Specification and Automatic Verification of Computational Reductions

Julien Grange[1], Fabian Velken[2], Nils Vortmeier[2], Thomas Zeume[2]

[1]LACL, Université Paris-Est Créteil, France
[2]Ruhr Universität Bochum, Germany

Complexity Days
December 14th, 2023

Teaching reductions

Goal : develop a platform to help students learn complexity theory

- ✓ Understand the classic reductions
- ☠ Design their own reductions, and get feedback

Teaching reductions

Goal : develop a platform to help students learn complexity theory

- ✓ Understand the classic reductions
- ☠ Design their own reductions, and get feedback
  - easy-to-grasp specification language for reductions
  - automatic tools to check the validity of such reductions
  - produce a counter-example if the reduction is incorrect

Specification language

To specify formally a reduction $P \leq P^\star$, either

- give an algorithmic procedure

$$\text{instance of } P \quad \mapsto \quad \text{instance of } P^\star$$

Specification language

To specify formally a reduction $P \leq P^{\star}$, either

- give an algorithmic procedure

$$\text{instance of } P \quad \mapsto \quad \text{instance of } P^{\star}$$

  ☠ procedural
  ✓ left-to-right

Specification language

To specify formally a reduction $P \leq P^\star$, either

- give an algorithmic procedure

$$\text{instance of } P \quad \mapsto \quad \text{instance of } P^\star$$

  - procedural
  - ✓ left-to-right

- give an FO-interpretation to define instances of $P^\star$ in instances of P

Definition (FO-interpretation between graphs)

$\rho = \left(\varphi_{\text{domain}}(\bar{x}), \varphi_{\sim}(\bar{x}, \bar{y}), \varphi_E(\bar{x}, \bar{y})\right)$ transforms

- a graph $\mathcal{G} = (V, E)$
- into the graph $\rho(\mathcal{G}) := \left({}^{\varphi_{\text{domain}}(\mathcal{G})}\big/_{\varphi_{\sim}(\mathcal{G})}, \varphi_E(\mathcal{G}) \right)$

Specification language

To specify formally a reduction $P \leq P^\star$, either

- give an algorithmic procedure

$$\text{instance of } P \quad \mapsto \quad \text{instance of } P^\star$$

  - ☠ procedural
  - ✓ left-to-right
- give an FO-interpretation to define instances of $P^\star$ in instances of P
  - ✓ declarative
  - ☠ right-to-left

Definition (FO-interpretation between graphs)

$\rho = \left(\varphi_{\text{domain}}(\bar{x}), \varphi_{\sim}(\bar{x}, \bar{y}), \varphi_E(\bar{x}, \bar{y})\right)$ transforms

- a graph $\mathcal{G} = (V, E)$
- into the graph $\rho(\mathcal{G}) := \left({}^{\varphi_{\text{domain}}(\mathcal{G})}\!\big/_{\varphi_{\sim}(\mathcal{G})}, \varphi_E(\mathcal{G}) \right)$

Specification language

To specify formally a reduction $P \leq P^\star$, either

- give an algorithmic procedure

$$\text{instance of } P \quad \mapsto \quad \text{instance of } P^\star$$

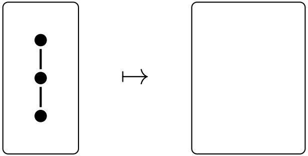
  - ☠ procedural
  - ✓ left-to-right
- give an FO-interpretation to define instances of $P^\star$ in instances of P
  - ✓ declarative
  - ☠ right-to-left

Best of both worlds :

- ✓ declarative
- ✓ left-to-right

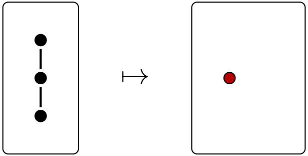
Specification language : *cookbook reductions*

Cookbook reduction



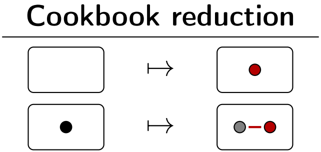
$$k\!-\!\textsc{Clique} \leq (k\!+\!1)\!-\!\textsc{Clique}$$

