Introduction à la logique

François Schwarzentruber
ENS Cachan, Antenne de Bretagne, France

Outline

Un cauchemar

- Albert : Bonjour, j'aimerais un logiciel pour résoudre des sudokus.

	2					4	8	
1			8		2	3		7
				9	3	2	1	
	4	8						1
			6	1	8			
7						9	3	
	1	5	9	4				
4		2	3		6			9
	3	7					5	

Un cauchemar

- Albert : Bonjour, j'aimerais un logiciel pour résoudre des sudokus.

	2					4	8	
1			8		2	3		7
				9	3	2	1	
	4	8						1
			6	1	8			
7						9	3	
	1	5	9	4				
4		2	3		6			9
	3	7					5	

- Informaticien : J'ai travaillé toute la nuit. Le voici.

```
// HelloWin.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"

#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst;                                // current instance
TCHAR szTitle[MAX_LOADSTRING];                  // The title b
TCHAR szWindowClass[MAX_LOADSTRING];                        // The

// Foward declarations of functions included in this code module:
ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK    About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_HELLOWIN, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
```

Un cauchemar

- Albert : Bonjour, j'aimerais un logiciel pour gérer l'emploi du temps de l'ENS.

Un cauchemar

- Albert : Bonjour, j'aimerais un logiciel pour gérer l'emploi du temps de l'ENS.

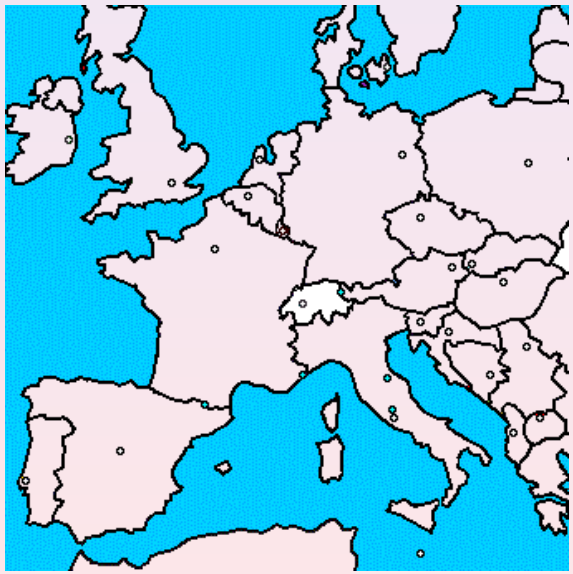
- Informaticien : Nouvelle nuit blanche ! Le voici.

```
// HelloWin.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"

#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst;                                // current instance
TCHAR szTitle[MAX_LOADSTRING];                  // The title b
TCHAR szWindowClass[MAX_LOADSTRING];            // The

// Foward declarations of functions included in this code module:
ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK    About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_HELLOWIN, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
```


4/32

Un cauchemar

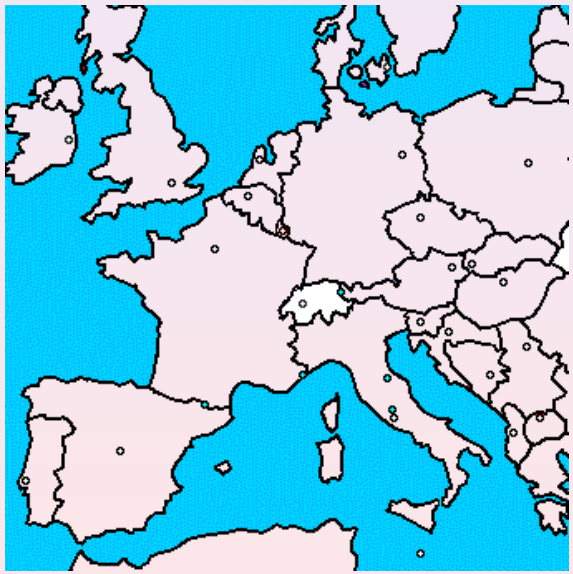
- Albert : Bonjour, j'ai besoin d'un logiciel qui colorie automatiquement mes cartes...



