

# Mon album photo



*Jean Goubault-Larrecq*

# Disclaimer

---

❖ I'm going to talk about me, me, and me again...

« Si tu ne parles pas de toi, qui va en parler? »



Laurent F.



# What Google Images says

+Vous Recherche **Images** Vidéos Maps Actualités Shopping Gmail Plus Connexion

Google

Recherche Environ 637 résultats (0,43 secondes)

Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair.  
[En savoir plus](#) | [Ignorer](#)

Tout  
**Images**  
Maps  
Vidéos  
Actualités  
Shopping  
Plus

Tous les résultats  
Par sujet

Toutes les tailles  
Grandes  
Moyennes  
Icônes  
Supérieure à...  
Égale à...

Toutes les couleurs  
En couleur  
Noir et blanc

# What Google Images says

OK...

The screenshot shows a Google Images search interface. At the top, there's a navigation bar with links for '+Vous Recherche Images Vidéos Maps Actualités Shopping Gmail Plus' and a 'Connexion' button. The search bar contains the text 'jean goubault-larrecq'. Below the search bar, it says 'Recherche Environ 637 résultats (0,43 secondes)'. On the left side, there's a sidebar with navigation options: 'Images', 'Maps', 'Vidéos', 'Actualités', 'Shopping', and 'Plus'. Below these are filters for 'Tous les résultats Par sujet', 'Toutes les tailles' (Grandes, Moyennes, Icônes, Supérieure à..., Égale à...), and 'Toutes les couleurs' (En couleur, Noir et blanc). The main area displays a grid of search results. A purple oval highlights two portrait photos of a man. Other results include the LSU logo, a coin, a book cover titled 'Proof Theory and Automated Deduction', a chessboard, a man speaking at a podium, a man in a white shirt, a man in a suit, a 'Next' button, a sunflower, and a woman's portrait. A yellow box in the top right corner contains the text: 'Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair. En savoir plus | Ignorer'.

# What Google Images says

The screenshot shows the Google Images search interface. At the top, the navigation bar includes '+Vous Recherche Images Vidéos Maps Actualités Shopping Gmail Plus' and 'Connexion'. The Google logo is on the left, and a search bar contains the text 'The right place!' in purple. A yellow notification box on the right states: 'Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair. En savoir plus | Ignorer'. The search results are displayed in a grid. A purple circle with the letters 'LSU' in blue is overlaid on the grid, highlighting a specific image. The grid contains various images related to LSU, including portraits of people, a chess set, a book cover titled 'Proof Theory and Automated Deduction', a coin, a presentation slide, a group photo, a 'Next' button, a sunflower, and a person in a dark shirt. On the left side, there are filters for 'Tous les résultats Par sujet', 'Toutes les tailles' (Grandes, Moyennes, Icônes, Supérieure à..., Égale à...), and 'Toutes les couleurs' (En couleur, Noir et blanc).

# What Google Images says

The image shows a screenshot of the Google Images search interface. At the top, the navigation bar includes '+Vous Recherche Images Vidéos Maps Actualités Shopping Gmail Plus' and 'Connexion'. The search bar contains the text 'jean goubault-larrecq'. Below the search bar, the results are displayed in a grid. A large orange box with the text 'Half-true' is overlaid on the search results. A purple circle highlights a specific image in the grid, which is a seal or logo. The search results include various images: portraits of people, a chessboard, a book cover titled 'Proof Theory and Automated Deduction', a person speaking at a podium, a group of people, a sunflower, and a person in a dark shirt. On the left side, there are filters for 'Tous les résultats', 'Toutes les tailles', and 'Toutes les couleurs'. A 'Next' button is visible on the right side of the grid.



# What Google Images says

+Vous Recherche **Images** Vidéos Maps Actualités Shopping Gmail Plus Connexion

Google

**Recherche** Environ 637 résultats (0,43 secondes)

Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair.  
[En savoir plus](#) | [Ignorer](#)

Tout  
**Images**  
Maps  
Vidéos  
Actualités  
Shopping  
Plus

Tous les résultats  
Par sujet

Toutes les tailles  
Grandes  
Moyennes  
Icônes  
Supérieure à...  
Égale à...

Toutes les couleurs  
En couleur  
Noir et blanc

# What Google Images says

The screenshot shows a Google Images search for 'jean goubault-larrecq'. The search bar contains the text 'jean goubault-larrecq' and a search button. Below the search bar, it says 'Recherche Environ 637 résultats (0,43 secondes)'. The results are displayed in a grid format. On the left, there is a sidebar with navigation options: 'Tout', 'Images', 'Maps', 'Vidéos', 'Actualités', 'Shopping', and 'Plus'. Below these are filters for 'Tous les résultats Par sujet', 'Toutes les tailles' (Grandes, Moyennes, Icônes, Supérieure à..., Égale à...), and 'Toutes les couleurs' (En couleur, Noir et blanc). The search results include several images: two portraits of a man, the 'LSU' logo, two coins, a book cover titled 'Proof Theory and Automated Deduction', a chessboard, a man speaking at a podium, a man in a white shirt, a man in a suit, a woman in a black dress, a 'Next' button, a man in a white shirt, a yellow flower, and a woman in a black shirt. A yellow notification box in the top right corner contains the text: 'Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair. En savoir plus | Ignorer'. There are several purple question marks and lines pointing to various elements in the image.

No, I'm lousy at chess...

(?)

(?)

(?)

4

(?)

(?)

(?)

# What Google Images says

+Vous Recherche **Images** Vidéos Maps Actualités Shopping Gmail Plus Connexion

Google

**Recherche** Environ 637 résultats (0,43 secondes)

Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair.  
[En savoir plus](#) | [Ignorer](#)

Tout  
**Images**  
Maps  
Vidéos  
Actualités  
Shopping  
Plus

Tous les résultats  
Par sujet

Toutes les tailles  
Grandes  
Moyennes  
Icônes  
Supérieure à...  
Égale à...

Toutes les couleurs  
En couleur  
Noir et blanc

# What Google Images says

Une introduction à l'architecture des ordinateurs

Béatrice Bérard, Jean Goubault-Larrecq

LSV/UMR 8643, CNRS, ENS Cachan & INRIA Futurs projet SECSI

61 avenue du président-Wilson, F-94235 Cachan Cedex

[goubault@lsv.ens-cachan.fr](mailto:goubault@lsv.ens-cachan.fr)

Phone: +33-1 47 40 75 68 Fax: +33-1 47 40 75 21

October 6, 2005

Ce document est la retranscription de notes de cours par Béatrice Bérard sur l'architecture des ordinateurs, datant de 1993. Ces notes s'inspiraient du livre d'A. Tannenbaum, "architecture des micro-ordinateurs". Ce cours présente les choses de façon suffisamment simple pour qu'on puisse se faire une idée raisonnablement claire de la façon dont fonctionnent les ordinateurs, depuis les circuits électroniques jusqu'à la micro-programmation.

Les processeurs aujourd'hui sont infiniment plus compliqués, les mémoires vives ne fonctionnent plus à base de flip-flops, soyez prévenus. Mais ceci reste une excellente introduction. Je mettrai quelques commentaires en notes en bas de page au sujet de certains points que je souhaite préciser précédées de "[IGL]"

A few lecture notes

Toutes les couleurs  
En couleur  
Noir et blanc

Centre de recherche Inria Saclay - Île-de-France





# What Google Images says



- HOME
- BROWSE**
- SEARCH
- ABOUT
- RESEARCHERS
- LIBRARIANS
- PUBLISHERS

## Homology, Homotopy, and Applications

My most noteworthy paper?

