A File System Based on Concept Analysis

Sébastien Ferré and Olivier Ridoux

IRISA / CNRS / IFSIC

DOOD'2000

A Logical File System

- Replace paths with logical formulas

  ```
  cd Louis ∨ Ella ; ls black ∧ blue
  ```

- A seemingly hierarchical navigation in a non-hierarchical structure

  ```
  % cd x ; ls
  y       z/       . . .
  % cd z
  ```

  y is "in" x

  z is "under" x

- Replace working-directory with working-query

  PWD := PWD ∧ (Louis ∨ Ella)

  Keep provision for a notion of "absolute" formula

  ```
  cd /Louis∨Ella
  ```

  PWD := Louis ∨ Ella

Formulas as paths

- A path is a way to some *place*

  what is the place designated by a formula?

- A directory name is a path element

  what are the formula elements?

⟶ An answer is in *Formal Concept Analysis* [Wille & Ganter],

and in extensions of it [Ferré & Ridoux]

- A path gives information on where it leads to

  what is the information given by a formula?

  - intrinsic properties: something one can and is willing to pay for
  - extrinsic properties: “interesting books”, “correct programs”, “keywords”, . . .

Logical Concept Analysis (1/2)

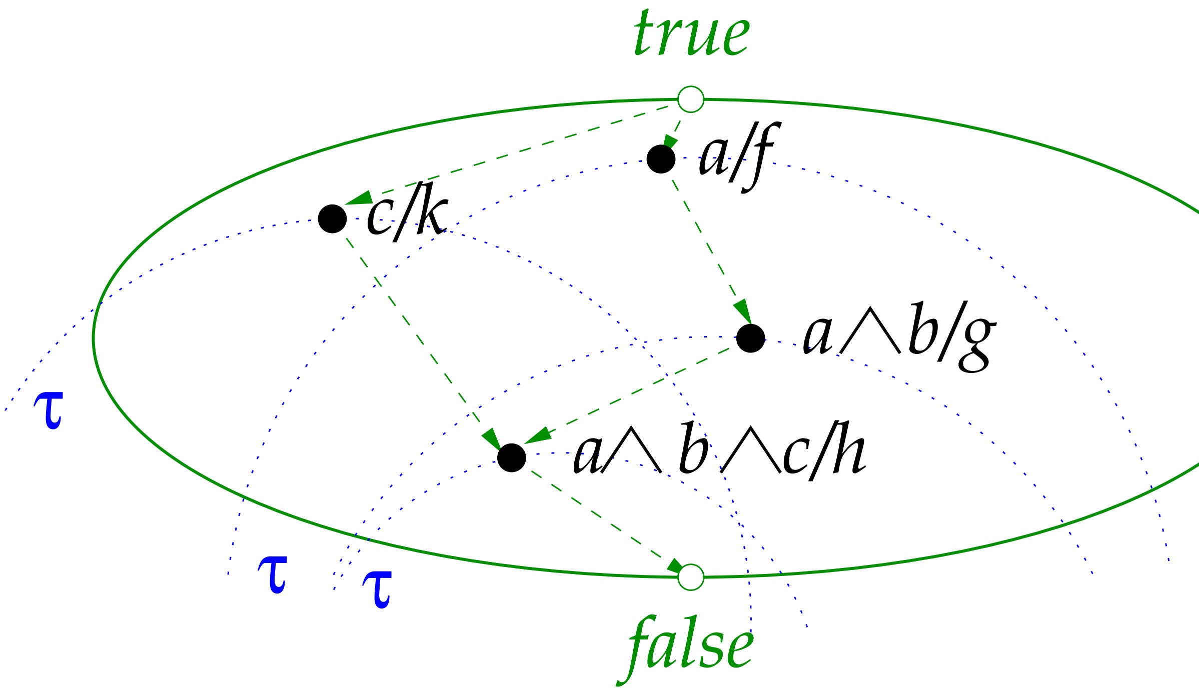
– [Ferré & Ridoux, ICCS-2000]

