Detecting useless critical pairs

Cyrille Chenavier

Université Paris-Est Marne-la-Vallée

Laboratoire d'informatique Gaspard-Monge

September 10, 2017

Plan

I. Motivations

- ▷ The CPC procedure
- ▷ Useless reductions for completion
- ▷ Linear algebra and useless reductions

II. Reduction operators

- ▷ Definition of reduction operators
- ▷ Reduction operators and labelled reductions

III. Lattice criterion for rejecting useless reductions

- ▷ Lattice structure of reduction operators
- ▷ The criterion

Plan

I. Motivations

The CPC procedure

- The Critical Pair/Completion (CPC) procedure appears in:
  - Theorem proving,
  - Polynomial ideal theory,
  - Term rewriting,
  - $\ldots$

The CPC procedure

- The Critical Pair/Completion (CPC) procedure appears in:
  - Theorem proving,
  - Polynomial ideal theory,
  - Term rewriting,
  - ...
- Two directions for improving the CPC procedure:

The CPC procedure

- The Critical Pair/Completion (CPC) procedure appears in:
  - Theorem proving,
  - Polynomial ideal theory,
  - Term rewriting,
  - . . .
- Two directions for improving the CPC procedure:
  - Find selection strategies:
    - Add the new critical pairs in a stack or in queue,
    - Reduce the critical pairs in parallel,
    - . . .

The CPC procedure

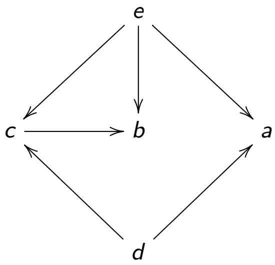
- The Critical Pair/Completion (CPC) procedure appears in:
  - Theorem proving,
  - Polynomial ideal theory,
  - Term rewriting,
  - ...
- Two directions for improving the CPC procedure:
  - Find selection strategies:
    - Add the new critical pairs in a stack or in queue,
    - Reduce the critical pairs in parallel,
    - ...
  - Find criteria for avoiding useless critical pairs.
    - A first type is based on the structure of branchings.
    - A second type consists in rejecting useless reductions.

Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.

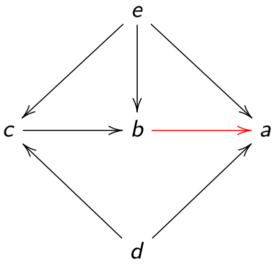
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



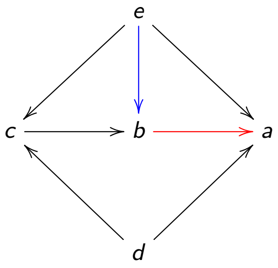
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



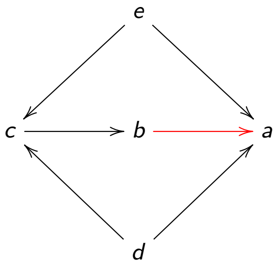
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



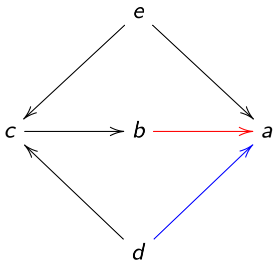
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



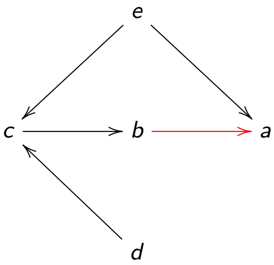
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



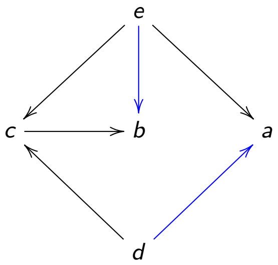
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



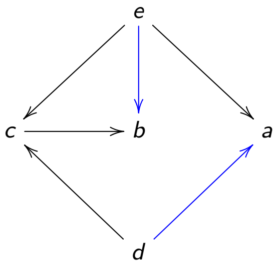
Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



Useless reductions for completion

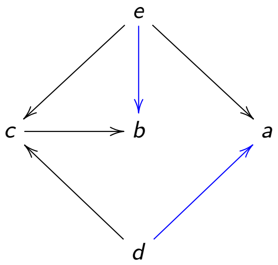
- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



- Our goal: introduce a lattice criterion for detecting useless reductions.

