My Humble Journey Through Academia

Michael Blondin

Université de Sherbrooke

My journey in computer science

My journey in computer science

- **DEC @ Collège Lionel-Groulx, 2004 – 2007**

My journey in computer science

- **DEC @ Collège Lionel-Groulx, 2004 – 2007**
- **B.Sc. @ U. Montreal, 2007 – 2009**

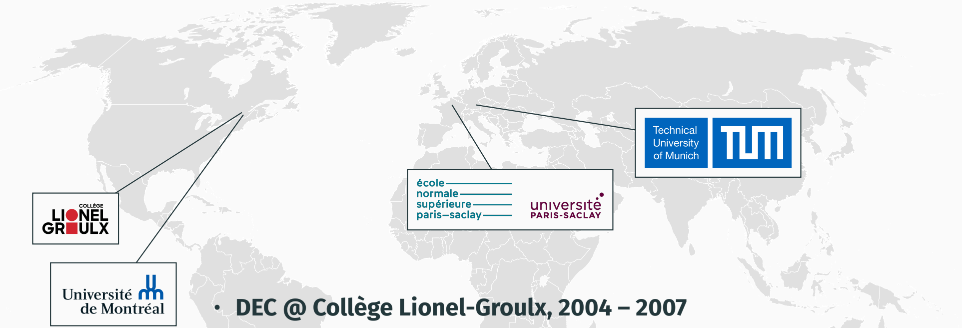
My journey in computer science

- **DEC @ Collège Lionel-Groulx, 2004 – 2007**
- **B.Sc. @ U. Montreal, 2007 – 2009**
- **M.Sc. @ U. Montreal, 2009 – 2011**

My journey in computer science

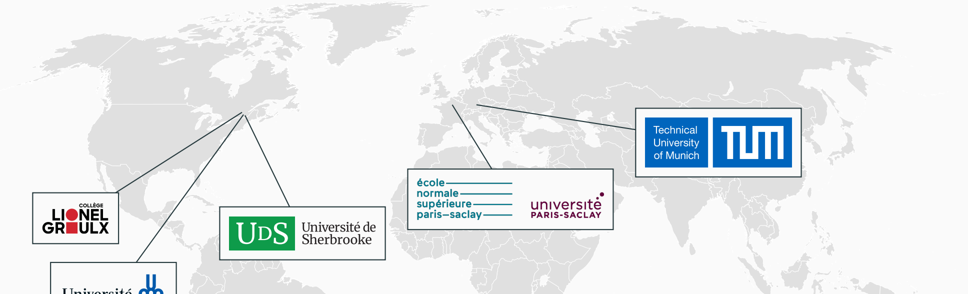
- DEC @ Collège Lionel-Groulx, 2004 – 2007
- B.Sc. @ U. Montreal, 2007 – 2009
- M.Sc. @ U. Montreal, 2009 – 2011
- Ph.D. @ U. Montreal and ENS Paris-Saclay, 2012 – 2016

My journey in computer science



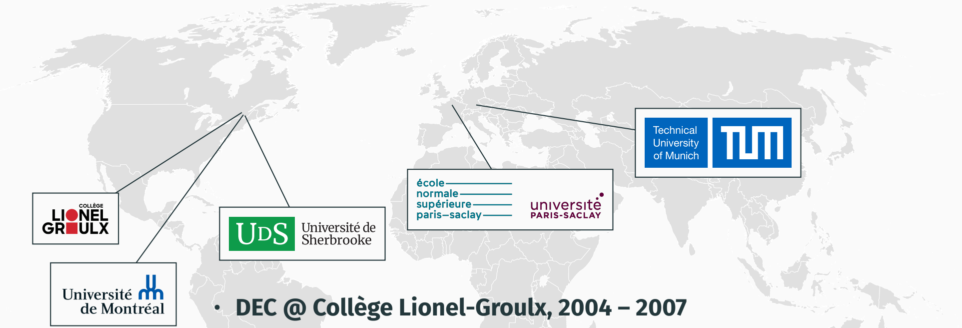
- **DEC @ Collège Lionel-Groulx, 2004 – 2007**
- **B.Sc. @ U. Montreal, 2007 – 2009**
- **M.Sc. @ U. Montreal, 2009 – 2011**
- **Ph.D. @ U. Montreal and ENS Paris-Saclay, 2012 – 2016**
- **Postdoc @ TU Munich, 2016 – 2018**

My journey in computer science



- **DEC @ Collège Lionel-Groulx, 2004 – 2007**
- **B.Sc. @ U. Montreal, 2007 – 2009**
- **M.Sc. @ U. Montreal, 2009 – 2011**
- **Ph.D. @ U. Montreal and ENS Paris-Saclay, 2012 – 2016**
- **Postdoc @ TU Munich, 2016 – 2018**
- **Assistant professor @ U. Sherbrooke, 2018 – 2023**

My journey in computer science



- DEC @ Collège Lionel-Groulx, 2004 – 2007
- B.Sc. @ U. Montreal, 2007 – 2009
- M.Sc. @ U. Montreal, 2009 – 2011
- Ph.D. @ U. Montreal and ENS Paris-Saclay, 2012 – 2016
- Postdoc @ TU Munich, 2016 – 2018
- Assistant professor @ U. Sherbrooke, 2018 – 2023
- Associate (tenured) professor @ U. Sherbrooke, 2023 – Now

B.Sc. and M.Sc.: interests

Interested in theoretical CS

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)
- Cryptography
- Quantum computing

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)
- Cryptography
- Quantum computing
- Flow networks
- Optimization
- Group representation theory (@ math departement)
- Advanced cryptography (@ McGill University)

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)
- Cryptography
- Quantum computing
- Flow networks
- Optimization
- Group representation theory (@ math departement)
- Advanced cryptography (@ McGill University)
- (Also for fun: linguistics and history of math)

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)
- Cryptography
- Quantum computing
- Flow networks
- Optimization
- Group representation theory (@ math departement)
- Advanced cryptography (@ McGill University)
- (Also for fun: linguistics and history of math)

Regret: not doing more math...

B.Sc. and M.Sc.: interests

Interested in theoretical CS, e.g. I took these optional courses:

- Computational complexity theory
- Numerical algorithms
- Graph theory (@ math departement)
- Group theory (@ math departement)
- Cryptography
- Quantum computing
- Flow networks
- Optimization
- Group representation theory (@ math departement)
- Advanced cryptography (@ McGill University)
- (Also for fun: linguistics and history of math)

~~Regret: not doing more math...~~

Possible to learn more along the way!

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−
```

Too low for undergrad research internship 😱

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−
```

Too low for undergrad research internship 😱 😠

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−



A
A+
A+
B

3,3
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−



A
A+
A+
B

3,3
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−



A
A+
A+
B

3,3

A+
A+
A+
A

4,2
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−



A
A+
A+
B

3,3

A+
A+
A+
A

4,2
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

```
A
C
D+
B+
A−



A
A+
A+
B

3,3

A+
A+
A+
A

4,2

A+
A
A−
A+

A
A+
A+

A

4,1

A
A+
A
A−


4,0
```

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

A
C
D+
B+
A−

A
A+
A+
B

3,3

A+
A+
A+
A

4,2

A+
A
A−
A+

A
A+
A+

A

4,1

A
A+
A
A−

4,0

Just enough for scholarships 😊

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

A
C
D+
B+
A−

A
A+
A+
B

3,3

A+
A+
A+
A

4,2

A+
A
A−
A+

A
A+
A+

A

4,1

A
A+
A
A−

4,0

Regret: not taking 1st semester more seriously
(2.86/4.3 vs. 4.07/4.3 → 3.82/4.3)

B.Sc. and M.Sc.: grades

Grades (B.Sc.)

A
C
D+
B+
A−

A
A+
A+
B

3,3

A+
A+
A+
A

4,2

A+
A
A−
A+

A
A+
A+

A

4,1

A
A+
A
A−

4,0

~~Regret: not taking 1st semester more seriously (2.86/4.3 vs. 4.07/4.3 → 3.82/4.3)~~

Possible to make your own way!

Initiation to research

- Undergrad research scholarship (B.Sc. last summer): *arithmetic circuits and proof trees* 

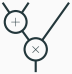
B.Sc. and M.Sc.: research

Initiation to research

- Undergrad research scholarship (B.Sc. last summer): *arithmetic circuits and proof trees* 

  Outcome: read a dozen of papers/books on circuit complexity, algebra and parameterized complexity

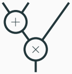
Initiation to research

- Undergrad research scholarship (B.Sc. last summer):
  arithmetic circuits and proof trees
  
  Outcome: read a dozen of papers/books on circuit complexity, algebra and parameterized complexity

- Research project (B.Sc. last semester):
  complexity of the automata intersection problem
  

B.Sc. and M.Sc.: research

Initiation to research

- Undergrad research scholarship (B.Sc. last summer):
  arithmetic circuits and proof trees 

  Outcome: read a dozen of papers/books on circuit complexity, algebra and parameterized complexity

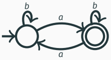
- Research project (B.Sc. last semester):
  complexity of the automata intersection problem

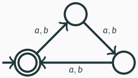
  Outcome: springboard for a master's 

B.Sc. and M.Sc.: research

Master's thesis: complexity of the automata intersection problem

Master's thesis: complexity of the automata intersection problem

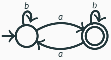
Problem: $\exists w \in$ 

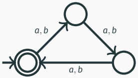
$\cap \cdots \cap$ 

?

B.Sc. and M.Sc.: research

Master's thesis: complexity of the automata intersection problem

Problem: $\exists w \in$ 

$\cap \cdots \cap$ 

?

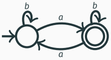
Results:

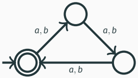
	Max size b of S_i ; max # of final states		
	1	2	3 or more
Single generator; $\lvert\Sigma\rvert = 1$	L	L	NP
Elementary 2-groups	$\oplus$L	$\oplus$L	NP (Beaudry 1988b)
Elementary p-groups	Mod_pL	NP	NP (Beaudry 1988b)
All abelian groups	$\in \mathrm{NC}^3, \in \mathrm{FL}^{\mathrm{ModL}}/\mathrm{poly}$	NP	NP (Beaudry 1988b)

	Max size b of S_i ; max # of final states		
	1	2	3 or more
Idempotent commutative	$\in \mathrm{AC}^0$	NP	NP (Beaudry 1988b)
Groups	$\in$ NC (Luks 1990)	NP	NP
Commutative	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)
Idempotent	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)
Aperiodic	NP (Beaudry *et al.* 1992)	NP	NP (Beaudry *et al.* 1992)
All monoids	PSPACE (Kozen 1977)	PSPACE (Kozen 1977)	PSPACE (Kozen 1977)

B.Sc. and M.Sc.: research

Master's thesis: complexity of the automata intersection problem

Problem: $\exists w \in$ 

$\cap \cdots \cap$ 

?