### On the geometry of intuitionistic $S_4$ proofs

Jean Goubault-Larrecq and Éric Goubault

Source: [Homology Homotopy Appl.](#) Volume 5, Number 2 (2003), 137-209.

#### Abstract

The Curry-Howard correspondence between formulas and types, proofs and programs, proof simplification and program execution, also holds for intuitionistic modal logic  $S_4$ . It turns out that the  $S_4$  modalities translate as a monoidal comonad on the space of proofs, giving rise to a canonical

Supérieure à...  
Égale à...

Toutes les couleurs  
En couleur  
Noir et blanc



Centre de recherche Inria Saclay - Île-de-France



Next

Connexion

Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair.

[En savoir plus](#) | [Ignorer](#)

# What Google Images says

+Vous Recherche **Images** Vidéos Maps Actualités Shopping Gmail Plus Connexion

Google

Recherche Environ 637 résultats (0,43 secondes)

Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair.  
[En savoir plus](#) | [Ignorer](#)

Tout  
**Images**  
Maps  
Vidéos  
Actualités  
Shopping  
Plus

Tous les résultats  
Par sujet

Toutes les tailles  
Grandes  
Moyennes  
Icônes  
Supérieure à...  
Égale à...

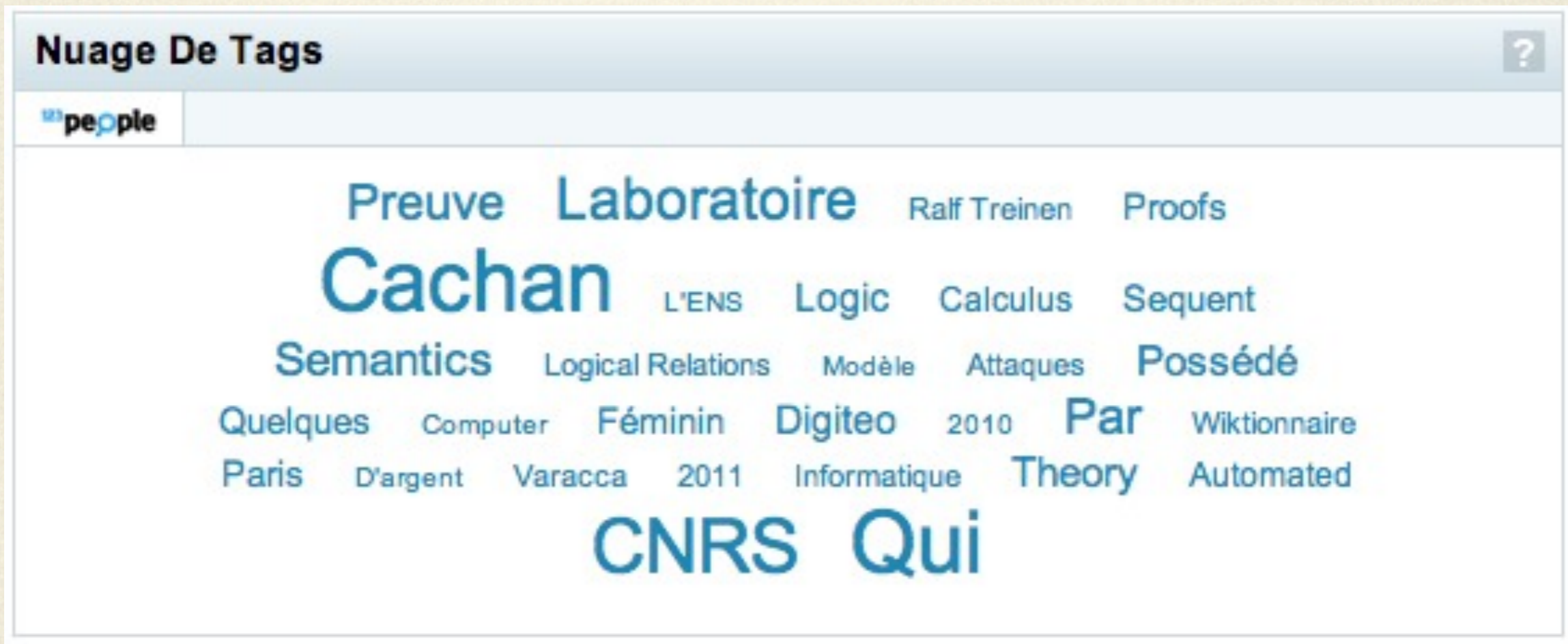
Toutes les couleurs  
En couleur  
Noir et blanc

# What Google Images says

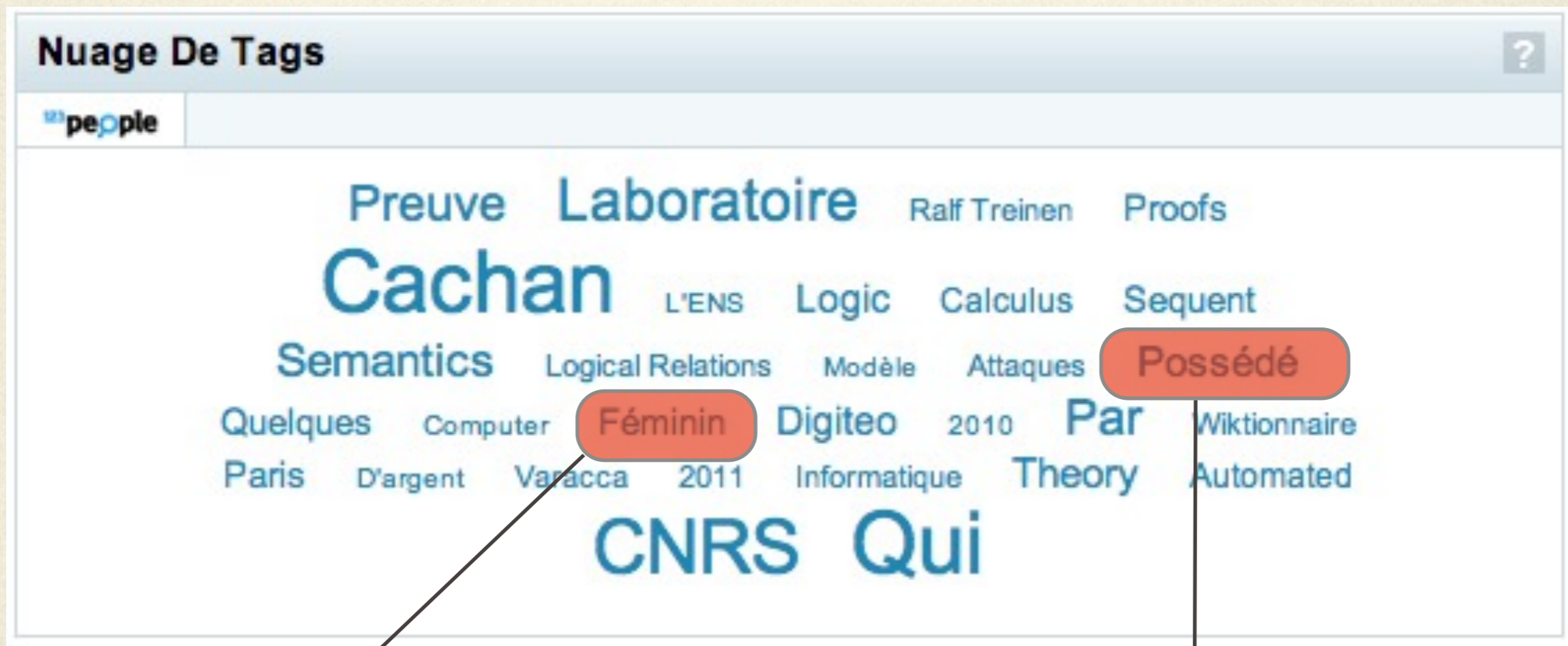
The image shows a screenshot of the Google Images search interface. At the top, there is a navigation bar with links for '+Vous', 'Recherche', 'Images', 'Vidéos', 'Maps', 'Actualités', 'Shopping', 'Gmail', and 'Plus'. The search bar contains the text 'jean goubault-larrecq'. Below the search bar, the results are displayed in a grid. A large purple text box is overlaid on the center of the page, containing the text: 'Funny, I did not know I had a 123 People account'. The search results include a portrait of a man, a chessboard, a book cover titled 'Proof Theory and Automated Deduction', a man speaking at a podium, a man in a white shirt, a blue-tinted image of a person, a man in a suit, a man in a white shirt, a group of people, a book cover, a blue 'Next' button, a man in a white shirt, a yellow flower, a document, and a woman's portrait. On the left side, there are filters for 'Recherche', 'Tout', 'Images', 'Maps', 'Vidéos', 'Actualités', 'Shopping', 'Plus', 'Tous les résultats', 'Par sujet', 'Toutes les tailles', 'Grandes', 'Moyennes', 'Icônes', 'Supérieure à...', 'Égale à...', 'Toutes les couleurs', 'En couleur', and 'Noir et blanc'. On the right side, there is a yellow box with text: 'Nos règles de confidentialité et conditions d'utilisation évoluent. En plus concis et plus clair. En savoir plus | Ignorer'.