– Logical context: $i : \mathcal{O} \longrightarrow \mathcal{L}$
all objects such that . . .: $\tau : \mathcal{L} \longrightarrow 2^{\mathcal{O}}, \quad f \mapsto \{o \in \mathcal{O} \mid i(o) \dot{\models} f\}$
the property common to all . . .: $\sigma : 2^{\mathcal{O}} \longrightarrow \mathcal{L}, \quad O \mapsto \dot{\bigvee}_{o \in O} i(o)$

theorem : $\sigma(\tau(\sigma(O))) \dot{\equiv} \sigma(O)$ and $\tau(\sigma(\tau(f))) = \tau(f)$

⟶ the contents of a LFS is a *logical context*

Logical contexts



legend: *i(f)/f*

Logical Concept Analysis (2/2)

– Formal concept : $\{\langle\ \tau(f)\ \ ,\ \sigma(\tau(f))\ \rangle \mid f \in \mathcal{L}\}$

$\langle$ extension, intention $\rangle$

→ f *labels* the concept $\langle\tau(f),\sigma(\tau(f))\rangle$

subconcept : $C_1 \leq C_2 \overset{def}{=} int(C_1)\dot{\models}int(C_2)$ or $ext(C_1) \subseteq ext(C_2)$

theorem : complete lattice for the subconcept≤concept order

→ *formal concepts* serve as directories

→ ≤ serves as a pseudo-hierarchy

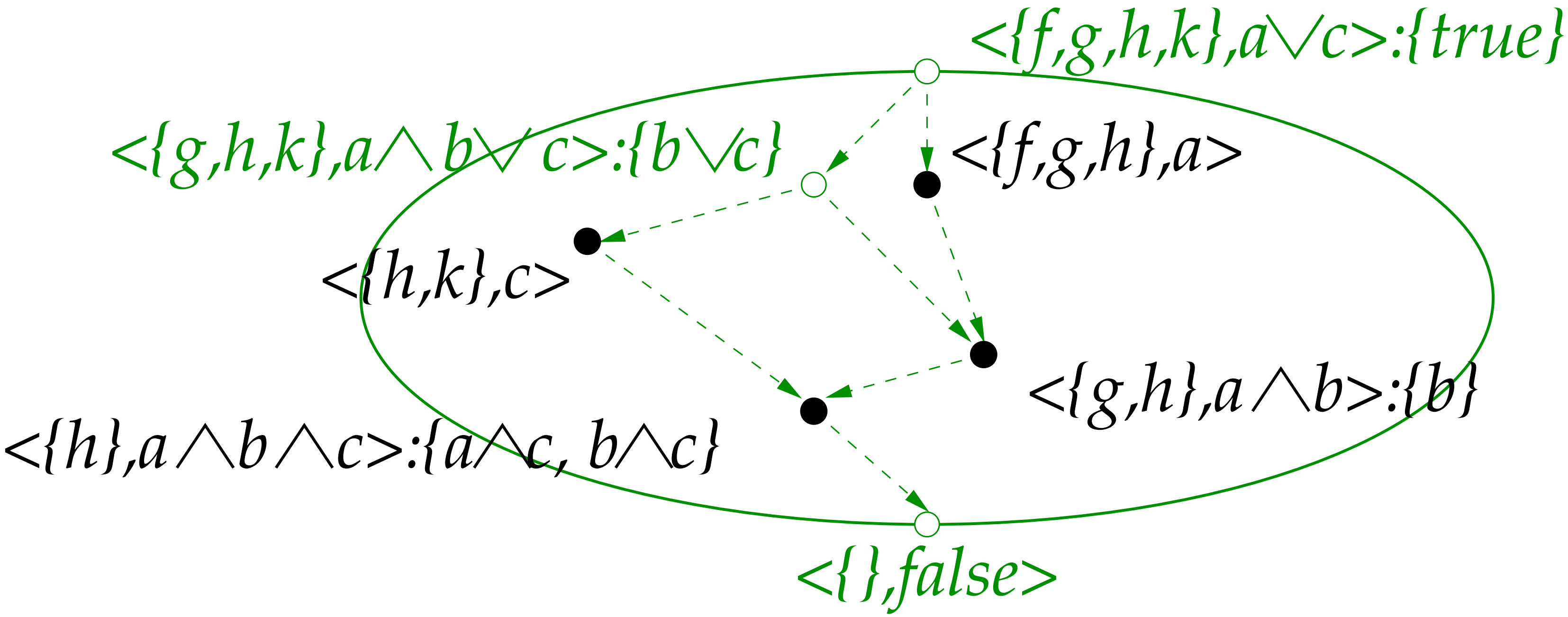
– Description increment :

