The Curry-Howard-isomorphism and Cartesian Closed Categories

Florian Rabe

Part of the course on Computational Logic by Michael Kohlhase

Fall 2007, Jacobs University Bremen

06.11.2007

Review: Categories of Contexts

- A recurring pattern in logic
- Essentially the same for *FOL* and *STT*; in the following $Con^{STT}(\Sigma)$ as an example
- Objects: Contexts $x_1 : A_1, \ldots, x_n : A_n$
- Morphisms from $\Gamma' = x_1 : A_1, \ldots, x_m : A_m$ to $\Gamma = y_1 : B_1, \ldots, y_n : B_n$: tuples $(t_1, \ldots, t_n)$ such that $x_1 : A_1, \ldots, x_m : A_m \vdash_\Sigma t_i : B_i$ for $i = 1, \ldots, n$
- Morphism $(t_1, \ldots, t_n)$ from Γ' to Γ can be seen as substitutions $\sigma = [t_1/y_1, \ldots, t_n/y_n]$ (where, intutitively, the substitutions go against the morphism direction)
- Identity $id_{x_1:A_1,\ldots,x_m:A_m} = (x_1, \ldots, x_m)$
- Composition $\sigma \bullet (t_1, \ldots, t_n) = (\sigma(t_1), \ldots, \sigma(t_n))$

References: Curry-Howard-correspondence

```
@Book{curry,
  author = "H. Curry and R. Feys",
  title = "Combinatory Logic",
  publisher = "North-Holland",
  address = "Amsterdam",
  year = "1958",
}

@InCollection{howard,
  author = "W. Howard",
  title = "The formulas-as-types notion of construction
    ",
  booktitle = "To H.B. Curry: Essays on Combinatory
    Logic, Lambda-Calculus and Formalism",
  editors = "J. Seldin and J. Hindley",
  publisher = "Academic Press",
  year = "1980",
  pages = "479--490",
}
```

Review: First-order logic

Summary:

- Inference rules for FOL-fragment consisting of true, conjunction and implication
- Natural deduction with context, e.g., $\Gamma \vdash A$ for $\Gamma = A_1, \ldots, A_n$
- Rules: truth introduction, conjunction introduction and elimination, implication introduction and elimination

Review: Simply-typed lambda calculus

Summary:

- Typing of terms for lambda calculus with unit type, product type, and function type
- Type inference rules with context, e.g., $\Gamma \vdash t : A$ for $\Gamma = x_1 : A_1, \ldots, x_n : A_n$
- Rules: unit element, pairing, projections, lambda abstraction, application

The Curry-Howard-correspondence

- Observation: Rules look almost alike
- formulas correspond to types
- context (= list of hypotheses) corresponds to context (= list of variable declarations)
- inference rules correspond to typing rules
- What FOL-entity corresponds to STT-terms? Answer: Proofs.
- Thus: Proof-checking corresppnds to type-checking; proof-search corresponds to type-inhabitation (where a type is inhabited if there is a term of it).

Proof term assignment

Summary:

- In the FOL-inference rules, replace $A_1, \ldots, A_n \vdash A$ with $x_1 : A_1, \ldots, x_n : A_n \vdash p : A$ where
  - x_i: names for the assumed proofs of the A_i
  - p proof of A possibly using the assumed proofs x_i, i.e., p is a (proof) term with variables x_i
- Give every rule a name, so that a rule becomes a constructor for proof terms. For example, call modus ponens MP.
- Then every proof is a term, for example $MP(p_1, p_2)$ is a proof of B if p_1 is a proof of $A \longrightarrow B$ and p_2 is a proof of A.
- Then the rules for *FOL* with proofs and for *STT* become the same.

07.11.2007

Review: Constructions in Categories

- Pairs terminal/initial, product/coproduct, pullback/pushout, equalizer/coequalizer of dual constructions
- All constructions exist in $\mathcal{S}\mathrm{et}$.
- Results:
  - If U is an X, then: U' is an X iff $U \cong U'$, where X is one of the above constructions.
  - If $\mathcal{C}$ has all X, then $\mathcal{C}^{op}$ has all duality partners X.
  - If $\mathcal{C}$ has a terminal object, then having all pullbacks is equivalent to having all products and all equalizers.
  - Dually: If $\mathcal{C}$ has an initial object, then having all pushouts is equivalent to having all coproducts and all coequalizers.
- Categories relevant for logic: Sig^I, Th^I, Con^I_Σ, $Mod^I(\Sigma)$
- Candidates for constructions in the relevant categories
  - Sig^I and Th^I: initial - empty, coproduct - union, pushout - union with sharing
  - Con^I_Σ: terminal - empty, product - concatenation
  - $Mod^I(\Sigma)$: terminal - singleton, product - cartesian product, initial - term model