# 123 People

.....



# 123 People



?

I would like something to explain this to me...

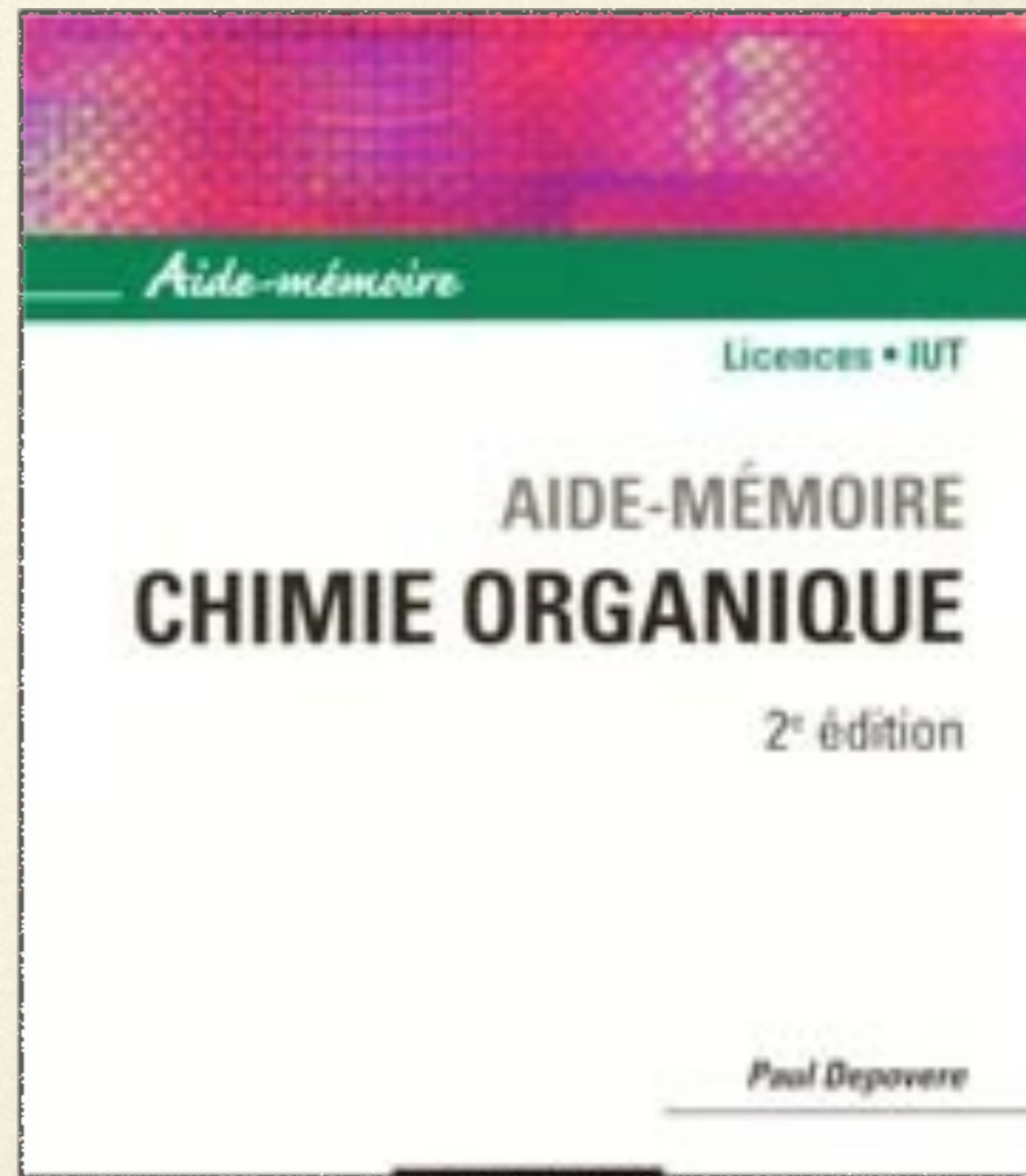
# The real me

- ❖ At age 11, my math teacher made me discover computers, and **programming**



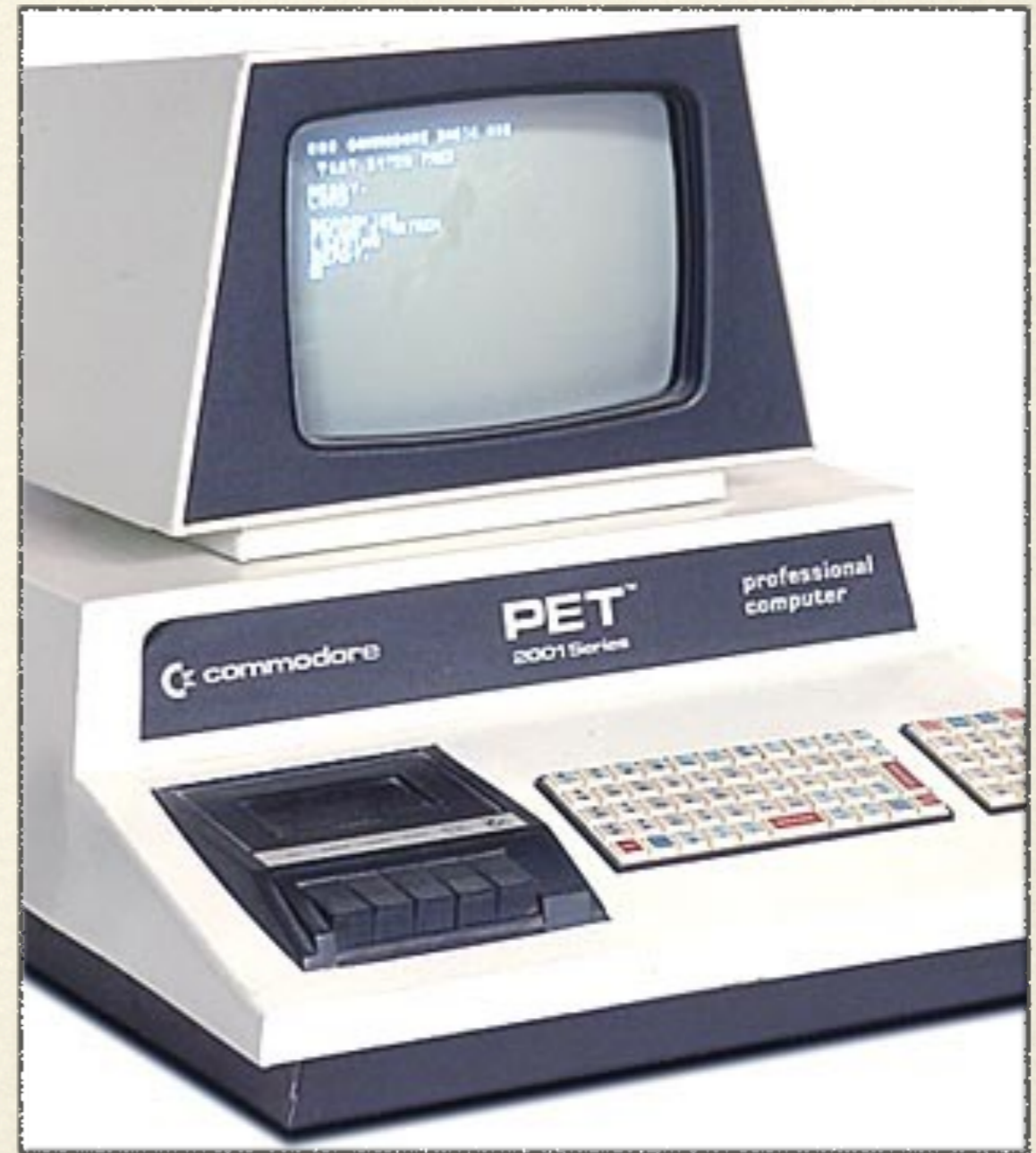
# The real me

- ❖ At 13, I was interested in:
- ❖ **chemistry**
- ❖ mechanical engineering
- ❖ ancient greek
- ~~❖ not computer science~~



# The real me

- ❖ At age 15, I rediscovered **computers**, thanks to my parents



# The real me

At age 20 (at X), I was undecided:  
mathematics, physics,  
**computer science?**



Some of my computers in 1985

# The real me

At age 20 (at X), I was undecided:  
mathematics, physics,  
**computer science?**

**Minus:** every computer scientist (or geek) I knew was more or less nuts



Some of my computers in 1985

# The real me

At age 20 (at X), I was undecided:  
mathematics, physics,  
**computer science?**

**Minus:** every computer scientist (or geek) I knew was more or less nuts

**Plus:** anyway, I was spending 100% of my leisure time programming

(being paid for what I would do anyway: a dream?)



Some of my computers in 1985

# The real me

❖ At age 46, one might say I am still undecided

(although I have given up on physics)

❖ I still am sort of a **geek**

The screenshot shows a code editor with two windows. The top window, titled 'bcic' and 'clause.ml', contains ML code. A green vertical bar highlights a list of files: 'queue.mlx heap.m', 'mlx fixbool.mlx', 'intersym\_lex.mlx', 'x gclause.mlx mo', 'mlx model.mlx