Useless reductions for completion

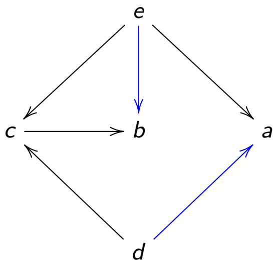
- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



- Our goal: introduce a lattice criterion for detecting useless reductions.
  - We use linear algebra techniques.

Useless reductions for completion

- We are interested in rejecting useless reductions.
  - We consider abstract rewriting systems.
- Example of useless reductions:



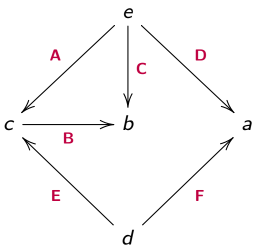
- Our goal: introduce a lattice criterion for detecting useless reductions.
  - We use linear algebra techniques.

Remarks.

- Buchberger's algorithm is interpreted as Gaussian elimination [Lazard, 1983].
- A set of terms can be embedded into the vector space it spans.

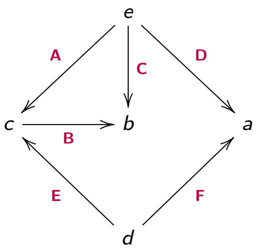
Linear algebra and useless reductions

- We consider labelled reductions.
  - e.g.,



Linear algebra and useless reductions

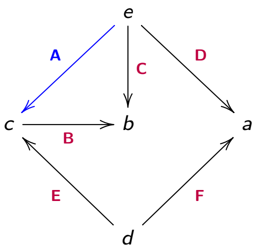
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.

Linear algebra and useless reductions

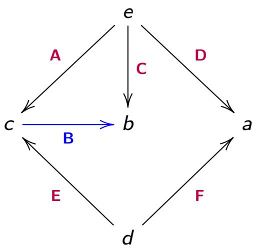
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$

Linear algebra and useless reductions

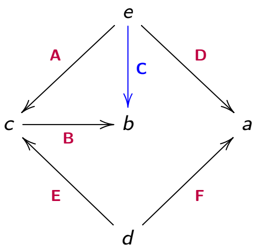
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$

Linear algebra and useless reductions

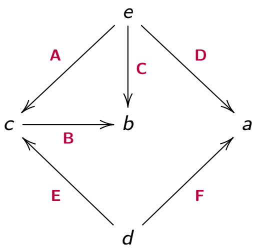
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$

Linear algebra and useless reductions

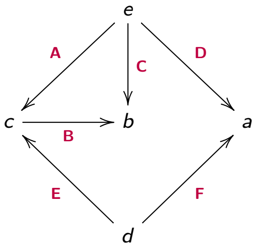
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$

Linear algebra and useless reductions

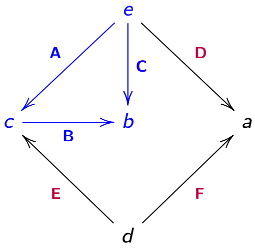
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$
- The loops in the reduction diagram are interpreted as algebraic relations.

Linear algebra and useless reductions

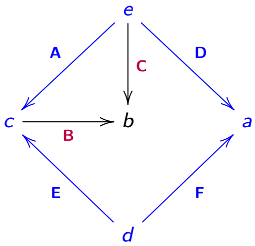
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$
- The loops in the reduction diagram are interpreted as algebraic relations.
  - e.g., **C'** $-$ **B'** $-$ **A'** $= 0$

Linear algebra and useless reductions

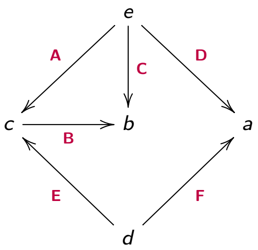
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$
- The loops in the reduction diagram are interpreted as algebraic relations.
  - e.g., **C'** $-$ **B'** $-$ **A'** $= 0$ and **F'** $-$ **E'** $-$ **D'** $+$ **A'** $= 0$.

