

CONCOURS MCF SECTION 27

OFFRE 25I8I6

LOGIQUE MONADIQUE DU SECOND ORDRE ET BEAUX PRÉORDRES
POUR LES
MÉTHODES FORMELLES



ZYGMUNT
ZALESKI
STICHTING



Aliaume Lopez
Université de Varsovie

à Bordeaux
le 2025-05-06



<https://www.irif.fr/~alopez/>

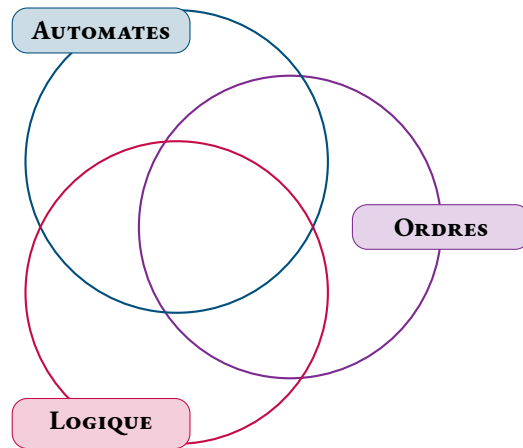
PARCOURS ACADÉMIQUE

ENS Paris-Saclay	(2015 – 2019)	LMF & IRIF	(2019 – 2023)	Varsovie	(2023 – 2025)
Agrégation de Mathématiques		Thèse sous la direction de		Postdoctorat	
<i>Option Info. Classé 5e</i>		GOUBAULT-LARRECQ & SCHMITZ		MIKOŁAJ BOJAŃCZYK	
Stages de L3 et M1		2 Prix de Thèse		Co-organisation Autobóz 2024	
<i>Birmingham et Ljubljana</i>		<i>Ackermann Award &</i>		Membre du comité de programme	
Stage M2 au LSV		<i>E. W. Beth Dissertation Prize</i>		de CSL'26	
GOUBAULT-LARRECQ & SCHMITZ		Césure (1 an)		Co-encadrement de 2 stagiaires	
		<i>Autorité de Sécurité Nucléaire</i>			