Results:

	Max size b of S_i ; max # of final states		
	1	2	3 or more
Single generator; $\lvert\Sigma\rvert = 1$	L	L	NP
Elementary 2-groups	$\oplus$L	$\oplus$L	NP (Beaudry 1988b)
Elementary p-groups	Mod$_p$L	NP	NP (Beaudry 1988b)
All abelian groups	$\in \mathrm{NC}^3, \in \mathrm{FL}^{\mathrm{ModL}}/\mathrm{poly}$	NP	NP (Beaudry 1988b)

	Max size b of S_i ; max # of final states		
	1	2	3 or more
Idempotent commutative	$\in \mathrm{AC}^0$	NP	NP (Beaudry 1988b)
Groups	$\in$ NC (Luks 1990)	NP	NP
Commutative	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)
Idempotent	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)	NP (Beaudry *et al.* 1992)
Aperiodic	NP (Beaudry *et al.* 1992)	NP	NP (Beaudry *et al.* 1992)
All monoids	PSPACE (Kozen 1977)	PSPACE (Kozen 1977)	PSPACE (Kozen 1977)

Outcome: CSR'12 + Computational Complexity'16

Ph.D.: naive plan

Applied for a scholarship to do this:

Super-quadratic lower bounds on the size of branching programs

Ph.D.: naive plan

Applied for a scholarship to do this:

Super-quadratic lower bounds on the size of branching programs

Instead, did this for > 6 months:

Ph.D.: naive plan

Applied for a scholarship to do this:

Super-quadratic lower bounds on the size of branching programs

Instead, did this for > 6 months:

What to do with experience in theoretical CS and strikes?

Ph.D.: naive plan

Applied for a scholarship to do this:

Super-quadratic lower bounds on the size of branching programs

Instead, did this for > 6 months:

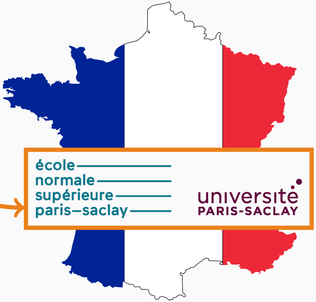
What to do with experience in theoretical CS and strikes?

Ph.D.: naive plan

Applied for a scholarship to do this:

Super-quadratic lower bounds on the size of branching programs

Instead, did this for > 6 months:



What to do with experience in theoretical CS and strikes?

Ph.D.: new horizons

Ph.D.: new horizons

Pierre McKenzie
(Toronto ’84)

Alain Finkel
(Paris-Sud XI ’86)

Ph.D.: new horizons

Complexity

Stephen Cook (Harvard ’66)

Allan Borodin (Cornell ’69)

Pierre McKenzie (Toronto ’84)

Concurrency and verification

Guy Vidal-Naquet (Paris-VI ’81)

Alain Finkel (Paris-Sud XI ’86)

Ph.D.: new horizons

Algorithms and complexity of counter-based systems

Complexity

Stephen Cook
(Harvard ’66)

Allan Borodin
(Cornell ’69)

Pierre McKenzie
(Toronto ’84)

Concurrency and verification

Guy Vidal-Naquet
(Paris-VI ’81)

Alain Finkel
(Paris-Sud XI ’86)

Ph.D.: new horizons

Algorithms and complexity of counter-based systems

Complexity

Stephen Cook (Harvard '66) — Allan Borodin (Cornell '69)

Pierre McKenzie (Toronto '84)

Concurrency and verification

Guy Vidal-Naquet (Paris-VI '81)

Alain Finkel (Paris-Sud XI '86)

Watch out:
finding an apartment
is the first open problem!

Ph.D.: contribution (∼3 years)

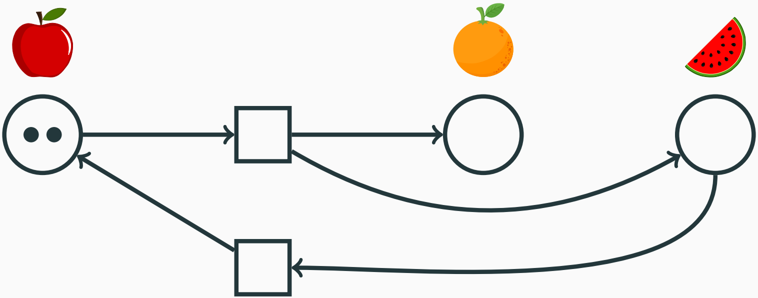
Conference papers

- M. Blondin, A. Finkel, P. McKenzie. ***Handling Infinitely Branching WSTS***. **ICALP 2014**.
- M. Blondin, A. Finkel, S. Göller, C. Haase, P. McKenzie. ***Reachability in 2D Vector Addition Systems with States is PSPACE-complete***. **LICS 2015**.
- M. Blondin, A. Finkel, C. Haase, S. Haddad. ***Approaching the Coverability Problem Continuously***. **TACAS 2016**.

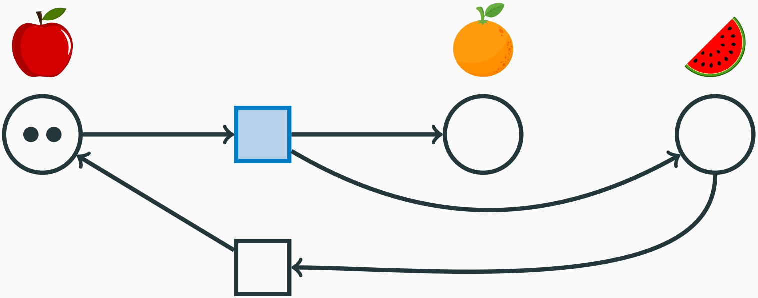
Journal papers

- M. Blondin, A. Finkel, C. Haase, S. Haddad. *The Logical View on Continuous Petri Nets*. ACM Transactions on Computational Logic (TOCL), 2017.
- M. Blondin, A. Finkel, P. McKenzie. *Well Behaved Transition Systems*. Logical Methods in Computer Science (LMCS), 2017.
- M. Blondin, A. Finkel, P. McKenzie. *Handling Infinitely Branching Well-structured Transition Systems*. Information and Computation, 2018.
- M. Blondin, M. Englert, A. Finkel, S. Göller, C. Haase, R. Lazić, P. McKenzie, P. Totzke. *The Reachability Problem for Two-Dimensional Vector Addition Systems with States*. Journal of the ACM (JACM), 2021.

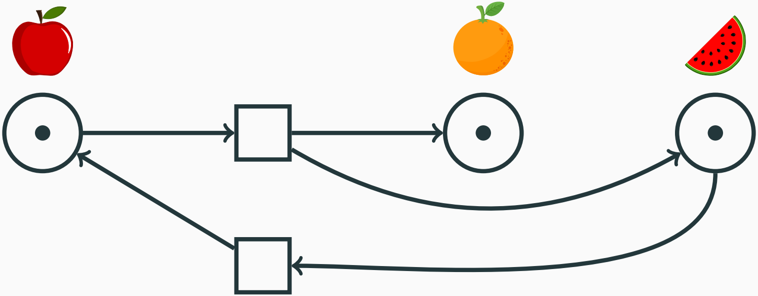
Ph.D.: reachability in Petri nets



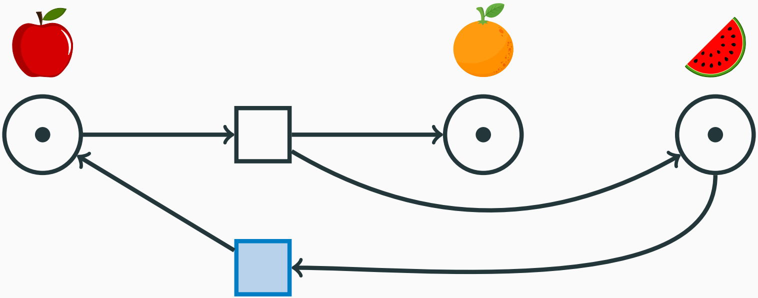
Ph.D.: reachability in Petri nets

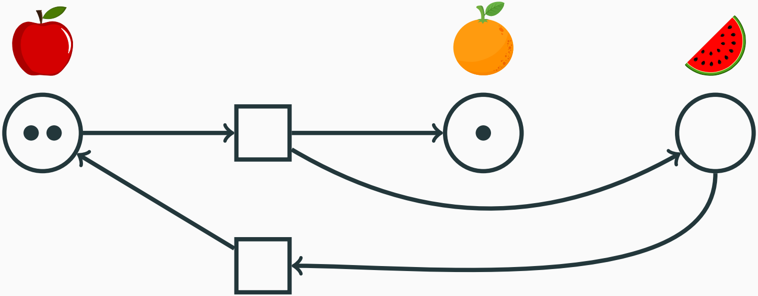


Ph.D.: reachability in Petri nets

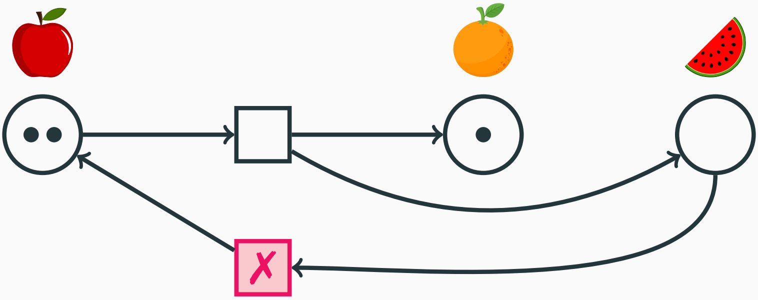


Ph.D.: reachability in Petri nets

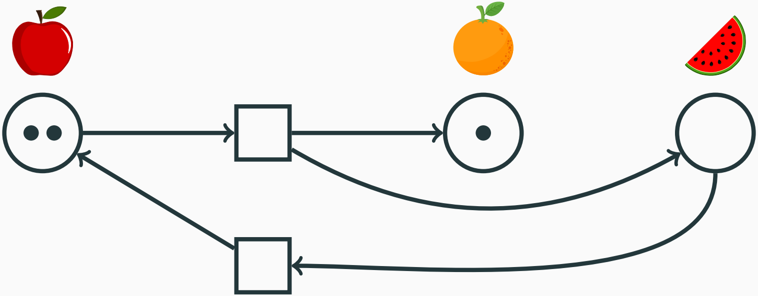




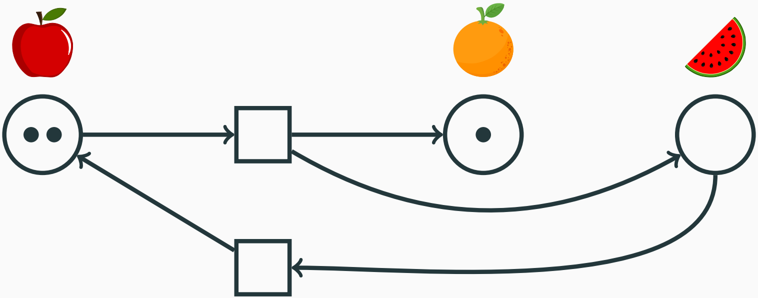
Ph.D.: reachability in Petri nets



Ph.D.: reachability in Petri nets

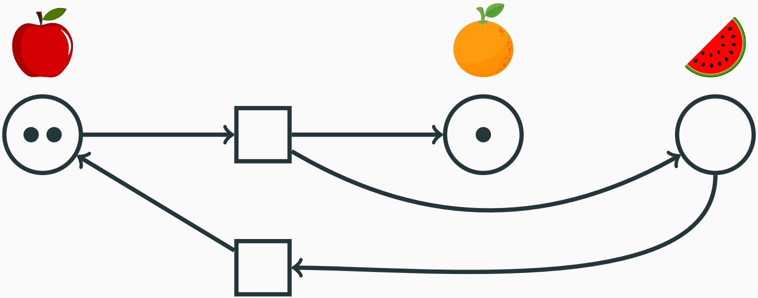


$$\{🍎 : 2\} \xrightarrow{*} \{🍎 : 2, 🍊 : 1\}$$



Reachability: $\boldsymbol{x} \xrightarrow{*} \boldsymbol{y}$?

