Deduction and Fractions

Dominique Duval

Grenoble, November 26., 2009
Categorical Computer Science 2009

Outline

Motivations

The semantics of computational effects

Cf. the talk by Jean-Guillaume Dumas:
a framework for dealing with the order of evaluation
of the arguments in a language with effects

Fact 1. Syntax, models, proofs,... : this is logic...

Fact 2. Categories and limit sketches provide tools
for dealing with the semantics *and* with the syntax.

Fact 3. A logic is, essentially, a (bi)category of fractions.

What is a logic?

A logic should have

- a syntax
  which are the sentences of interest?
- a notion of models
  what is the meaning of each sentence?
- a system for proofs
  how can a sentence be infered from another one?

In this talk we focus on proofs.

In this talk

A deduction rule, written AS a fraction

$$\frac{\mathcal{H}}{\mathcal{C}}$$

actually IS a fraction (in the categorical sense)

$$\frac{\mathcal{C}}{\mathcal{H}}$$

Propositional logic

Hilbert calculus, restricted to the connector “$\Rightarrow$”.

Syntax. Propositions (formulas) are made of symbols $p, q, ...$ and a binary operation “$\Rightarrow$”.

Models. Given a set of propositions Σ, a model (interpretation) of Σ associates to each proposition $p \in \Sigma$ a truth value $v(p) \in \{0, 1\}$ in accordance with the truth table for “$\Rightarrow$”:

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Deduction rules. *modus ponens*

$$\frac{A \quad A \Rightarrow B}{B}$$

and two rules with “empty” premisses

$$A \Rightarrow (B \Rightarrow A) \qquad (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$$

Outline

Theories and specifications: two categories

For a given logic:

- A theory Θ is a saturated class of sentences, called theorems:
  every sentence derived from Θ with the rules of the logic is in Θ.
- A specification Σ is a class of sentences, called axioms:
  new sentences may be derived from Σ with the rules of the logic (generally).

This provides two categories:

- **T** for theories
- **S** for specifications

Theories and specifications: two adjoint functors

For a given logic:

- Every theory Θ can be seen as a (huge) specification $\mathbf{R}\Theta$.
- Every specification Σ generates a theory $\mathbf{L}\Sigma$ using the rules of the logic.

This provides two adjoint functors:

$$\begin{tikzcd}\mathbf{S} \arrow[r, "\mathbf{L}"] \arrow[r, phantom, shift right=3, "\perp"] & \mathbf{T} \arrow[l, bend left, "\mathbf{R}"]\end{tikzcd}$$

In addition, every theory Θ is saturated:

$$\mathbf{LR}\Theta \cong \Theta$$

Propositional logic: theories

A propositional theory Θ is:

- a set $\Theta(F)$ of formulas
- a subset $\Theta(T)$ of true formulas
- a binary operation "$\Rightarrow$": $\Theta(F)^2 \to \Theta(F)$ which satisfies the rules: for all p, q, r in $\Theta(F)$:
  - if $p, p \Rightarrow q \in \Theta(T)$ then $q \in \Theta(T)$
  - $p \Rightarrow (q \Rightarrow p) \in \Theta(T)$
  - $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)) \in \Theta(T)$

Example. The theory of booleans $\mathcal{B}$:

$$\mathcal{B}(F) = \{0, 1\}, \mathcal{B}(T) = \{1\},$$
$$B(\Rightarrow)(1, 0) = 0, \text{ otherwise } B(\Rightarrow)(p, q) = 1$$

Propositional logic: specifications

A propositional specification Σ is:

- a set $\Sigma(F)$ of formulas
- a subset $\Sigma(T)$ of true formulas
- a partial binary operation "$\Rightarrow$": $\Sigma(F)^2 \rightharpoonup \Sigma(F)$

Example.

$$\Sigma_0(F) = \{p, q\},\ \Sigma_0(T) = \emptyset,\ q = (p \Rightarrow p).$$

Propositional logic is an adjunction

- $\mathbf{R} : \mathbf{T} \to \mathbf{S}$ is the inclusion
- $\mathbf{L} : \mathbf{S} \to \mathbf{T}$ generates theorems from axioms

A specification Σ_0:

$$\Sigma_0(F) = \{p, q\},\ \Sigma_0(T) = \emptyset,\ q = (p \Rightarrow p).$$

The models of Σ_0:

v_0: $v_0(p) = 0$, $v_0(q) = 1$.
v_1: $v_1(p) = 1$, $v_1(q) = 1$.