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

PUBLICATIONS

Conférences	9 (dont 4 en seul auteur)
Journaux	2
Soumissions	1



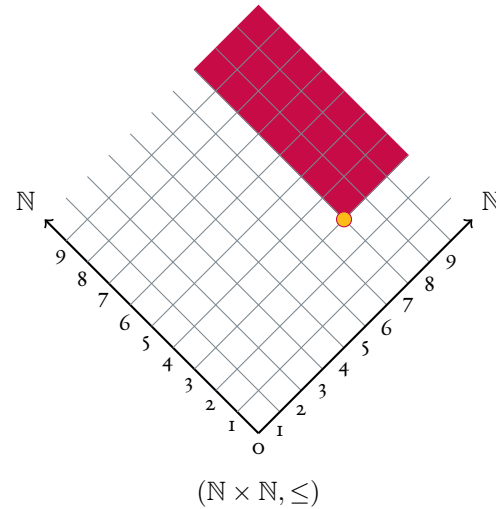
LOGIQUE



ORDRES

AUTOMATES

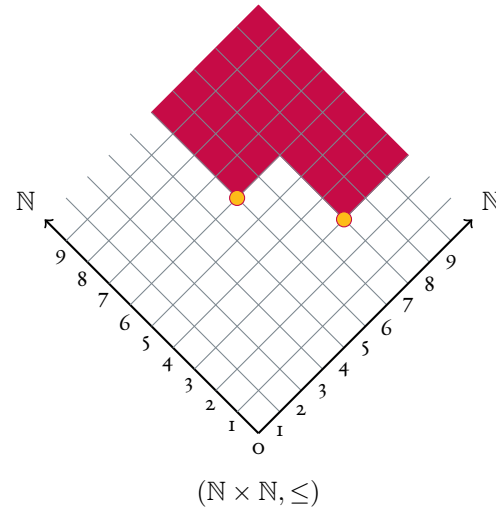
LOGIQUE



ORDRES

AUTOMATES

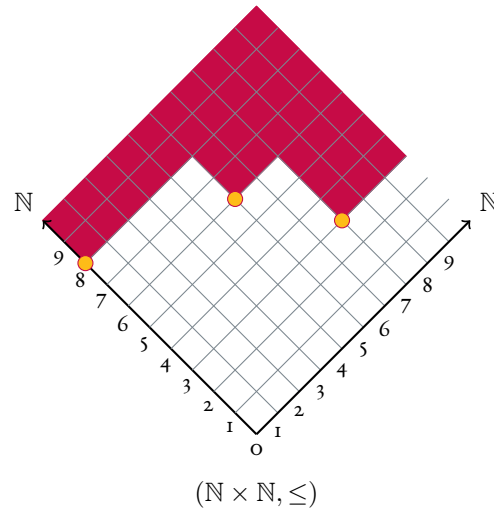
LOGIQUE



ORDRES

AUTOMATES

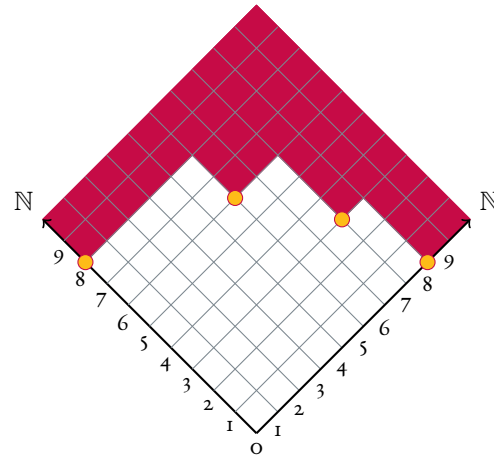
LOGIQUE



ORDRES

AUTOMATES

LOGIQUE

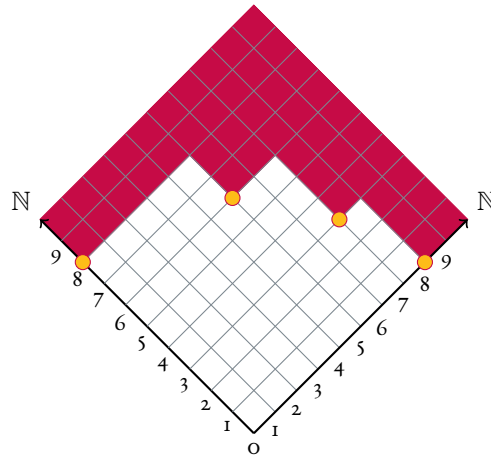
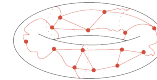


$(\mathbb{N} \times \mathbb{N}, \leq)$

ORDRES

AUTOMATES

LOGIQUE



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

GRAPHES

ROBERTSON & SEYMOUR
Mineurs de graphes

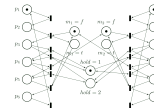
ALGÈBRE

HILBERT / GRÖBNER
Calcul symbolique

VÉRIFICATION

Systèmes de transition bien structurés (WSTS)

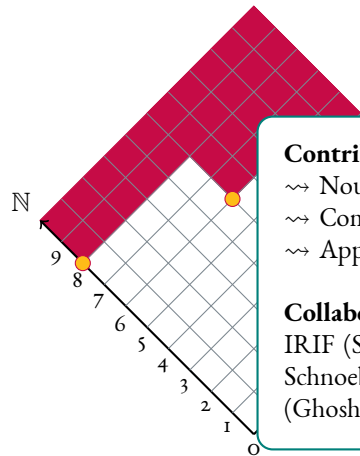
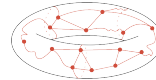
$(\mathbb{N}^k, \rightarrow, \leq) \rightsquigarrow$ réseau de Pétri



ORDRES

AUTOMATES

LOGIQUE



Contributions : 3 axes

- ~> Nouveaux beaux préordres
- ~> Complexité des préordres
- ~> Approche topologique

[FoSSaCS'23]

[MSCS]

[Coll. Mat.]

Collaborations :

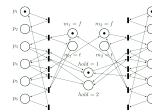
IRIF (Schmitz), LMF (Goubault-Larrecq, Vialard, Schnoebelen), LIS (Lhote), MPI-SWS (Schütze), **LaBRI** (Ghosh).

ALGÈBRE

HILBERT / GRÖBNER

Calcul symbolique

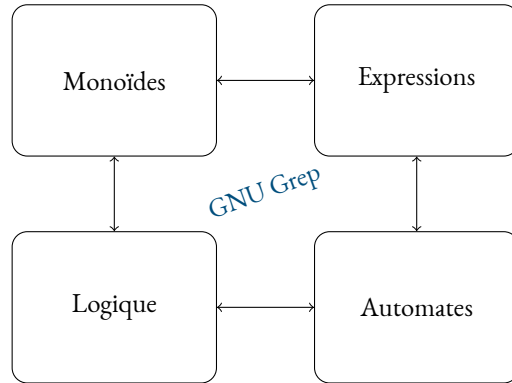
Structurés (WSTS)

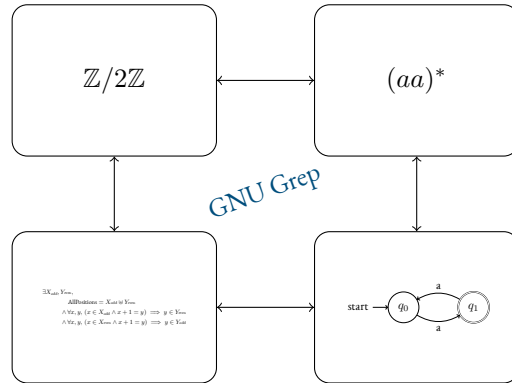
 $(\mathbb{N} \times \mathbb{N}, \leq)$


ORDRES

AUTOMATES

LOGIQUE


Langages $L: \Sigma^* \rightarrow \text{Bool}$

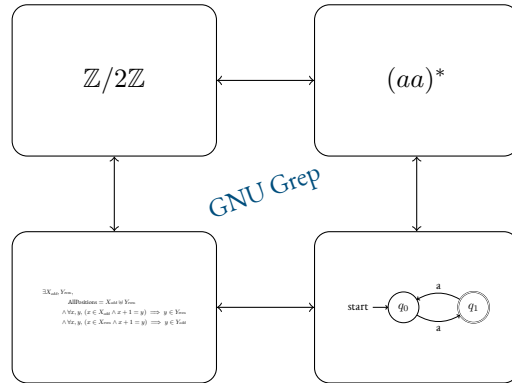


Langages $L: \Sigma^* \rightarrow \text{Bool}$

ORDRES

AUTOMATES

LOGIQUE



Langages $L: \Sigma^* \rightarrow \text{Bool}$

$$w \mapsto 2^{|w|} \times |w|_a$$

QUANTITATIF

$$f: \Sigma^* \rightarrow \mathbb{N}$$

Automates pondérés

$$w \mapsto w\#w$$

QUALITATIF

$$f: \Sigma^* \rightarrow \Gamma^*$$

Fonctions (poly)régulières

DÉCIDABILITÉ

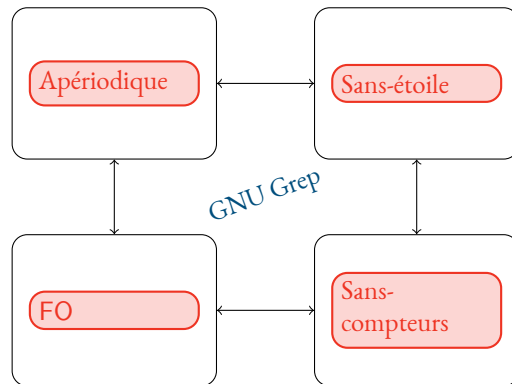
Équivalence? FO-définissabilité? etc.