Linear algebra and useless reductions

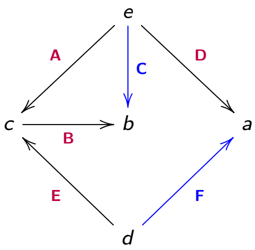
- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$
- The loops in the reduction diagram are interpreted as algebraic relations.
  - e.g., **C'** $-$ **B'** $-$ **A'** $= 0$ and **F'** $-$ **E'** $-$ **D'** $+$ **A'** $= 0$.
- The useless reductions correspond to leading terms of relations for a given order.

Linear algebra and useless reductions

- We consider labelled reductions.
  - e.g.,



- For every reduction, we consider the difference between its source and its target.
  - e.g., **A'** $= e - c$, **B'** $= c - b$, **C'** $= e - b$, $\cdots$
- The loops in the reduction diagram are interpreted as algebraic relations.
  - e.g., **C'** $-$ **B'** $-$ **A'** $= 0$ and **F'** $-$ **E'** $-$ **D'** $+$ **A'** $= 0$.
- The useless reductions correspond to leading terms of relations for a given order.
  - e.g., for **A** $\sqsubset$ **B** $\sqsubset \cdots \sqsubset$ **F** the useless reductions are **C** and **F**.

Plan

II. Reduction operators

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\mathrm{lt}(T(g)) < g$.

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\mathrm{lt}(T(g)) < g$.
- The set of reduction operators is denoted by $\mathbf{RO}(G, <)$.

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\mathrm{lt}(T(g)) < g$.
- The set of reduction operators is denoted by $\mathbf{RO}(G, <)$.

Reduction operators and labelled reductions.

- A reduction operator T encodes the labelled reductions $v \xrightarrow{\ell_{T,v}} T(v)$.

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\operatorname{lt}(T(g)) < g$.
- The set of reduction operators is denoted by $\mathbf{RO}(G,<)$.

Reduction operators and labelled reductions.

- A reduction operator T encodes the labelled reductions $v \xrightarrow{\ell_{T,v}} T(v)$.
- The labels of reductions induced by $F = \{T_1, \cdots, T_n\} \subset \mathbf{RO}(G,<)$ are ordered by:

$$\ell_{T_i,u} \sqsubset \ell_{T_j,v} := (i < j) \vee (i = j \wedge \operatorname{lt}(u) < \operatorname{lt}(v)).$$

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\mathrm{lt}(T(g)) < g$.
- The set of reduction operators is denoted by $\mathbf{RO}(G, <)$.

Reduction operators and labelled reductions.

- A reduction operator T encodes the labelled reductions $v \xrightarrow{\ell_{T,v}} T(v)$.
- The labels of reductions induced by $F = \{T_1, \cdots, T_n\} \subset \mathbf{RO}(G, <)$ are ordered by:

$$\ell_{T_i,u} \sqsubset \ell_{T_j,v} := (i < j) \vee (i = j \wedge \mathrm{lt}(u) < \mathrm{lt}(v)).$$

Definition and labelled reductions

- We fix V is a vector space equipped with a well-ordered basis $(G, <)$.

Definition.

- An endomorphism T of V is a reduction operator if
  - T is a projector,
  - $\forall g \in G$, we have either $T(g) = g$ or $\mathrm{lt}(T(g)) < g$.
- The set of reduction operators is denoted by $\mathbf{RO}(G, <)$.

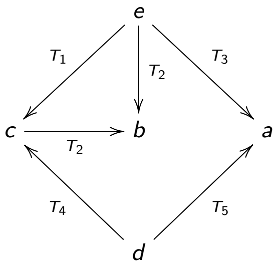
Reduction operators and labelled reductions.

