CNRS Concours 06/02

Logique Monadique du Second Ordre
et
Beaux Préordres
pour les
Méthodes Formelles

Aliaume Lopez
Université de Varsovie

à Paris
le 2025-03-14

Parcours Académique

ENS Paris-Saclay (2015 – 2019)

Agrégation de Mathématiques

Stages de L3 et M1 à Birmingham et Ljubljana

Stage M2 au LSV
Goubault-Larrecq & Schmitz

LMF & IRIF (2019 – 2023)

Thèse sous la direction de
Goubault-Larrecq & Schmitz

2 Prix de Thèse
Ackermann Award &
E. W. Beth Dissertation Prize

Césure (1 an)
Autorité de Sûreté Nucléaire

Varsovie (2023 – 2025)

Postdoctorat
Mikołaj Bojańczyk

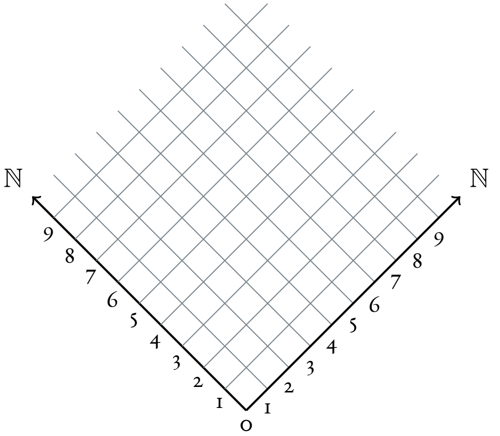
Co-organisation Autobóz 2024

Membre du comité de programme de **CSL'26**

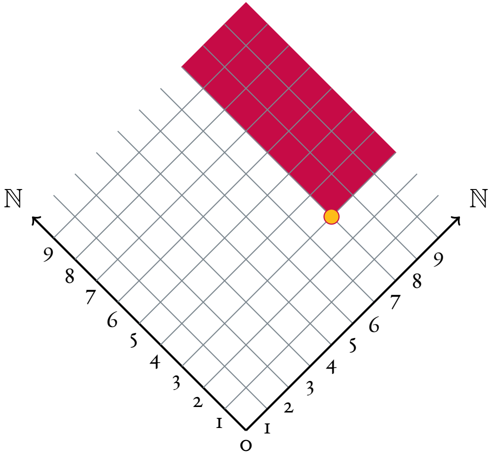
Co-encadrement de 2 stagiaires

« Théorèmes de préservation pour la logique au premier ordre : localité, topologie et constructions limites. »

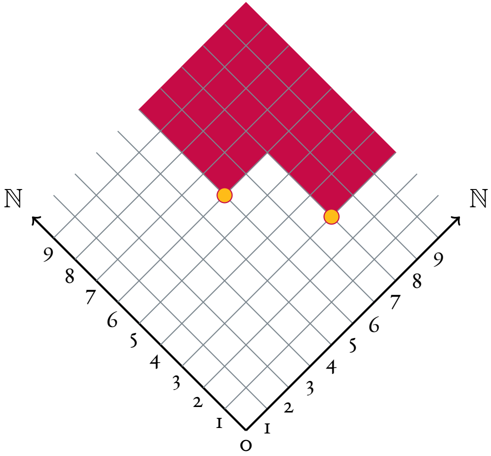
Conférences	8 (dont 4 en seul auteur)
Journaux	2
Soumissions	2



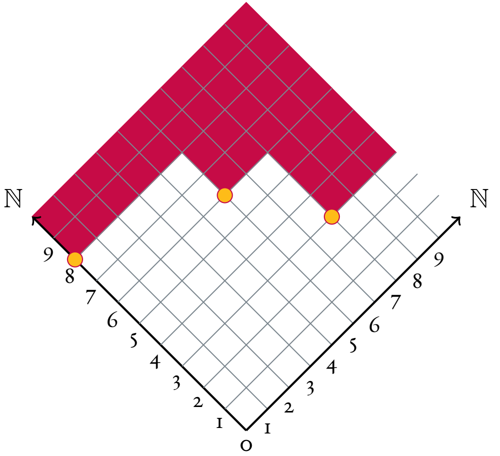
$(\mathbb{N} \times \mathbb{N}, \leq)$



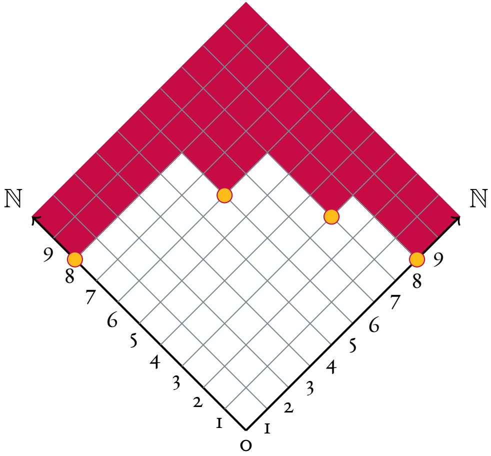
$(\mathbb{N} \times \mathbb{N}, \leq)$



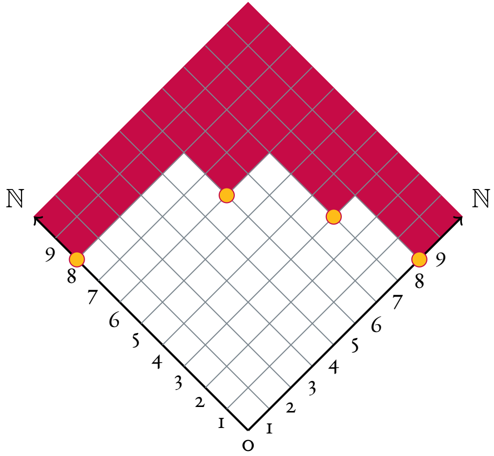
$(\mathbb{N} \times \mathbb{N}, \leq)$



$(\mathbb{N} \times \mathbb{N}, \leq)$



$(\mathbb{N} \times \mathbb{N}, \leq)$



$(\mathbb{N} \times \mathbb{N}, \leq)$

Graphes

Robertson & Seymour

Mineurs de graphes

Algèbre

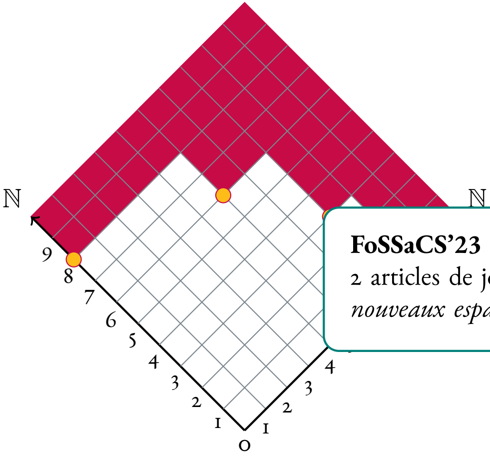
Hilbert / Gröbner

Calcul symbolique

Vérification

Systèmes de transition bien structurés (WSTS)