tp', 'clauses.mla plau', and '-progress -all'. The code below includes a function definition for 'resolve\_P\_f' and a public function 'bcEval0'. The bottom window, titled 'Fabrice - watch - 71x21', shows the output of a program, displaying various statistics such as 'derived clauses: 798170', 'subsumed clauses [forward]: 635056', and 'Total generated (derived+subsumed forward+frozen): 2893169'.

```
bcic clause.ml
7
4009
H1.1
queue.mlx heap.m
mlx fixbool.mlx
intersym_lex.mlx
x gclause.mlx mo
mlx model.mlx tp
clauses.mla plau
-progress -all

clause.ml 89% (4212,78) (HimML)
clause.ml 1 bcic

#ifdef HAS_STDARG
PUBLIC ExprPtr bcEval0(IN CONST byteCodeChunk *bc,
                      IN CONST ExprPtr env, ...)

Fabrice - watch - 71x21
Sat Feb 4 20:40:03 2012
ary 2,0s: ../hlmon hl.pgr
derived clauses: 798170 (50875 splitting defs, 870 ne facts)
alt with, not subsumed: 163114
bsumed clauses [forward] : 635056 (of which 558016 automata clauses)
bsumed clauses [backward]: 9233 (of which 9176 automata clauses)
rozen clauses: 1459943, awaken: 17273.
tersection predicates: 0.
lly defined predicate symbols: 510
Backward subsumed clauses because of fully defined symbols: 0
Subsumed parent clauses because of fully defined symbols: 3535
revisions: 178
rt simplifications: 725
ound facts: 0
ned clauses: 0
tomata clauses: 81962
Total generated (derived+subsumed forward+frozen): 2893169
Total subsumed (forward+backward+fully defined): 647824
```

# The real me

❖ At age 46, one might say I am still undecided

(although I have given up on physics)

