Deduction and Fractions

Dominique Duval

Grenoble, November 26., 2009
Categorical Computer Science 2009
Introduction

A deduction rule is a fraction

Deduction is the composition of fractions

Conclusion
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 inferred from another one?*

In this talk we focus on proofs.
In this talk

A deduction rule, written AS a fraction

\[
\frac{\mathcal{H}}{C}
\]

actually IS a fraction (in the categorical sense)

\[
\frac{C}{\mathcal{H}}
\]
Propositional logic

Hilbert calculus, restricted to the connector “⇒”.

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

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

\[
\begin{array}{ccc}
A & B & A \Rightarrow B \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

Deduction rules. modus ponens

\[
\begin{array}{c}
A \\
A \Rightarrow B \\
\hline
B \\
\end{array}
\]

and two rules with “empty” premisses

\[
A \Rightarrow (B \Rightarrow A) \quad (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))
\]
Outline

Introduction

A deduction rule is a fraction

Deduction is the composition of fractions

Conclusion
Theories and specifications: two categories

For a given logic:

- A **theory** $\Theta$ is a saturated class of sentences, called **theorems**: every sentence derived from $\Theta$ with the rules of the logic is in $\Theta$.
- A **specification** $\Sigma$ is a class of sentences, called **axioms**: new sentences may be derived from $\Sigma$ 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 $\Theta$ can be seen as a (huge) specification $R\Theta$.
- Every specification $\Sigma$ generates a theory $L\Sigma$ using the rules of the logic.

This provides two adjoint functors:

\[
\begin{array}{c}
S \xleftarrow{\perp} L \xrightarrow{R} T
\end{array}
\]

In addition, every theory $\Theta$ is saturated:

\[LR\Theta \cong \Theta\]
Propositional logic: theories

A propositional theory $\Theta$ is:

- a set $\Theta(F)$ of formulas
- a subset $\Theta(T)$ of true formulas
- a binary operation “$\Rightarrow$”: $\Theta(F)^2 \rightarrow \Theta(F)$
  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 $\mathcal{B}$:

$\mathcal{B}(F) = \{0, 1\}$, $\mathcal{B}(T) = \{1\}$,
$\mathcal{B}(\Rightarrow)(1, 0) = 0$, otherwise $\mathcal{B}(\Rightarrow)(p, q) = 1$
Propositional logic: specifications

A propositional specification $\Sigma$ is:
- a set $\Sigma(F)$ of formulas
- a subset $\Sigma(T)$ of true formulas
- a partial binary operation “$\Rightarrow$”: $\Sigma(F)^2 \rightarrow \Sigma(F)$

Example.

$\Sigma_0(F) = \{p, q\}$, $\Sigma_0(T) = \emptyset$, $q = (p \Rightarrow p)$. 
Propositional logic is an adjunction

- **R : T → S** is the inclusion
- **L : S → T** generates theorems from axioms

A specification $\Sigma_0$:

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

The models of $\Sigma_0$:

- $\nu_0$: $\nu_0(p) = 0, \nu_0(q) = 1$.
- $\nu_1$: $\nu_1(p) = 1, \nu_1(q) = 1$.

The theory $L\Sigma_0$:

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

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

The models of $\Sigma_0$ are the morphisms of theories $L\Sigma_0 \rightarrow B$. 
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 $R : T \rightarrow S$
- a generating functor $L : S \rightarrow T$

which form an adjunction

$$
S \xrightarrow{R} T \xleftarrow{L} S
$$

with $R$ full and faithful, i.e., $L R \Theta \cong \Theta$ for every theory $\Theta$
Entailments and fractions

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

- An entailment

$\Sigma \xrightarrow{\tau} \Sigma'$

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

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

$\Sigma \xrightarrow{\sigma} \Sigma' \xleftarrow{\tau} \Sigma_1$

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

\[
\begin{array}{ccc}
L\Sigma & \xrightarrow{L\sigma} & L\Sigma' \\
& & \xleftarrow{L\tau} L\Sigma_1 \\
& & \downarrow_{L(\tau)^{-1}} \\
& & L(\frac{\sigma}{\tau})
\end{array}
\]
“The” theorem

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

Remark. Every theory $\Theta$ is $\Theta = L\Sigma$ for some specification $\Sigma$.

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

(because $\Sigma_1$ is “too small”)

Theorem. Every morphism of theories $\theta : L\Sigma \to L\Sigma_1$ is $\theta = L(\frac{\sigma}{\tau})$ 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 $L : S \rightarrow T$

**Definition.**

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

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

\[
\frac{A \land A \Rightarrow B}{B}
\]

- **Static.** A theory $\Theta$ is a saturated set of theorems. Let $\Theta$ be a theory with theorems $p$ and $p \Rightarrow q$. Then theorem $q$ is also in $\Theta$.