$$f = g? f = 0? \exists g \in \text{FO}, f = g? \dots$$

ORDRES

AUTOMATES

LOGIQUE


Langages $L: \Sigma^* \rightarrow \text{Bool}$

$$w \mapsto 2^{|w|} \times |w|_a$$

QUANTITATIF

$$f: \Sigma^* \rightarrow \mathbb{N}$$

Automates pondérés

$$w \mapsto w\#w$$

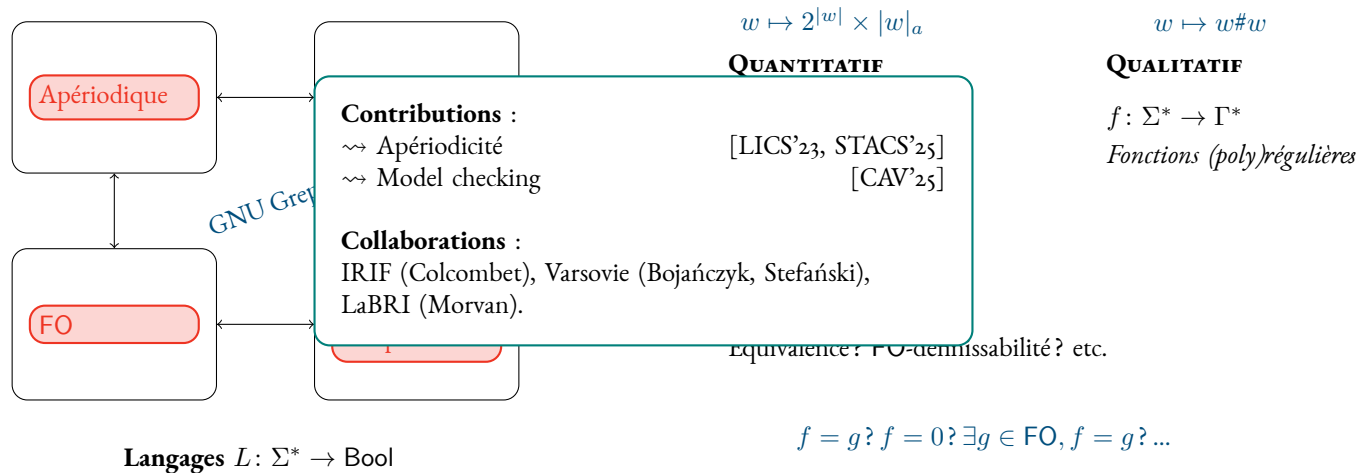
QUALITATIF

$$f: \Sigma^* \rightarrow \Gamma^*$$

Fonctions (poly)régulières
DÉCIDABILITÉ

Équivalence? FO-définissabilité? etc.

$$f = g? f = 0? \exists g \in \text{FO}, f = g? \dots$$



ORDRES

AUTOMATES

LOGIQUE



ORDRES AUTOMATES LOGIQUE

GRAPHES



ORDRES

AUTOMATES

LOGIQUE

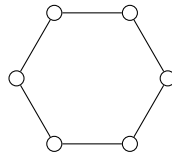
GRAPHES

 $\varphi = \ll \text{contient un chemin } \mathbf{induit} \text{ de longueur } 2 \gg$

GRAPHS

$$\varphi = \text{« contient un chemin \textbf{induit} de longueur 2 »}$$

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

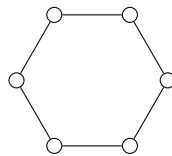
GRAPHES

$\models \varphi$? oui

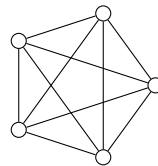
$\varphi = \ll \text{contient un chemin } \mathbf{induit} \text{ de longueur } 2 \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$

GRAPHES

$\varphi = \ll \text{contient un chemin } \mathbf{induit} \text{ de longueur } 2 \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



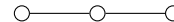
$\models \varphi ?$ oui



$\models \varphi ?$ non



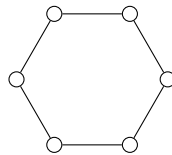
$\models \varphi ?$ non



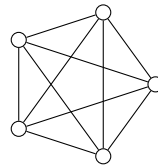
$\models \varphi ?$ oui

GRAPHES

$\varphi = \ll \text{contient un chemin } \mathbf{induit} \text{ de longueur } 2 \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



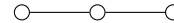
$\models \varphi ? \text{ oui}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ non}$

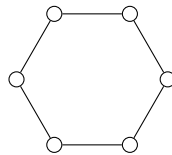


$\models \varphi ? \text{ oui}$

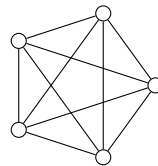
$$\begin{array}{l} G \subseteq_i H \\ \wedge \\ G \models \varphi \end{array} \implies H \models \varphi$$