❖ I still am sort of a **geek**

HimML, my own version of the ML language

h1, an efficient prover / tree automata library

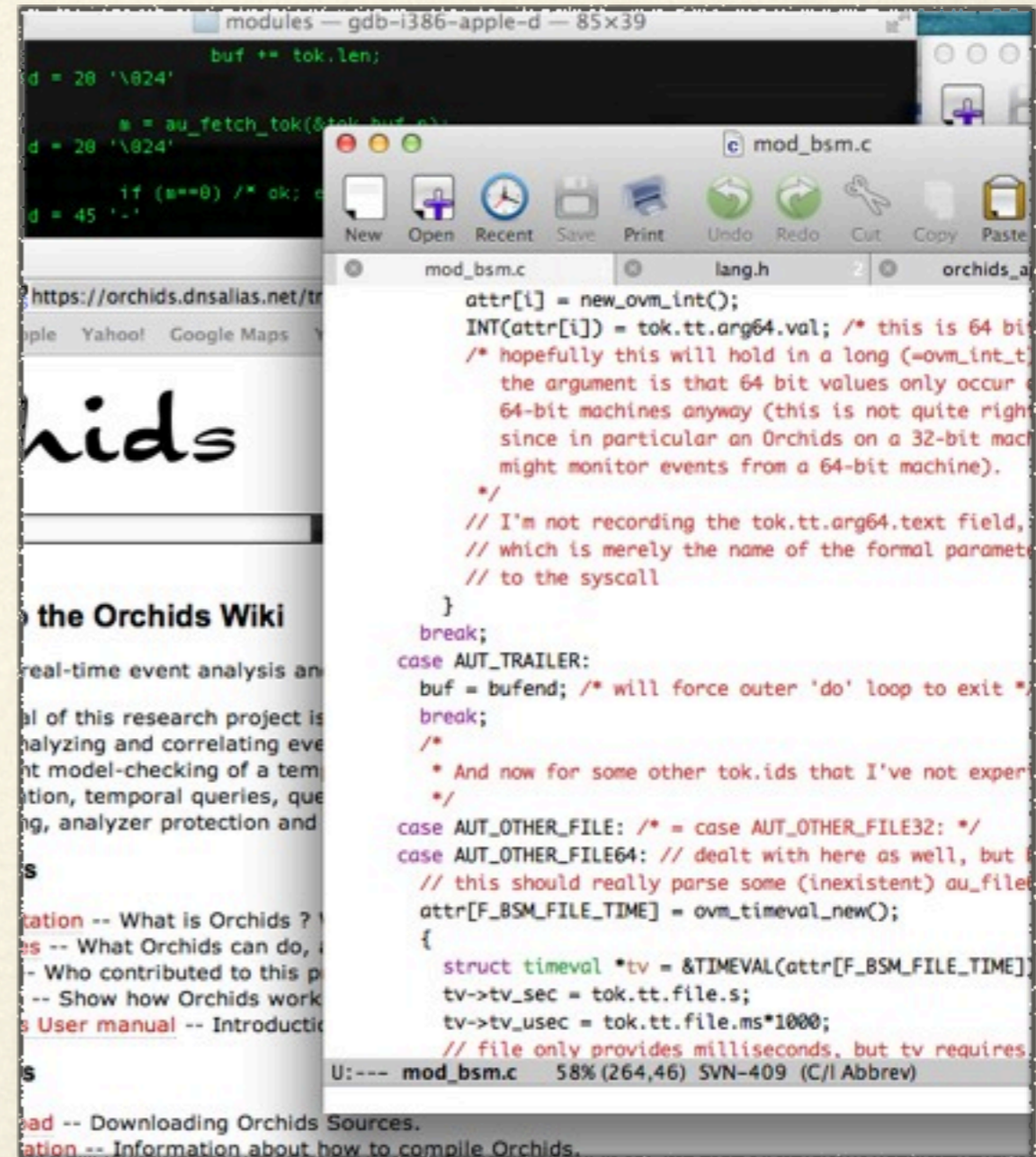
```
bc.c | clause.ml
7
4009
H1.1
queue.mlx heap.m
mlx fixbool.mlx
ntersym_lex.mlx
x gclause.mlx mo
mlx model.mlx tp
clauses.mla plau
-progress -all
clause.ml 89% (4212,78) (HimML)
clause.ml | bc.c
#IFDEF HAS_STDARG
PUBLIC ExprPtr bcEval0(IN CONST byteCodeChunk *bc,
IN CONST ExprPtr env, ...)
Fabrice -- watch -- 71x21
y 2,0s: ../hlmon h1.pgr Sat Feb 4 20:40:03 2012
rived clauses: 798170 (50875 splitting defs, 870 ne facts)
alt with, not subsumed: 163114
bsumed clauses [forward] : 635056 (of which 558016 automata clauses)
bsumed clauses [backward]: 9233 (of which 9176 automata clauses)
rozen clauses: 1459943, awoken: 17273.
tersection predicates: 0.
lly defined predicate symbols: 510
Backward subsumed clauses because of fully defined symbols: 0
Subsumed parent clauses because of fully defined symbols: 3535
revious: 178
rt simplifications: 725
und facts: 0
ned clauses: 0
tomata clauses: 81962
Total generated (derived+subsumed forward+frozen): 2893169
Total subsumed (forward+backward+fully defined): 647824
```

# The real me

❖ At age 46, one might say I am still undecided

(although I have given up on physics)

❖ I still am sort of a **geek**



# The real me

❖ At age 46, one might say I am still undecided  
(although I have given up on physics)

❖ I still am sort of a **geek**

**orchids**

our intrusion detection system,  
with **Julien Olivain**



A screenshot of a computer screen. The top part shows a terminal window with green text on a black background, displaying code snippets like 'buf += tok.len;', 'd = 20 '\024'', and 'if (m==0) /\* ok;'. Below the terminal is a browser window showing the 'orchids' website. The website has a white background with the word 'orchids' in a large, black, cursive font. Below the logo, there is a section titled 'the Orchids Wiki' with some text. On the right side of the screen, there is a code editor window titled 'mod\_bsm.c' showing C code. The code includes comments and function calls like 'new\_ovm\_int()', 'INT(attr[i]) = tok.tt.arg64.val;', and 'ovm\_timeval\_new()'. At the bottom of the code editor, there is a status bar showing 'U:--- mod\_bsm.c 58% (264,46) SVN-409 (C/I Abbrev)'. An arrow points from the 'orchids' logo in the text above to the website in the screenshot.

# The real me

❖ At age 46, one might say I am still undecided

(although I have given up on physics)

❖ I still am sort of a geek

❖ I'm a computer scientist working in **security**

(here, intrusion detection—related to previous code snippet)

## A Smell of ORCHIDS

Jean Goubault-Larrecq<sup>1</sup> and Julien Olivain<sup>1,2</sup>

<sup>1</sup> LSV, ENS Cachan, CNRS, INRIA  
LSV, 61 avenue du président Wilson, F-94235 Cachan Cedex  
{olivain,goubault}@lsv.ens-cachan.fr

<sup>2</sup> Above Security, Suite 203  
1919 Lionel-Bertrand boulevard, Boisbriand, Québec, Canada, J7H 1N8  
julien.olivain@abovesecurity.com

**Abstract.** ORCHIDS is an intrusion detection tool based on techniques for fast, on-line model-checking. ORCHIDS detects complex, correlated strands of events with very low overhead in practice, although its detection algorithm has worst-case exponential time complexity.

The purpose of this paper is twofold. First, we explain the salient features of the basic model-checking algorithm in an intuitive way, as a form of dynamically-spawned monitors. One distinctive feature of the ORCHIDS algorithm is that fresh monitors need to be spawned at a possibly alarming rate.

The second goal of this paper is the efficiency of the procedure, using abstract monitors. This includes monitoring monitors that are subsumed by the so-called shortest run criterion. The ORCHIDS algorithm maintains its monitoring operation is effected with no annoying bug in its core algorithm.



### 1 Introduction

It is a *lieu commun* that the security of computers is more challenged by new threats. Viruses and worms have been used to infect computers since the early 1980s.

# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security using **automated deduction**

## Finite models for formal security proofs

Jean Goubault-Larrecq \*

*LSV, ENS Cachan, CNRS, INRIA, Cachan, France*

First-order logic models of security for cryptographic protocols, based on variants of the Dolev–Yao model, are now well-established tools. Given that we have checked a given security protocol  $\pi$  using a given first-order prover, how hard is it to extract a formally checkable proof of it, as required in, e.g., common criteria at the highest evaluation level (EAL7)? We demonstrate that this is surprisingly hard in the general case: the problem is non-recursive. Nonetheless, we show that we can instead extract finite models  $\mathcal{M}$  from a set  $S$  of clauses representing  $\pi$ , automatically, and give two ways of doing so. We then define a model-checker testing  $\mathcal{M} \models S$ , and show how we can instrument it to output a formally checkable proof, e.g., in Coq. Experience on a number of protocols shows that this is practical, and that even complex (secure) protocols modulo equational theories have small finite models, making our approach suitable.