$(\mathbb{N}^k, \rightarrow, \leq) \rightsquigarrow$ réseau de Pétri



$(\mathbb{N} \times \mathbb{N}, \leq)$

FoSSaCS'23 : théorème de point fixe
2 articles de journaux : *types d'ordres* et *nouveaux espaces*

Graphes

Robertson & Seymour

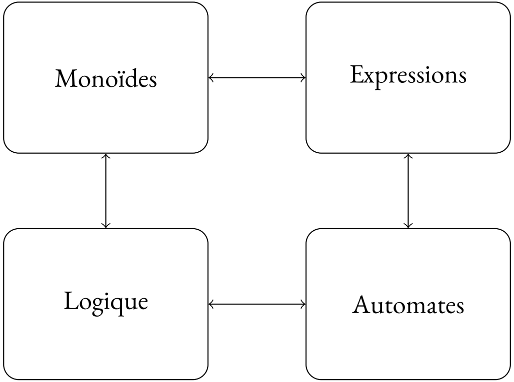
Algèbre

Hilbert / Gröbner

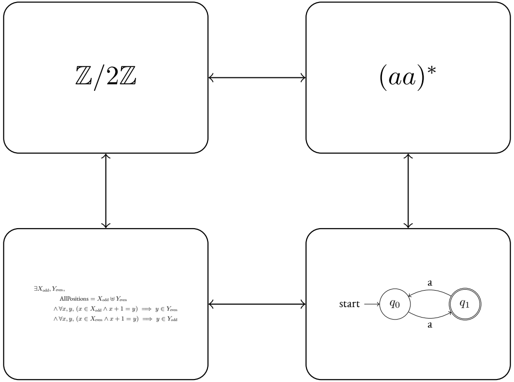
Calcul symbolique

Systèmes de transition bien structurés (WSTS)

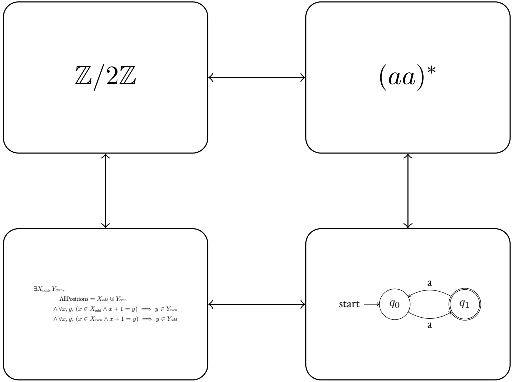
$(\mathbb{N}^k, \to, \leq) \rightsquigarrow$ réseau de Pétri



Langages $L \colon \Sigma^* \to \mathsf{Bool}$



Langages $L\colon \Sigma^* \to \mathsf{Bool}$



Langages $L\colon \Sigma^* \to \mathsf{Bool}$

Quantitatif

$f\colon \Sigma^* \to \mathbb{N}$

Automates pondérés

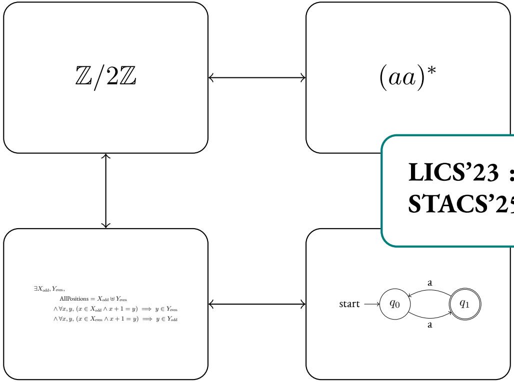
Qualitatif

$f\colon \Sigma^* \to \Gamma^*$

Fonctions (poly)régulières

Décidabilité

Équivalence ? FO-définissabilité ? etc.



Langages $L\colon \Sigma^* \to \mathsf{Bool}$

Quantitatif

$f\colon \Sigma^* \to \mathbb{N}$

Automates pondérés

Qualitatif

$f\colon \Sigma^* \to \Gamma^*$

Fonctions (poly)régulières

LICS'23 : fonctions $\mathbb{Z}$-polyrégulières
STACS'25 : fonctions $\mathbb{N}$-polyrégulières

Décidabilité

Équivalence ? FO-définissabilité ? etc.

Graphes

Graphes

φ = « contient un chemin **induit** de longueur 2 »

Graphes

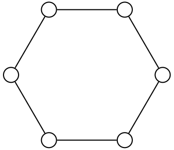
φ = « contient un chemin **induit** de longueur 2 »

$$\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$$

Graphes

φ = « contient un chemin **induit** de longueur 2 »

$$\exists x, y, z, E(x,y) \land E(y,z) \land \neg E(x,z)$$

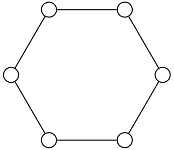


$\models \varphi$? oui

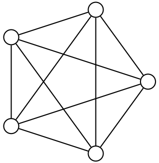
Graphes

φ = « contient un chemin **induit** de longueur 2 »

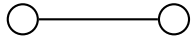
$$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$$



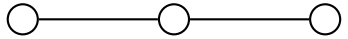
$\models \varphi$? oui



$\models \varphi$? non



$\models \varphi$? non

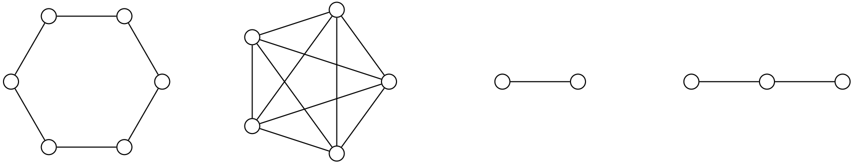


$\models \varphi$? oui

Graphes

φ = « contient un chemin **induit** de longueur 2 »

$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$



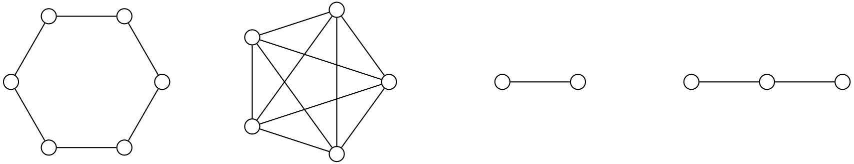
$\models \varphi$? oui $\models \varphi$? non $\models \varphi$? non $\models \varphi$? oui

$$\begin{array}{c} G \subseteq_i H \\ \wedge \\ G \models \varphi \end{array} \implies H \models \varphi$$

Graphes

φ = « contient un chemin **induit** de longueur 2 »

$$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$$



$\models \varphi$? oui | $\models \varphi$? non | $\models \varphi$? non | $\models \varphi$? oui

$$\wedge \begin{array}{l} G \subseteq_i H \\ G \models \varphi \end{array} \implies H \models \varphi$$