- **Dynamic.** A specification $\Sigma$ is a set of axioms, which generates a theory $L\Sigma$. Let $\Sigma$ be a specification with axioms $p$ and $p \Rightarrow q$. Then the specification $\Sigma'$ made of $\Sigma$ and the axiom $q$ is equivalent to $\Sigma$, i.e., $L\Sigma = L\Sigma'$.
The modus ponens fraction

Propositional specifications:
\[ \mathcal{H} : \mathcal{H}(F) = \{ A, B, A \Rightarrow B \}, \quad \mathcal{H}(T) = \{ A, A \Rightarrow B \} \]
\[ \mathcal{C} : \mathcal{C}(F) = \{ B \}, \quad \mathcal{C}(T) = \{ B \} \]
\[ \mathcal{H}' : \mathcal{H}'(F) = \{ A, B, A \Rightarrow B \}, \quad \mathcal{H}'(T) = \{ A, B, A \Rightarrow B \} \]

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

\[
\begin{array}{ccc}
\mathcal{H} & \xrightarrow{h} & \mathcal{H}' \\
\downarrow & & \downarrow \\
\mathcal{C} & \xleftarrow{c} & \mathcal{H}'
\end{array}
\]

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

<table>
<thead>
<tr>
<th>RULES</th>
<th>FRACTIONS</th>
<th>numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{H}, \mathcal{C} : \text{rules} )</td>
<td>( \mathcal{H}, \mathcal{C} : \text{fractions} )</td>
<td>( 2, 3 \in \mathbb{Z} )</td>
</tr>
<tr>
<td>( \frac{\mathcal{H}}{\mathcal{C}} )</td>
<td>( \mathcal{H} \xrightarrow{\mathcal{H}} \mathcal{H}' \xleftarrow{\mathcal{C}} \mathcal{C} )</td>
<td>( \frac{3}{2} \left( \frac{3}{2} \neq \frac{6}{4} \right) ) “syntactically”</td>
</tr>
<tr>
<td>( \frac{\mathcal{H}}{\mathcal{C}} )</td>
<td>( \mathcal{LH} \xrightarrow{\mathcal{LH}} \mathcal{LH}' \xleftarrow{\mathcal{LC}} \mathcal{LC} )</td>
<td>( \frac{3}{2} \in \mathbb{Q} \left( \frac{3}{2} = \frac{6}{4} \right) )</td>
</tr>
</tbody>
</table>
Outline

Introduction

A deduction rule is a fraction

Deduction is the composition of fractions

Conclusion
The deduction process is a succession of deduction steps.

A deduction step:

- **Input.** A rule $\frac{H}{C}$, a specification $\Sigma$, an instance $i$ of $H$ in $\Sigma$.

- **Output.** The instance $j$ of $C$ in $\Sigma$ which corresponds to “applying $\frac{H}{C}$ to $i$”.
What is a deduction step?

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

**Definition.**

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

**Definition.**

The step applying a rule $p : C \rightarrow \mathcal{H}$ to an instance $i : \mathcal{H} \rightarrow \Sigma$ of $\mathcal{H}$ in $\Sigma$ is the composition of fractions $i \circ p : C \rightarrow \Sigma$
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 $\Sigma_0 : \Sigma_0(F) = \{p, q\}$, $\Sigma_0(T) = \emptyset$, $q = (p \Rightarrow p)$ generates the theorem $q$.

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

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

\[
\begin{array}{c}
\{A, A \Rightarrow B\} \xrightarrow{h} \{A, A \Rightarrow B, B\} \xleftarrow{c} \{B\}
\end{array}
\]

\[
\begin{array}{c}
\{A, A \Rightarrow B\} \xrightarrow{A \rightarrow (p \Rightarrow q)} \{B\}
\end{array}
\]

\[
\begin{array}{c}
\{A, A \Rightarrow B\} \xrightarrow{B \rightarrow q} \{B\}
\end{array}
\]

\[
\begin{array}{c}
\{(p \Rightarrow q) \Rightarrow q, p \Rightarrow q\} \xrightarrow{} \{(p \Rightarrow q) \Rightarrow q, p \Rightarrow q, q\}
\end{array}
\]

\[
\begin{array}{c}
\emptyset \xleftarrow{} \{(p \Rightarrow q) \Rightarrow q, p \Rightarrow q, q\}
\end{array}
\]
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*. 