GRAPHES

$\varphi = \ll \text{contient un chemin } \mathbf{induit} \text{ de longueur } 2 \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



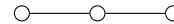
$\models \varphi ? \text{ oui}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ oui}$

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

φ préservée par *extensions*

Théorème Łoś-Tarski

$$\varphi \in \text{FO}$$

Facile $\Uparrow$ φ préservée par extensions $\Downarrow$ Difficile
 φ équivalente à ψ existentielle

GRAPHES

$\varphi = \ll \text{contient un chemin induit de longueur 2} \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



$\models \varphi ? \text{ oui}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ oui}$

$G \subseteq H$
 $\wedge G \models \varphi \implies H \models \varphi$
 φ préservée par extensions

Théorème Łoś-Tarski

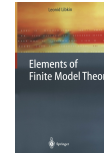
$$\varphi \in \text{FO}$$

Facile $\Uparrow$ φ préservée par extensions $\Downarrow$ Difficile
 φ équivalente à ψ existentielle

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 structurales



GRAPHES

$\varphi = \llcorner \text{contient un chemin induit de longueur 2} \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



$\models \varphi ? \text{ oui}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ oui}$

$G \subseteq H$
 $\wedge G \models \varphi \implies H \models \varphi$
 φ préservée par extensions

Théorème Łoś-Tarski

$\varphi \in \text{FO}$

Facile $\Uparrow$

φ préservée par extensions

$\equiv$ Difficile

φ équivalente à ψ ex

BASES DE DONNÉES

Contributions : méthodes *génériques*

$\rightsquigarrow$ Composition (topologie)

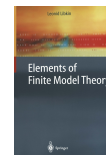
[CSL'21]

$\rightsquigarrow$ Localité

[LICS'22]

Collaborations :

IRIF (Schmitz), LMF (Goubault-Larrecq).



Bases de données

raisonnement *par homomorphisme*

GRAPHES

$\varphi = \llcorner \text{contient un chemin induit de longueur 2} \gg$
 $\exists x, y, z, E(x, y) \wedge E(y, z) \wedge \neg E(x, z)$



$\models \varphi ? \text{ oui}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ non}$



$\models \varphi ? \text{ oui}$

$G \subseteq H$
 $\wedge G \models \varphi$

$\implies H \models \varphi$

φ préservée par extensions

statut des théorèmes dans le cas fini?

Décompositions structurales



ENSEIGNEMENTS EFFECTUÉS

TD Transducteurs	24h	M1
TP de Programmation en Python	12h	L1
TP de Programmation en Java	12h	L1
TP d'Introduction aux Systèmes d'Exploitation	24h	L1
TP de Programmation Fonctionnelle	24h	L3
TP/TD de Gestion de Projet	12h	L2
TP de Programmation Réseau	24h	L3
TD de Complexité Avancée	24h	M1
TD de Programmation 1	24h	L3

TOTAL : 180H

Université de Varsovie
2023 – 2024



Université Paris Cité
2022 – 2023



Université Paris Cité
2021 – 2022



École Normale Supérieure Paris-Saclay
2019 – 2020



ENSEIGNEMENTS EFFECTUÉS

TD Transducteurs	24h	M1
TP de Programmation en Python	12h	L1
TP de Programmation en Java	12h	L1
TP d'Introduction aux Systèmes d'Exploitation	24h	L1
TP de Programmation Fonctionnelle	24h	L3
TP/TD de Gestion de Projet	12h	L2
TP de Programmation Réseau	24h	L3
TD de Complexité Avancée	24h	M1
TD de Programmation I	24h	L3

AGRÉGATION MATH. INFO.

DÉVELOPPEUR FULLSTACK (ASN)

PROJETS PERSONNELS

ENCADREMENT (2 MOIS, L2)



ENSEIGNEMENTS EFFECTUÉS

TD Transducteurs	24h	M1
TP de Programmation en Python	12h	L1
TP de Programmation en Java	12h	L1
TP d'Introduction aux Systèmes d'Exploitation	24h	L1
TP de Programmation Fonctionnelle	24h	L3
TP/TD de Gestion de Projet	12h	L2
TP de Programmation Réseau	24h	L3
TD de Complexité Avancée	24h	M1
TD de Programmation 1	24h	L3

- AGRÉGATION MATH. INFO.
- DÉVELOPPEUR FULLSTACK (ASN)
- PROJETS PERSONNELS
- ENCADREMENT (2 MOIS, L2)



BESOINS

Informatique Théorique
Mathématiques Automates
Complexité Algorithmique
Logique

Programmation
Génie Logiciel Impérative
Fonctionnelle Système
Objet Web

Systèmes et Réseaux
Systèmes BDD
Réseaux Cloud

Sécurité
Cryptographie Systèmes
Réseaux

Semestre 9 : Méthodes Formelles

ENSEIGNEMENTS EFFECTUÉS

TD Transducteurs	24h	M1
TP de Programmation en Python	12h	L1
TP de Programmation en Java	12h	L1
TP d'Introduction aux Systèmes d'Exploitation	24h	L1
TP de Programmation Fonctionnelle	24h	L3
TP/TD de Gestion de Projet	12h	L2
TP de Programmation Réseau	24h	L3
TD de Complexité Avancée	24h	M1
TD de Programmation 1	24h	L3

- AGRÉGATION MATH. INFO.
- DÉVELOPPEUR FULLSTACK (ASN)
- PROJETS PERSONNELS
- ENCADREMENT (2 MOIS, L2)