- A reduction operator T encodes the labelled reductions $v \xrightarrow{\ell_{T,v}} T(v)$.
- The labels of reductions induced by $F = \{T_1, \cdots, T_n\} \subset \mathbf{RO}(G, <)$ are ordered by:

$$\ell_{T_i,u} \sqsubset \ell_{T_j,v} := (i < j) \vee (i = j \wedge \mathrm{lt}(u) < \mathrm{lt}(v)).$$

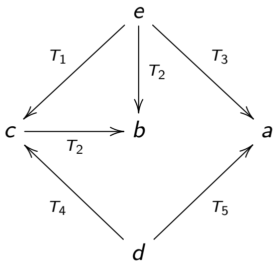
Example

- $G = \left\{ a < b < c < d < e \right\}$ and



Example

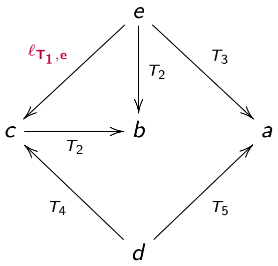
- $G = \left\{a < b < c < d < e\right\}$ and



- We obtain the following order on labels:

Example

- $G = \{a < b < c < d < e\}$ and

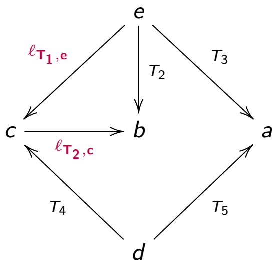


- We obtain the following order on labels:

$$\ell_{T_1,e}$$

Example

- $G = \{a < b < c < d < e\}$ and

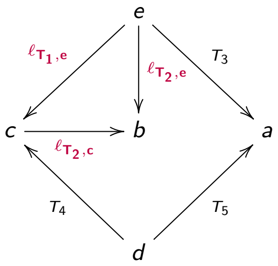


- We obtain the following order on labels:

$$\ell_{\mathbf{T_1},\mathbf{e}} \sqsubset \ell_{\mathbf{T_2},\mathbf{c}}$$

Example

- $G = \{a < b < c < d < e\}$ and

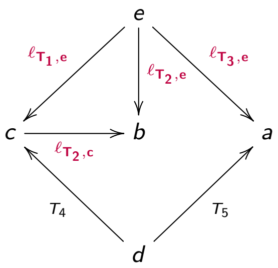


- We obtain the following order on labels:

$$\ell_{\mathbf{T_1},\mathbf{e}} \sqsubset \ell_{\mathbf{T_2},\mathbf{c}} \sqsubset \ell_{\mathbf{T_2},\mathbf{e}}$$

Example

- $G = \{a < b < c < d < e\}$ and

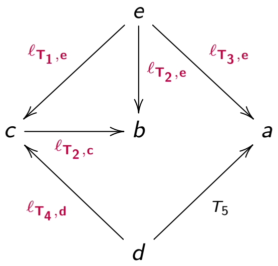


- We obtain the following order on labels:

$$\ell_{T_1,e} \sqsubset \ell_{T_2,c} \sqsubset \ell_{T_2,e} \sqsubset \ell_{T_3,e}$$

Example

- $G = \{a < b < c < d < e\}$ and

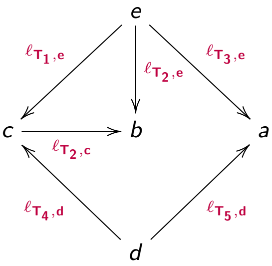


- We obtain the following order on labels:

$$\ell_{T_1,e} \sqsubset \ell_{T_2,c} \sqsubset \ell_{T_2,e} \sqsubset \ell_{T_3,e} \sqsubset \ell_{T_4,d}$$

Example

- $G = \{a < b < c < d < e\}$ and



- We obtain the following order on labels:

$$\ell_{T_1,e} \sqsubset \ell_{T_2,c} \sqsubset \ell_{T_2,e} \sqsubset \ell_{T_3,e} \sqsubset \ell_{T_4,d} \sqsubset \ell_{T_5,d}.$$

Plan

III. Lattice criterion for rejecting useless reductions

Lattice structure on $\mathbf{RO}(G, <)$ and the criterion

Lattice structure.

- The map $\mathbf{RO}(G,<) \longrightarrow \text{Subspaces}(V),\ T \longmapsto \ker(T)$ is a bijection [C. 2016].

Lattice structure on $\mathbf{RO}(G,<)$ and the criterion

Lattice structure.

- The map $\mathbf{RO}(G,<) \longrightarrow \text{Subspaces}(V), \ T \longmapsto \ker(T)$ is a bijection [C. 2016].
- $\mathbf{RO}(G,<)$ admits a lattice structure where:
  - $T_1 \preceq T_2$ if $\ker(T_2) \subseteq \ker(T_1)$,
  - $T_1 \wedge T_2 := \ker^{-1}(\ker(T_1) + \ker(T_2))$,
  - $T_1 \vee T_2 := \ker^{-1}(\ker(T_1) \cap \ker(T_2))$.