$\boxed{\varphi \textit{ préservée par extensions}}$

Théorème Łoś-Tarski

$$\varphi \in \mathsf{FO}$$

φ préservée par extensions

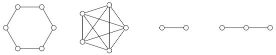
Facile $\Uparrow$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle

Graphes

φ = « contient un chemin **induit** de longueur 2 »

$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$



$$\left.\begin{array}{l} G \subseteq_i H \\ G \models \varphi \end{array}\right\} \wedge \implies H \models \varphi$$

$\boxed{\varphi \textit{ préservée par extensions}}$

Théorème Łoś-Tarski

$\varphi \in \mathsf{FO}$

φ préservée par extensions

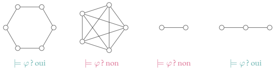
Facile $\Uparrow$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle

Graphes

$\varphi =$ « contient un chemin **induit** de longueur 2 »

$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$



$$\begin{matrix} G \subseteq_i H \\ \wedge \\ G \models \varphi \end{matrix} \implies H \models \varphi$$

$\boxed{\varphi \text{ } \textit{préservée par extensions}}$

Bases de données

Structures relationnelles $\rightsquigarrow$ bases de données

Bases incomplètes $\rightsquigarrow$ *préservation par homomorphisme*

Théorie des modèles finis

Statut des théorèmes dans le cas fini ?

Décompositions structurelles

Théorème Łoś-Tarski

$\varphi \in \mathsf{FO}$

φ préservée par extensions

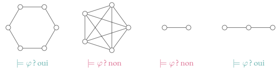
Facile $\Uparrow$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle

Graphes

φ = « contient un chemin **induit** de longueur 2 »

$\exists x, y, z, E(x,y) \wedge E(y,z) \wedge \neg E(x,z)$



$$\begin{array}{c} G \subseteq_i H \\ \wedge \\ G \models \varphi \end{array} \implies H \models \varphi$$

$\boxed{\varphi \textit{ préservée par extensions}}$

Bases de données

Structures relationnelles $\rightsquigarrow$ bases de données

Bases incomplètes $\rightsquigarrow$ *préservation par homomorphisme*

CSL'21 : topologie de la préservation
<u>LICS'22 :</u> localité et préservation

Théorie des modèles finis

Statut des théorèmes dans le cas fini ?

Décompositions structurelles

Relativisation à une classe $\mathcal{C}$?

$\varphi \in \mathsf{FO}$

Facile $\Uparrow$ φ préservée par extensions **sur** $\mathcal{C}$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle **sur** $\mathcal{C}$

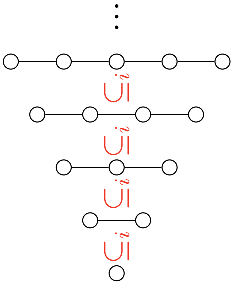
Cycles $\subseteq$ Cycles et Chemins

Chemins $\subseteq$ Cycles et Chemins

Cycles et Chemins $\subseteq$ Degré$_{\leq 2}$ $\subseteq$ Graphes Finis

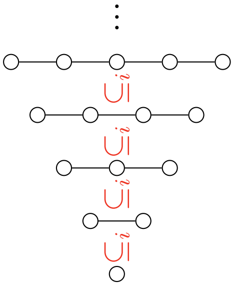
[Tait'59]

Relativisation à une classe $\mathcal{C}$?



Chemins

Relativisation à une classe $\mathcal{C}$?



Chemins

φ préservée par extensions

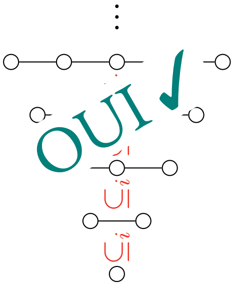
$$\Downarrow$$

$\varphi \equiv$ « Au moins x éléments »

OU

$\varphi \equiv \bot$

Relativisation à une classe $\mathcal{C}$?



Chemins

φ préservée par extensions

$$\Downarrow$$

$\varphi \equiv$ « Au moins x éléments »

OU

$\varphi \equiv \bot$

Relativisation à une classe $\mathcal{C}$?

$\varphi \in \mathsf{FO}$

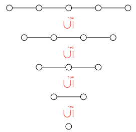
φ préservée par extensions **sur** $\mathcal{C}$

Facile $\Uparrow$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle **sur** $\mathcal{C}$

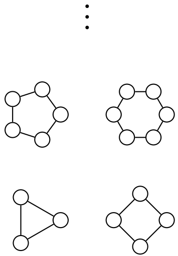
Cycles $\subseteq$ Cycles et Chemins $\subseteq$ Degré$_{\leq 2}$ $\subseteq$ Graphes Finis

[Tait'59]



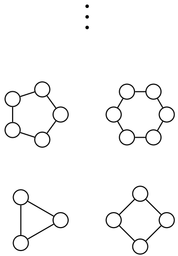
Beau Préordre $\subseteq$ Cycles et Chemins

Relativisation à une classe $\mathcal{C}$?



Cycles

Relativisation à une classe $\mathcal{C}$?



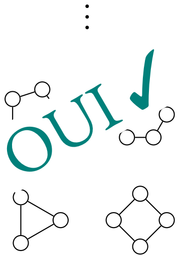
Cycles

Toute formule est préservée par extensions !

mais…

Toute formule est équivalente à une formule existentielle !

Relativisation à une classe $\mathcal{C}$?



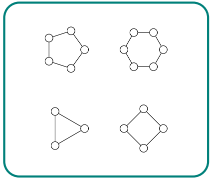
Cycles

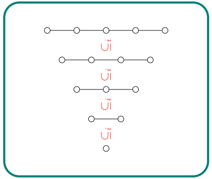
Toute formule est préservée par extensions !

mais…

Toute formule est équivalente à une formule existentielle !

Logique





Beau Préordre

Cycles et Chemins $\subseteq$ Degré$_{\leq 2}$ $\subseteq$ Graphes Finis

[Tait'59]

Relativisation à une classe $\mathcal{C}$?

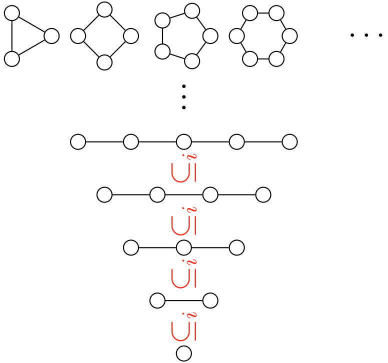
$\varphi \in \mathsf{FO}$

φ préservée par extensions **sur** $\mathcal{C}$

Facile $\Uparrow$ $\Downarrow$ Difficile

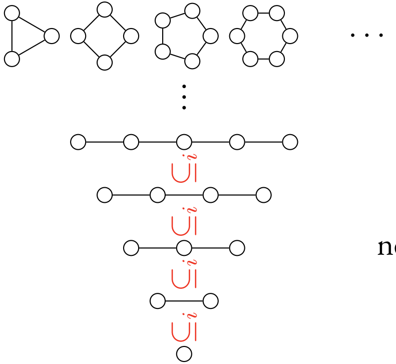
φ équivalente à ψ existentielle **sur** $\mathcal{C}$

Relativisation à une classe $\mathcal{C}$?