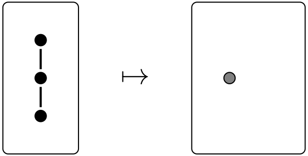
Cookbook reduction



$$k-\textsc{Clique} \leq (k+1)-\textsc{Clique}$$

Specification language : *cookbook reductions*

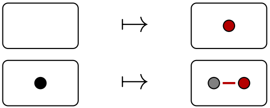


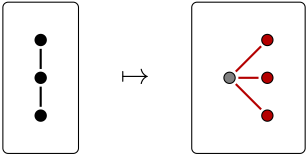


$$k\!-\!\textsc{Clique} \leq (k\!+\!1)\!-\!\textsc{Clique}$$

Specification language : *cookbook reductions*

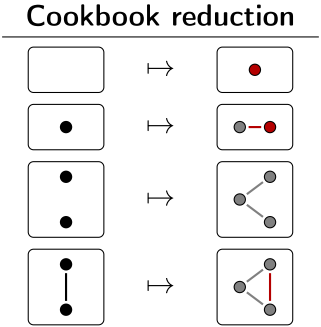
Cookbook reduction

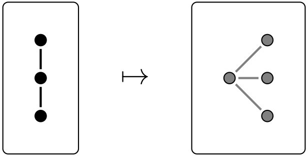




$$k-\textsc{Clique} \leq (k+1)-\textsc{Clique}$$

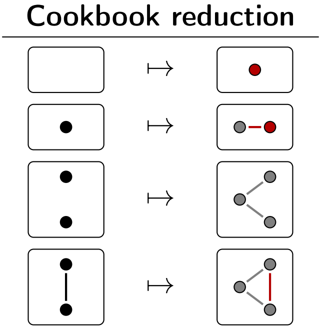
Specification language : *cookbook reductions*

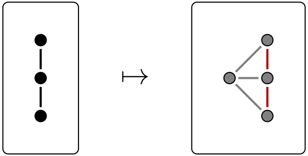




$$k\text{−}\textsc{Clique} \leq (k+1)\text{−}\textsc{Clique}$$

Specification language : *cookbook reductions*

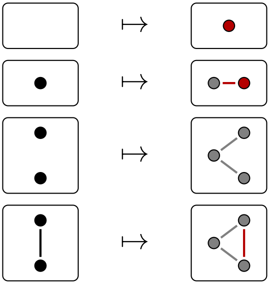


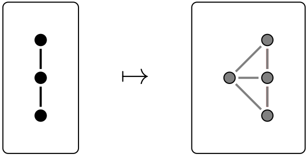


$k-\textsc{Clique} \leq (k+1)-\textsc{Clique}$

Specification language : *cookbook reductions*

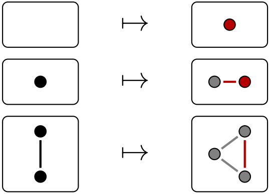
Cookbook reduction



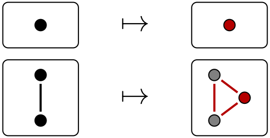


$$k-\textsc{Clique} \leq (k+1)-\textsc{Clique}$$

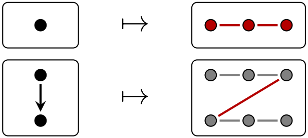
Specification language : *cookbook reductions*



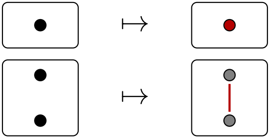
$k-\textsc{Clique} \leq (k+1)-\textsc{Clique}$



$k-\textsc{VertexCover} \leq k-\mathrm{FVS}$



$\textsc{HamCycle}_d \leq \textsc{HamCycle}_u$



$k-\textsc{Clique} \leq k-\textsc{IndepSet}$

Specification language : *cookbook reductions*

Theorem

Every cookbook reduction is equivalent to a quantifier-free interpretation

...but not all QF-interpretations are cookbook reductions

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- ☠ for $\mathcal{R} = \{\text{FO-interpretations}\}$ and any non-trivial $P^\star$

Checking reduction correctness

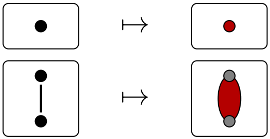
For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO
- for $\mathcal{R} = \{$edge-gadget reductions$\}$ and some $P^\star \in \mathrm{AC}^0$



Edge-gadget reductions

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- ☠ for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- ☠ for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO
- ☠ for $\mathcal{R} = \{$edge-gadget reductions$\}$ and some $P^\star \in \mathsf{AC}^0$

For **fixed** $P, P^\star$, whether $r \in \mathcal{R}$ is a reduction $P \leq P^\star$ is **decidable**

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- ☠ for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- ☠ for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO
- ☠ for $\mathcal{R} = \{$edge-gadget reductions$\}$ and some $P^\star \in \mathrm{AC}^0$

For **fixed** $P, P^\star$, whether $r \in \mathcal{R}$ is a reduction $P \leq P^\star$ is **decidable**

- ✓ for $\mathcal{R} = \{$cookbook reductions of arity $\leq r\}$, any P, and $P^\star \in$ FO

