p' : C_k \hookrightarrow H$ be a morphism. Because (**??**) is satisfied, t does not create any elements of C_k and there must exist a morphism $p : C_k \hookrightarrow G$ with $\mathrm{tr}_t \circ p = p'$. If follows that (2.2) is satisfied.

Third, we show that (2.3) is satisfied. We consider the cases that firstly, ρ is a deleting minimal potentially improving rule and secondly, that ρ is an inserting and not a deleting minimal potentially improving rule.

1. If ρ is a deleting potentially minimal improving rule, the condition $\mathrm{ap_{pi}}(k, P) = \exists(b : L \hookrightarrow C_k, \neg\exists(a_{k+2} : C_k \hookrightarrow C_{k+1}, \text{true}))$ is satisfied by m. Therefore a morphism $p : C_k \hookrightarrow G$ with $p \not\models \neg\exists(a_{k+1} : C_k \hookrightarrow C_{k+1}, \text{true}) = \mathrm{part}(1, e, C_{k+1})$ and $m = p \circ b$ must exist. Since ρ is a deleting rule, at least one element of $p(C_k)$ has been deleted by t and $\mathrm{tr}_t \circ p$ is not total. It follows that (2.3) is satisfied.

2. If ρ is an inserting and not a deleting potentially minimal improving rule, $\mathrm{ap_{pi}}(k, P) = \neg\exists(b : L \hookrightarrow P, \text{true})$ is satisfied by m. Because $L = C_k$, $m \models \neg\exists(b : C_k \hookrightarrow P, \text{true}) = \mathrm{part}(1, e, P)$. Since (**??**) is satisfied, $\mathrm{tr} \circ p$ is total and $\mathrm{tr}_t \circ p \models \mathrm{part}(1, e, P)$. Therefore, (2.3) is satisfied.

23

Last, since (**??**) and (**??**) are satisfied, ρ cannot create any occurrence of C_j with $j \leq k$ and j being even and ρ cannot delete any occurrences of C_j with $j \leq k$ and j being odd. Therefore, (2.4) and (2.5) are satisfied.

In total follows that t is a direct minimal consistency improving transformation and ρ' is a direct minimal consistency improving rule. $\qquad\square$

Theorem 3.3. *Let a constraint c in EANF, a basic increasing rule ρ with P and a consistency increasing transformation $t : G \Longrightarrow_{\rho,m} H$ w.r.t c be given. Then, $m \models \mathrm{ap}_{\mathrm{pi}}(c_{\max}^G, P)$.*

Proof. Let $k = c_{\max}^G$.

1. If c is a inserting rule, then $\mathrm{ap}_{\mathrm{pi}}(c_{\max}^G, P) = \neg \exists (a_k^p : L \hookrightarrow P, \mathsf{true})$. Let $n : R \hookrightarrow H$ be the comatch of t. Since t is a consistency increasing transformation, there does exist a morphism $p : C_k \hookrightarrow G$ with $p \not\models \exists (a_k^p : C_k \hookrightarrow P, \mathsf{true})$, $\mathrm{tr}_t \circ p \models \exists (a_k^p : C_k \hookrightarrow P, \mathsf{true})$ such that $p'(P) \cap n(R \setminus K)$ for the morphism $p' : P \hookrightarrow H$ with $p = p \circ a_k^p$. Assume that $m \models \exists (a_k^p : L \hookrightarrow P, \mathsf{true})$. Then, every morphism $p : C_k \hookrightarrow G$ with $\mathrm{tr}_t \circ p \models \exists (a_k^p : C_k \hookrightarrow P, \mathsf{true})$ with $p'(P) \cap n(R \setminus K)$ for the morphism $p' : P \hookrightarrow H$ with $p = p' \circ a_k^p$ already satisfies $\exists (a_k^p : C_k \hookrightarrow P, \mathsf{true})$. This is a contradiction.