Reference: Cartesian Closed Categories

```
@book{ccc,
    author = {J. Lambek and P. Scott},
    publisher = {Cambridge University Press},
    series = {Cambridge Studies in Advanced Mathematics
        },
    title = {Introduction to Higher-Order Categorical
        Logic},
    volume = {7},
    year = {1986}
}
```

Recommended read: Chapter 6.1 in Steve Awodey's notes

Cartesian Closed Categories

- Assume a category $\mathcal{C}$ with products $(A_1 \times A_2, \pi_1^{A_1,A_2}, \pi_2^{A_1,A_2})$ for all $A_1, A_2 \in |\mathcal{C}|$.
- Then for all $f_i \in \mathcal{C}(O, A_i)$ for $i = 1, 2$, there is a unique morphism $\langle f_1, f_2 \rangle : \mathcal{C}(O, A_1 \times A_2)$.
- For $g_i \in \mathcal{C}(A_i, B_i)$ for $i = 1, 2$, we put $g_1 \times g_2 := \langle \pi_1^{A_1,A_2} \bullet f_1, \pi_2^{A_1,A_2} \bullet f_2 \rangle \in \mathcal{C}(A_1 \times A_2, B_1 \times B_2)$.
- For $A, B \in |\mathcal{C}|$, $(B^A, eval^{A,B})$ where $eval^{A,B} \in \mathcal{C}\left(B^A \times A, B\right)$ is called an exponential of A and B iff:

$$\mathcal{C}(\Gamma \times A, B) \cong \mathcal{C}\left(\Gamma, B^A\right), \qquad f \mapsto \bar{f}$$

  for all $\Gamma \in |\mathcal{C}|$ and $(\bar{f} \times id_A) \bullet eval^{A,B} = f$ for all $f \in \mathcal{C}(\Gamma \times A, B)$.
- A category is called cartesian closed if it has a terminal object, products for all pairs of objects, and exponentials for all pairs of objects.

Cartesian Closed Categories (2)

- CCC is the category with cartesian closed categories as objects.
- CCC-morphisms from $\mathcal{C}$ to $\mathcal{C}'$: Functors F from $\mathcal{C}$ to $\mathcal{C}'$ that preserve CCC-structure, i.e.,
  - F maps the terminal object of $\mathcal{C}$ to the terminal object of $\mathcal{C}'$,
  - $(F(A \times B), F(\pi_1^{A,B}), F(\pi_2^{A,B}))$ is a product in $\mathcal{C}'$ of $F(A)$ and $F(B)$,
  - $(F(B^A), F(\mathit{eval}^{A,B}))$ is an exponential in $\mathcal{C}'$ of $F(A)$ and $F(B)$.
- Identity and composition are as in $\mathcal{C}\mathrm{at}$.

Example: $\mathcal{S}\mathrm{et}$

- Terminal: Singletons
- Product: Cartesian product with projections
- Exponential of A and B:
  - $B^A = \mathcal{S}\mathrm{et}(A, B)$
  - $eval^{A,B} : (\varphi, \alpha) \mapsto \varphi(\alpha)$ for $\varphi \in \mathcal{S}\mathrm{et}(A, B)$ and $\alpha \in A$.

Example: Con_{Σ}^{STT}

- Here: STT with unit type T, product types $A \times B$, and function types $A \to B$
- Only the case with one variable needed because $x_1 : A_1, \ldots, x_m : A_m \cong x : A_1 \times \ldots \times A_m$.
- Then: Morphisms from $x : A$ to $x : B$ are simply terms t such that $x : A \vdash t : B$.
- Terminal object is $x : T$ (or isomorphically: empty context).
- Product object of $x : A$ and $x : B$ is $x : A \times B$ (or isomorphically: $x : A, y : B$).
- Exponential object of $x : A$ and $x : B$ is $x : A \to B$.
- Still missing: projection morphisms for the product, evaluation morphism for the exponential
- Needed for the proof: unique morphisms into the terminal and product object and the isomorphism from the definition of exponential

Example: *FOL*

- Use the fragment of *FOL* with truth, conjunction, implication; call it P.
- This is even a fragment of propositional logic, i.e., no quantifiers, no terms; signatures contain only nullary predicate symbols.
- Let Σ' be the STT-signature containing one constant for every inference rule of P. We use one base type for every nullary predicate symbol of Σ.
- Build a category Pf_Σ^P
  - objects are named lists of hypotheses, i.e., $x_1 : A_1, \ldots, x_n : A_m$ for formulas $A_1, \ldots, A_m \in Sen^P(\Sigma)$
  - morphisms from $x_1 : A_1, \ldots, x_n : A_m$ to $y_1 : B_1, \ldots, y_n : B_n$ are tuples $(p_1, \ldots, p_n)$ where p_i is a proof of B_i using the assumptions $A_1, \ldots, A_m$
  - identity and composition just like for $Con_{\Sigma'}^{STT}$

Then clearly: $Pf_\Sigma^P \cong Con_{\Sigma'}^{STT}$ in CCC.