Keywords: Dolev–Yao model, formal security proof, finite model, tree automaton,  $\mathcal{H}_1$ , inductionless induction

### 1. Introduction

So far, automated verification of cryptographic protocols in models in the style of Dolev and Yao [36] has been considered under a variety of angles: (un)decidability results [37,49], practical decision procedures [6,65,84], extension to security properties other than secrecy and authentication (e.g., [20]), to protocols requiring equational theories, to soundness with respect to computational models (e.g., [56] for the latter two points), in particular.

We consider yet another angle: producing formally checkable proofs of security,

# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security using **automated deduction**

## Finite models for formal security proofs

Jean Goubault-Larrecq \*

*LSV, ENS Cachan, CNRS, INRIA, Cachan, France*

First-order logic models of security for cryptographic protocols, based on variants of the Dolev-Yao model, are now well-established tools. Given that we have checked a given security protocol  $\pi$  using a given first-order prover, how hard is it to extract a formally checkable proof of it, as required in, e.g., common criteria at the highest evaluation level (EAL7)? We demonstrate that this is surprisingly hard in the general case: the problem is non-recursive. Nonetheless, we show that we can instead extract finite models  $\mathcal{M}$  from a set  $S$  of clauses representing  $\pi$ , automatically, and give two ways of doing so. We then define a model-checker testing  $\mathcal{M} \models S$ , and show how we can instrument it to output a formally checkable proof, e.g., in Coq. Experience on a number of protocols shows that this is practical, and that even complex (secure) protocols modulo equational theories have small finite models, making our approach suitable.

Keywords: Dolev-Yao model, formal security proof, finite model, tree automaton,  $\mathcal{H}_1$ , inductionless induction

### 1. Introduction

So far, automated verification of cryptographic protocols in models in the style of Dolev and Yao [36] has been considered under a variety of angles: (un)decidability results [37,49], practical decision procedures [6,65,84], extension to security properties other than secrecy and authentication (e.g., [20]), to protocols requiring equational theories, to soundness with respect to computational models (e.g., [56] for the latter two points), in particular.

We consider yet another angle: producing formally checkable proofs of security,

# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, using automated deduction and **static analysis**

## Cryptographic Protocol Analysis on Real C Code<sup>★</sup>

Jean Goubault-Larrecq<sup>a,\*</sup> Fabrice Parrennes<sup>b,1</sup>

<sup>a</sup>LSV/UMR CNRS & ENS Cachan, INRIA Futurs projet SECSI  
61, av. du président-Wilson, 94235 Cachan Cedex, France

<sup>b</sup>RATP, EST/ISF/QS LAC VC42  
40 bis Roger Salengro, F-94734 Fontenay-sous-Bois, France

### Abstract

Implementations of cryptographic protocols often contain bugs affecting security, which cannot be detected by standard tools (e.g. or TLS). We describe how cryptographic protocol clause sets can be applied to detect such bugs statically. This involves integrating the analysis of which messages an external attacker can see with concrete run-time data with abstract interpretation. We make use of so-called *trust assertions* to define a decidable class  $\mathcal{H}_1$ , which can be used to check the secrecy properties, and to detect security vulnerabilities.

*Key words:* cryptographic protocols, static security analysis, trust assertion,  $\mathcal{H}_1$

*PACS:*



# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, using automated deduction and **static analysis**

## Cryptographic Protocol Analysis on Real C Code<sup>★</sup>

Jean Goubault-Larrecq<sup>a,\*</sup> Fabrice Parrennes<sup>b,1</sup>

<sup>a</sup>LSV/UMR CNRS & ENS Cachan, INRIA Futurs projet SECSI  
61, av. du président-Wilson, 94235 Cachan Cedex, France

<sup>b</sup>RATP, EST/ISF/QS LAC VC42  
40 bis Roger Salengro, F-94734 Fontenay-sous-Bois, France

### Abstract

Implementations of cryptographic protocols often affect security, which cannot be verified (e.g., for TLS). We describe how cryptographic protocol clause sets can be applied to detect security vulnerabilities statically. This involves integrating the analysis of which messages an external attacker can see with concrete run-time data with abstract interpretation. We make use of so-called *trust assertions* to define a decidable class  $\mathcal{H}_1$ , which can be used to verify secrecy properties, and to detect security vulnerabilities.

**Key words:** cryptographic protocols, static security analysis, trust assertion,  $\mathcal{H}_1$

**PACS:**



# The real me

❖ At age 46, one might say I am still undecided

(although I have given up on physics)

❖ I still am sort of a geek

❖ I'm a computer scientist working in security and automated deduction, and **tree automata**

## Alternating Two-Way AC-Tree Automata\*

Jean Goubault-Larrecq Kumar Neeraj Verma

LSV/CNRS UMR 8643 & INRIA projet Futurs & ENS Cachan  
61 avenue du président-Wilson, F-94235 Cachan Cedex  
{goubault|verma}@lsv.ens-cachan.fr  
Phone: +33-1 47 40 75 68 Fax: +33-1 47 40 24 64

**Abstract.** We explore the notion of alternating two-way tree automata modulo the theory of finitely many associative-commutative (AC) symbols. This was prompted by questions arising in the theory of alternating two-way tree automata, which is fundamental in modeling group key distribution protocols, where the emptiness problem is fundamental. This also has independent interest in the theory of alternating two-way tree automata, or of alternating two-way tree automata with push clauses, or of alternating two-way tree automata with AC symbol, with only function symbols. The emptiness problem is undecidable in the general case of alternating two-way tree automata with AC symbols, provided push clauses are allowed. To this end, extensive use is made of the natural extension to vector addition machines of alternating two-way tree automata.

### Introduction

Automata and in particular tree automata have been used in hardware verification, in particular in hardware verification of tree automata with various features. In this paper, we study alternating two-way tree automata, where transitions may not just guard terms, but also



# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in **security**, automated deduction, tree automata, **logic**

**Complete Lax Logical Relations for Cryptographic Lambda-Calculus**

Jean Goubault-Larrecq<sup>1</sup> Sławomir Lasota<sup>2</sup> David Nowak<sup>1</sup> Yu Zhang<sup>1</sup>  
<sup>1</sup> LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan  
61, av. du Président Wilson, 94235 Cachan Cedex, France  
{goubault, nowak, zhang}@lsv.ens-cachan.fr  
<sup>2</sup> Institute of Informatics, Warsaw University, ul. Banacha 2, 02-097 Warszawa, Poland

**Abstract**

Properties are profitably expressed through logical relations, and logical relations are used as a powerful tool of technique to establish contextual equivalence in lambda calculi, see e.g. Sumii and Pierce's approach, showing that logical relations, or lax logical relations, should be lax at encryption types, and strict at decryption types. In this paper, as a difficult aspect of fresh name creation, we use logical relations to establish results. Two approaches are used: the first is based on the lambda-calculus with constant names, and Stark's name creation approach, and the second is based on logical relations which are lax at encryption types but strict (non-lax) at decryption types. They are sound and complete for all types.

