Theory of Programming Languages
程序设计语言理论

张昱

Department of Computer Science and Technology
University of Science and Technology of China

December, 2008

第六章 命题和类型

6.1 Curry-Howard同构[PFPL, 32]

6.2 经典证明和控制算符[PFPL, 33]

6.1 Curry-Howard同构(Isomorphism)

Curry-Howard同构：在命题(**Proposition**)和类型(**type**)之间存在对应，使得证明对应于程序。

对于每个命题ϕ，存在一个关联的类型τ，使得对ϕ的每个证明，存在一个对应的类型为τ的表达式。

证明有可计算的内容，程序是证明的一种形式。

程序语言中的概念可以引起逻辑中的概念，相反亦然。

最初的同构由**Curry**、**Howard**观测到，适用于构造逻辑；后来得到扩展和丰富。

6.1.1 构造逻辑(Constructive Logic) (直觉主义逻辑)

不接受<u>排中律(命题非真即假)</u>

6.1.2 命题作为类型

6.1.1 构造逻辑(Constructive Logic)-1

- 构造逻辑(直觉主义逻辑)的语义
  - 构造逻辑关心两种断言
    - ϕ **prop**：表示ϕ是一个命题
    - ϕ **true**：表示ϕ是一个真命题
  - 命题是描述要解决的问题的规范(specification)
  - 对命题所引起的问题的解决是证明(proof)

    如果命题有一个证明，则称该命题为真。
  - 构造逻辑的特征：只有存在命题的证明，才能判断该命题为真——可构造性。
  - 如何评判命题为假？用反证法证明(假设命题为真,推出与假设相矛盾)

6.1.1 构造逻辑(Constructive Logic)-2

❖ 构造逻辑的语义

- 构造逻辑不接受排中律，即不接受 $\phi \vee \neg\phi$为定理

  对于一个命题，不一定是真、假之一。

Why? 总是存在一些尚未解决的命题!

未解决：表示尚没有对此命题的证明或反例(refutation)

例如，P=NP?

- 命题ϕ是可判定的(decidable)是指存在对ϕ的证明或反例.

例：如果ϕ表示两个自然数的不相等命题，则ϕ是可判定的。因为对于给定的自然数m和n，总可以算出$m = n$或者$m \neq n$。

6.1.1 构造逻辑(Constructive Logic)-3

❖ 构造逻辑的语义

- 断言ϕ **prop**和ϕ **true**是基本的，是直言断言
- 一般地，会更关注假言断言 ϕ_1 true, ..., ϕ_n true $\vdash \phi$ true 表示ϕ为真是以$\phi_1, \ldots, \phi_n$都为真为假设的。
- 假言断言满足以下结构性质(Γ是一组命题为真的假设)

$$\frac{}{\Gamma, \phi \text{ true} \vdash \phi \text{ true}}$$

$$\frac{\Gamma \vdash \phi \text{ true} \quad \Gamma, \phi \text{ true} \vdash \psi \text{ true}}{\Gamma \vdash \psi \text{ true}}$$

$$\frac{\Gamma \vdash \psi \text{ true}}{\Gamma, \phi \text{ true} \vdash \psi \text{ true}}$$

$$\frac{\Gamma, \phi \text{ true}, \phi \text{ true} \vdash \theta \text{ true}}{\Gamma, \phi \text{ true} \vdash \theta \text{ true}}$$

$$\frac{\Gamma, \psi \text{ true}, \phi \text{ true}, \Gamma' \vdash \theta \text{ true}}{\Gamma, \phi \text{ true}, \psi \text{ true}, \Gamma' \vdash \theta \text{ true}}$$

(32.1)

6.1.1 构造逻辑(Constructive Logic)-4

❖ 命题逻辑

- 命题逻辑的联结词：永真(truth)、永假(falsehood)、合取(conjunction)、析取(disjunction)、蕴涵(implication)、否(negation)
- 命题逻辑的语法：由以下推导形如ϕ **prop**的断言的规则给出

$$\frac{}{\texttt{true}\ \mathsf{prop}}$$

$$\frac{}{\texttt{false}\ \mathsf{prop}}$$

蕴涵

$$\frac{\phi\ \mathsf{prop}\quad \psi\ \mathsf{prop}}{\texttt{imp}(\phi,\psi)\ \mathsf{prop}}$$