Lattice structure on $\mathbf{RO}(G,<)$ and the criterion

Lattice structure.

- The map $\mathbf{RO}(G,<) \longrightarrow \text{Subspaces}(V), \ T \longmapsto \ker(T)$ is a bijection [C. 2016].
- $\mathbf{RO}(G,<)$ admits a lattice structure where:
  - $T_1 \preceq T_2$ if $\ker(T_2) \subseteq \ker(T_1)$,
  - $T_1 \wedge T_2 := \ker^{-1}(\ker(T_1) + \ker(T_2))$,
  - $T_1 \vee T_2 := \ker^{-1}(\ker(T_1) \cap \ker(T_2))$.

The criterion.

- Let $T_1, \cdots, T_n \in \mathbf{RO}(G,<)$.

Lattice structure on $\mathbf{RO}(G,<)$ and the criterion

Lattice structure.

- The map $\mathbf{RO}(G,<) \longrightarrow \text{Subspaces}(V),\ T \longmapsto \ker(T)$ is a bijection [C. 2016].
- $\mathbf{RO}(G,<)$ admits a lattice structure where:
  - $T_1 \preceq T_2$ if $\ker(T_2) \subseteq \ker(T_1)$,
  - $T_1 \wedge T_2 := \ker^{-1}(\ker(T_1) + \ker(T_2))$,
  - $T_1 \vee T_2 := \ker^{-1}(\ker(T_1) \cap \ker(T_2))$.

The criterion.

- Let $T_1, \cdots, T_n \in \mathbf{RO}(G,<)$. Let $\tilde{T}_i \in \mathbf{RO}(G,<)$ defined by
  - $\tilde{T}_i(g) = g$ if $g \notin \text{im}((T_1 \wedge \cdots \wedge T_{i-1}) \vee T_i)$,
  - $\tilde{T}_i(g) = T_i(g)$, otherwise.

Lattice structure on $\mathbf{RO}(G,<)$ and the criterion

Lattice structure.

- The map $\mathbf{RO}(G,<) \longrightarrow \mathrm{Subspaces}(V)$, $T \longmapsto \ker(T)$ is a bijection [C. 2016].
- $\mathbf{RO}(G,<)$ admits a lattice structure where:
  - $T_1 \preceq T_2$ if $\ker(T_2) \subseteq \ker(T_1)$,
  - $T_1 \wedge T_2 := \ker^{-1}(\ker(T_1) + \ker(T_2))$,
  - $T_1 \vee T_2 := \ker^{-1}(\ker(T_1) \cap \ker(T_2))$.

The criterion.

- Let $T_1, \cdots, T_n \in \mathbf{RO}(G,<)$. Let $\tilde{T}_i \in \mathbf{RO}(G,<)$ defined by
  - $\tilde{T}_i(g) = g$ if $g \notin \mathrm{im}((T_1 \wedge \cdots \wedge T_{i-1}) \vee T_i)$,
  - $\tilde{T}_i(g) = T_i(g)$, otherwise.
- A completion of $\{\tilde{T}_1, \cdots, \tilde{T}_n\}$ leads to a completion of $\{T_1, \cdots, T_n\}$.

Lattice structure on $\mathbf{RO}(G,<)$ and the criterion

Lattice structure.

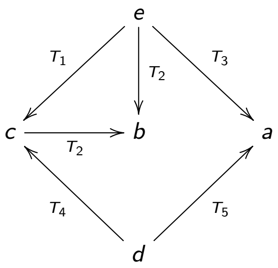
- The map $\mathbf{RO}(G,<) \longrightarrow \mathrm{Subspaces}(V)$, $T \longmapsto \ker(T)$ is a bijection [C. 2016].
- $\mathbf{RO}(G,<)$ admits a lattice structure where:
  - $T_1 \preceq T_2$ if $\ker(T_2) \subseteq \ker(T_1)$,
  - $T_1 \wedge T_2 := \ker^{-1}(\ker(T_1) + \ker(T_2))$,
  - $T_1 \vee T_2 := \ker^{-1}(\ker(T_1) \cap \ker(T_2))$.