5/32

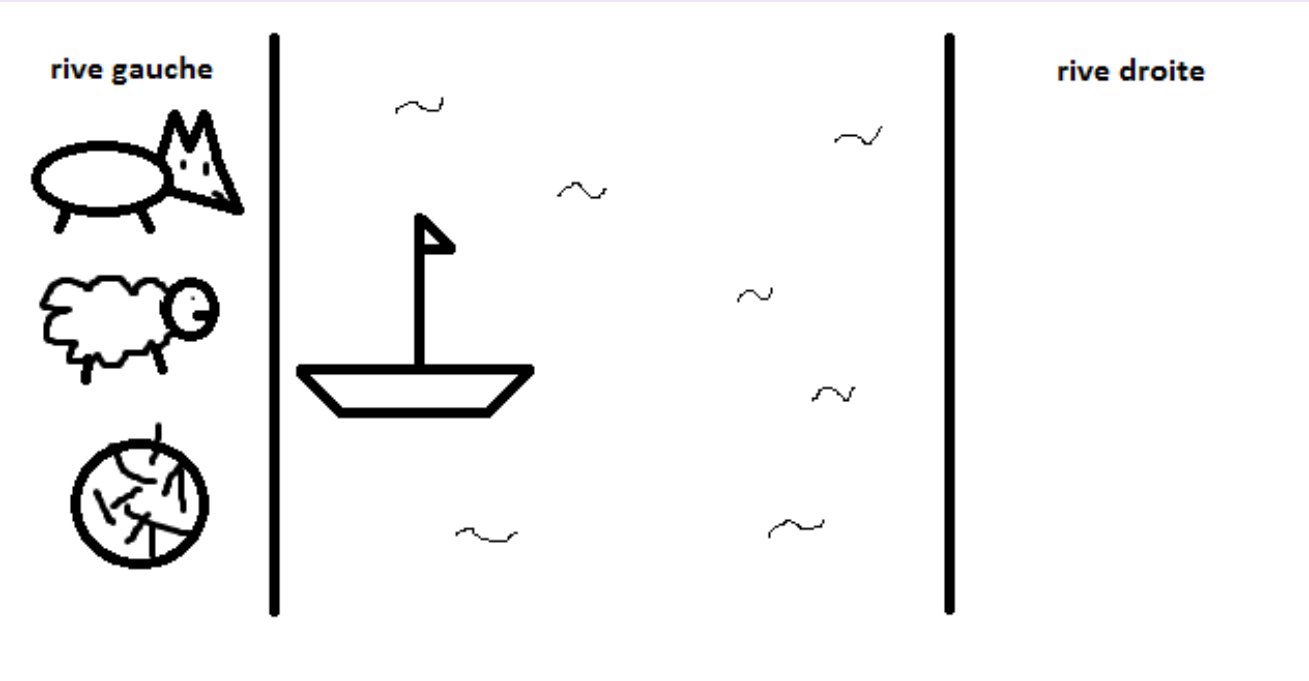
Un cauchemar

- Albert : Bonjour, j'ai besoin d'un logiciel qui colorie automatiquement mes cartes...