合取

$$\frac{\phi\ \mathsf{prop}\quad \psi\ \mathsf{prop}}{\texttt{and}(\phi,\psi)\ \mathsf{prop}}$$

析取

$$\frac{\phi\ \mathsf{prop}\quad \psi\ \mathsf{prop}}{\texttt{or}(\phi,\psi)\ \mathsf{prop}}$$

Abstract	*Concrete*
`true`	$\top$
`false`	$\bot$
$\texttt{and}(\phi_1;\phi_2)$	$\phi_1 \wedge \phi_2$
$\texttt{or}(\phi_1;\phi_2)$	$\phi_1 \vee \phi_2$
$\texttt{imp}(\phi_1;\phi_2)$	$\phi_1 \supset \phi_2$

6.1.1 构造逻辑(Constructive Logic)-5

❖命题逻辑

- ➢ 用于证明的规则
  - – 引入规则：由给定的连接词形成命题的"直接"证明
  - – 消去规则：由其他命题的"间接"证明形成命题的证明

证明守恒(conservation of proof)原理

这些规则是相互逆转的：

- – 消去规则只能提取引入规则所引入的信息（证明形式）
- – 可以使用引入规则构造证明，供消去形式使用。

- ➢ 永真规则：只有引入形式，没有消去形式

$$\frac{}{\Gamma \vdash \top \ \mathsf{true}} \quad (32.2)$$

6.1.1 构造逻辑(Constructive Logic)-6

❖ 命题逻辑

➢ 合取规则

– 引入规则

$$\frac{\Gamma \vdash \phi \text{ true} \quad \Gamma \vdash \psi \text{ true}}{\Gamma \vdash \phi \wedge \psi \text{ true}} \quad (32.3)$$

– 消去规则

$$\frac{\Gamma \vdash \phi \wedge \psi \text{ true}}{\Gamma \vdash \phi \text{ true}} \qquad \frac{\Gamma \vdash \phi \wedge \psi \text{ true}}{\Gamma \vdash \psi \text{ true}}$$

➢ 蕴涵规则

– 引入规则

$$\frac{\Gamma, \phi \text{ true} \vdash \psi \text{ true}}{\Gamma \vdash \phi \supset \psi \text{ true}} \quad (32.4)$$

– 消去规则

$$\frac{\Gamma \vdash \phi \supset \psi \text{ true} \quad \Gamma \vdash \phi \text{ true}}{\Gamma \vdash \psi \text{ true}}$$

6.1.1 构造逻辑(Constructive Logic)-7

❖ 命题逻辑

➢ 永假规则：没有引入形式，只有消去形式

$$\frac{\Gamma \vdash \perp \text{ true}}{\Gamma \vdash \phi \text{ true}} \quad (32.5)$$

➢ 析取规则

– 引入规则

$$\frac{\Gamma \vdash \phi \text{ true}}{\Gamma \vdash \phi \vee \psi \text{ true}} \qquad \frac{\Gamma \vdash \psi \text{ true}}{\Gamma \vdash \phi \vee \psi \text{ true}} \quad (32.6)$$

– 消去规则

$$\frac{\Gamma \vdash \phi \vee \psi \text{ true} \quad \Gamma, \phi \text{ true} \vdash \theta \text{ true} \quad \Gamma, \psi \text{ true} \vdash \theta \text{ true}}{\Gamma \vdash \theta \text{ true}}$$

6.1.1 构造逻辑(Constructive Logic)-8

- 使证明显式化——Curry-Howard同构的关键
  - 断言ϕ **true**表示ϕ有一个证明
  - 可用断言$p : \phi$取代ϕ **true**，表示p是ϕ的一个证明
  - 假言断言修改为 $x_1 : \phi_1, \ldots, x_n : \phi_n \vdash p : \phi$
  - 构造逻辑的规则可以用证明项重写如下
    Γ是一组形如$x_i : \phi_i$的假设，且Γ中不存在重复的变量x_i
    - 永真规则

$$\frac{}{\Gamma \vdash \mathtt{trueI} : \top}$$

(32.7)