Ph.D.: reachability in Petri nets

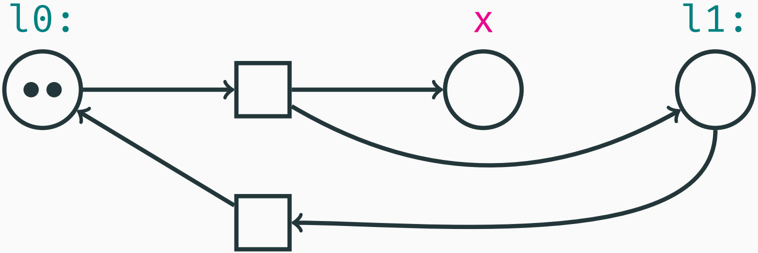


Useful for the formal verification of

- Concurrent programs
- Protocols
- Business processes
- Biological processes

⋮

Ph.D.: reachability in Petri nets



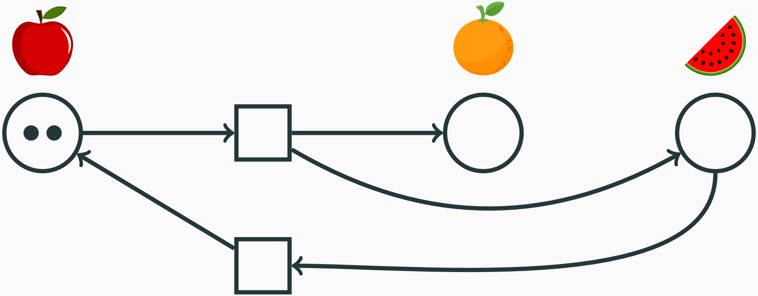
Useful for the formal verification of

- Concurrent programs
- Protocols
- Business processes
- Biological processes

⋮

```
      spawn proc()
      spawn proc()

      proc():
l0:     x++
l1:     goto l0
```

Ph.D.: reachability in Petri nets



Useful for the formal verification of

- Concurrent programs
- Protocols
- Business processes
- Biological processes

⋮

I did not care much about this...

Ph.D.: reachability in Petri nets

Quanta magazine — Physics · Mathematics · Biology · Computer Science · Topics · Archive

COMPUTATIONAL COMPLEXITY

An Easy-Sounding Problem Yields Numbers Too Big for Our Universe

Researchers prove that navigating certain systems of vectors is among the most complex computational problems.

Ph.D.: reachability in Petri nets

In general	Two unbounded counters
EXPSPACE-hard (1976)	
	Decidable (1979)
Decidable (1981–2011)	
	$\in$ 2-EXPTIME (1986)

Ph.D.: reachability in Petri nets

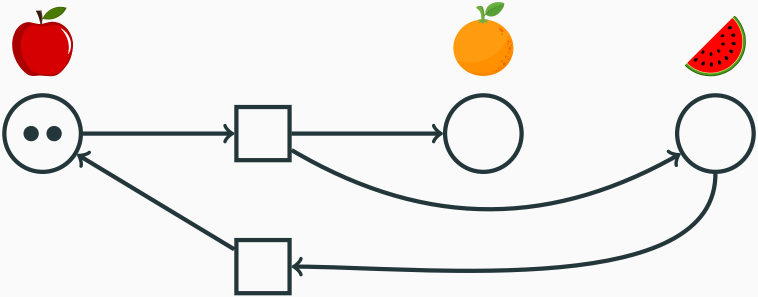
In general	Two unbounded counters
EXPSPACE-hard (1976)	
	Decidable (1979)
Decidable (1981–2011)	
	$\in$ 2-EXPTIME (1986)
$\in$ Ackermann (2015)	**PSPACE-complete (2015)**

Ph.D.: reachability in Petri nets

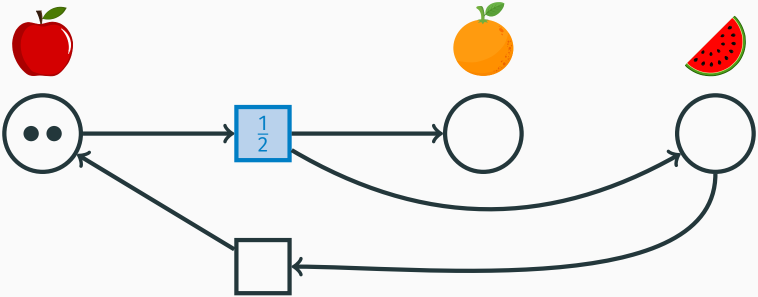
In general	Two unbounded counters
EXPSPACE-hard (1976)	
	Decidable (1979)
Decidable (1981–2011)	
	$\in$ 2-EXPTIME (1986)
$\in$ Ackermann (2015)	**PSPACE-complete (2015)**
$\mathcal{O}\left(2^{\cdot^{\cdot^{\cdot^{2}}}}\right)$-hard (2019)	
Ackermann-hard (2021)	

A(5) > # atoms in the universe

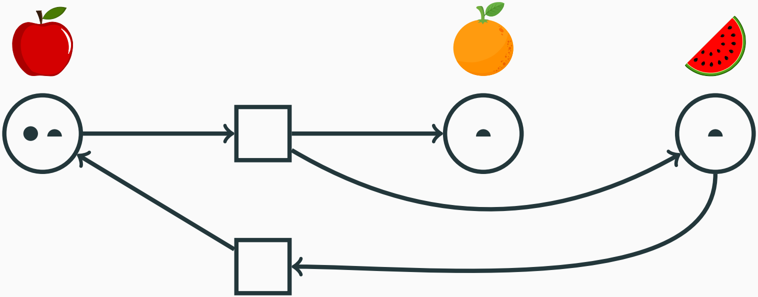
Ph.D.: continuous reachability Petri nets



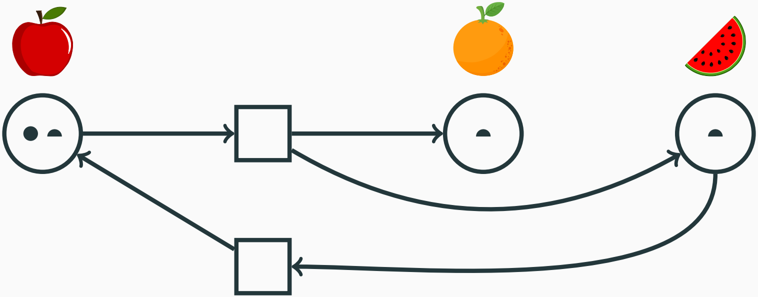
Ph.D.: continuous reachability Petri nets



Ph.D.: continuous reachability Petri nets



Ph.D.: continuous reachability Petri nets



Continuous reachability is solvable in polynomial time

Ph.D.: continuous reachability Petri nets

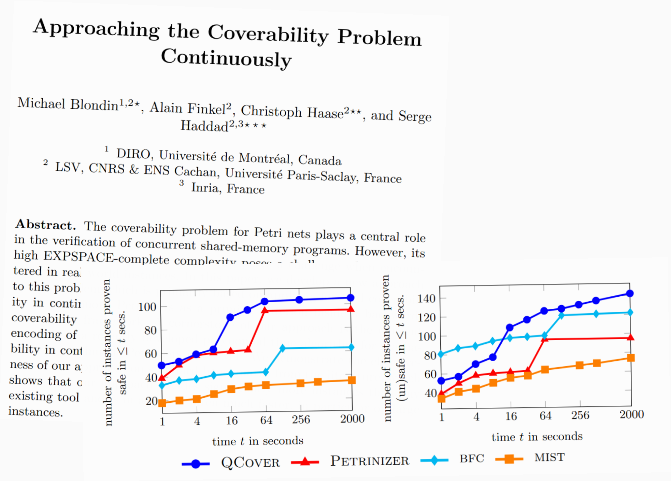
Approaching the Coverability Problem Continuously

Michael Blondin[1,2⋆], Alain Finkel[2], Christoph Haase[2⋆⋆], and Serge Haddad[2,3⋆⋆⋆]

[1] DIRO, Université de Montréal, Canada
[2] LSV, CNRS & ENS Cachan, Université Paris-Saclay, France
[3] Inria, France

Abstract. The coverability problem for Petri nets plays a central role in the verification of concurrent shared-memory programs. However, its high EXPSPACE-complete complexity poses a challenge when encountered in real-world instances. In this paper, we develop a new approach to this problem which is primarily based on applying forward coverability in continuous Petri nets as a pruning criterion inside a backward-coverability framework. A cornerstone of our approach is the efficient encoding of a recently developed polynomial-time algorithm for reachability in continuous Petri nets into SMT. We demonstrate the effectiveness of our approach on standard benchmarks from the literature, which shows that our approach decides significantly more instances than any existing tool and is in addition often much faster, in particular on large instances.