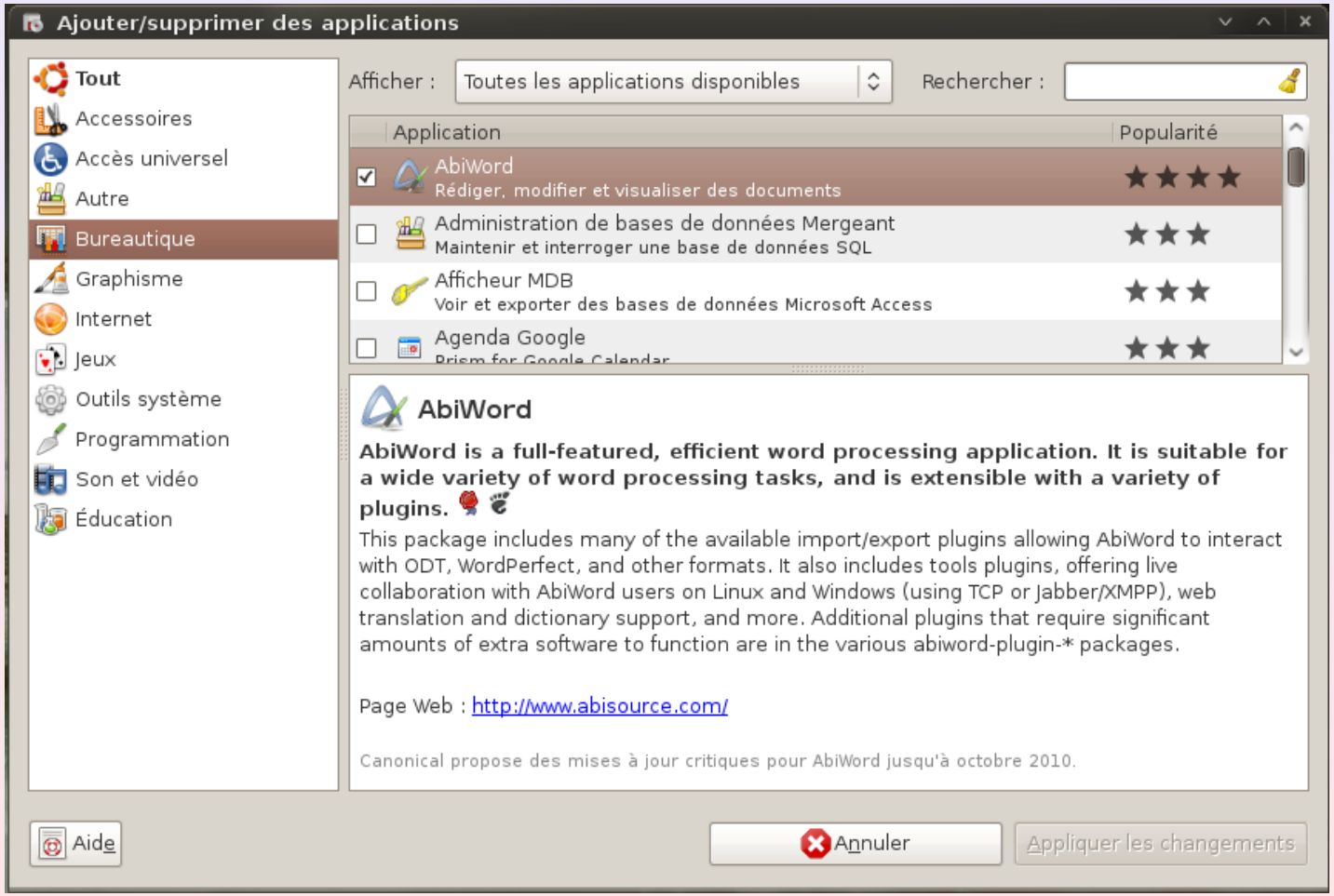


- Informaticien : Désolé, je suis fatigué... je n'en peux plus...