Cycles ∪ Chemins

Relativisation à une classe $\mathcal{C}$?



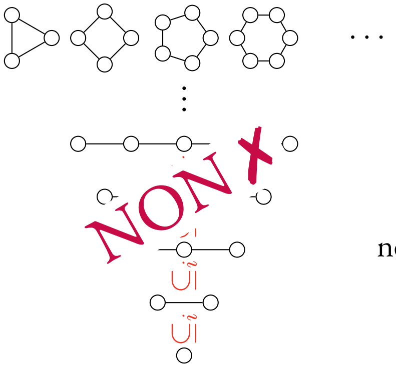
Cycles $\cup$ Chemins

« $\forall x.\text{degré}(x) = 2$ »
est préservée par extensions

et

ne peut pas être écrite comme une formule existentielle !

Relativisation à une classe $\mathcal{C}$?



Cycles ∪ Chemins

« $\forall x.\text{degré}(x) = 2$ »
est préservée par extensions

et

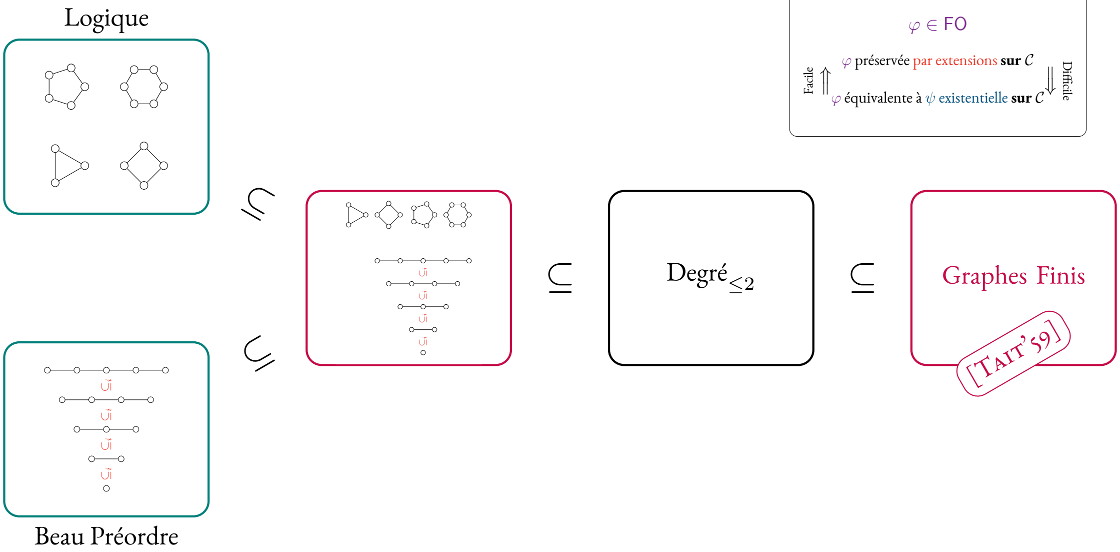
ne peut pas être écrite comme une formule existentielle !

Relativisation à une classe $\mathcal{C}$?

$\varphi \in \mathsf{FO}$

Facile $\Uparrow$ φ préservée par extensions **sur** $\mathcal{C}$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle **sur** $\mathcal{C}$

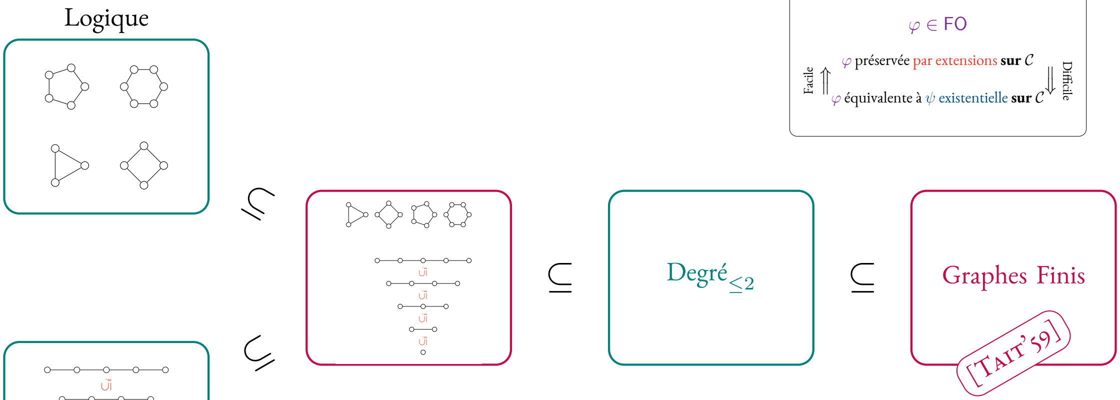


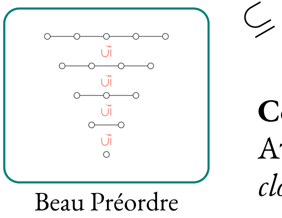
Relativisation à une classe $\mathcal{C}$?

$\varphi \in \mathsf{FO}$

Facile $\Uparrow$ φ préservée par extensions **sur** $\mathcal{C}$ $\Downarrow$ Difficile

φ équivalente à ψ existentielle **sur** $\mathcal{C}$





Condition <u>suffisante</u>

Atserias, Dawar et Grohe (2008) :

clos par $\subseteq_i$, *clos par* $\uplus$, *<u>degré borné</u>*

Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

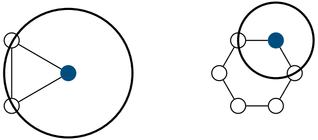
1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

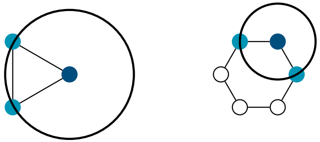
Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$



Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

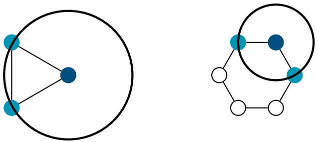
1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$



Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

$$\mathrm{Loc}_{1,1}(\text{Cycles}) = \{C_3, P_3\}$$

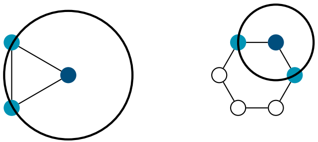


Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

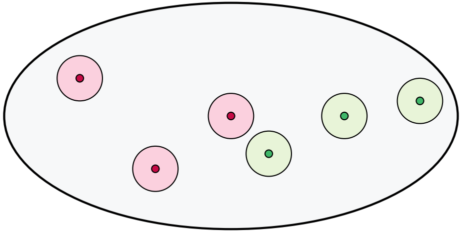
$\Uparrow$

$\mathrm{Loc}_{1,1}(\mathrm{Cycles}) = \{C_3, P_3\}$



Ré-écriture des formules

$\varphi \quad \rightsquigarrow \quad \bigvee \bigwedge (\neg)\, \exists^{\geq k}_{r} x . \psi_{|\mathcal{N}(x,r)}(x)$ Localité