BESOINS

✓

Informatique Théorique

Mathématiques Automates
Complexité Algorithmique
Logique

✓

Programmation

Génie Logiciel Impérative
Fonctionnelle Système
Objet Web

•

Systèmes et Réseaux

Systèmes BDD
Réseaux Cloud

Sécurité

Cryptographie Systèmes
Réseaux

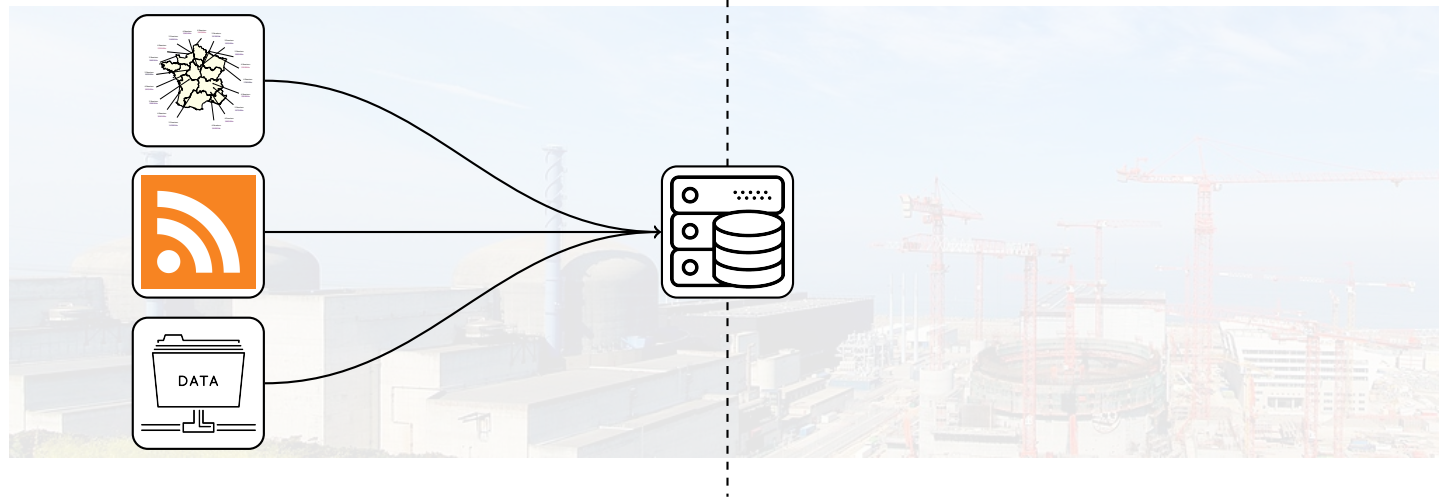
Semestre 9 : Méthodes Formelles

UN EXEMPLE DE PROGRAMME UTILISANT DES DONNÉES



UN EXEMPLE DE PROGRAMME UTILISANT DES DONNÉES

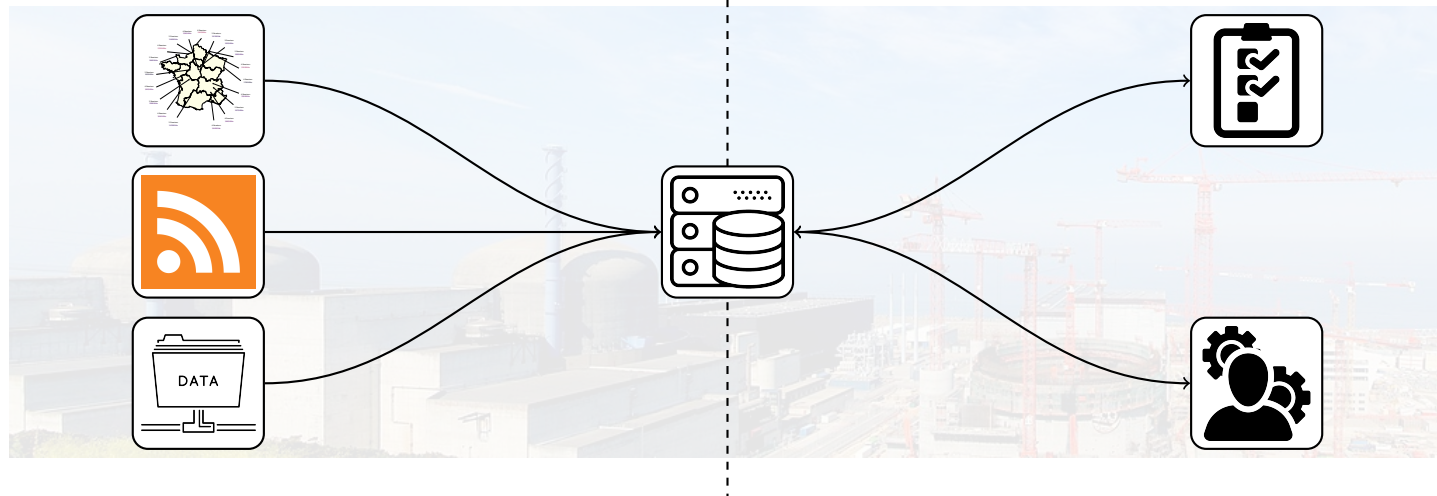
AMONT : PIPELINE D'INGESTION



UN EXEMPLE DE PROGRAMME UTILISANT DES DONNÉES

AMONT : PIPELINE D'INGESTION

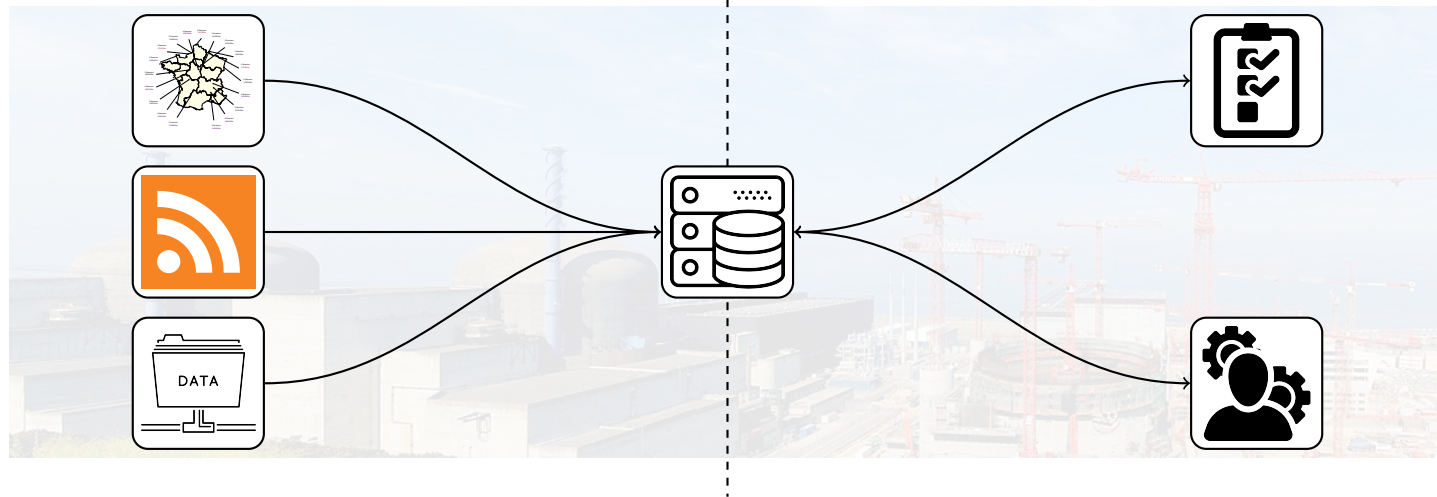
AVAL : INTERACTION VIA UNE API



UN EXEMPLE DE PROGRAMME UTILISANT DES DONNÉES

AMONT : PIPELINE D'INGESTION

AVAL : INTERACTION VIA UNE API



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

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

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

Cadre : classes de structures relationnelles finies

$\mathcal{C}$ avec une relation de transtion $\rightarrow$

SYSTÈMES DE TRANSITION

FONCTIONS POLYRÉGULIÈRES

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

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\tilde{\cup} \tilde{\cup}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

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

PROBLÈME À LONG TERME

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

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\bigcup \mathfrak{A} \quad \bigcup \mathfrak{B}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION

FONCTIONS POLYRÉGULIÈRES

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

PROBLÈME À LONG TERME

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

Dans un premier temps, ignorer $\rightarrow$
 $\rightsquigarrow (\mathcal{C}, \subseteq_i)$ est un *beau préordre*?

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\tilde{\cup} \quad \tilde{\cup}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

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

PROBLÈME À LONG TERME

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

Dans un premier temps, ignorer $\rightarrow$
 $\rightsquigarrow (\mathcal{C}, \subseteq_i)$ est un *beau préordre*?

Conjecture [Daligault et al., 2010] :

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

$\implies$