Ph.D.: continuous reachability Petri nets



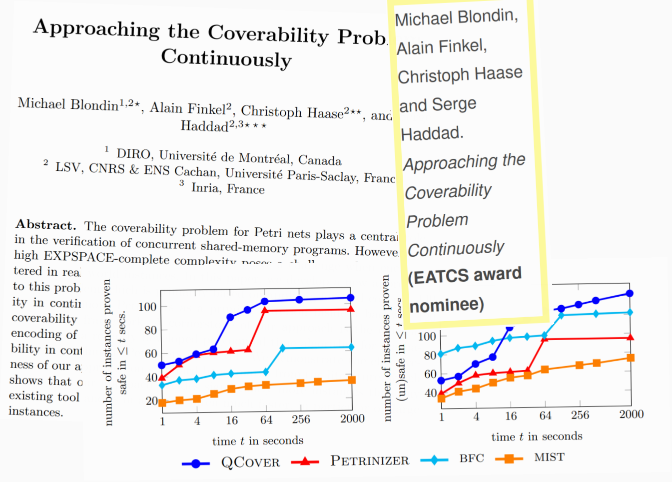
Approaching the Coverability Problem Continuously

Michael Blondin[1,2⋆], Alain Finkel[2], Christoph Haase[2⋆⋆], and Serge Haddad[2,3⋆⋆⋆]

[1] DIRO, Université de Montréal, Canada
[2] LSV, CNRS & ENS Cachan, Université Paris-Saclay, France
[3] Inria, France

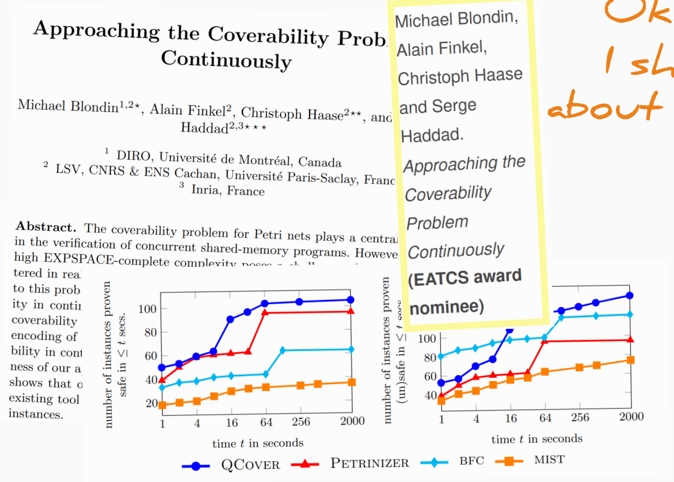
Abstract. The coverability problem for Petri nets plays a central role in the verification of concurrent shared-memory programs. However, its high EXPSPACE-complete complexity poses a ch
tered in rea
to this prob
ity in conti
coverability
encoding of
bility in con
ness of our a
shows that o
existing tool
instances.

Ph.D.: continuous reachability Petri nets



Michael Blondin, Alain Finkel, Christoph Haase and Serge Haddad. *Approaching the Coverability Problem Continuously* **(EATCS award nominee)**

Ph.D.: continuous reachability Petri nets


Approaching the Coverability Prob[lem]
Continuously

Michael Blondin[1,2*], Alain Finkel[2], Christoph Haase[2**], and
Haddad[2,3***]

[1] DIRO, Université de Montréal, Canada
[2] LSV, CNRS & ENS Cachan, Université Paris-Saclay, Franc[e]
[3] Inria, France

Abstract. The coverability problem for Petri nets plays a central
in the verification of concurrent shared-memory programs. Howeve[r]
high EXPSPACE-complete complexity
tered in rea
to this prob
ity in conti
coverability
encoding of
bility in cont
ness of our a
shows that o
existing tool
instances.

Michael Blondin, Alain Finkel, Christoph Haase and Serge Haddad. *Approaching the Coverability Problem Continuously* **(EATCS award nominee)**

Okay, maybe I should care about applications!

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer ☹ (Apr. 13, 2016)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer ☹ (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer 😟 (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer ☹ (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔

PhD and tenure are the hardest
Finding a postdoc is the easiest
— Colleague

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer 😟 (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔
- Contact Javier Esparza (May 4, 2016 @ 13:14)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer 😟 (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔
- Contact Javier Esparza (May 4, 2016 @ 13:14)
- Javier Esparza replies (May 4, 2016 @ 14:05)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer 😟 (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔
- Contact Javier Esparza (May 4, 2016 @ 13:14)
- Javier Esparza replies (May 4, 2016 @ 14:05)
- Skype interview (May 6, 2016)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer 😟 (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I should just quit 💔
- Contact Javier Esparza (May 4, 2016 @ 13:14)
- Javier Esparza replies (May 4, 2016 @ 14:05)
- Skype interview (May 6, 2016)
- Onsite interview and offer 🥂 (May 11, 2016)

End of Ph.D.: now what?

- Plan: apply 1 year ahead for a scholarship
  to go to Oxford (Oct. 2, 2015)
- Meanwhile: offer at U. of Warwick (Oct. 19, 2015)
- Scholarship rejected, too late for other offer ☹ (Apr. 13, 2016)
- Ph.D. thesis submission (Apr. 22, 2016)
- Unsure what to do, maybe I shou
- Contact Javier Esparza (May 4, 2016
- Javier Esparza replies (May 4, 2016
- Skype interview (May 6, 2016)
- Onsite interview and offer 🥂 (May 11, 2016)

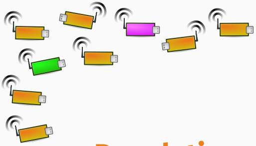
First open problem:
Obtaining visas

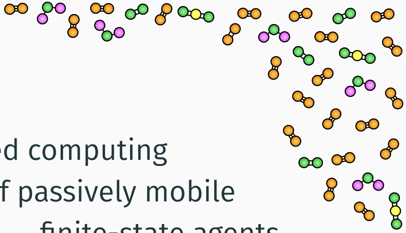
Second open problem:
Finding an apartment

Postdoc: population protocols

Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

Postdoc: population protocols

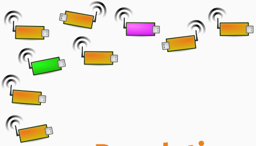


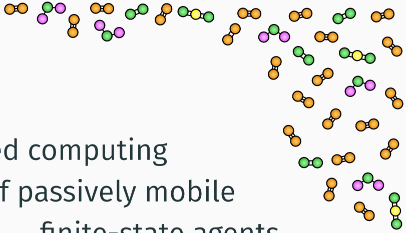


Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

Model *e.g.* networks of passively mobile sensors and chemical reaction networks

Postdoc: population protocols





Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

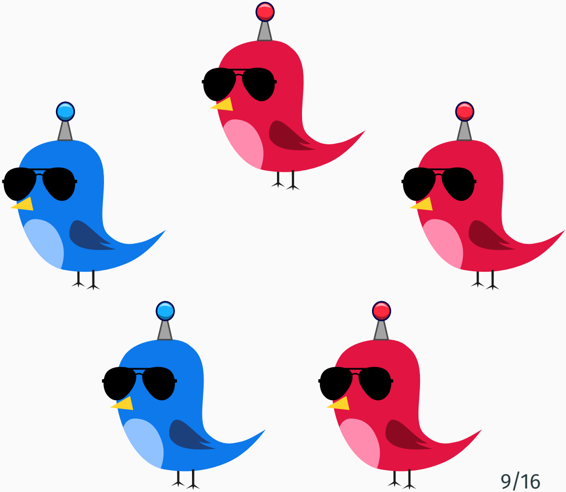
Model *e.g.* networks of passively mobile sensors and chemical reaction networks

Protocols compute predicates of the form $f\colon \mathbb{N}^d \to \{0, 1\}$

e.g. $f(m, n)$ is computed by $m + n$ agents

Postdoc: population protocols (majority)

red agents ≥ # blue agents?

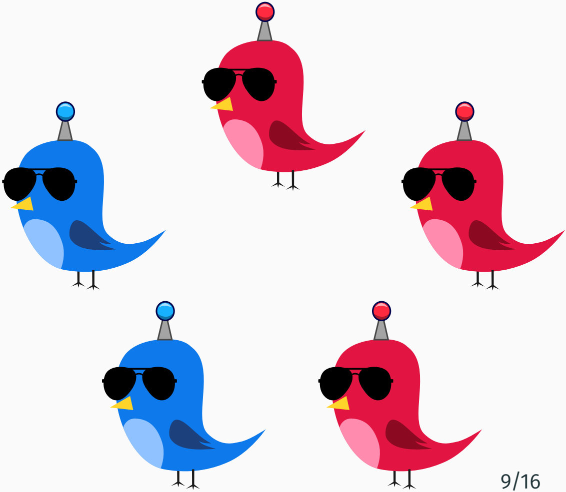


red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents

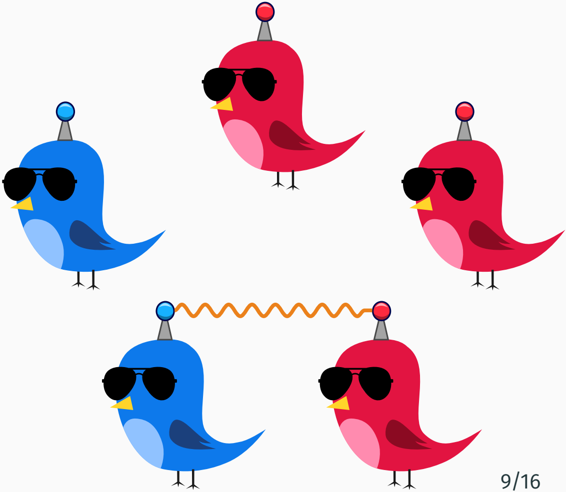
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

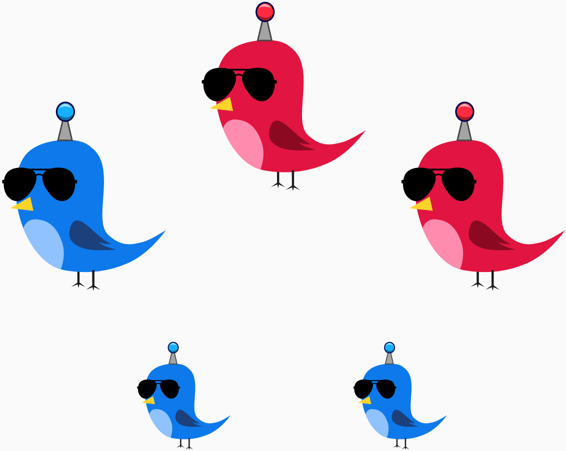
- Two large agents become small blue agents
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

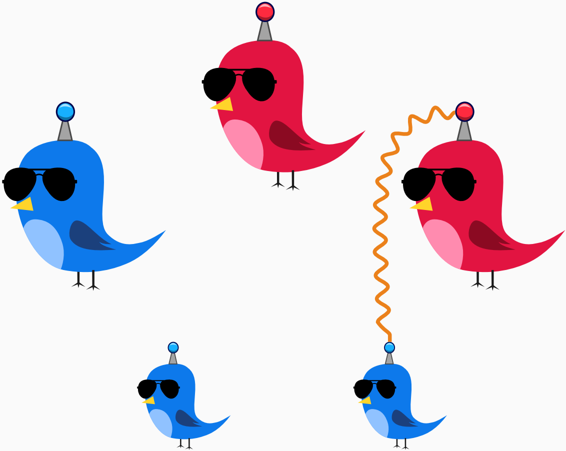
- Two large agents become small blue agents
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