The theory $\mathbf{L}\Sigma_0$:

$$\mathbf{L}\Sigma_0(F) = \{p, q, p \Rightarrow q, ...\},\ \mathbf{L}\Sigma_0(T) = \{q, ...\}.$$

$q \in \mathbf{L}\Sigma_0(T)$ because $p \Rightarrow p$ can be deduced,
using the propositional rules.

The models of Σ_0 are the morphisms of theories $\mathbf{L}\Sigma_0 \to \mathcal{B}$.

Diagrammatic logic

Definition. (without syntax...)

A diagrammatic logic is functor with a full and faithful right adjoint

So, a logic is

- a category of theories **T**
- a category of specifications **S**
- a forgetful functor **R** : **T** → **S**
- a generating functor **L** : **S** → **T**

which form an adjunction

$$\begin{tikzcd} \mathbf{S} \arrow[r, "\mathbf{L}"] \arrow[r, phantom, "\perp"', shift right=2] & \mathbf{T} \arrow[l, bend left, "\mathbf{R}"] \end{tikzcd}$$

with **R** full and faithful, *i.e.*, $\mathbf{LR}\Theta \cong \Theta$ for every theory Θ

Entailments and fractions

With respect to a logic $\mathbf{L} : \mathbf{S} \to \mathbf{T}$

- An entailment

$$\begin{tikzcd}\Sigma \arrow[r, shift left, "\tau"] & \Sigma' \arrow[l, dashed, shift left]\end{tikzcd}$$

  is a morphism $\tau : \Sigma \to \Sigma'$ in $\mathbf{S}$ such that $\mathbf{L}\tau$ is invertible in $\mathbf{T}$.

- A fraction $\frac{\sigma}{\tau}$ is a cospan in $\mathbf{S}$ made of
  a morphism σ (the numerator)
  and an entailment τ (the denominator)

$$\begin{tikzcd}\Sigma \arrow[r, "\sigma"] & \Sigma'_1 \arrow[r, dashed, shift right] & \Sigma_1 \arrow[l, shift right, "\tau"']\end{tikzcd}$$

  Then $\mathbf{L}(\frac{\sigma}{\tau}) = (\mathbf{L}\tau)^{-1} \circ \mathbf{L}\sigma$ in $\mathbf{T}$

$$\begin{tikzcd}\mathbf{L}\Sigma \arrow[r, "\mathbf{L}\sigma"] \arrow[rr, bend right, "\mathbf{L}(\frac{\sigma}{\tau})"'] & \mathbf{L}\Sigma'_1 \arrow[r, shift right, "\mathbf{L}\tau^{-1}"'] & \mathbf{L}\Sigma_1 \arrow[l, shift right, "\mathbf{L}\tau"']\end{tikzcd}$$

“The” theorem

Gabriel and Zisman (1967)
Calculus of Fractions and Homotopy Theory. Ch. 1.

Remark. Every theory Θ is $\Theta = \mathbf{L}\Sigma$ for some specification Σ.

Remark. In general, a morphism of theories $\theta\colon \mathbf{L}\Sigma \to \mathbf{L}\Sigma_1$ is not $\theta = \mathbf{L}\sigma$ for a morphism of specifications $\sigma\colon \Sigma \to \Sigma_1$.

(because Σ_1 is “too small”)

Theorem. Every morphism of theories $\theta\colon \mathbf{L}\Sigma \to \mathbf{L}\Sigma_1$ is $\theta = \mathbf{L}(\frac{\sigma}{\tau})$ for some fraction $\frac{\sigma}{\tau} : \Sigma \to \Sigma_1$.