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

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\tilde{\cup} \quad \tilde{\cup}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

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

PROBLÈME À LONG TERME

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

Dans un premier temps, ignorer $\rightarrow$
 $\rightsquigarrow (\mathcal{C}, \subseteq_i)$ est un *beau préordre*?

Conjecture [Daligault et al., 2010] :

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

$\implies$

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

Différentes représentations

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

$\rightsquigarrow$ Classe de largeur de clique bornée

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\tilde{\cup} \quad \tilde{\cup}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

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

PROBLÈME À LONG TERME

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

Dans un premier temps, ignorer $\rightarrow$
 $\rightsquigarrow (\mathcal{C}, \subseteq_i)$ est un *beau préordre*?

Conjecture [Daligault et al., 2010] :

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

$\implies$

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

Différentes représentations

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

$\rightsquigarrow$ Classe de largeur de clique bornée

Et plus tard?

$\rightsquigarrow$ *homomorphismes*?

$\rightsquigarrow$ relations $\rightarrow$?

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\tilde{\cup} \quad \tilde{\cup}$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

SYSTÈMES DE TRANSITION**FONCTIONS POLYRÉGULIÈRES**

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

PROBLÈME À LONG TERME

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

Dans un premier temps, ignorer $\rightarrow$
 $\rightsquigarrow (\mathcal{C}, \subseteq_i)$ est un *beau préordre*?

Conjecture [Daligault et al., 2010] :

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

$\implies$

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

Différentes représentations

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

$\rightsquigarrow$ Classe de largeur de clique bornée

Et plus tard?

$\rightsquigarrow$ *homomorphismes*?

$\rightsquigarrow$ relations $\rightarrow$?