    - 合取规则

$$\frac{\Gamma \vdash p : \phi \quad \Gamma \vdash q : \psi}{\Gamma \vdash \mathtt{andI}(p;q) : \phi \wedge \psi}$$

$$\frac{\Gamma \vdash p : \phi \wedge \psi}{\Gamma \vdash \mathtt{andEl}(p) : \phi}$$

$$\frac{\Gamma \vdash p : \phi \wedge \psi}{\Gamma \vdash \mathtt{andEr}(p) : \psi}$$

6.1.1 构造逻辑(Constructive Logic)-9

- 构造逻辑的规则可以用证明项重写如下

  Γ是一组形如$x_i : \phi_i$的假设，且Γ中不存在重复的变量x_i

  – 蕴涵规则 **(32.7)**

$$\frac{\Gamma, x:\phi \vdash p:\psi}{\Gamma \vdash \texttt{impI}[\phi](x.p):\phi \supset \psi} \qquad \frac{\Gamma \vdash p:\phi \supset \psi \quad \Gamma \vdash q:\phi}{\Gamma \vdash \texttt{impE}(p;q):\psi}$$

  – 永假规则

$$\frac{\Gamma \vdash p:\bot}{\Gamma \vdash \texttt{falseE}[\phi](p):\phi}$$

  – 析取规则

$$\frac{\Gamma \vdash p:\phi}{\Gamma \vdash \texttt{orIl}[\psi](p):\phi \vee \psi}$$

$$\frac{\Gamma \vdash p:\psi}{\Gamma \vdash \texttt{orIr}[\phi](p):\phi \vee \psi}$$

$$\frac{\Gamma \vdash p:\phi \vee \psi \quad \Gamma, x:\phi \vdash q:\theta \quad \Gamma, y:\psi \vdash r:\theta}{\Gamma \vdash \texttt{orE}[\phi;\psi](p;x.q;y.r):\theta}$$

6.1.2 命题作为类型-1

❖ 命题ϕ和其类型ϕ^*之间的对应关系

命题	类型	
$\top$	unit	空积类型
$\bot$	void	空和类型
$\phi \wedge \psi$	$\phi^* \times \psi^*$	二元积类型
$\phi \supset \psi$	$\phi^* \to \psi^*$	函数类型
$\phi \vee \psi$	$\phi^* + \psi^*$	二元和类型

❖ 证明和程序之间的对应关系

证明	程序	
trueI	triv	空积的引入形式
falseE[ϕ](p)	abort[ϕ^*](p^*)	空和的消去形式
andI(p;q)	pair(p^*,q^*)	二元积的引入形式
andEl(p)	fst(p^*)	二元积的消去形式
andEr(p)	snd(p^*)	二元积的消去形式

6.1.2 命题作为类型-2

❖命题ϕ和其类型ϕ^*之间的对应关系

命题	类型	
$\top$	unit	空积类型
$\bot$	void	空和类型
$\phi \wedge \psi$	$\phi^* \times \psi^*$	二元积类型
$\phi \supset \psi$	$\phi^* \to \psi^*$	函数类型
$\phi \vee \psi$	$\phi^* + \psi^*$	二元和类型

❖证明和程序之间的对应关系

证明	程序	
$\mathtt{impI}[\phi](x.p)$	$\mathtt{lam}[\phi^*](x.p^*)$	函数的引入形式
$\mathtt{impE}(p;q)$	$\mathtt{ap}(p^*,q^*)$	函数的消去形式
$\mathtt{orIl}[\psi](p)$	$\mathtt{in}[\mathtt{l}][\psi^*](p^*)$	二元和的引入形式
$\mathtt{orIr}[\phi](p)$	$\mathtt{in}[\mathtt{r}][\phi^*](p^*)$	二元和的引入形式
$\mathtt{orE}[\phi;\psi](p;x.q;y.r)$	$\mathtt{case}[\phi^*,\psi^*](p^*,x.q^*,y.r^*)$	二元和的消去形式

6.1.2 命题作为类型-3

❖ Curry-Howard同构(PFPL Theorem 32.1)

1.如果ϕ **prop**，则ϕ^* **type**;