Corollary. (Up to equiv.,) **T** is the category of fractions of **S** with denominators the entailments.

What is a deduction rule?

With respect to a logic **L** : **S** $\to$ **T**

Definition.

A rule $\frac{\mathcal{H}}{\mathcal{C}}$ is a fraction $\frac{c}{h} : \mathcal{C} \to \mathcal{H}$

$$\begin{tikzcd}
\mathcal{H} \arrow[r, shift left, "h"] & \mathcal{H}' \arrow[l, shift left, dashed] & \mathcal{C} \arrow[l, "c"']
\end{tikzcd}$$

This definition includes both elementary rules
and derived rules (or proofs)

(the distinction is provided by the syntax of **L**).

According to [GZ68], the rules are
the morphisms of theories, expressed as fractions.

The modus ponens rule

$$\frac{\mathcal{H}}{\mathcal{C}} = \frac{A \quad A \Rightarrow B}{B}$$

- Static. A theory Θ is a saturated set of theorems.
  Let Θ be a theory with theorems p and $p \Rightarrow q$.
  Then theorem q is also in Θ.
- Dynamic. A specification Σ is a set of axioms,
  which generates a theory $\mathbf{L}\Sigma$.
  Let Σ be a specification with axioms p and $p \Rightarrow q$.
  Then the specification Σ' made of Σ and the axiom q
  is equivalent to Σ, *i.e.*, $\mathbf{L}\Sigma = \mathbf{L}\Sigma'$.

The modus ponens fraction

Propositional specifications:

$$\mathcal{H}:\ \mathcal{H}(F)=\{A,B,A\Rightarrow B\},\ \mathcal{H}(T)=\{A,A\Rightarrow B\}$$
$$\mathcal{C}:\ \mathcal{C}(F)=\{B\},\ \mathcal{C}(T)=\{B\}$$
$$\mathcal{H}':\ \mathcal{H}'(F)=\{A,B,A\Rightarrow B\},\ \mathcal{H}'(T)=\{A,B,A\Rightarrow B\}$$

The inclusions of $\mathcal{H}$ and $\mathcal{C}$ in $\mathcal{H}'$ are morphisms of specifications and h is an entailment

$$\begin{tikzcd}
\mathcal{H} \arrow[r, hook, shift left, "h"] & \mathcal{H}' \arrow[l, dashed, shift left] & \mathcal{C} \arrow[l, hook', "c"']
\end{tikzcd}$$

$\mathbf{L}h$ is an isomorphism of theories, $\mathbf{L}(\frac{c}{h})=\mathbf{L}h^{-1}\circ\mathbf{L}c$

$$\begin{tikzcd}
\mathbf{L}\mathcal{H} \arrow[r, shift left, "\mathbf{L}h"] & \mathbf{L}\mathcal{H}' \arrow[l, shift left, "\mathbf{L}h^{-1}"] & \mathbf{L}\mathcal{C} \arrow[l, "\mathbf{L}c"'] \arrow[ll, bend left, "\mathbf{L}(\frac{c}{h})"]
\end{tikzcd}$$

Rules are fractions

RULES	FRACTIONS	numbers
$\mathcal{H},\ \mathcal{C}:$ rules	$\mathcal{H},\ \mathcal{C}:$ fractions	$2,\ 3 \in \mathbb{Z}$
$\frac{\mathcal{H}}{\mathcal{C}}$	$\begin{tikzcd} \mathcal{H} \arrow[r, shift left, "h"] & \mathcal{H}' \arrow[l, dashed, shift left] & \mathcal{C} \arrow[l, "c"'] \arrow[ll, dotted, bend left, "\frac{c}{h}"] \end{tikzcd}$	$\frac{3}{2}\ \ (\frac{3}{2} \neq \frac{6}{4})$ “syntactically”
$\frac{\mathcal{H}}{\mathcal{C}}$	$\begin{tikzcd} \mathbf{L}\mathcal{H} \arrow[r, shift left, "\mathbf{L}h"] & \mathbf{L}\mathcal{H}' \arrow[l, shift left] & \mathbf{L}\mathcal{C} \arrow[l, "\mathbf{L}c"'] \arrow[ll, bend left, "\mathbf{L}(\frac{c}{h})"] \end{tikzcd}$	$\frac{3}{2} \in \mathbb{Q}\ \ (\frac{3}{2} = \frac{6}{4})$

Outline

Deduction

The deduction process is
a succession of deduction steps.