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

Système de transition bien structuré

$\mathfrak{A}' \dashrightarrow \mathfrak{B}'$

$\mathfrak{U} \quad \mathfrak{U}'$

$\mathfrak{A} \longrightarrow \mathfrak{B}$

Couvrabilité :

$D_0 \rightarrow^* D'_i \supseteq D_{\text{err}}$ décidable

(Sous hypothèses de *calculabilité*)

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 ? »

$(\mathcal{C}, \rightarrow_{\Delta}, \rightarrow_{\text{hom}}) \rightsquigarrow D_0 \rightarrow_{\Delta}^* D_{\text{err}} \models \Delta \wedge \neg q?$

Objet : programmes Python « simples »

Point de départ : *fonctions polyrégulières*

```
1 def getBetween(l, i, j):
2     """ Get elements between i and j """
3     for (k, c) in enumerate(l):
4         if i <= k and k <= j: ①
5             yield c ②
6
7 def containsAB(w):
8     """ Contains "ab" as a subsequence """
9     seen_a = False ③
10    for (x, c) in enumerate(w):
11        if c == "a": ④
12            seen_a = True ⑤
13        elif seen_a and c == "b":
14            return True
15    return False
16
17 def subwordsWithAB(word):
18     """ Get subwords that contain "ab" """
19    for (i,c) in enumerate(word): ⑥
20        for (j,d) in reversed(enumerate(word)): ⑦
21            s = getBetween(word, i, j) ⑧
22            if containsAB(s):
23                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*

PROBLÈMES THÉORIQUES À LONG TERME

Décider l'équivalence de fonctions

Décider la FO-définissabilité

```
def getBetween(l, i, j):
    for (c) in enumerate(l):
        if i <= c <= j:
            yield c

7 def containsAB(w):
8     """ Contains "ab" as a subsequence """
9     seen_a = False
10    for (x, c) in enumerate(w):
11        if c == "a":
12            seen_a = True
13        elif seen_a and c == "b":
14            return True
15    return False

16
17 def subwordsWithAB(word):
18     """ Get subwords that contain "ab" """
19    for (i,c) in enumerate(word):
20        for (j,d) in reversed(enumerate(word)):
21            s = getBetween(word, i, j)
22            if containsAB(s):
23                yield s
```

Poly
↑
FOPoly

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

SYSTÈMES DE TRANSITION

FONCTIONS POLYRÉGULIÈRES

Objet : programmes Python « simples »

Point de départ : *fonctions polyrégulières*

PROBLÈMES THÉORIQUES À LONG TERME

Décider l'équivalence de fonctions

Décider la FO-définissabilité

Dans un premier temps

↪ sorties unaires $\{a\}^* \simeq \mathbb{N}$

↪ entrées unaires / commutatives?

↪ lien quantitatif / qualitatif?

```

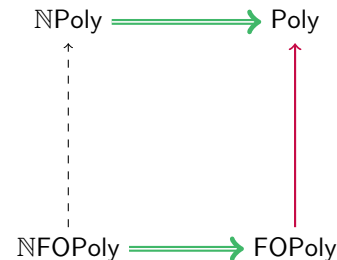
def getBetween(l, i, j):
    for (c) in enumerate(l):
        if i <= c <= j:
            return True
    return False

def containsAB(w):
    """ Contains "ab" as a subsequence """
    for (i, c) in enumerate(w):
        if c == 'a':
            seen_a = True
    for (j, d) in reversed(enumerate(w)):
        if d == 'b' and seen_a:
            return True
    return False

def subwordsWithAB(word):
    """ Get subwords that contain "ab" """
    for (i, c) in enumerate(word):
        if c == 'a':
            for (j, d) in reversed(enumerate(word)):
                if d == 'b':
                    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



SYSTÈMES DE TRANSITION

FONCTIONS POLYRÉGULIÈRES

Objet : programmes Python « simples »

Point de départ : fonctions *polyrégulières*

PROBLÈMES THÉORIQUES À LONG TERME

Décider l'équivalence de fonctions

Décider la FO-définissabilité

Dans un premier temps

↪ sorties unaires $\{a\}^* \simeq \mathbb{N}$

↪ entrées unaires / commutatives?

↪ lien quantitatif / qualitatif?

```

17 def subwordsWithAB(word):
18     """ Get subwords that contain "ab" """
19     for (i,c) in enumerate(word):
20         for (j,d) in reversed enumerate(word):
21             s = getBetween(word, i, j)
22             if containsAB(s):
23                 yield s

```

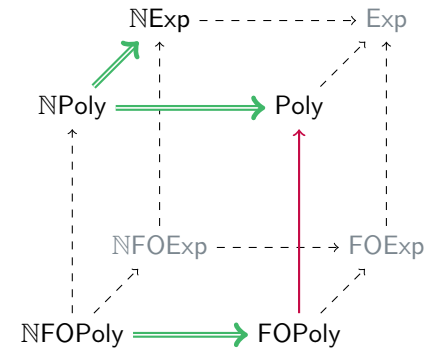


Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword

Objet : programmes Python « simples »

Point de départ : fonctions *polyrégulières*

PROBLÈMES THÉORIQUES À LONG TERME

Décider l'équivalence de fonctions

Décider la FO-définissabilité

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?

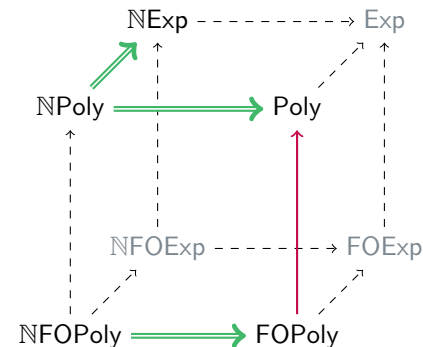


Fig. 1. A small Python program that outputs all subwords of a given word containing ab as a scattered subword

SYSTÈMES DE TRANSITION

FONCTIONS POLYRÉGULIÈRES

Objet : programmes Python « simples »

Point de départ : fonctions *polyrégulières*

PROBLÈMES THÉORIQUES À LONG TERME

Décider l'équivalence de fonctions

Décider la FO-définissabilité

Dans un premier temps

↪ sorties unaires $\{a\}^* \simeq \mathbb{N}$

↪ entrées unaires / commutatives?