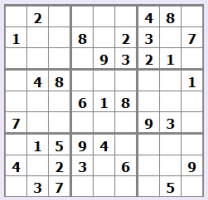
Un cauchemar



Un cauchemar

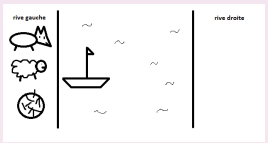


Un cauchemar

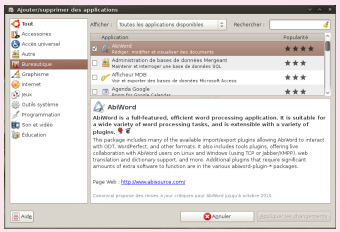


$\longrightarrow$ Solveur de sudokus

$\longrightarrow$ Logiciel de gestion d'emploi du temps



$\longrightarrow$ un solveur automatique de planification



$\longrightarrow$ un solveur automatique de gestion de packages

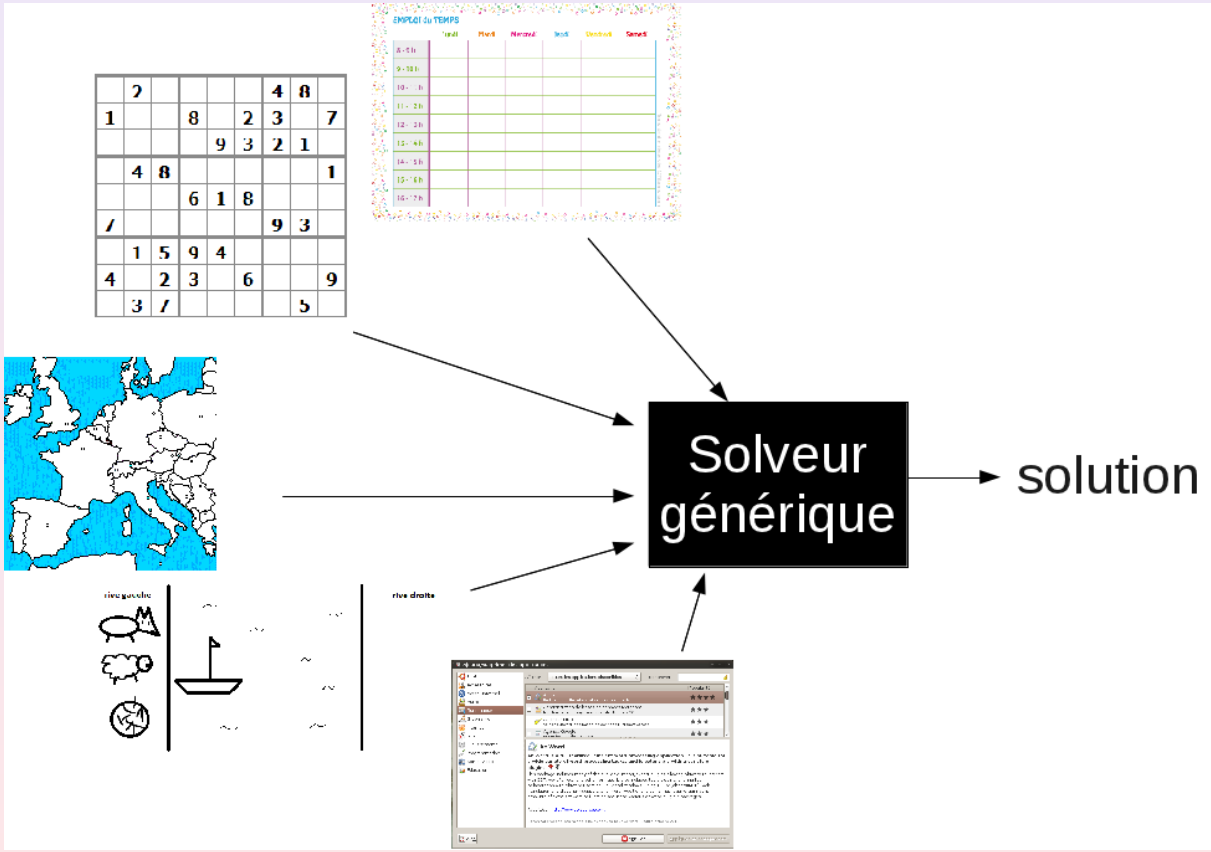
8/32

Outline

Outline

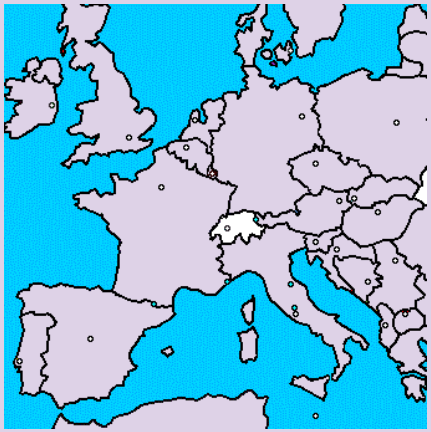
Retour à la réalité

L'informaticien se réveille...



Solveur générique

Exemple



'Tout pays a une et une seule couleur.
Deux pays voisins n'ont pas la même couleur.

→ **Solveur générique** → *solution*

Qualité pour le langage formel utilisé

```
‘Tout pays a une et une seule
couleur.
Deux pays voisins n’ont pas la même
couleur.
```

Balance entre :

- l’expressivité, un langage avec beaucoup de vocabulaire
- la complexité du raisonnement automatique

Outline

La logique propositionnelle

$$(\text{la France est en bleu} \ \textbf{ou}\ \text{la France est en rouge} \ \textbf{ou}\ \text{la France est en vert})$$

$$\textbf{et}\ (\text{l'Allemagne est en bleu} \ \textbf{ou}\ \text{l'Allemagne est en rouge} \ \textbf{ou}\ \text{l'Allemagne est en vert})$$

$$\textbf{et}\ (\text{la France est en bleu} \Rightarrow \textbf{non}\ \text{la France est en rouge})$$

$$\textbf{et}\ (\text{la France est en bleu} \Rightarrow \textbf{non}\ \text{la France est en vert})$$

$$\vdots$$

$$\textbf{et}\ (\text{la France est en bleu} \Rightarrow \textbf{non}\ \text{l'Allemagne est en bleu})$$

$$\textbf{et}\ (\text{la France est en rouge} \Rightarrow \textbf{non}\ \text{l'Allemagne est en rouge})$$

$$\textbf{et}\ (\text{la France est en vert} \Rightarrow \textbf{non}\ \text{l'Allemagne est en vert})$$

$$\vdots$$

La logique du premier ordre

- $\forall x \in P, \exists c \in C,$ x est de la couleur c ;
- $\forall x \in P, \forall y \in P,$ x voisin de y

$$\Rightarrow \forall c \in C, (\textbf{non } x \text{ est de la couleur } c \textbf{ ou non } y \text{ est de la couleur } c) .$$

La logique du premier ordre instanciée pour les réels

- $\forall x \in \mathbb{R}, x > 0 \Rightarrow \exists y \in \mathbb{R} \mid (y > 0 \text{ et } x = y^2)$;
- $\exists x \in \mathbb{R}, x^3 + 5x - 10 = 0$.