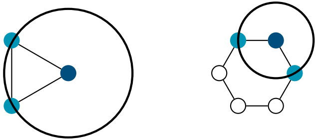


Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

$\Uparrow$

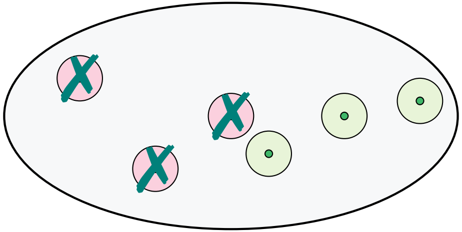
$\mathrm{Loc}_{1,1}(\text{Cycles}) = \{C_3, P_3\}$



Ré-écriture des formules

$$\varphi \rightsquigarrow \bigvee \bigwedge (\neg)\, \exists_r^{\geq k} x. \psi_{|\mathcal{N}(x,r)}(x) \qquad \text{Localité}$$

$$\rightsquigarrow \bigvee \bigwedge \exists_r^{\geq k} x. \psi_{|\mathcal{N}(x,r)}(x) \qquad (\subseteq_i)$$

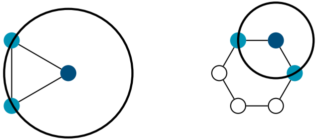


Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

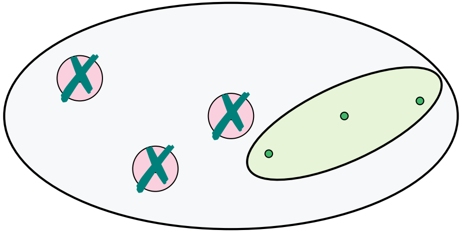
$\Uparrow$

$\mathrm{Loc}_{1,1}(\mathrm{Cycles}) = \{C_3, P_3\}$



Ré-écriture des formules

φ	$\rightsquigarrow$	$\bigvee \bigwedge (\neg)\, \exists_r^{\geq k} x. \psi_{\lvert \mathcal{N}(x,r)}(x)$	Localité
	$\rightsquigarrow$	$\bigvee \bigwedge \exists_r^{\geq k} x. \psi_{\lvert \mathcal{N}(x,r)}(x)$	$(\subseteq_i)$
	$\rightsquigarrow$	$\exists \vec{x}.\, \theta_{\lvert \mathcal{N}(\vec{x},r)}(\vec{x})$	Voisinages

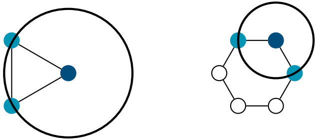


Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

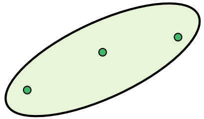
$\Uparrow$

$\mathrm{Loc}_{1,1}(\text{Cycles}) = \{C_3, P_3\}$



Ré-écriture des formules

φ	$\rightsquigarrow$	$\bigvee\bigwedge (\neg)\, \exists^{\geq k}_{r} x. \psi_{\mid\mathcal{N}(x,r)}(x)$	Localité
	$\rightsquigarrow$	$\bigvee\bigwedge \exists^{\geq k}_{r} x. \psi_{\mid\mathcal{N}(x,r)}(x)$	$(\subseteq_i)$
	$\rightsquigarrow$	$\exists \vec{x}.\, \theta_{\mid\mathcal{N}(\vec{x},r)}(\vec{x})$	Voisinages
	$\rightsquigarrow$	$\exists \vec{x}.\, \exists \vec{y}.\, \gamma(\vec{x}, \vec{y})$	$\mathrm{Loc}_{r,\mid\vec{x}\mid}(\mathcal{C})$

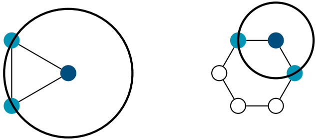


Théorème [Lopez, LICS'22]
Soit $\mathcal{C}$ *close par* $\subseteq_i$ et $\uplus$. Les propriétés suivantes sont équivalentes :

1. Łoś-Tarski relativise à $\mathcal{C}$
2. Łoś-Tarski relativise à $\mathrm{Loc}_{r,k}(\mathcal{C})$, pour tout $r, k \in \mathbb{N}$

$\Uparrow$

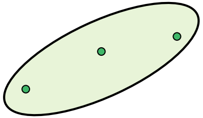
$$\mathrm{Loc}_{1,1}(\mathrm{Cycles}) = \{C_3, P_3\}$$



Nouvelles classes : « *localement X* »
$\rightsquigarrow$ *localement fini* $\iff$ degré borné

Ré-écriture des formules

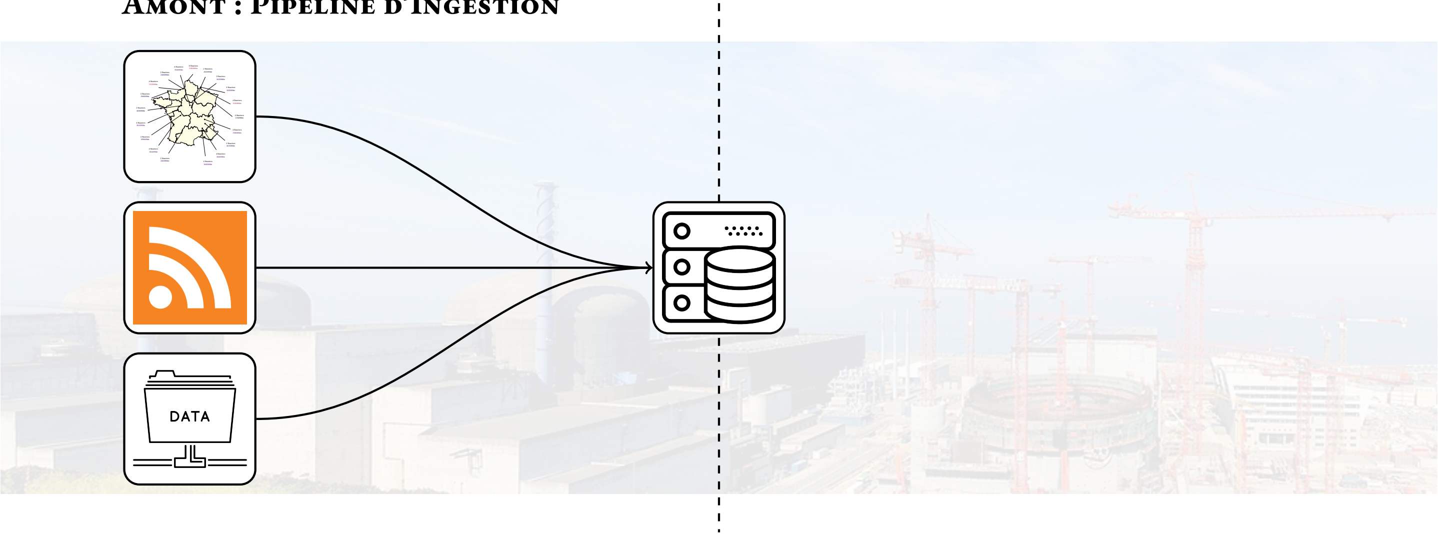
φ	$\rightsquigarrow$	$\bigvee \bigwedge (\neg)\, \exists^{\geq k}_{r} x. \psi_{\mid \mathcal{N}(x,r)}(x)$	Localité
	$\rightsquigarrow$	$\bigvee \bigwedge\ \exists^{\geq k}_{r} x. \psi_{\mid \mathcal{N}(x,r)}(x)$	$(\subseteq_i)$
	$\rightsquigarrow$	$\exists \vec{x}.\, \theta_{\mid \mathcal{N}(\vec{x},r)}(\vec{x})$	Voisinages
	$\rightsquigarrow$	$\exists \vec{x}.\, \exists \vec{y}.\, \gamma(\vec{x}, \vec{y})$	$\mathrm{Loc}_{r,\lvert\vec{x}\rvert}(\mathcal{C})$