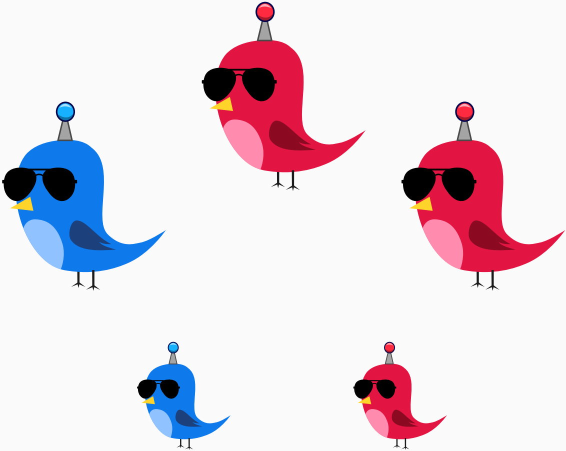


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

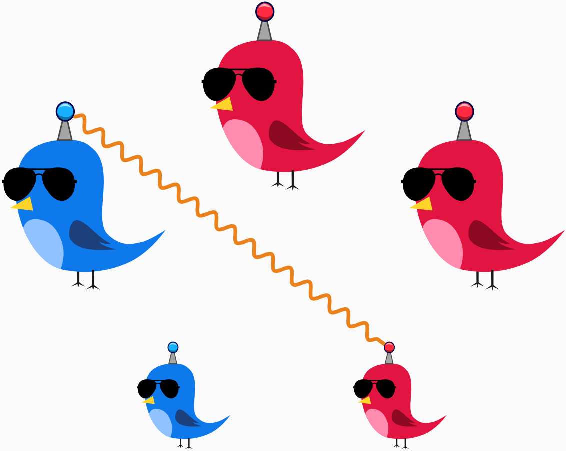


red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents

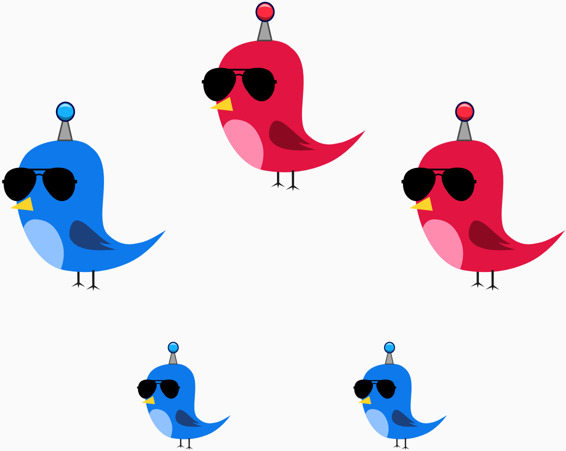
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour



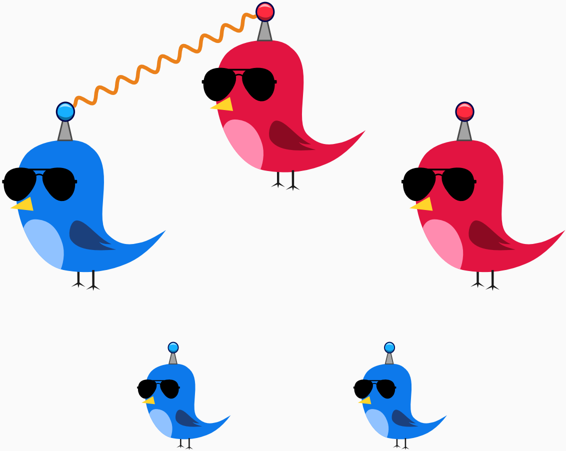
Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents

- Large agents convert small agents to their colour

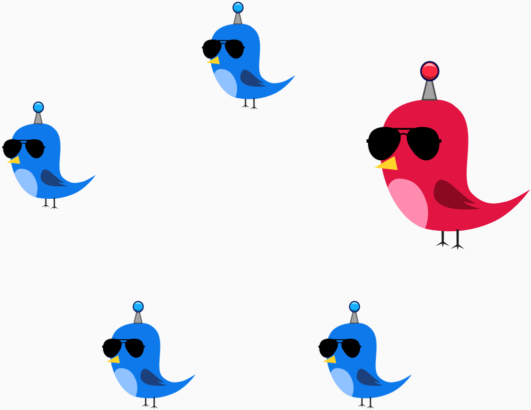


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

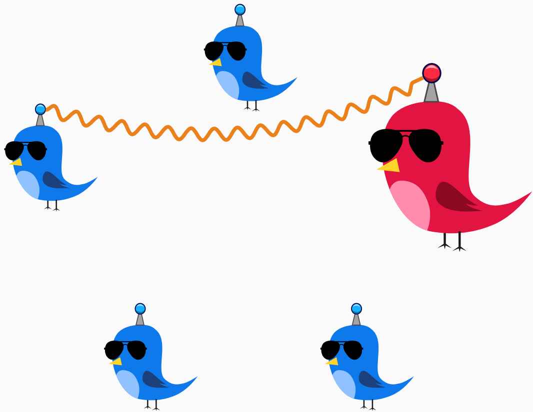


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

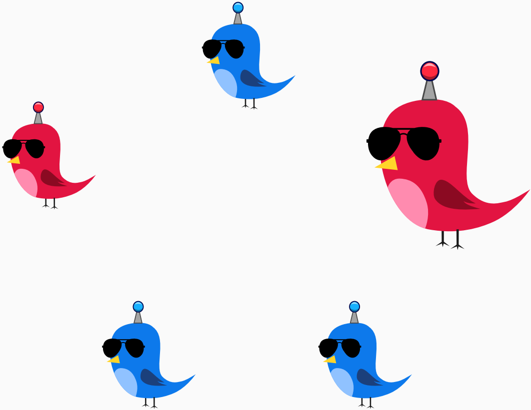


red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents

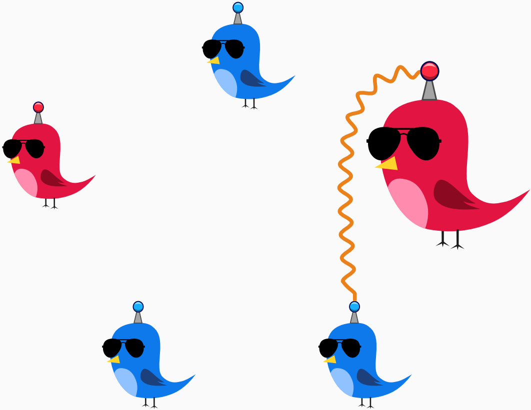
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

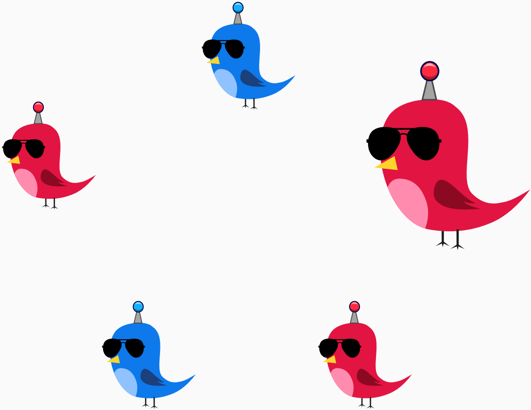


red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents

- Large agents convert small agents to their colour

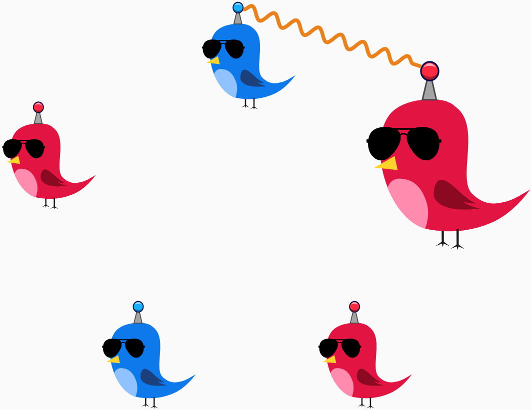


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

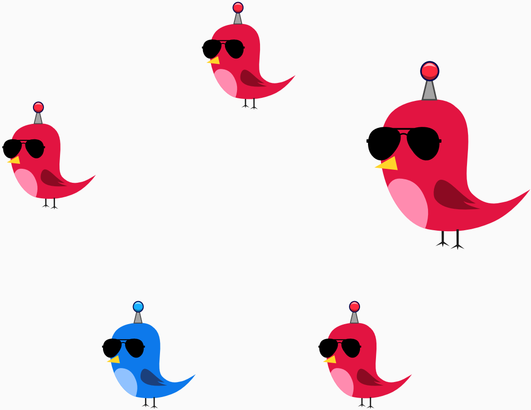


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

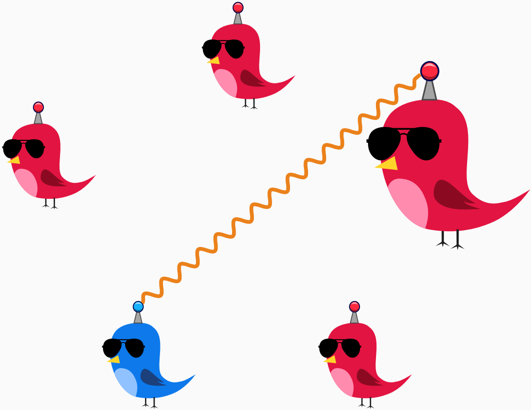
- Two large agents become small blue agents
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

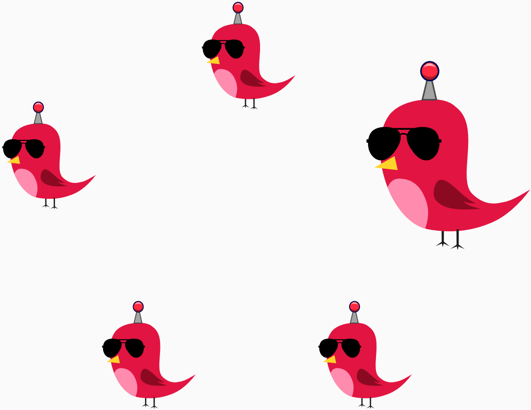
- Two large agents become small blue agents
- Large agents convert small agents to their colour



red agents ≥ # blue agents?

Protocol:

- Two large agents become small blue agents
- Large agents convert small agents to their colour

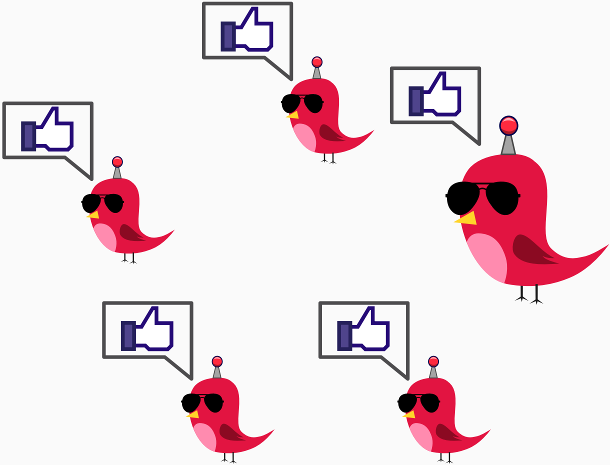


Postdoc: population protocols (majority)

red agents ≥ # blue agents?