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- ☠ for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- ☠ for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO
- ☠ for $\mathcal{R} = \{$edge-gadget reductions$\}$ and some $P^\star \in \mathrm{AC}^0$

For **fixed** $P, P^\star$, whether $r \in \mathcal{R}$ is a reduction $P \leq P^\star$ is **decidable**

- ✓ for $\mathcal{R} = \{$cookbook reductions of arity $\leq r\}$, any P, and $P^\star \in$ FO
- ✓ for $\mathcal{R} = \{$edge-gadget reductions$\}$, any P, and $P^\star \in$ MSO

Checking reduction correctness

For **fixed** $P^\star$, whether $r \in \mathcal{R}$ is a reduction $\emptyset \leq P^\star$ is **undecidable**

- ☠ for $\mathcal{R} = \{$FO-interpretations$\}$ and any non-trivial $P^\star$
- ☠ for $\mathcal{R} = \{$QF-interpretations$\}$ and some $P^\star \in$ FO
- ☠ for $\mathcal{R} = \{$edge-gadget reductions$\}$ and some $P^\star \in \mathrm{AC}^0$

For **fixed** $P, P^\star$, whether $r \in \mathcal{R}$ is a reduction $P \leq P^\star$ is **decidable**

- ✓ for $\mathcal{R} = \{$cookbook reductions of arity $\leq r\}$, any P, and $P^\star \in$ FO
- ✓ for $\mathcal{R} = \{$edge-gadget reductions$\}$, any P, and $P^\star \in$ MSO

For **input** $P, P^\star$, whether $r \in \mathcal{R}$ is a reduction $P \leq P^\star$ is **decidable**

- ✓ for $\mathcal{R} = \{$QF-interpretations$\}$, $P, P^\star \in \exists$FO

Elements of proof

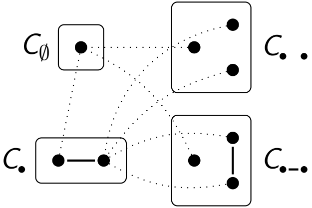
Theorem

Fix any problem P and any $P^\star \in \mathsf{FO}$. One can decide whether a cookbook reduction of arity $\leq r$ is a valid reduction $P \leq P^\star$.

Elements of proof

Theorem

Fix any problem P and any $P^\star \in \mathsf{FO}$. One can decide whether a cookbook reduction of arity $\leq r$ is a valid reduction $P \leq P^\star$.

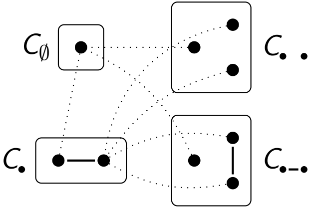


The recipe for the cookbook reduction ρ of arity 2 from $k-$Clique to $(k+1)-$Clique

Elements of proof

Theorem

Fix any problem P and any $P^\star \in$ FO. One can decide whether a cookbook reduction of arity $\leq r$ is a valid reduction $P \leq P^\star$.



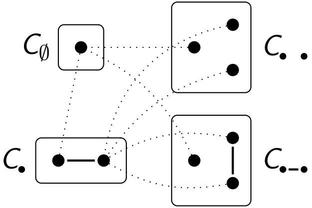
The recipe for the cookbook reduction ρ of arity 2 from $k-\textsc{Clique}$ to $(k+1)-\textsc{Clique}$

$\rho(\mathcal{A})$ can be FO-interpreted in $\mathcal{A} \uplus \text{recipe}(\rho)$.

Elements of proof

Theorem

Fix any problem P and any $P^\star \in \mathsf{FO}$. One can decide whether a cookbook reduction of arity $\leq r$ is a valid reduction $P \leq P^\star$.

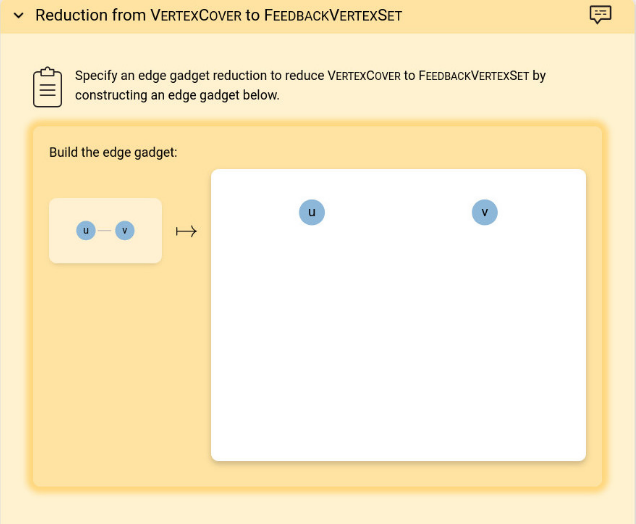


The recipe for the cookbook reduction ρ of arity 2 from $k-\textsc{Clique}$ to $(k+1)-\textsc{Clique}$

$\rho(\mathcal{A})$ can be FO-interpreted in $\mathcal{A} \uplus \mathrm{recipe}(\rho)$.