Les logiques modales

$\Box_p$ où $\Box$ peut se lire :

- Albert croit que ...
- demain, ...
- il est obligatoire que ...
- on peut prouver que ...
- ...

Example

Albert croit que demain $x = 2$ et ne croit pas qu'il est obligatoire que $y = 3$

Raisonnement automatique

Logique des propositions	possible (!)
Logiques modales	souvent possible [Halpern...]
Logique du premier ordre	impossible [Church]
Logique du premier ordre instanciée avec les entiers	impossible [Tarski]
Logique du premier ordre instanciée avec les réels	possible (!!) [Tarski]

`http://www.irisa.fr/prive/fschwarz/realqelim/`

Outline

Outline

Formule satisfiable

- p **et** $(p \Rightarrow q)$ **et non** q n'est pas satisfiable.
- p **et** $(p \Rightarrow (q$ **ou** $s))$ **et non** q est satisfiable.

Le problème SAT

Le problème SAT

- entrée : une formule φ construite à partir de propositions atomiques, **ou** , **et** , **non** , $\Rightarrow$;
- sortie : oui ssi la formule φ est satisfiable.

Algorithme naif

- parcourir toutes les valeurs possibles pour les propositions atomiques

Le problème SAT

Le problème SAT

- entrée : une formule φ construite à partir de propositions atomiques, **ou** , **et** , **non** , $\Rightarrow$;
- sortie : oui ssi la formule φ est satisfiable.

Algorithme naif

- parcourir toutes les valeurs possibles pour les propositions atomiques

Theorem

Le problème SAT est NP-complet.

En pratique, on peut faire des optimisations

Illustrons tout ça avec des formules qui ressemble à ça :

(p **ou** q) **et** (s **ou non** q **ou non** r) **et** (**non** s **ou** r) **et non** s

Si on connait la valeur d'une proposition atomique, on en profite !

Example

(p **ou** q) **et** (s **ou non** q **ou non** r) **et** (**non** s **ou** r) **et non** s

En pratique, on peut faire des optimisations

Illustrons tout ça avec des formules qui ressemble à ça :

(p **ou** q) **et** (s **ou non** q **ou non** r) **et** (**non** s **ou** r) **et non** s

Si on connait la valeur d'une proposition atomique, on en profite !

Example

(p **ou** q) **et** (s **ou non** q **ou non** r) **et** (**non** s **ou** r) **et non** s

↓

(p **ou** q) **et** (**non** q **ou non** r) **et non** s

En pratique, on peut faire des optimisations

Si une proposition atomique apparat positivement (négativement) partout, on en profite aussi !

Example

(p **ou** q) **et** (s **ou non** q **ou** p) **et** (**non** s **ou** r)

En pratique, on peut faire des optimisations

Si une proposition atomique apparat positivement (négativement) partout, on en profite aussi !

Example

(p **ou** q) **et** (s **ou non** q **ou** p) **et** (**non** s **ou** r)

↓

(**non** s **ou** r)

Outline

Utilisation d'un solveur

Outline

Qu'est ce que la logique ?

- L'algèbre linéaire est la discipline qui étudie les espaces vectoriels.
  On manipule des vecteurs comme $(2,5)$, des applications linéaires, des matrices, etc.

- La logique est la discipline qui étudie les raisonnements.
  On manipule des formules comme a **ou** b.

Exemple de résultat en logique

Theorem (Compacité de la logique propositionnelle)

Soit Σ un ensemble de formules construites à partir de propositions atomiques, **ou** , **et** , **non** , $\Rightarrow$.

On a équivalence entre :

- *Σ est satisfiable;*
- *Toute les parties finies de Σ sont satisfiables.*

Une discipline... pluridisciplinaire

Applications

- en maths, philosophie : formaliser le raisonnement
- en informatique : résoudre des problèmes, vérification de programmes, base de données

Au niveau technique

- maths : topologie, théorie des jeux, etc.
- informatique : algorithmique, théorie de la complexité
- philosophie : logiques temporelles, etc.
- linguistique : syntaxe, sémantique

Merci de votre attention !