$$Inc : \mathcal{L} \longrightarrow 2^{\mathcal{L}},$$

$$i(o)\dot{\models}f \Rightarrow \exists g \in Inc(f)[\tau(i(o))\ \subset\ \tau(f) \cap \tau(g)\ \subset\ \tau(f)]$$

→ Inc returns a choice of what to "cd" next

Concept lattices



legend: $<\tau(f),\sigma(\tau(f))>$:*labels*

note: $Inc(a) \ni c$, but $Inc(b) \not\ni a$

Primitive operations

- All objects such that . . .

  $\tau(f) = \{o \in \mathcal{O} \mid i(o) \dot{\models} f\}$

- Who is here?

  $t(q) = \{o \mid \tau(i(o)) = \tau(q)\}$

- What next?

  $i(o) \dot{\models} f \Rightarrow \exists g \in Inc(f)[\tau(i(o)) \ \subset \ \tau(f) \cap \tau(g) \ \subset \ \tau(f)]$

Commands (cd)

- cd f : PWD $:= f \wedge$ PWD

  cd $/f$: PWD $:= f$

  cd a ; cd b $\equiv$ cd b ; cd a

  cd /usr/local/bin

- pwd : PWD

- cd .. : interpreted in the navigation history rather than in the data-structure

  see button <BACK> in WWW navigators

Commands (ls)

- ls -R f : $\underline{\tau}(f)$ (querying)
- ls f : $\underline{t}(f) + \underline{Inc}(f)$ (navigation)

Commands (mv/cp)

- mv -r $from\ to$:

  forall $o \in \underline{\tau}(from)$ do $i(o) := to \dot{\wedge} (from \dot{\Rightarrow} i(o))$ od

- mv $from\ to$:

  forall $o \in \underline{t}(from)$ do $i(o) := to \dot{\wedge} (from \dot{\Rightarrow} i(o))$ od

reminder: $\dfrac{a \wedge (a \Rightarrow b)}{b}$

⟶ "⇒" must exist ; the deduction relation must form a complemented lattice

Implementation

- Is it feasible?

  challenges:

  - avoid representing the concept lattice (exponential in the worst case)
  - avoid calling a theorem prover for every access

  ⟶ a logical cache

- Two prototypes:
  - simple logic / non-naive system: valued attributes / NFS server
    - MAN page explorer: ls change/directory → cd
    - City resource explorer
    - File explorer
  - (more) sophisticated logic / naive system: generic / single file
    - Vietnamese cookbook: ls noodle/fish-sauce
    - Personal notebook: cd todo/july
    - Bibtex explorer: ls author:jones/year:1980..1990/keywords:horn&-music

A logical cache

- A Hasse diagram of *useful* formulas for the $\dot{\models}$ order: $(\mathcal{F}, \rightarrow)$
  - all objects description: $i(\mathcal{O})$
  - all possible increments: e.g. elementary formulas submitted by the user

  the $\dot{\models}$ order does not depend on the context;