2.如果$\Gamma \vdash p : \phi$，则$\Gamma^* \vdash p^* : \phi^*$。

➢ 上述定理反映出命题和类型,以及证明和程序之间的静态对应关系

➢ 进一步扩展得到动态对应关系：按下面方法(规定了证明的可计算内容)撤消消去和引入规则，可以产生程序的执行行为

$$
\begin{array}{rcl}
\mathtt{andEl}(\mathtt{andI}(p;q)) & \mapsto & p \\
\mathtt{andEr}(\mathtt{andI}(p;q)) & \mapsto & q \\
\mathtt{impE}(\mathtt{impI}[\phi](x.q);p) & \mapsto & [p/x]q \\
\mathtt{orE}[\phi;\psi](\mathtt{orIl}[\psi](p);x.q;y.r) & \mapsto & [p/x]q \\
\mathtt{orE}[\phi;\psi](\mathtt{orIr}[\phi](p);x.q;y.r) & \mapsto & [p/y]r
\end{array}
$$

6.2 经典证明和控制算符

构造逻辑：不接受排中律，判断命题为真的唯一标准是要为该命题构造一个证明。

经典逻辑：接受排中律，即$\phi \vee \neg\phi$ true。

经典逻辑比构造逻辑所能表达的范围要小。

例如，在经典逻辑中，命题$\phi \vee \neg\phi$不能说明存在ϕ的一个证明或反例，而只能说明不可能同时存在对ϕ的证明和反例。

6.2.1 经典逻辑

6.2.1 经典逻辑(Classical Logic)-1

❖ 经典逻辑

➢ 关心三种断言

– ϕ **true**: 表示ϕ是一个真命题

– ϕ **false**: 表示ϕ是一个假命题

– #: 表示一个已经推导出的矛盾

➢ 假言断言 $\phi_1 \text{ false}, \cdots, \phi_m \text{ false}; \psi_1 \text{ true}, \cdots, \psi_n \text{ true} \vdash J$

J: 是上述三种断言之一

➢ 三种断言的显式证明表示

– $p : \phi$ 表示p 是ϕ的一个证明

– $k \div \phi$ 表示k 是ϕ的一个反例

– $k \# p$ 表示k 和p 相矛盾

➢ 假言断言改为 $\underbrace{u_1 \div \phi_1, \cdots, u_m \div \phi_m}_{\Delta}; \underbrace{x_1 : \psi_1, \cdots, x_n : \psi_n}_{\Gamma} \vdash J$

J: 是上述显式证明形式之一

6.2.1 经典逻辑(Classical Logic)-2

❖ 静态语义

➢ 矛盾产生于证明和反例之间的冲突

$$\frac{\Delta;\Gamma \vdash k \div \phi \quad \Delta;\Gamma \vdash p : \phi}{\Delta;\Gamma \vdash k \# p} \qquad \textbf{(33.1)}$$

➢ 自反性

$$\frac{}{\Delta, u \div \phi;\Gamma \vdash u \div \phi} \qquad \frac{}{\Delta;\Gamma, x : \psi \vdash x : \phi}$$

➢ 永真与永假互补

ccr: call with current refutation
ccp: call with current proof

$$\frac{\Delta, u \div \phi;\Gamma \vdash k \# p}{\Delta;\Gamma \vdash \mathtt{ccr}(u \div \phi . k \# p) : \phi} \qquad \frac{\Delta;\Gamma, x : \phi \vdash k \# p}{\Delta;\Gamma \vdash \mathtt{ccp}(x : \phi . k \# p) \div \phi}$$

➢ 联结词规则组织成永真和永假的引入规则，后者相当于构造逻辑中的消去规则

$$\frac{}{\Delta;\Gamma \vdash \langle\rangle : \top} \qquad \frac{}{\Delta;\Gamma \vdash \mathtt{abort} \div \bot} \qquad \textbf{(33.1)}$$