Un exemple de programme utilisant des données

Un exemple de programme utilisant des données

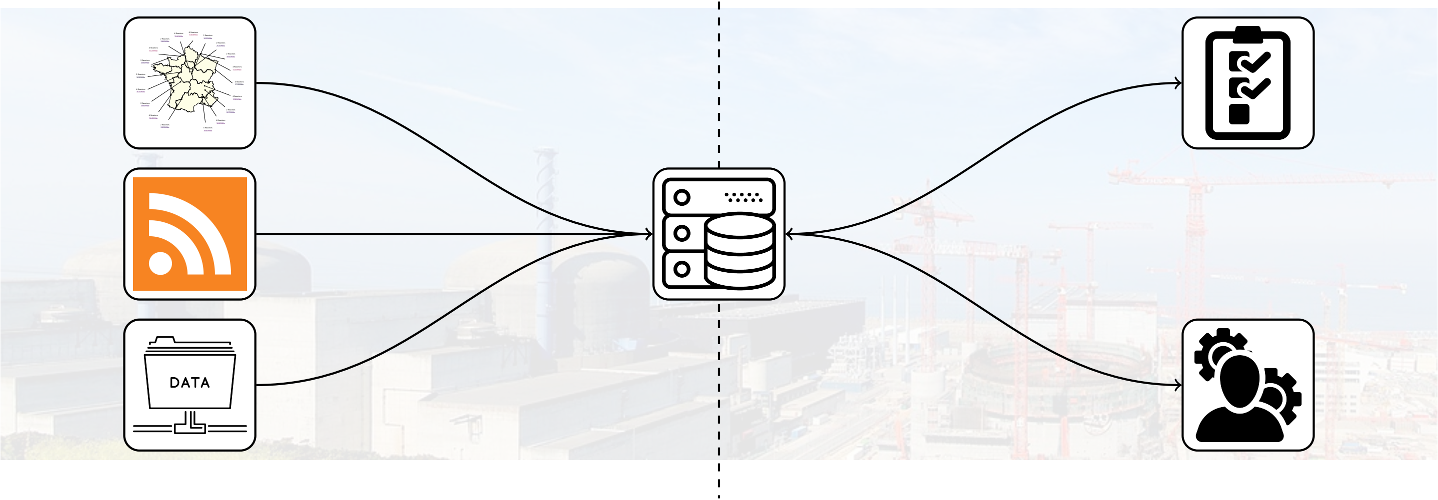
Amont : Pipeline d'Ingestion



Un exemple de programme utilisant des données

Amont : Pipeline d'Ingestion

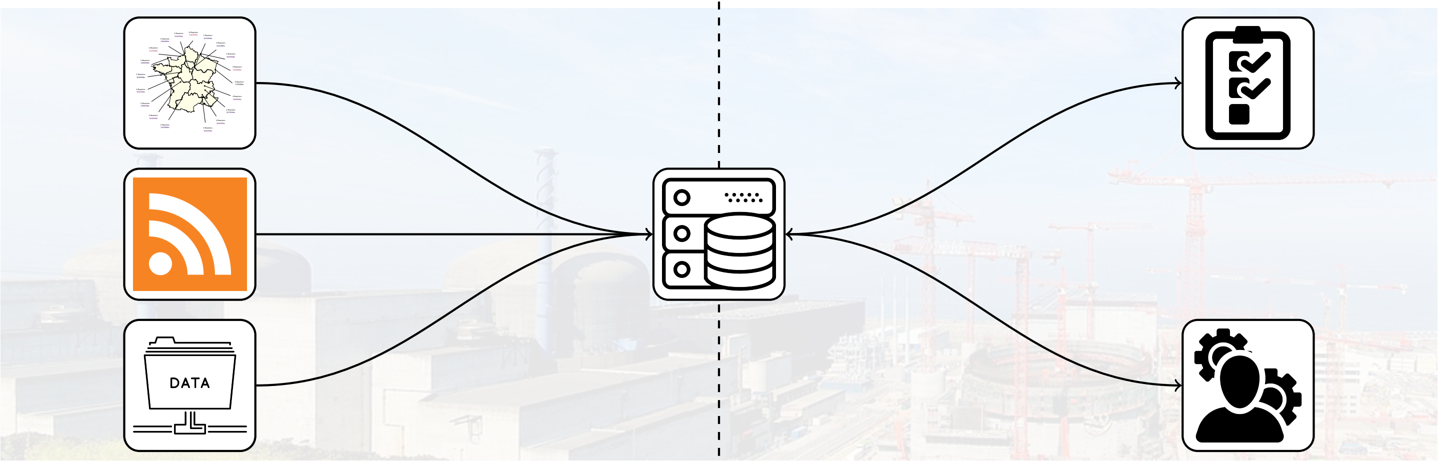
Aval : Interaction via une API



Un exemple de programme utilisant des données

Amont : Pipeline d'Ingestion

Aval : Interaction via une API



Vérification : $\{\varphi\}\ P\ \{\psi\}$? (Hoare)

Vérification : $D_0 \to^* D_{\text{err}}$? (Accessibilité)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\rightarrow$

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$
Système de transition bien structuré

$$\begin{array}{ccc} \mathfrak{A}' & \dashrightarrow & \mathfrak{B}' \\ \cup\!\!\mid_i & & \cup\!\!\mid_i \\ \mathfrak{A} & \longrightarrow & \mathfrak{B} \end{array}$$

Couvrabilité :
$D_0 \to^* D'_i \supseteq D_{\mathrm{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$

Système de transition bien structuré

$$\begin{tikzcd}
\mathfrak{A}' \arrow[r, dashed] & \mathfrak{B}' \\
\mathfrak{A} \arrow[u, phantom, sloped, "\subseteq_i"] \arrow[r] & \mathfrak{B} \arrow[u, phantom, sloped, "\subseteq_i"]
\end{tikzcd}$$

> Couvrabilité :
> $D_0 \to^* D'{}_i\!\supseteq D_{\mathrm{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Dans un premier temps, ignorer $\to$

$\rightsquigarrow$ $(\mathcal{C}, \subseteq_i)$ est un *beau préordre* ?