Protocol:

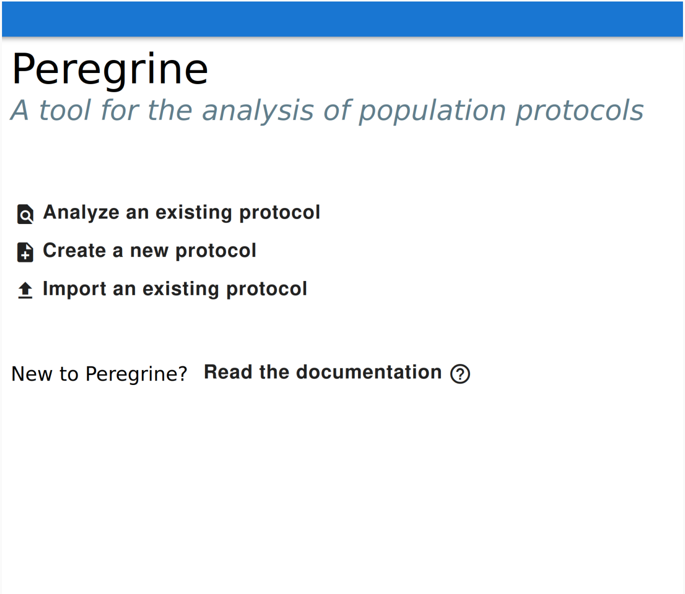
- Two large agents become small blue agents
- Large agents convert small agents to their colour



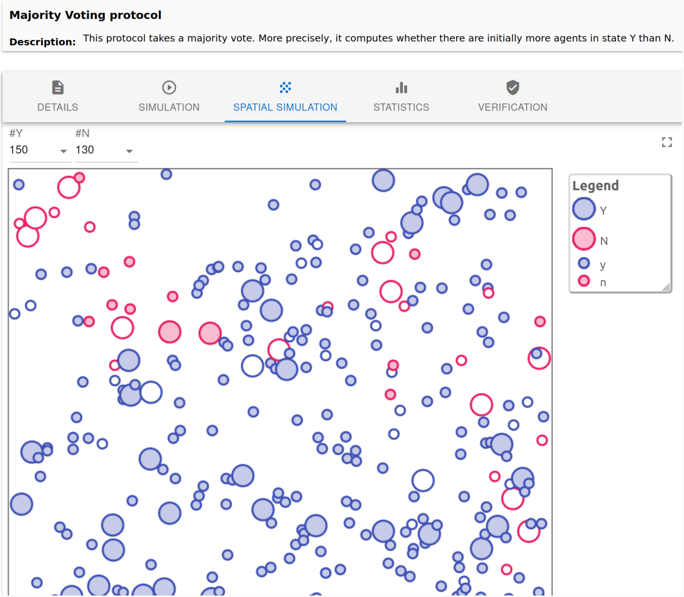
Postdoc: work

- Formally and automatically verify that a protocol works as intended (PODC'17)
- Automatic analysis of the expected running time of a protocol (CONCUR'18)
- Tool support (CAV'18)
- State complexity of protocols (STACS'18)

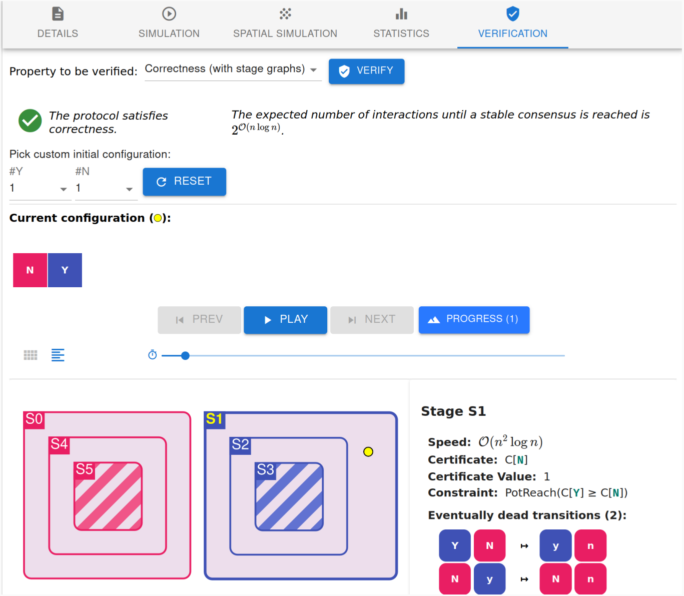
Postdoc: work



Postdoc: work



Postdoc: work



Postdoc: contribution ($\sim$2 years)

- Work with former Ph.D. advisor on WSTS: FSTTCS'17
- Work with collaborators on Petri net relaxations:
  LICS'17, CONCUR'18
- Work with J. Esparza on population protocols:
  PODC'17, STACS'18, CAV'18, CONCUR'18 (and LICS'18)
  + a journal version

Postdoc: contribution ($\sim$2 years)

- Work with former Ph.D. advisor on WSTS: FSTTCS'17
- Work with collaborators on Petri net relaxations:
  LICS'17, CONCUR'18
- Work with J. Esparza on population protocols:
  PODC'17, STACS'18, CAV'18, CONCUR'18 (and LICS'18)
  + a journal version

Nonregret: going to Munich!

Postdoc: contribution (∼2 years)

- Work with former Ph.D. advisor on WSTS: FSTTCS'17
- Work with collaborators on Petri net relaxations:
  LICS'17, CONCUR'18
- Work with J. Esparza on population protocols:
  PODC'17, STACS'18, CAV'18, CONCUR'18 (and LICS'18)
  + a journal version

Tip: do something at least a bit different during a postdoc

Prof: finding a position

- Wanted to come back to Quebec

  (family, friends, partner, language, sense of duty, etc.)

- Restricted pool of 3-5 interesting universities

- Luck: U. Sherbrooke opened a position one year into my postdoc

Prof: finding a position

- Wanted to come back to Quebec

  (family, friends, partner, language, sense of duty, etc.)

- Restricted pool of 3-5 interesting universities

- Luck: U. Sherbrooke opened a position one year into my postdoc

Only 2 positions in TCS across Quebec in 8 years

Prof: finding a position

- Wanted to come back to Quebec

  (family, friends, partner, language, sense of duty, etc.)

- Restricted pool of 3-5 interesting universities

- Luck: U. Sherbrooke opened a position one year into my postdoc

Watch out for such narrow expectations!

Prof: finding a position

- Wanted to come back to Quebec

  (family, friends, partner, language, sense of duty, etc.)

- Restricted pool of 3-5 interesting universities

- Luck: U. Sherbrooke opened a position one year into my postdoc

Beware of scientific isolation!

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)
- Student applies in Sherbrooke (May 2019)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)
- Student applies in Sherbrooke (May 2019)
- Okay for joint supervision with MPI (Sept. 2019)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)
- Student applies in Sherbrooke (May 2019)
- Okay for joint supervision with MPI (Sept. 2019)
- Beginning of Ph.D. in Sherbrooke (Jan. 2020)

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)
- Student applies in Sherbrooke (May 2019)
- Okay for joint supervision with MPI (Sept. 2019)
- Beginning of Ph.D. in Sherbrooke (Jan. 2020)

His publications during Ph.D.:
4 × LICS, 2 × CAV, 1 × TACAS + 2 × journals

Prof: finding a first Ph.D. student

- Twice TA + supervised undergrad project in Munich (2017)
- Student asks for reference letter for MPI (Dec. 12, 2018)
- Rejected. Reference letter for local university? (Apr. 14, 2019)
- Sure. By the way I have money in Sherbrooke (Apr. 15, 2019)
- No thanks, my French sucks (Apr. 16, 2019)
- Student applies in Sherbrooke (May 2019)
- Okay for joint supervision with MPI (Sept. 2019)
- Beginning of Ph.D. in Sherbrooke (Jan. 2020)

Tried to pull this a second time, but student went to ENS Paris-Saclay (and then Oxford)

Prof: scientific contribution ($\sim$6 years)

- *Formal verifcation methods for the development of reliable dynamic networks*
- *Efficient verification of concurrent and distributed infinite-state systems*
- *Automatic incorrectness analysis of counter-based systems*

Prof: scientific contribution ($\sim$6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

Prof: scientific contribution ($\sim$6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

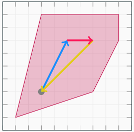
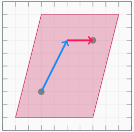
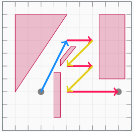
For example, $M = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$ where

$\boldsymbol{m}_1$	$\boldsymbol{m}_2$	$\boldsymbol{m}_3$
$(1, 2)$	$(1, 0)$	$(-1, -1)$

Prof: scientific contribution ($\sim$6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

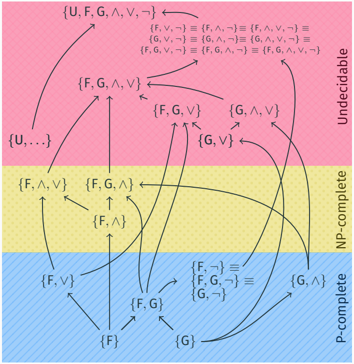
- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

Safe scheduling	**Safe reachability**	**Safe planning**	**etc.**
G Z	Z U target	$(\neg Z_1 \wedge \cdots \wedge \neg Z_4)$ U target	
			

Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR’19, STACS’20, CAV’20)

- **Relaxations of Petri nets**
  (LICS’20, TACAS’21, LICS’21, FoSSaCS’22, LICS’23
  + ACM SIGLOG News’20, LMCS’21, TOCL’23, LMCS’24, TOCL’25)

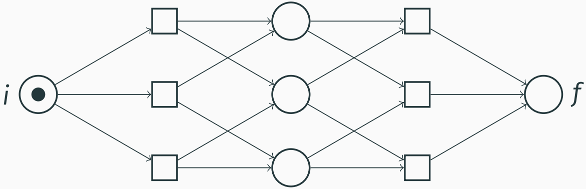


Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

- Verification of **workflow nets** (LICS'22, CAV'22, LICS'24)



Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

- Verification of **workflow nets** (LICS'22, CAV'22, LICS'24)

- Population protocols with **unordered data** (ICALP'23)

e.g. some color has absolute majority?

Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)

- **Relaxations of Petri nets**
  (LICS'20, TACAS'21, LICS'21, FoSSaCS'22, LICS'23
  + ACM SIGLOG News'20, LMCS'21, TOCL'23, LMCS'24, TOCL'25)

- Verification of **workflow nets** (LICS'22, CAV'22, LICS'24)

- Population protocols with **unordered data** (ICALP'23)

- **Data structures** for formal verification (TACAS'25)

Prof: scientific contribution (∼6 years)

- State complexity, expressiveness and verification of **population protocols** (CONCUR'19, STACS'20, CAV'20)
- **Relaxations of Petri nets**
  (L
  +
- V 4)
- P 23)
- **Data structures** for formal verification (TACAS'25)

Students on the papers mentioned:

- M.Sc. → data engineer
- Ph.D. → R&D @ Informal Systems
- M.Sc. → programmer

Prof: my tenure application after 4 years

- **Publications:** 9 × conf. + 6 × journals + 1 × invited + 1 × book
- **Mentoring:** 1 × Ph.D. + 3 × M.Sc. + 4 × interns + 10 × projects
- **Talks:** 9

Prof: my tenure application after 4 years

- **Publications:** 9 $\times$ conf. + 6 $\times$ journals + 1 $\times$ invited + 1 $\times$ book
- **Mentoring:** 1 $\times$ Ph.D. + 3 $\times$ M.Sc. + 4 $\times$ interns + 10 $\times$ projects
- **Talks:** 9
- **Teaching:** $\sim$3 courses / year (4 distinct courses overall)

Prof: my tenure application after 4 years

- **Publications:** 9 $\times$ conf. + 6 $\times$ journals + 1 $\times$ invited + 1 $\times$ book
- **Mentoring:** 1 $\times$ Ph.D. + 3 $\times$ M.Sc. + 4 $\times$ interns + 10 $\times$ projects
- **Talks:** 9
- **Teaching:** $\sim$3 courses / year (4 distinct courses overall)
- **Juries:** 3 $\times$ Ph.D. + 3 $\times$ M.Sc. + 8 $\times$ predoc. + 3 $\times$ scholarships
- **Reviewing:** 3 $\times$ program committees + 18 reviews
- **Administrative:** 7 $\times$ committees + president + union rep.

On taking risks

I think this is a sign that I don't take enough risks:

	Conf.	Journals
# submissions	27	12
# accepted	24	12
# rejected	3	0

On taking risks

I think this is a sign that I don't take enough risks:

	Conf.	CORE ranking
# submissions	27	
# accepted	24	11 × A*, 11 × A, 2 × national
# rejected	3	3 × A*

On taking risks

I think this is a sign that I don't take enough risks:

	Conf.	CORE ranking
# submissions	27	
# accepted	24	$11 \times$ A*, $11 \times$ A, $2 \times$ national
# rejected	3	$3 \times$ A*

Tip: take some risks!

On taking risks

I think this is a sign that I don't take enough risks:

	Conf.	CORE ranking
# submissions	27	
# accepted	24	11 × A*, 11 × A, 2 × national
# rejected	3	3 × A*

But: science takes time!

Teaching: back as a student

Teaching assistant:

Degree	# times as TA	# distinct courses
B.Sc.	2	2
M.Sc.	5	3
Ph.D.	8	4
Postdoc	6	4
Total	21	11

Teaching: back as a student

Teaching assistant:

Degree	# times as TA	# distinct courses
B.Sc.	2	2
M.Sc.	5	3
Ph.D.	8	4
Postdoc	6	4
Total	21	11

Time consuming!
+ I really liked to write
detailed exercise sheets

Teaching: back as a student

Teaching assistant:

Degree	# times as TA	# distinct courses
B.Sc.	2	2
M.Sc.	5	3
Ph.D.	8	4
Postdoc	6	4
Total	21	11

Prize for best TA of U. Montreal

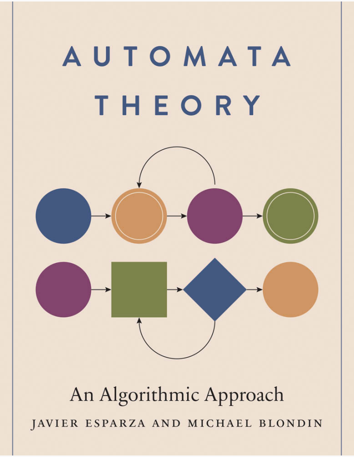
Teaching: back as a student

Teaching assistant:

Degree	# times as TA	# distinct courses
B.Sc.	2	2
M.Sc.	5	3
Ph.D.	8	4
Postdoc	6	4
Total	21	11

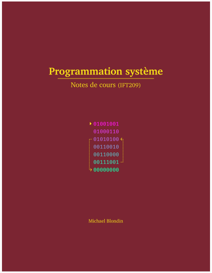
Not needed to get a position!
(but a bit of experience won't hurt)

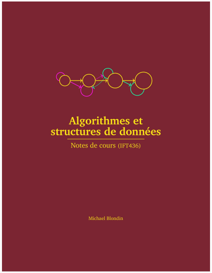
Teaching: now

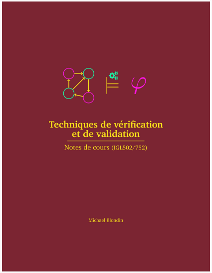


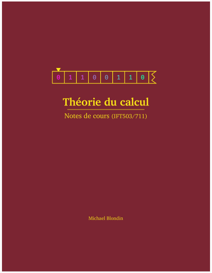
MIT Press, 2023

Teaching: now

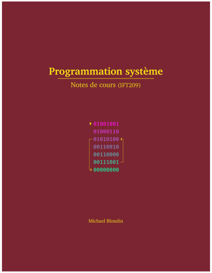


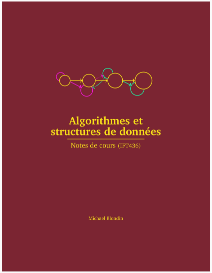


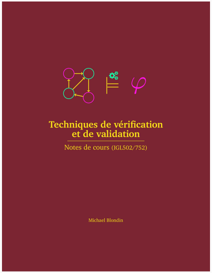


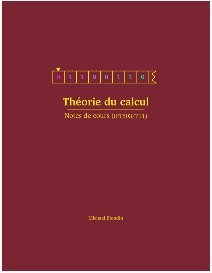


Teaching: now



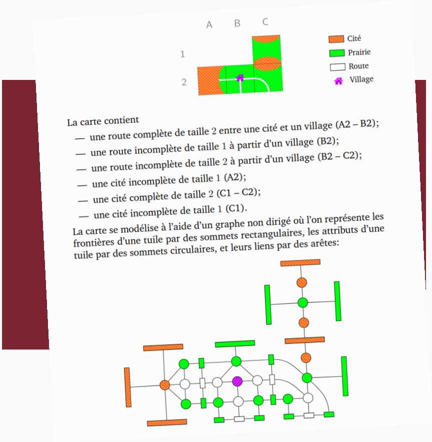






≥ 764 pages, 336 exercises, 275 sol.
100 code implem. with visual demo.
77 videos, 14 quizzes, etc.

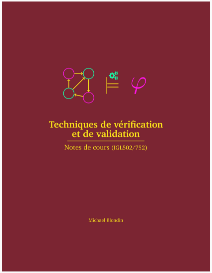
Teaching: now

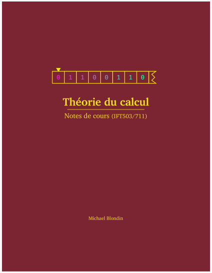


La carte contient

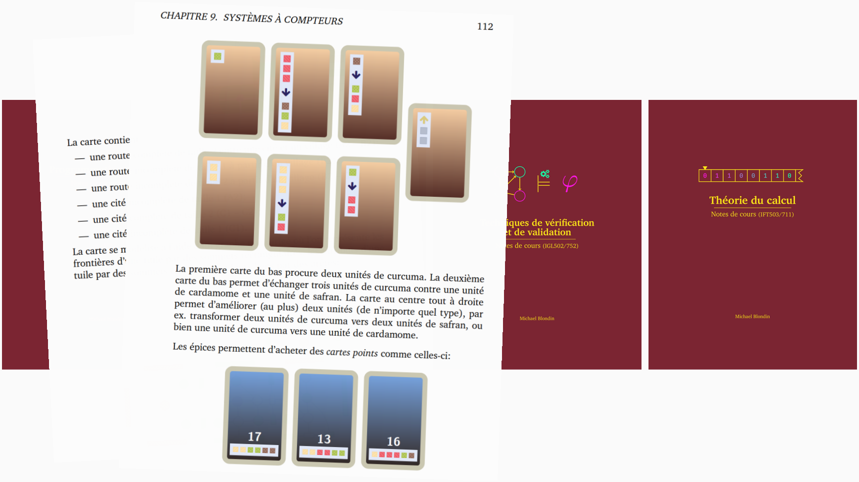
— une route complète de taille 2 entre une cité et un village (A2 – B2);
— une route incomplète de taille 1 à partir d'un village (B2);
— une route incomplète de taille 2 à partir d'un village (B2 – C2);
— une cité incomplète de taille 1 (A2);
— une cité complète de taille 2 (C1 – C2);
— une cité incomplète de taille 1 (C1).

La carte se modélise à l'aide d'un graphe non dirigé où l'on représente les frontières d'une tuile par des sommets rectangulaires, les attributs d'une tuile par des sommets circulaires, et leurs liens par des arêtes:

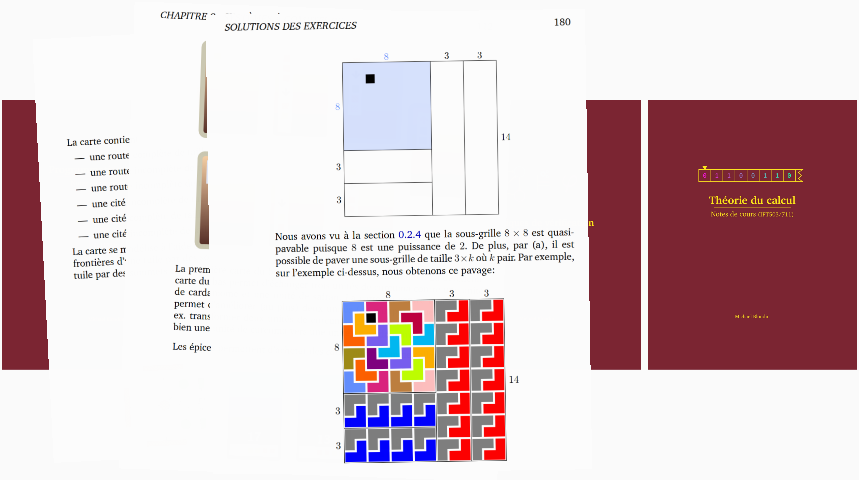




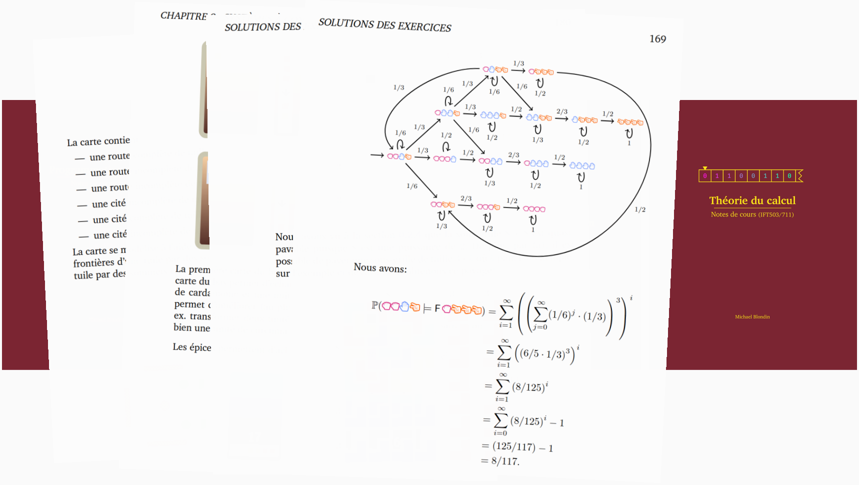
Teaching: now



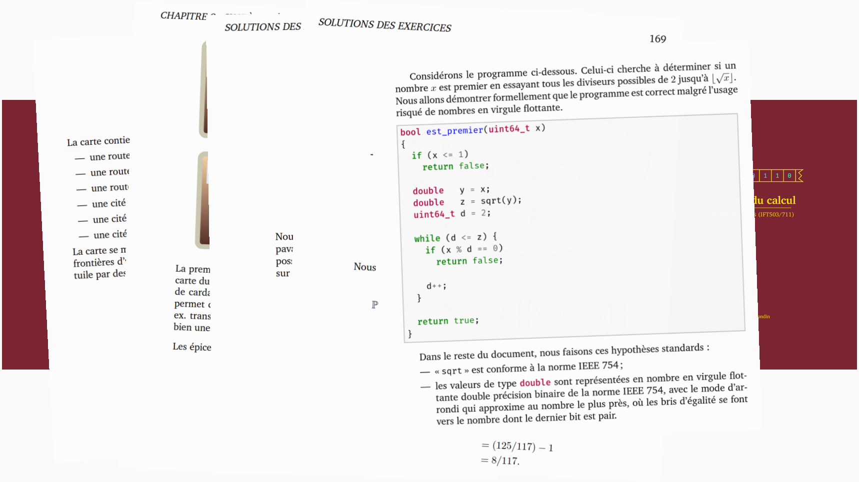
Teaching: now



Teaching: now



Teaching: now



SOLUTIONS DES EXERCICES 169

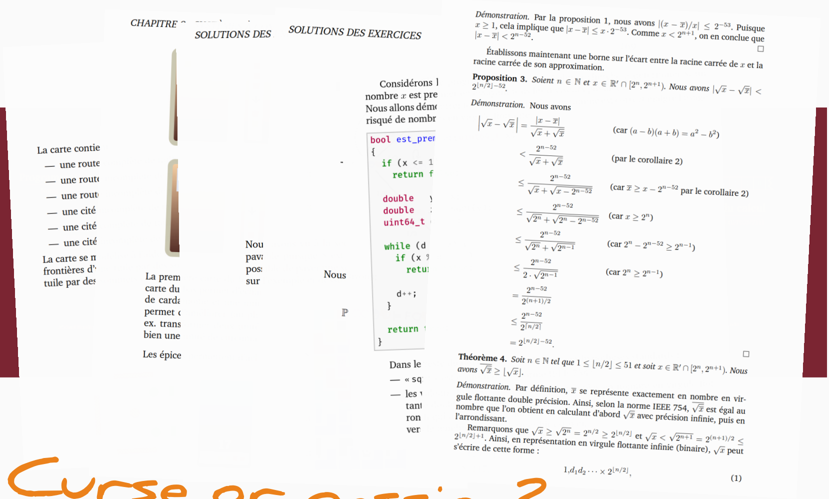
Considérons le programme ci-dessous. Celui-ci cherche à déterminer si un nombre x est premier en essayant tous les diviseurs possibles de 2 jusqu'à $\lfloor\sqrt{x}\rfloor$. Nous allons démontrer formellement que le programme est correct malgré l'usage risqué de nombres en virgule flottante.

```
bool est_premier(uint64_t x)
{
  if (x <= 1)
    return false;

  double   y = x;
  double   z = sqrt(y);
  uint64_t d = 2;

  while (d <= z) {
    if (x % d == 0)
      return false;

    d++;
  }

  return true;
}
```

Dans le reste du document, nous faisons ces hypothèses standards :

— « sqrt » est conforme à la norme IEEE 754 ;

— les valeurs de type double sont représentées en nombre en virgule flottante double précision binaire de la norme IEEE 754, avec le mode d'arrondi qui approxime au nombre le plus près, où les bris d'égalité se font vers le nombre dont le dernier bit est pair.


Démonstration. Par la proposition 1, nous avons $|(x-\overline{x})/x| \leq 2^{-53}$. Puisque $x \geq 1$, cela implique que $|x-\overline{x}| \leq x \cdot 2^{-53}$. Comme $x < 2^{n+1}$, on en conclue que $|x-\overline{x}| < 2^{n-52}$. □

Établissons maintenant une borne sur l'écart entre la racine carrée de x et la racine carrée de son approximation.

Proposition 3. *Soient $n \in \mathbb{N}$ et $x \in \mathbb{R}' \cap [2^n, 2^{n+1})$. Nous avons $|\sqrt{x} - \sqrt{\overline{x}}| < 2^{\lfloor n/2 \rfloor - 52}$.*

Démonstration. Nous avons

$$\begin{aligned}
\left|\sqrt{x} - \sqrt{\overline{x}}\right| &= \frac{|x-\overline{x}|}{\sqrt{x}+\sqrt{\overline{x}}} && \text{(car } (a-b)(a+b) = a^2 - b^2\text{)}\\
&< \frac{2^{n-52}}{\sqrt{x}+\sqrt{\overline{x}}} && \text{(par le corollaire 2)}\\
&\leq \frac{2^{n-52}}{\sqrt{x}+\sqrt{x-2^{n-52}}} && \text{(car } \overline{x} \geq x - 2^{n-52} \text{ par le corollaire 2)}\\
&\leq \frac{2^{n-52}}{\sqrt{2^n}+\sqrt{2^n-2^{n-52}}} && \text{(car } x \geq 2^n\text{)}\\
&\leq \frac{2^{n-52}}{\sqrt{2^n}+\sqrt{2^{n-1}}} && \text{(car } 2^n - 2^{n-52} \geq 2^{n-1}\text{)}\\
&\leq \frac{2^{n-52}}{2\cdot\sqrt{2^{n-1}}} && \text{(car } 2^n \geq 2^{n-1}\text{)}\\
&= \frac{2^{n-52}}{2^{(n+1)/2}}\\
&\leq \frac{2^{n-52}}{2^{\lceil n/2 \rceil}}\\
&= 2^{\lfloor n/2 \rfloor - 52}.
\end{aligned}$$
□

Théorème 4. *Soit $n \in \mathbb{N}$ tel que $1 \leq \lfloor n/2 \rfloor \leq 51$ et soit $x \in \mathbb{R}' \cap [2^n, 2^{n+1})$. Nous avons $\sqrt{\overline{x}} \geq \lfloor \sqrt{x} \rfloor$.*

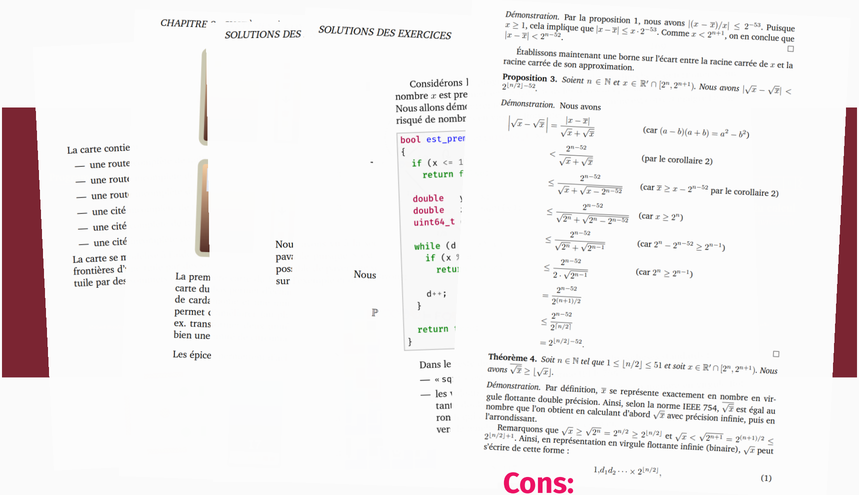
Démonstration. Par définition, $\overline{x}$ se représente exactement en nombre en virgule flottante double précision. Ainsi, selon la norme IEEE 754, $\sqrt{\overline{x}}$ est égal au nombre que l'on obtient en calculant d'abord $\sqrt{\overline{x}}$ avec précision infinie, puis en l'arrondissant.

Remarquons que $\sqrt{x} \geq \sqrt{2^n} = 2^{n/2} \geq 2^{\lfloor n/2 \rfloor}$ et $\sqrt{x} < \sqrt{2^{n+1}} = 2^{(n+1)/2} \leq 2^{\lfloor n/2 \rfloor + 1}$. Ainsi, en représentation en virgule flottante infinie (binaire), $\sqrt{x}$ peut s'écrire de cette forme :

$$1{,}d_1 d_2 \cdots \times 2^{\lfloor n/2 \rfloor}, \quad (1)$$

Curse or passion?

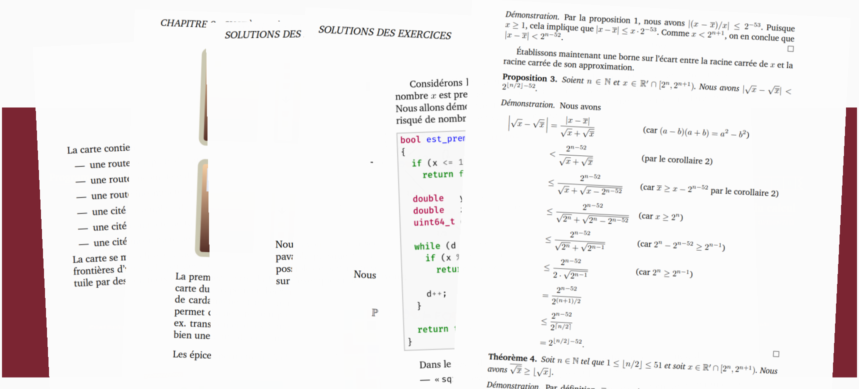
Teaching: now



Cons:

- Might hurt research
- or your personal life

Teaching: now



Pros:

- Academia also about education
- Grad student recruitment
- May spark research ideas
- ***Fun!***

Cons:

- Might hurt research
- or your personal life

Teaching: now

CHAPITRE

SOLU

La carte contie

— une route

— une route

ition 1, nous avons

présente exactement en nombre en vir-
lon la norme IEEE 754, $\sqrt{x}$ est égal au
bord $\sqrt{x}$ avec précision infinie, puis en

$\geq 2^{\lfloor n/2 \rfloor}$ et $\sqrt{x} < \sqrt{2^{n+1}} = 2^{(n+1)/2} \leq$
ule flottante infinie (binaire), $\sqrt{x}$ peut

$1{,}d_1d_2\cdots \times 2^{\lfloor n/2 \rfloor}$, (1)

Remember to have fun! 😊

Thank you!