6.2.1 经典逻辑(Classical Logic)-3

❖ 静态语义

➢ 联结词规则组织成永真和永假的引入规则，后者相当于构造逻辑中的消去规则 **(33.1)**

$$\frac{\Delta;\Gamma \vdash p:\phi \quad \Delta;\Gamma \vdash q:\psi}{\Delta;\Gamma \vdash \langle p,q\rangle:\phi\wedge\psi}$$

$$\frac{\Delta;\Gamma,x:\phi \vdash p:\psi}{\Delta;\Gamma \vdash \lambda(x{:}\phi.\,p):\phi\supset\psi}$$

$$\frac{\Delta;\Gamma \vdash k\div\phi}{\Delta;\Gamma \vdash \mathtt{fst};k\div\phi\wedge\psi}$$

$$\frac{\Delta;\Gamma \vdash p:\phi \quad \Delta;\Gamma \vdash k\div\psi}{\Delta;\Gamma \vdash \mathtt{app}(p);k\div\phi\supset\psi}$$

$$\frac{\Delta;\Gamma \vdash k\div\psi}{\Delta;\Gamma \vdash \mathtt{snd};k\div\phi\wedge\psi}$$

$$\frac{\Delta;\Gamma \vdash p:\phi}{\Delta;\Gamma \vdash \mathtt{inl}(p):\phi\vee\psi}$$

$$\frac{\Delta;\Gamma \vdash p:\psi}{\Delta;\Gamma \vdash \mathtt{inr}(p):\phi\vee\psi}$$

$$\frac{\Delta;\Gamma \vdash k\div\phi \quad \Delta;\Gamma \vdash l\div\psi}{\Delta;\Gamma \vdash \mathtt{case}(k,l)\div\phi\vee\psi}$$

$$\frac{\Delta;\Gamma \vdash k\div\phi}{\Delta;\Gamma \vdash \mathtt{not}(k):\neg\phi}$$

$$\frac{\Delta;\Gamma \vdash p:\phi}{\Delta;\Gamma \vdash \mathtt{not}(p)\div\neg\phi}$$

6.2.1 经典逻辑(Classical Logic)-4

❖ 动态语义：识别冲突的过程

➢ 抽象机的状态： ***k* # *p***

➢ 执行：对***k* # *p***约简

用矛盾状态之间的转换关系来归纳定义 **(33.2)**

$\texttt{fst};k \# \langle p,q\rangle \mapsto k \# p$	合取（二元积）
$\texttt{snd};k \# \langle p,q\rangle \mapsto k \# q$	合取（二元积）
$\texttt{case}(k,l) \# \texttt{inl}(p) \mapsto k \# p$	析取（二元和）
$\texttt{case}(k,l) \# \texttt{inr}(q) \mapsto l \# q$	析取（二元和）
$\texttt{app}(p);k \# \lambda(x:\phi.\,q) \mapsto k \# [p/x]q$	蕴涵（函数）
$\texttt{not}(p) \# \texttt{not}(k) \mapsto k \# p$	否定
$\texttt{ccp}(x:\phi.\,k \# p) \# q \mapsto [q/x]k \# [q/x]p$	
$k \# \texttt{ccr}(u \div \phi.\,l \# p) \mapsto [k/u]l \# [k/u]p$	

6.2.1 经典逻辑(Classical Logic)-5

❖ 动态语义：识别冲突的过程

$$\mathrm{ccp}(x:\phi.k\ \#\ p)\ \#\ q \mapsto [q/x]k\ \#\ [q/x]p$$

$$k\ \#\ \mathrm{ccr}(u \div \phi.l\ \#\ p) \mapsto [k/u]l\ \#\ [k/u]p$$

➢ 上面的规则会导致对 $\mathrm{ccp}(x:\phi.k\ \#\ p)\ \#\ \mathrm{ccr}(u \div \phi.l\ \#\ q)$ 有两种状态转换

– 第一种 $[r/x]k\ \#\ [r/x]p$　　r is $\mathrm{ccr}(u \div \phi.l\ \#\ q)$

– 第二种 $[m/u]l\ \#\ [m/u]q$　　m is $\mathrm{ccp}(x:\phi.k\ \#\ p)$

➢ 如何看待上述情况

– 将动态语义视为是非确定性的，即上述两种情况都可能

– 确定的动态语义：规定优先级

- 选择第一种：惰性证明语义
- 选择第二种：急切证明语义

6.2.1 经典逻辑(Classical Logic)-6

- 动态语义：识别冲突的过程
  - 执行的初始状态或结束状态是什么？
    - 对证明的急切解释
      假定初始(或终结)的反例为**halt**，然后形成 **halt** # p
    - 对反例的急切解释
      假定初始(或终结)的证明为**halt**，然后形成 k # **halt**

Thanks!