  the $\leq$ order on concept does

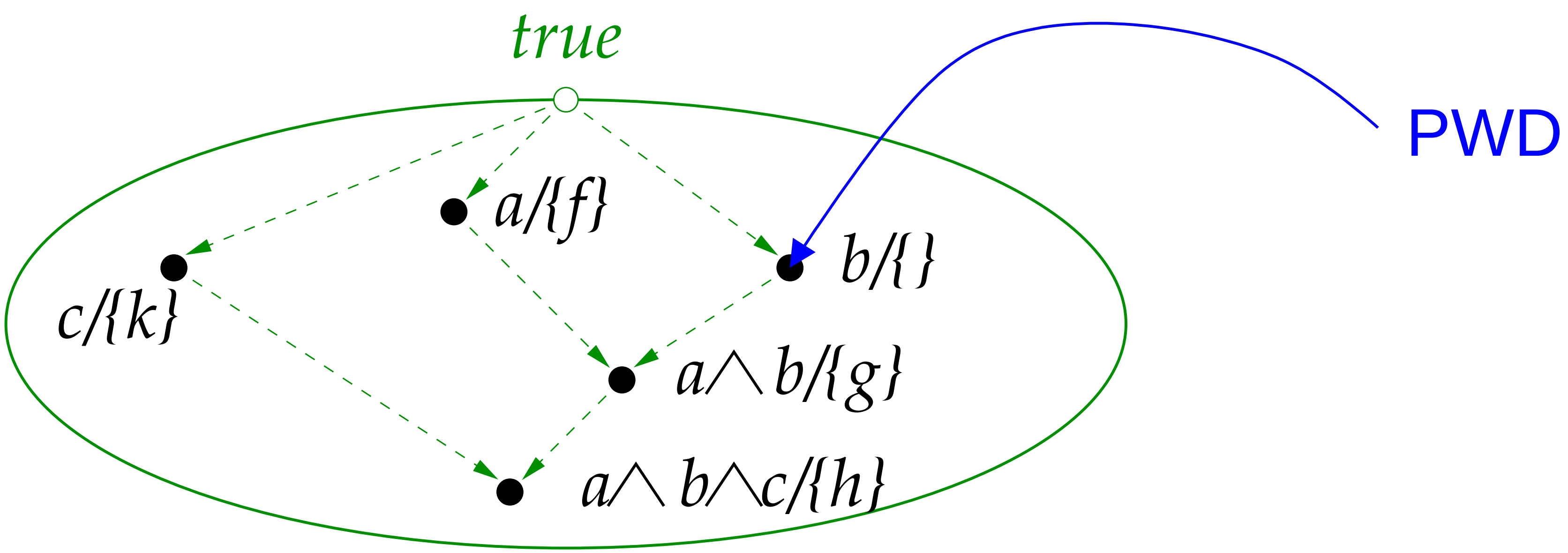
- Primitive operations revisited
  - $\tau(f) = \{o \in \mathcal{O} \mid i(o) \dot{\models} f\}$

    $\tau(f) = \{i^{-1}(g) \mid g \in \mathcal{F} \wedge f \rightarrow^* g\}$
  - $t(q) = \{o \mid \tau(i(o)) = \tau(q)\}$

    $t(q) = i^{-1}(max_{\rightarrow}\{g \mid g \in i(\mathcal{O}) \wedge f \rightarrow^* g\})$
  - $i(o) \dot{\models} f \Rightarrow \exists g \in Inc(f)[\tau(i(o)) \subset \tau(f) \cap \tau(g) \subset \tau(f)]$

    $Inc(q) = \lceil\{x \in \mathcal{F} \mid 0 < |\tau(f) \cap \tau(x)| < |\tau(f)|\}\rceil$

Logical caches



legend: $x/i^{-1}(x)$

```
% cd b ; ls
g      c/
```

Applications

- Consumer oriented

  guides, catalogues, encyclopedia, (address|phone|cook|...)books, ...

- Computer oriented

  personal files, mail folders, ...

  software engineering environment

  - documentation/requirements
  - configuration/version management
  - test datas, program elements
  - software architecture

- Computational linguistics

  sentence $\xrightarrow{nlp}$ formula $\xrightarrow{ls}$ individuals + formulas $\xrightarrow{nlp}$ sentences

- Web, XML

Future works

- Elaborate the logical framework
  - reflection, formulas described by formulas:

    e.g. navigation among formula increments
  - modality: e.g. "All I Know"
  - relations among objects
  - taxinomy

- Elaborate an efficient implementation

  Many issues:
  - intra-file navigation: e.g., bibtex files
  - access modes: e.g., RWE as formula
  - . . .