$\rightsquigarrow$ Ordres totaux $\simeq (\Sigma^*, \leq^*)$!

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$

Système de transition bien structuré

$$\begin{tikzcd} \mathfrak{A}' \arrow[r, dashed] & \mathfrak{B}' \\ \mathfrak{A} \arrow[r] \arrow[u, phantom, sloped, "\subseteq_i"] & \mathfrak{B} \arrow[u, phantom, sloped, "\subseteq_i"] \end{tikzcd}$$

Couvrabilité :
$D_0 \to^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Dans un premier temps, ignorer $\to$

$\rightsquigarrow$ $(\mathcal{C}, \subseteq_i)$ est un *beau préordre* ?

$\rightsquigarrow$ Ordres totaux $\simeq (\Sigma^*, \leq^*)$!

Conjecture [Daligault et al., 2010] :

$(\mathcal{C}, \subseteq_i)$ *beau préordre*

$\Longrightarrow$

$\mathcal{C}$ a *largeur de clique bornée*

(représentation arborescente)

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$

Système de transition bien structuré

$$\begin{array}{ccc} \mathfrak{A}' & \dashrightarrow & \mathfrak{B}' \\ \rotatebox{90}{$\subseteq_i$} & & \rotatebox{90}{$\subseteq_i$} \\ \mathfrak{A} & \longrightarrow & \mathfrak{B} \end{array}$$

Couvrabilité :
$D_0 \to^* D' \supseteq_i D_{\mathrm{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Dans un premier temps, ignorer $\to$

$\rightsquigarrow$ $(\mathcal{C}, \subseteq_i)$ est un *beau préordre* ?

$\rightsquigarrow$ Ordres totaux $\simeq (\Sigma^*, \leq^*)$!

Conjecture [Daligault et al., 2010] :
$(\mathcal{C}, \subseteq_i)$ *beau préordre*
$\Longrightarrow$
$\mathcal{C}$ a *largeur de clique bornée*
(représentation arborescente)

Différentes représentations

$\rightsquigarrow$ $(\Sigma^*, \leq_{\text{facteur}})$ représente les classes $(\mathcal{C}, \subseteq_i)$

$\rightsquigarrow$ représentation arborescente

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$
Système de transition bien structuré

$$\begin{tikzcd}
\mathfrak{A}' \arrow[r, dashed] & \mathfrak{B}' \\
\mathfrak{A} \arrow[u, phantom, sloped, "\subseteq_i"] \arrow[r] & \mathfrak{B} \arrow[u, phantom, sloped, "\subseteq_i"]
\end{tikzcd}$$

Couvrabilité :
$D_0 \to^* D' {}_i\!\supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Dans un premier temps, ignorer $\to$

$\rightsquigarrow$ $(\mathcal{C}, \subseteq_i)$ est un *beau préordre* ?

$\rightsquigarrow$ Ordres totaux $\simeq (\Sigma^*, \leq^*)$!

Conjecture [Daligault et al., 2010] :

$(\mathcal{C}, \subseteq_i)$ *beau préordre*

$\Longrightarrow$

$\mathcal{C}$ a *largeur de clique bornée*

(représentation arborescente)

Différentes représentations

$\rightsquigarrow$ $(\Sigma^*, \leq_{\text{facteur}})$ représente les classes $(\mathcal{C}, \subseteq_i)$

$\rightsquigarrow$ représentation arborescente

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$

Système de transition bien structuré

$$\begin{tikzcd}
\mathfrak{A}' \arrow[r, dashed] & \mathfrak{B}' \\
\mathfrak{A} \arrow[u, phantom, "\rotatebox{90}{$\subseteq_i$}"] \arrow[r] & \mathfrak{B} \arrow[u, phantom, "\rotatebox{90}{$\subseteq_i$}"]
\end{tikzcd}$$

Couvrabilité :
$D_0 \to^* D' {}_i\!\supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Et plus tard...

$\rightsquigarrow$ *homomorphismes*

$\rightsquigarrow$ relations $\to$

Cadre : classes de structures relationnelles finies $\mathcal{C}$ avec une relation de transition $\to$

Problème à long terme

Reconnaître les systèmes $(\mathcal{C}, \to, \subseteq_i)$ qui sont bien structurés

Dans un premier temps, ignorer $\to$

$\rightsquigarrow$ $(\mathcal{C}, \subseteq_i)$ est un *beau préordre* ?

$\rightsquigarrow$ Ordres totaux $\simeq (\Sigma^*, \leq^*)$!

Conjecture [Daligault et al., 2010] :

$(\mathcal{C}, \subseteq_i)$ *beau préordre*

$\Longrightarrow$

$\mathcal{C}$ a *largeur de clique bornée*

(représentation arborescente)

Différentes représentations

$\rightsquigarrow$ $(\Sigma^*, \leq_{\text{facteur}})$ représente les classes $(\mathcal{C}, \subseteq_i)$

$\rightsquigarrow$ représentation arborescente

Parti pris : $(\mathcal{C}, \to, \subseteq_i)$

Système de transition bien structuré

$$\begin{tikzcd}
\mathfrak{A}' \arrow[r, dashed] & \mathfrak{B}' \\
\mathfrak{A} \arrow[u, phantom, sloped, "\subseteq_i"] \arrow[r] & \mathfrak{B} \arrow[u, phantom, sloped, "\subseteq_i"]
\end{tikzcd}$$

Couvrabilité :
$D_0 \to^* D' {}_i\!\supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

Et plus tard...

$\rightsquigarrow$ *homomorphismes*

$\rightsquigarrow$ relations $\to$

Une ouverture sur les bases de données

Algorithmes de type *Chase*

D_0 base de donnée, Δ contraintes, q requête

« toute complétion de D_0 vérifiant Δ satisfait q ?»

Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

```python
def getBetween(l, i, j):
    """ Get elements between i and j """
    for (k, c) in enumerate(l):
        if i <= k and k <= j: ①
            yield c ②

def containsAB(w):
    """ Contains "ab" as a subsequence """
    seen_a = False ③
    for (x, c) in enumerate(w):
        if c == "a": ④
                seen_a = True ⑤
        elif seen_a and c == "b":
            return True
    return False

def subwordsWithAB(word):
    """ Get subwords that contain "ab" """
    for (i,c) in enumerate(word): ⑥
        for (j,d) in reversed(enumerate(word)): ⑦
            s = getBetween(word, i, j) ⑧
            if containsAB(s):
                yield s
```

Fig. 1. A small Python program that outputs all subwords of a given word containing `ab` as a scattered subword

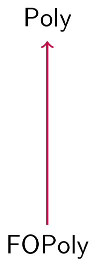
Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

```python
def getBetween(l, i, j):
    """ Get elements between i and j """
    for (k, c) in enumerate(l):
        if i <= k and k < j:
            yield c

def containsAB(w):
    """ Contains "ab" as a subsequence """
    seen_a = False
    for (x, c) in enumerate(w):
        if c == "a":
                seen_a = True
        elif seen_a and c == "b":
            return True
    return False

def subwordsWithAB(word):
    """ Get subwords that contain "ab" """
    for (i,c) in enumerate(word):
        for (j,d) in reversed(enumerate(word)):
            s = getBetween(word, i, j)
            if containsAB(s):
                yield s
```

Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword

Problèmes théoriques à long terme

Décider l'équivalence de fonctions
Décider la FO-définissabilité (`x := True`)



Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

Problèmes théoriques à long terme

Décider l'équivalence de fonctions
Décider la FO-définissabilité (`x := True`)

Dans un premier temps

- ⇝ sorties unaires $\{a\}^* \simeq \mathbb{N}$
- ⇝ entrées unaires / commutatives ?
- ⇝ lien quantitatif / qualitatif ?

$$\begin{tikzcd} \mathbb{N}\mathrm{Poly} \arrow[r, Rightarrow] & \mathrm{Poly} \\ \mathbb{N}\mathrm{FOPoly} \arrow[r, Rightarrow] \arrow[u, dashed] & \mathrm{FOPoly} \arrow[u] \end{tikzcd}$$

```
def subwordsWithAB(word):
    """ Get subwords that contain "ab" """
    for (i,c) in enumerate(word):
        for (j,d) in reversed(enumerate(word)):
            s = getBetween(word, i, j)
            if containsAB(s):
                yield s
```

Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword

Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

```
def getBetween(l, i, j):
def containsAB(w):
    """ Contains "ab" as a subsequence """
        seen_a = True
        return True
def subwordsWithAB(word):
    """ Get subwords that contain "ab" """
    for (i,c) in enumerate(word):
        for (j,d) in reversed(enumerate(word)):
            s = getBetween(word, i, j)
            if containsAB(s):
                yield s
```

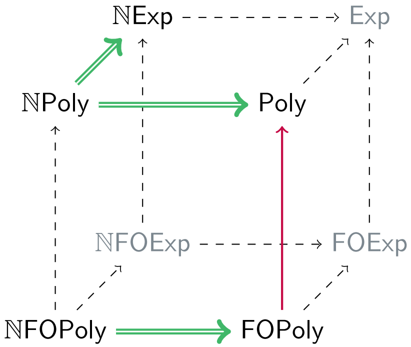
Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword

Problèmes théoriques à long terme

Décider l'équivalence de fonctions
Décider la FO-définissabilité (`x := True`)

Dans un premier temps

⇝ sorties unaires $\{a\}^* \simeq \mathbb{N}$
⇝ entrées unaires / commutatives ?
⇝ lien quantitatif / qualitatif ?



Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

Problèmes théoriques à long terme

Décider l'équivalence de fonctions
Décider la FO-définissabilité (`x := True`)

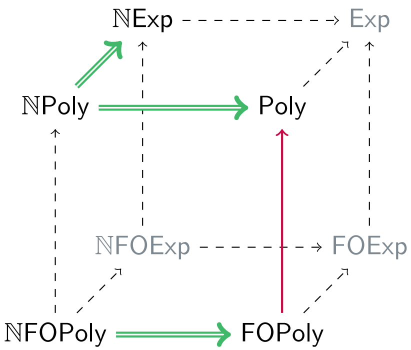
Dans un premier temps

- ⇝ sorties unaires $\{a\}^* \simeq \mathbb{N}$
- ⇝ entrées unaires / commutatives ?
- ⇝ lien quantitatif / qualitatif ?

Implémentation(s)

- ⇝ Vérification de triplets de Hoare ?
- ⇝ Optimisations de programmes ?

Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword



Objet : programmes Python « simples »
Point de départ : *fonctions polyrégulières*

Problèmes théoriques à long terme

Décider l'équivalence de fonctions
Décider la FO-définissabilité (`x := True`)

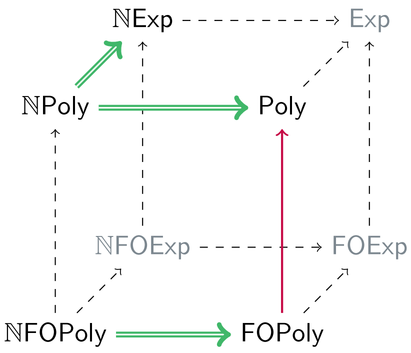
Dans un premier temps

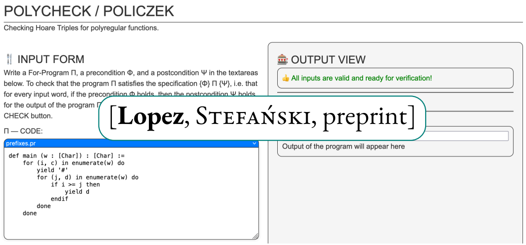
$\rightsquigarrow$ sorties unaires $\{a\}^* \simeq \mathbb{N}$
$\rightsquigarrow$ entrées unaires / commutatives ?
$\rightsquigarrow$ lien quantitatif / qualitatif ?

Implémentation(s)

$\rightsquigarrow$ Vérification de triplets de Hoare ?
$\rightsquigarrow$ Optimisations de programmes ?

Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword





[**Lopez**, Stefański, preprint]

Intégration du projet de recherche

Systèmes de transitions
$(\mathcal{C}, \rightarrow, \subseteq_i)$ *système de transition bien strucutré*

WQO

Programmes à états finis
$P : \Sigma^* \rightarrow \Gamma^*$ *polyrégulière*

MSO

LIS (Marseille)

Nathan Lhote, Pierre-Alain Reynier
Benjamin Monmege, Séverine Fratani, Pierre Ohlmann
Lê Thành Dũng (Tito) Nguyễn

LIRMM (Montpellier)

Christophe Paul, Dimitrios Thilikos
David Carral, Nofar Carmeli

LIGM (Marne-la-Vallée)

Arnaud Carayol, Léo Exibard
Victor Marsault, Nadime Francis, Claire David
Vincent Jugé, Marie-Pierre Béal

Expertise :
Logique, Beaux préordres, Automates

Co-organisation Autobóz 2024

Comité de programme de **CSL'26**

Co-encadrement de 2 stagiaires

2 Prix de Thèse
Ackermann Award &
E. W. Beth Dissertation Prize

Conférences	8 (dont 4 en seul auteur)
Journaux	2
Soumissions	2