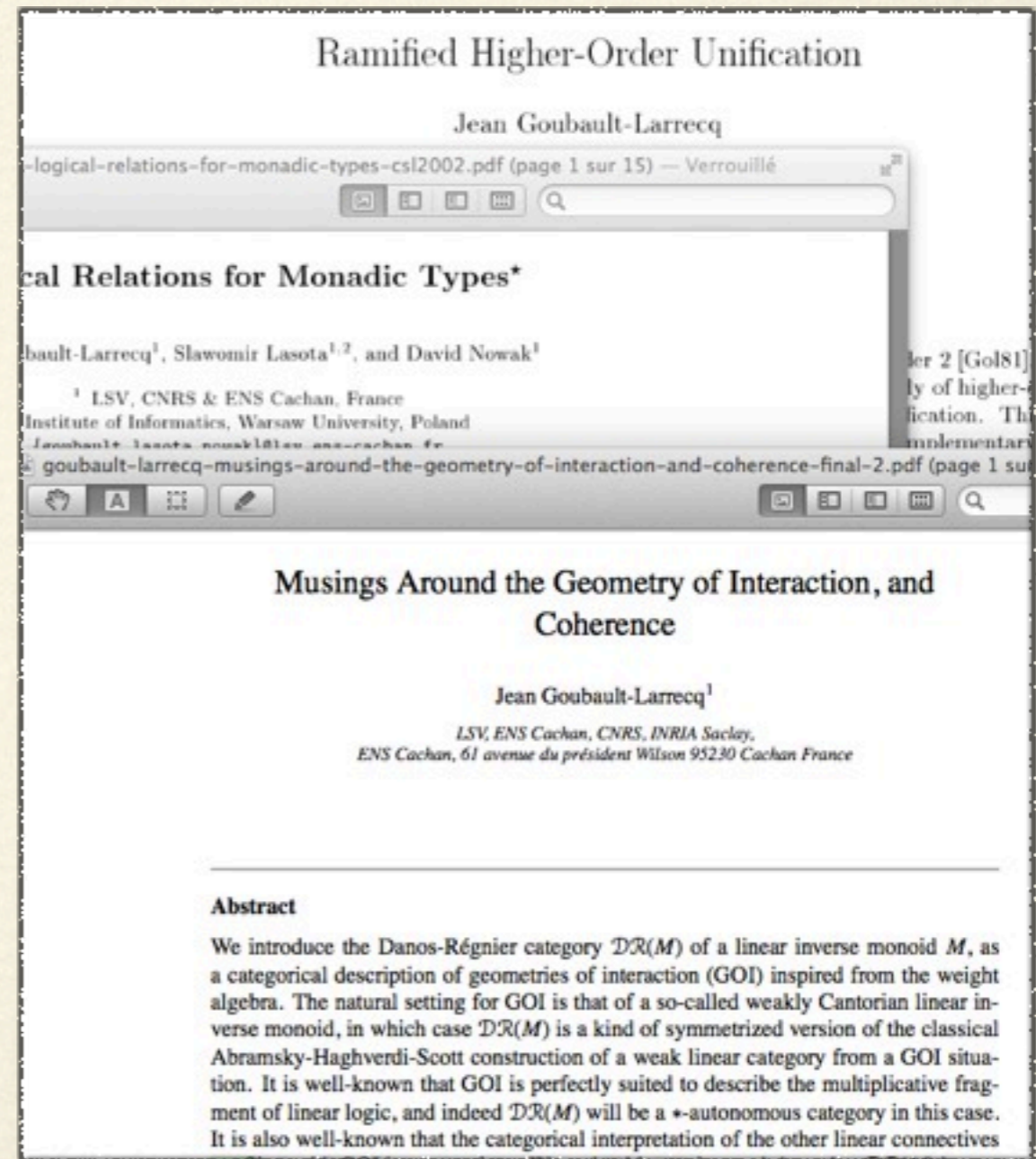
**Introduction**

Somedays many existing models of cryptographic lambda-calculus are based on the approach of Sumii and Pierce, in which complex logical relations as semantic equivalences. We take a new look at lambda-calculus, and gradually deconstruct the existing models. Here we show the power of prelogical relations in showing the soundness of fresh name creation.



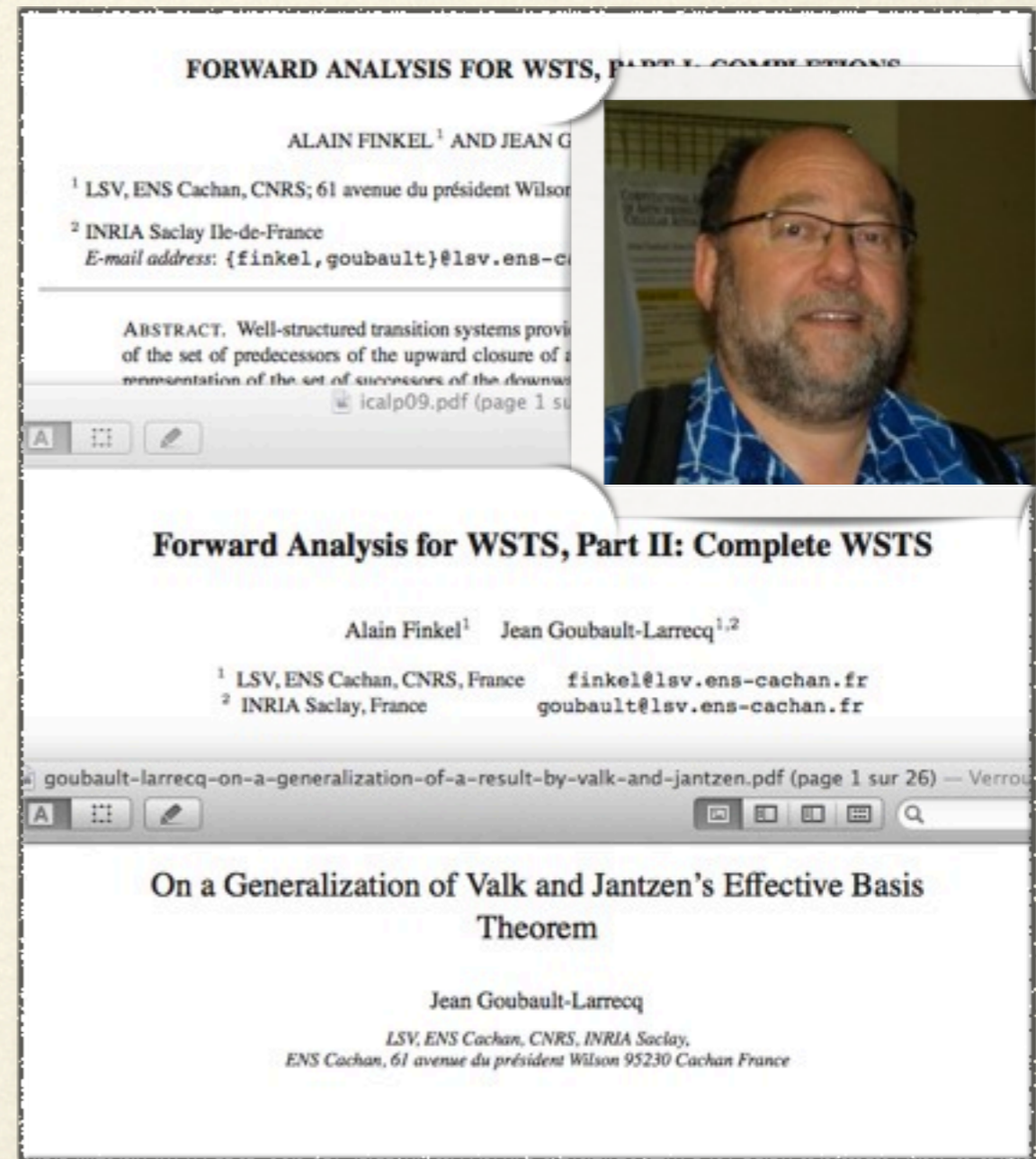
# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, automated deduction, tree automata, **logic**



# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, automated deduction, tree automata, logic, **verification**




# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, automated deduction, tree automata, logic, verification, **semantics**

**Continuous Random Variables**

Jean Goubault-Larrecq\*  
*LSV, ENS Cachan, CNRS, INRIA*

**Abstract**—We introduce the domain of continuous random variables (CRV) over a domain, as an algebraic structure. We show that this domain is a continuous lattice, and he established the remaining properties of the domain-closed category of FS-domains. (page 1 sur 10) — Verro



Daniele Varacca  
*Preuves, Programmes et Systèmes, UMR 7126, CNRS and University Paris Diderot*

**$\omega$ QRB-Domains and the Probabilistic Powerdomain**

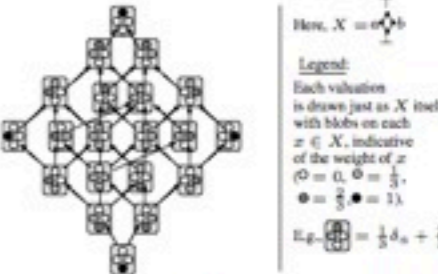
Jean Goubault-Larrecq\*  
\* *Preuves, Programmes et Systèmes, UMR 7126, CNRS and University Paris Diderot*  
† *LSV, ENS Cachan, CNRS, INRIA*

Abstract—Is there any cartesian-closed category of continuous domains that would be closed under Jones and Plotkin's probabilistic powerdomain construction? This is a major open question in the area of denotational semantics of probabilistic programming languages. We relax the question, and look for continuous dcpos instead. We introduce a natural class of quasi-continuous dcpos, the omega-QRB-domains. We show that they form a category omega-QRB with pleasing properties: omega-QRB is closed under the probabilistic powerdomain functor, has all finite products, all bilimits, and is closed under retracts, and even under so-called quasi-retracts. However, omega-QRB is not cartesian closed.

**Keywords**—Quasi-continuous domains, probabilistic powerdomain

I. INTRODUCTION

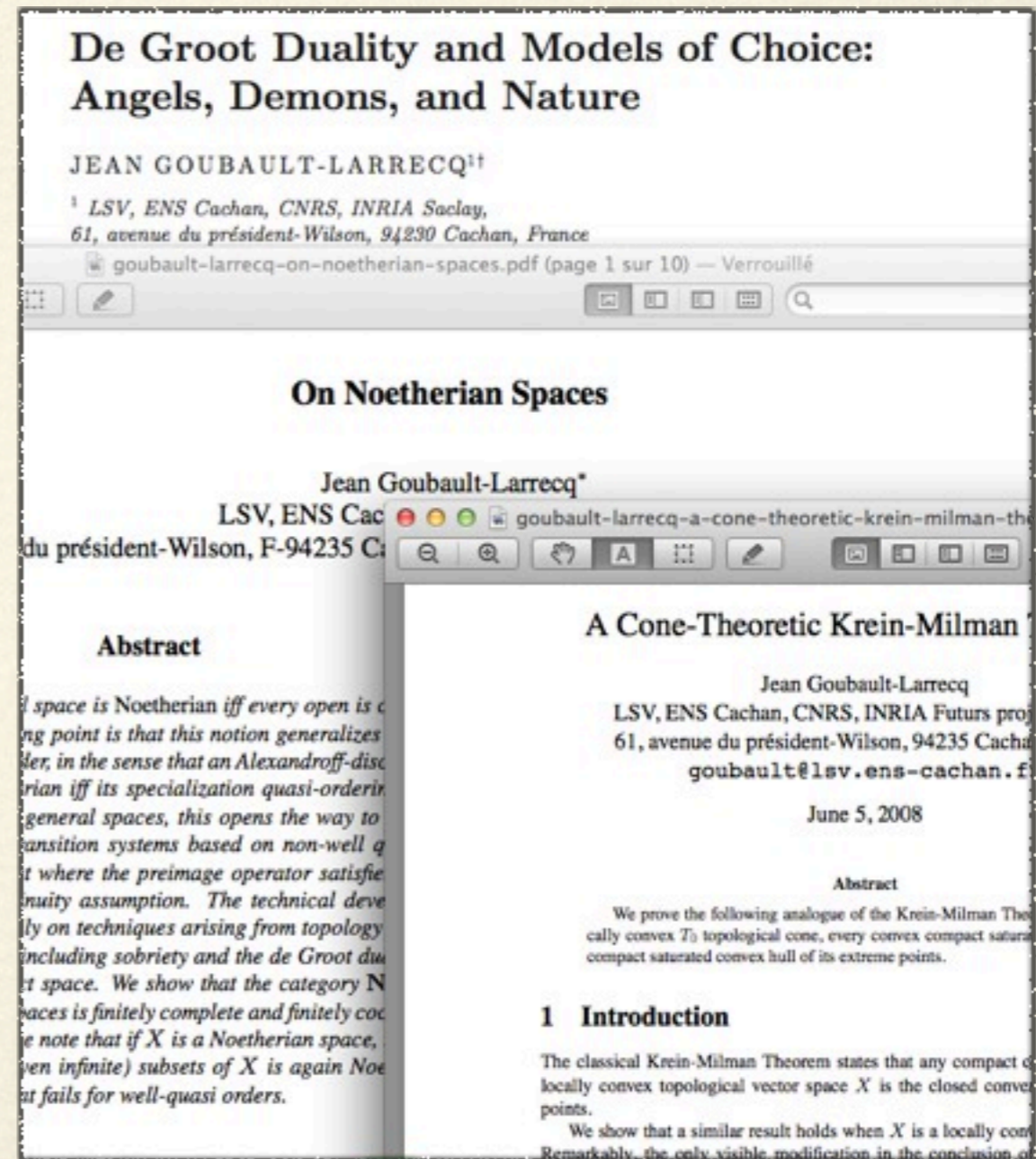
$\mathbf{V}_1(X)$  (resp.  $\mathbf{V}_\infty(X)$ ) the dcpo of all continuous



our contribution to bring some progress toward the question, and at least to understand the structure of  $\mathbf{V}_1(X)$  better. To appreciate this, recall what is

# The real me

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, automated deduction, tree automata, logic, verification, semantics, **topology**



# The real me

---

- ❖ At age 46, one might say I am still undecided  
(although I have given up on physics)
- ❖ I still am sort of a geek
- ❖ I'm a computer scientist working in security, automated deduction, tree automata, logic, verification, semantics, **topology**

## Non-Hausdorff Topology and Domain Theory

Selected Topics in Point-Set Topology

Jean Goubault-Larrecq  
LSV, ENS Cachan, CNRS, INRIA &  
61, avenue du président Wilson, 94230 Cachan, France  
goubault@lsv.ens-cachan.fr



(464 pages, to appear, 2012)

# Thanks!

---

- ❖ to all who have supported me throughout the years
- ❖ starting with **Dominique Bolignano**, to whom I owe much



# Thanks!

---

❖ to all of you who have supported me throughout the years, in particular:

# Thanks!

❖ to all of you who have supported me throughout the years, in particular:

- My students
- My coauthors
- And everybody at LSV.