A deduction step:

- *Input.* A rule $\frac{\mathcal{H}}{\mathcal{C}}$, a specification Σ, an instance *i* of $\mathcal{H}$ in Σ.
- *Output.* The instance *j* of $\mathcal{C}$ in Σ which corresponds to "applying $\frac{\mathcal{H}}{\mathcal{C}}$ to *i*".

What is a deduction step?

With respect to a logic $\mathbf{L} : \mathbf{S} \to \mathbf{T}$

Definition.

$\boxed{\text{An instance of } \mathcal{H} \text{ in } \Sigma \text{ is a fraction } i : \mathcal{H} \to \Sigma}$

Definition.

The step applying a rule $p : \mathcal{C} \to \mathcal{H}$
to an instance $i : \mathcal{H} \to \Sigma$ of $\mathcal{H}$ in Σ
is the composition of fractions $i \circ p : \mathcal{C} \to \Sigma$

$$\begin{tikzcd}
\mathcal{H} \arrow[dr, "i"'] & & \mathcal{C} \arrow[ll, "p"'] \arrow[dl, "j"] \\
& \Sigma &
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{H} \arrow[rr, "h"] \arrow[dr] & & \mathcal{H}' \arrow[ll, dashed] \arrow[dr] & & \mathcal{C} \arrow[ll, "c"'] \arrow[dl] \\
& \Sigma_H \arrow[rr] \arrow[dr, dashed] & & \Sigma_C \arrow[ll, dashed] \arrow[dl, dashed] & \\
& & \Sigma \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Deduction process

Since a deduction step is a composition,
the deduction process is (essentially)
a succession of compositions....

... combined with colimits of specifications
for grouping several hypotheses in a unique one...

... resulting in the usual tree-like representation
of the deduction.

Applying modus ponens

The specification Σ_0: $\Sigma_0(F) = \{p, q\}$, $\Sigma_0(T) = \emptyset$, $q = (p \Rightarrow p)$ generates the theorem q.

The last step in the proof is an application of *modus ponens*:

$$\frac{(p \Rightarrow q) \Rightarrow q \qquad p \Rightarrow q}{q}$$

$$\begin{tikzcd}
\{A, A \Rightarrow B\} \arrow[rr, shift left, "h"] \arrow[dr, "A \mapsto (p \Rightarrow q)"', "B \mapsto q"'] & & \{A, A \Rightarrow B, B\} \arrow[ll, shift left, dashed] \arrow[dr] & & \{B\} \arrow[ll, "c"'] \arrow[dl] \\
& \{(p \Rightarrow q) \Rightarrow q, p \Rightarrow q\} \arrow[rr, shift left] \arrow[dr, dashed, shift right] & & \{(p \Rightarrow q) \Rightarrow q, p \Rightarrow q, q\} \arrow[ll, shift left, dashed] \arrow[dl, dashed, shift left] & \\
& & \emptyset \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Outline

Introduction

A deduction rule is a fraction

Deduction is the composition of fractions

Conclusion

Conclusion

- The category of fractions is the quotient of a bicategory, and bicategories are technically difficult...

  Cf. the talk by Pawel Sobocinski.

- More about models, syntax, etc...
  • D.D. *How to combine diagrammatic logics*.

- More examples
  • Jean-Guillaume Dumas , D.D., Jean-Claude Reynaud. *Cartesian effect categories are Freyd-categories*.
  • Cesar Dominguez, D.D. *A parameterization process as a categorical construction*.