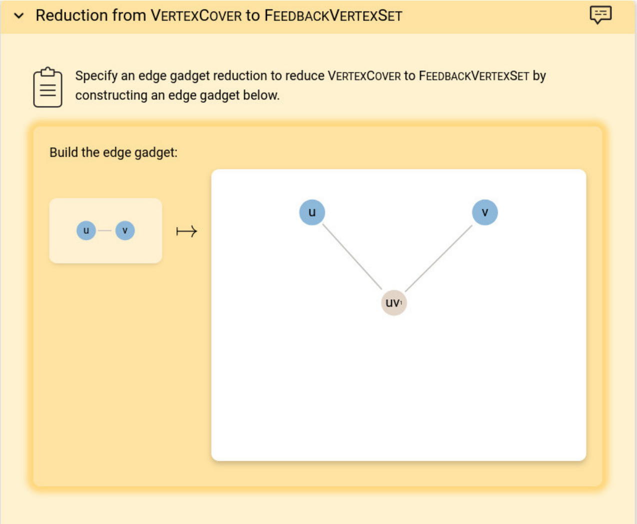
Hence, the correction of ρ only depends on the FO-type of its recipe at some depth.

Prototype on Iltis

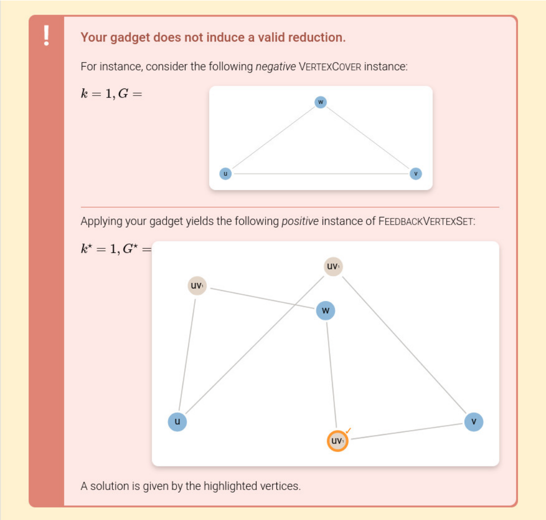


Enter your gadget-reduction

Prototype on Iltis



Enter your gadget-reduction



Wrong reduction: feedback via counter-example

Conclusion and perspectives

Cookbook reductions : specification langage for reductions

- ✓ intuitive
- ✓ powerful enough for many common reductions
- ✓ good decidable properties compared to other languages

Conclusion and perspectives

Cookbook reductions : specification langage for reductions

- ✓ intuitive
- ✓ powerful enough for many common reductions
- ✓ good decidable properties compared to other languages

To come :

- allow parameter manipulation (e.g. k in $k-\textsc{Clique}$)

Conclusion and perspectives

Cookbook reductions : specification langage for reductions

- ✓ intuitive
- ✓ powerful enough for many common reductions
- ✓ good decidable properties compared to other languages

To come :

- allow parameter manipulation (e.g. k in $k-\textsc{Clique}$)
- develop and deploy this feature on Iltis

Conclusion and perspectives

Cookbook reductions : specification langage for reductions

- ✓ intuitive
- ✓ powerful enough for many common reductions
- ✓ good decidable properties compared to other languages

To come :

- allow parameter manipulation (e.g. k in $k-\textsc{Clique}$)
- develop and deploy this feature on Iltis

Theorem

Fix any problem P and any $P^\star \in \mathrm{MSO}$. One can decide whether an edge-gadget reduction is a valid reduction $P \leq P^\star$.

Conclusion and perspectives

Cookbook reductions : specification langage for reductions

- ✓ intuitive
- ✓ powerful enough for many common reductions
- ✓ good decidable properties compared to other languages

To come :

- allow parameter manipulation (e.g. k in $k-\textsc{Clique}$)
- develop and deploy this feature on Iltis

~~Theorem~~ Conjecture

Fix any problem P and any $P^\star \in \cancel{\mathsf{MSO}}\ \mathsf{MSO}^2$. One can decide whether ~~an edge-gadget reduction~~ a cookbook reduction of arity $\leq r$ is a valid reduction $P \leq P^\star$.