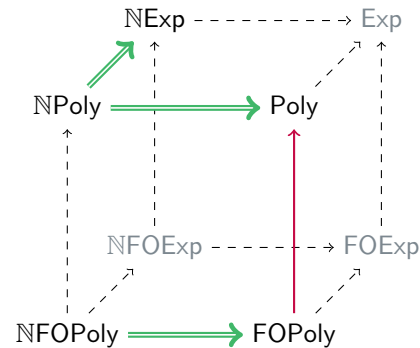
↪ lien quantitatif / qualitatif?

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



POLYCHECK / POLICZEK

Checking Hoare Triples for polyregular functions.

INPUT FORM

Write a For-Program Π , a precondition Φ , and a postcondition Ψ in the textareas below. To check that the program Π satisfies the specification $(\Phi) \Pi (\Psi)$, i.e. that for every input word, if the precondition Φ holds, then the postcondition Ψ holds, for the output of the program Π .

CHECK button.

Π --- CODE:

```
def main (w : [char]) : [char] :=
  for (i, c) in enumerate(w) do
    yield c
  for (j, d) in enumerate(w) do
    if i == j then
      yield d
    endif
  done
done
```

OUTPUT VIEW

All inputs are valid and ready for verification!

[Lopez, STEFAŃSKI, CAV'25]

Output of the program will appear here

INTÉGRATION



INTÉGRATION

LaBRI

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

INTÉGRATION

LaBRI



JOANNA
FIJALKOW



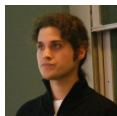
MARC
ZEITOUN



HUGO
GIMBERT



JÉRÔME
LEROUX



THOMAS
PLACE



NATHANAËL
FIJALKOW



SYLVAIN
LOMBARDY



VINCENT
PENELLE



IGOR
WALUKIEWICZ

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

INTÉGRATION

LaBRI



JOANNA
FIJALKOW



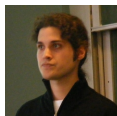
MARC
ZEITOUN



HUGO
GIMBERT



JÉRÔME
LEROUX



THOMAS
PLACE



NATHANAËL
FIJALKOW



SYLVAIN
LOMBARDY



VINCENT
PENELLE



IGOR
WALUKIEWICZ

Automates

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

INTÉGRATION

LaBRI



JOANNA
FIJALKOW



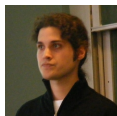
MARC
ZEITOUN



HUGO
GIMBERT



JÉRÔME
LEROUX



THOMAS
PLACE



NATHANAËL
FIJALKOW



SYLVAIN
LOMBARDY



VINCENT
PENELLE



IGOR
WALUKIEWICZ

Automates

Automates
Pondérés

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

INTÉGRATION

LaBRI

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory



JOANNA
FIJALKOW



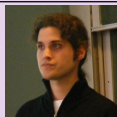
MARC
ZEITOUN



HUGO
GIMBERT



JÉRÔME
LEROUX



THOMAS
PLACE



NATHANAËL
FIJALKOW



SYLVAIN
LOMBARDY



VINCENT
PENELLE



IGOR
WALUKIEWICZ

Automates

Automates
Pondérés

Apériodicité

INTÉGRATION

LaBRI

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

Automates

Automates
Pondérés

Apériodicité

Beaux Préordres



JOANNA
FIJALKOW



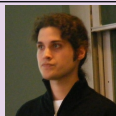
MARC
ZEITOUN



HUGO
GIMBERT



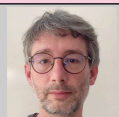
JÉRÔME
LEROIX



THOMAS
PLACE



NATHANAËL
FIJALKOW



SYLVAIN
LOMBARDY



VINCENT
PENELLE



IGOR
WALUKIEWICZ

INTÉGRATION

LaBRI

M2F

Modélisation et Méthodes Formelles

LX

Logical foundations of computing

Automata Theory

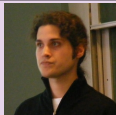
Formal Languages

Logic in computing

Algebraic techniques

Graphs and logics

Computability theory

 JOANNA FIJALKOW	 MARC ZEITOUN	 HUGO GIMBERT
 JÉRÔME LEROUX	 THOMAS PLACE	 NATHANAËL FIJALKOW
 SYLVAIN LOMBARDY	 VINCENT PENELLE	 IGOR WALUKIEWICZ

Automates

Automates
Pondérés

Apériodicité

Beaux Préordres

Modèles Finis

INTÉGRATION



		
JOANNA FIJALKOW	MARC ZEITOUN	ILAN
		
JÉRÔME LEROUX	THOMAS PLACE	
		
SYLVAIN LOMBARDY	VINCENT PENELLE	

Autre équipe : **RATIO**
Reasoning with data, knowledge, and constraints

Collaborations existantes (non-permanents)




Arka Ghosh Rémi Morvan

M2F
n et Méthodes Formelles

LX
undations of computing

- Formal Languages
- Algebraic techniques
- Computability theory

COURS TD/TP

Info. Théorique	72h
Programmation	60h
Réseaux / Système	48h

Niveaux : L1 / L2 / L3 / M1

Type : Université / École

Césure : Développeur Fullstack

Autorité de Sécurité Nucléaire

Agrégation :

Mathématiques option informatique.
Khôlleur de mathématiques (MPSI, MP)
de 2017 à 2019.

RÉSUMÉ**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

PUBLICATIONS

Conférences	9 (dont 4 en seul auteur)
Journaux	2
Soumissions	1