The criterion.

- Let $T_1, \cdots, T_n \in \mathbf{RO}(G,<)$. Let $\tilde{T}_i \in \mathbf{RO}(G,<)$ defined by
  - $\tilde{T}_i(g) = g$ if $g \notin \mathrm{im}((T_1 \wedge \cdots \wedge T_{i-1}) \vee T_i)$,
  - $\tilde{T}_i(g) = T_i(g)$, otherwise.
- A completion of $\{\tilde{T}_1, \cdots, \tilde{T}_n\}$ leads to a completion of $\{T_1, \cdots, T_n\}$.
- The rejected reductions have labels $\ell_{T_i,g}$, with $g \notin \mathrm{im}((T_1 \wedge \cdots \wedge T_{i-1}) \vee T_i)$.

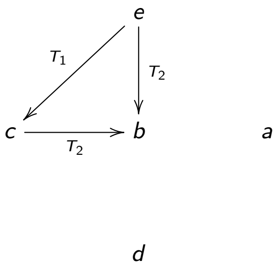
Illustration

- We consider:



Illustration

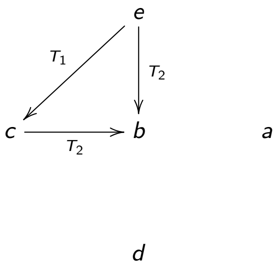
- We consider:



- Step 1: we have $e - c = (e - b) - (c - b)$.

Illustration

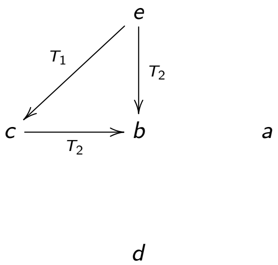
- We consider:



- Step 1: we have $e - c = (e - b) - (c - b)$.
  - $(e - c) \in \ker(T_1) \cap \ker(T_2) = \ker(T_1 \vee T_2)$.

Illustration

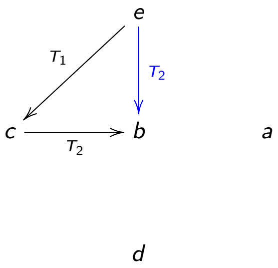
- We consider:



- Step 1: we have $e - c = (e - b) - (c - b)$.
  - $(e - c) \in \ker(T_1) \cap \ker(T_2) = \ker(T_1 \vee T_2)$.
  - $T_1 \vee T_2(e) = T_1 \vee T_2(c)$.
    - We have $\operatorname{lt}(T_1 \vee T_2(c)) \leq c$ and $c < e$.
    - Hence, $T_1 \vee T_2(e) \neq e$, so that $e \notin \operatorname{im}(T_1 \vee T_2)$.

Illustration

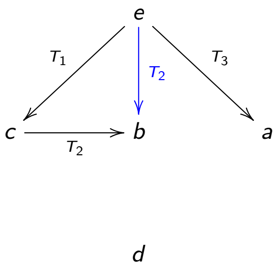
- We consider:



- Step 1: we have $e - c = (e - b) - (c - b)$.
  - $(e - c) \in \ker(T_1) \cap \ker(T_2) = \ker(T_1 \vee T_2)$.
  - $T_1 \vee T_2(e) = T_1 \vee T_2(c)$.
    - We have $\mathrm{lt}(T_1 \vee T_2(c)) \leq c$ and $c < e$.
    - Hence, $T_1 \vee T_2(e) \neq e$, so that $e \notin \mathrm{im}(T_1 \vee T_2)$.
  - The lattice criterion rejects the reduction $e \xrightarrow{T_2} b$.

Illustration

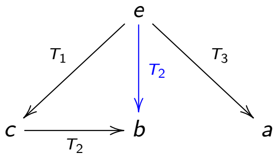
- We consider:



- Step 2: e and a are not connected by T_1 and T_2.

Illustration

- We consider:

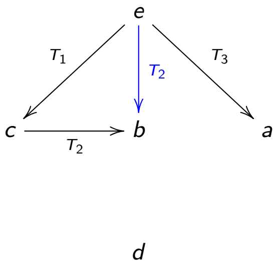


d

- Step 2: e and a are not connected by T_1 and T_2.
  - $T_1 \wedge T_2(e) \neq T_1 \wedge T_2(a)$.