2. If ρ is a deleting rule, $\mathrm{ap}_{\mathrm{pi}}(k, P) = \bigvee_{P \in \mathrm{ol}(L, C_k)} \bigwedge_{P' \in \mathrm{eol}(P, a_k)} \exists (i_L : L \hookrightarrow P, \neg \exists (i_P : P \hookrightarrow P', \mathsf{true}))$. Since t is a consistency increasing transformation, there does exist a morphism $p : C_k \hookrightarrow G$, such that $p \not\models \exists (a_k : C_k \hookrightarrow C_{k+1}, \mathsf{true})$ and $\mathrm{tr}_t \circ p$ is not total. Assume that $m \not\models \mathrm{ap}_{\mathrm{pi}}(k, P)$. Then, for each overlap Q of L and C_k, the inclusion i_{C_k} does satisfy $\exists (a_k : C_k \hookrightarrow C_{k+1}, \mathsf{true})$. This is a contradiction, because $p(C_k) \cap m(L \setminus K) \neq \emptyset$ has to hold.

$\qquad\square$

Lemma 3.8. *Let a constraint c in EANF and inserting basic increasing rules ρ_1, ρ_2 at layer k with P_1 and P_2, respectively, be given. Then, the rules $\rho_1' = (\rho_1, \mathrm{ap}_{\mathrm{pi}}(k, P_1))$ and $\rho_2' = (\rho_2, \mathrm{ap}_{\mathrm{pi}}(k, P_2))$ are sequentially independent.*

Proof. Let $G_1 \Longrightarrow_{\rho_1', m_1} G_2 \Longrightarrow_{\rho_2', m_2} G_3$ be a sequence of transformations. First, note that this sequnce can only exist if $P_1 \not\subseteq P_2$ and $P_2 \not\subseteq P_1$ Since both rules are do not insert elements of C_k, a morphism $d_1 : L_2 \hookrightarrow D_1$ with $m_2 = h_1 \circ d_1$ exists. Because both rules do not delete any elements a morphism $d_2 : R_1 \hookrightarrow D_2$ with $m_1^* = g_2 \circ d_2$ exists. Because ρ_2 is not a basic increasing rule with P_1, $h_2 \circ d_2 \models R(\mathrm{ap}_{\mathrm{pi}})$ and because ρ_1 does not delete any elements of P_2 is holds that $g_1 \circ d_1 \models R(\mathrm{ap}_{\mathrm{pi}})$. $\qquad\square$

Definition 3.9 (**repairing rule set**). *Let a constraint c in EANF 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 $k = \mathrm{c}_{\max}$ a sequence*

$$G = G_0 \Rightarrow_{r_0} \dots \Rightarrow_{r_{n-1}} G_n = H$$

exists, such that $r_j \in \mathcal{R}$ for all $j \in \{0, \dots, n-1\}$ and $H \models_{k+2} c$.

Corollary 3.10. *Let a constraint c in EANF and a repairing rule set $\mathcal{R}$ for c at layer k, for all $k \in \{1, 3, \ldots, \mathrm{nl}(c)\}$, be given. Then, for all graphs G, a sequence*

$$G = G_0 \Rightarrow_{r_0} \ldots \Rightarrow_{r_{n-1}} G_n = H$$

exists, such that $r_j \in \mathcal{R}$ for all $j \in \{0, \ldots, n-1\}$ and $H \models c$.

Definition 3.11 (decomposition of a graph). *Let graphs $G_0 \subset G_1$ be given. A decomposition of G_1 with G_0 is a minimal set*

$$\mathbf{P} \subseteq \{P_v \mid v \in V_{C_1 \setminus C_0}\}$$

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

Lemma 3.12. *Let graphs $G_0 \subset G_1$ and a decomposition $\mathbf{P}$ of G_1 with G_0 be given. Then, for each pair $P, P' \in \mathbf{P}$ with $P \neq P'$ the following holds:*

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

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

Lemma 3.13. *Let graphs $G_0 \subset G_1$ and a decomposition $\mathbf{P}$ 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 $E_{G_1} \subseteq E_H$. $\square$

Theorem 3.4. *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, $(\rho, \mathrm{ap}_{\mathrm{pi}}(j, C_{j+1})) \in \mathcal{R}$ 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 $(\rho, \mathrm{ap_{pi}}(k, P)) \in \mathcal{R}$ 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 $\rho_i = L_i \xleftarrow{l_i} K_i \xrightarrow{r_i} R_i$ at layer k with C_i exists such that $L_i \hookrightarrow R_{i-1}$ and $e_i = i_{Ci-1}^{C_i} \circ e_{i-1}$ 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 3.4 is satisfied.