Illustration

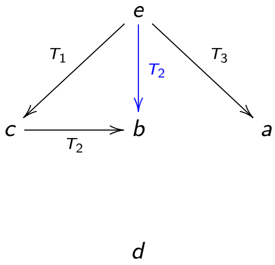
- We consider:



- Step 2: e and a are not connected by T_1 and T_2.
  - $T_1 \wedge T_2(e) \neq T_1 \wedge T_2(a)$.
  - $\ker(T_1 \wedge T_2) \cap \ker(T_3) = \{0\}$.

Illustration

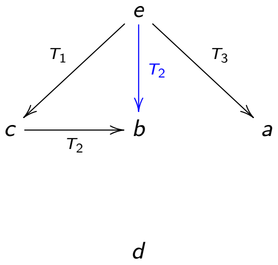
- We consider:



- Step 2: e and a are not connected by T_1 and T_2.
  - $T_1 \wedge T_2(e) \neq T_1 \wedge T_2(a)$.
  - $\ker(T_1 \wedge T_2) \cap \ker(T_3) = \{0\}$.
  - $(T_1 \wedge T_2) \vee T_3 = \mathrm{Id}_V$.

Illustration

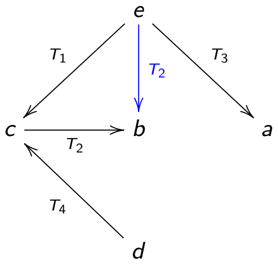
- We consider:



- Step 2: e and a are not connected by T_1 and T_2.
  - $T_1 \wedge T_2(e) \neq T_1 \wedge T_2(a)$.
  - $\ker(T_1 \wedge T_2) \cap \ker(T_3) = \{0\}$.
  - $(T_1 \wedge T_2) \vee T_3 = \mathrm{Id}_V$.
  - The criterion rejects no reduction.

Illustration

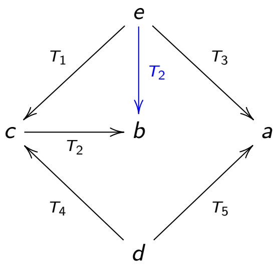
- We consider:



- Step 3: the criterion rejects no reduction.

Illustration

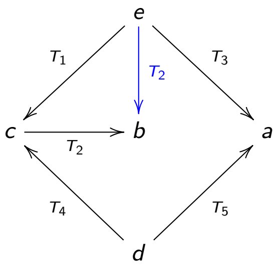
- We consider:



- Step 4: we have $d - a = (d - c) - (e - c) + (e - a)$.

Illustration

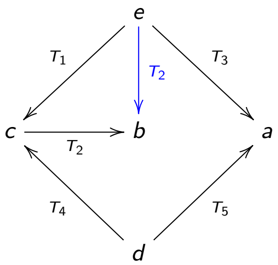
- We consider:



- Step 4: we have $d - a = (d - c) - (e - c) + (e - a)$.
  - $(d - a) \in \ker((T_1 \wedge \cdots \wedge T_4) \vee T_5)$.

Illustration

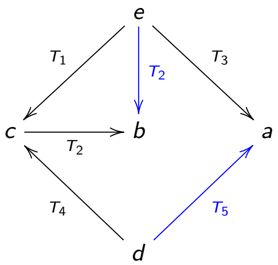
- We consider:



- Step 4: we have $d - a = (d - c) - (e - c) + (e - a)$.
  - $(d - a) \in \ker((T_1 \wedge \cdots \wedge T_4) \vee T_5)$.
  - $d \notin \operatorname{im}((T_1 \wedge \cdots \wedge T_4) \vee T_5)$.

Illustration

- We consider:



- Step 4: we have $d - a = (d - c) - (e - c) + (e - a)$.
  - $(d - a) \in \ker((T_1 \wedge \cdots \wedge T_4) \vee T_5)$.
  - $d \notin \operatorname{im}((T_1 \wedge \cdots \wedge T_4) \vee T_5)$.
  - The criterion rejects the reduction $d \xrightarrow{T_5} a$.

Thank you for listening!