1. Assume that 1. of theorem 3.4 holds. Let $(\rho, \mathrm{ap_{pi}}(j, C_{j+1})) \in \mathcal{R}$, such that $\rho = L \xleftarrow{l} K \xrightarrow{r} R$ 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, $\mathrm{ap_{pi}}(j, C_{j+1}) = \exists(b : L \hookrightarrow C_j, \neg\exists(a_j : C_j \hookrightarrow C_{j+1}, \mathsf{true}))$. 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 $m_1 \models \mathrm{ap_{pi}}(j, C_{j+1})$. Since r only deletes edges, a transformation $t : G = G_0 \Rightarrow_{r,m_1} G_1$ exists and $\mathrm{tr}_t \circ p$ 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 3.4 holds. Let $\rho_0 = (\rho, \mathrm{ap_{pi}}(k, P)) \in \mathcal{R}$, such that ρ is an inserting basic improving rule of c at layer k with $P \in \mathbf{P}$. Then, $\mathrm{ap_{pi}}(k, P) = \neg\exists(b : L \hookrightarrow P, \mathsf{true})$. 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 $m_0 \models \neg\exists(a_k' : C_k \hookrightarrow P, \mathsf{true}) = \mathrm{ap_{pi}}(k, P)$. Because r does not delete any elements, a transformation $t_0 : G \Longrightarrow_{r_0,m_0} G_1$ exists and $\mathrm{tr}_t \circ q \models \exists(a_k' : C_k \hookrightarrow P, \mathsf{true})$. We set $q_1 = \mathrm{tr}_{t_0} \circ q_0$ and apply the same method to q_1.

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

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

Algorithm 1: Floyd-Warshall Algorithmus

Data: A graph G, a constraint c in EANF and a set of rules $\mathcal{R}$

Result: A graph H with $H \models c$

1 extract basic rules of $\mathcal{R}$;

2 if $\mathcal{R}$ *is a repairing set* **then**

3 | repair

4 else

5 | repair with basic rules + application condition;

6 | repair with normal rules + application condition;

7 | generate a basic increasing rule set $\mathcal{R}'$;

8 | repair with rules of $\mathcal{R}'$;

9 end

---

such that $m_i = \mathrm{tr}_{t_{i-1}} \circ \ldots \circ \mathrm{tr}_{t_0} \circ m_0$ and $q \models \exists(b_i : C_k \hookrightarrow P_i, \mathsf{true})$ for all $P_i \in \mathbf{P}$ with $q = \mathrm{tr}_{t_n} \circ q_n$. 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 $H \models_{k+2} c$.

$\square$

Corollary 3.14. *If a set of rule $\mathcal{R}$ is a repairing set of c at layer $k \leq \mathrm{nl}(c)$ and 1. of theorem 3.4 applies, then $\mathcal{R}$ is a repairing set of c at layer j for all $k \leq j \leq \mathrm{nl}(c)$.*

Lemma 3.15. *Let a graph G_0, a constraint c in EANF and a repairing set $\mathcal{R}$ at layer $c_{\max}^{G_0}+1$ 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 $\rho_i = (\rho, \mathrm{ap}_{\mathrm{pi}}(c_{\max}^{G_0} + 1, P)$ with $\rho \in \mathcal{R}$ being a consistency increasing rule at layer $c_{\max}^{G_0} + 1$ with P, it holds that
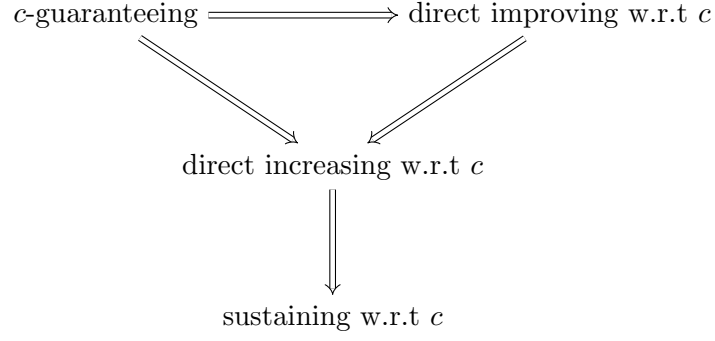
Figure 6: consistency relations for basic rules.

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.