

# 型と論理学と証明

## Imdexpr

ドーモ、ミナサン

今日は進捗発表大会ということで

進捗……？

は？  
ねえよそんなの…



あ!?

何だって  
聞こえないー!

怪我のせいで  
耳が遠くて



泣きそう

# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答



# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答

型とは

「値に対するフォーマット」である



#とは

そもそもマシンと人間の差は？

# マシン側から見る(一部)

|    |          |   |                    |
|----|----------|---|--------------------|
| 1  | 0000000: | ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 | .....JFIF.....     |
| 2  | 0000010: | 00 01 00 00 ff db 00 43 00 08 06 06 07 06 05 08 | .....C.....        |
| 3  | 0000020: | 07 07 07 09 09 08 0a 0c 14 0d 0c 0b 0b 0c 19 12 | .....              |
| 4  | 0000030: | 13 0f 14 1d 1a 1f 1e 1d 1a 1c 1c 20 24 2e 27 20 | ..... \$. '        |
| 5  | 0000040: | 22 2c 23 1c 1c 28 37 29 2c 30 31 34 34 34 1f 27 | ",#..(7),01444. '  |
| 6  | 0000050: | 39 3d 38 32 3c 2e 33 34 32 ff db 00 43 01 09 09 | 9=82<.342...C...   |
| 7  | 0000060: | 09 0c 0b 0c 18 0d 0d 18 32 21 1c 21 32 32 32 32 | .....2!!2222       |
| 8  | 0000070: | 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 | 2222222222222222   |
| 9  | 0000080: | 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 | 2222222222222222   |
| 10 | 0000090: | 32 32 32 32 32 32 32 32 32 32 32 32 32 32 ff c0 | 2222222222222222.. |
| 11 | 00000a0: | 00 11 08 01 4e 02 52 03 01 22 00 02 11 01 03 11 | ....N.R..".....    |
| 12 | 00000b0: | 01 ff c4 00 1c 00 00 01 05 01 01 01 00 00 00 00 | .....              |
| 13 | 00000c0: | 00 00 00 00 00 00 02 00 01 03 04 05 06 07 08 ff | .....              |
| 14 | 00000d0: | c4 00 4b 10 00 01 03 03 03 02 03 05 05 05 05 06 | ..K.....           |
| 15 | 00000e0: | 04 04 07 00 01 00 02 03 04 05 11 12 21 31 06 41 | .....!1.A          |
| 16 | 00000f0: | 13 22 51 14 32 61 71 81 07 15 23 42 91 52 a1 b1 | ."Q.2aq...#B.R..   |
| 17 | 0000100: | c1 d1 24 33 62 72 f0 16 34 54 82 92 e1 25 36 43 | ..\$3br..4T...%6C  |
| 18 | 0000110: | 73 26 28 44 94 35 53 63 83 93 a2 d3 ff c4 00 19 | s&(D.5Sc.....      |
| 19 | 0000120: | 01 00 03 01 01 01 00 00 00 00 00 00 00 00 00 00 | .....              |
| 20 | 0000130: | 00 00 01 02 03 04 05 ff c4 00 27 11 00 02 02 02 | .....'.....        |
| 21 | 0000140: | 02 02 01 05 01 00 03 00 00 00 00 00 00 01 02 11 | .....              |
| 22 | 0000150: | 03 21 12 31 04 41 13 22 32 33 51 61 05 23 52 71 | !.1.A."23Qa.#Rq    |
| 23 | 0000160: | ff da 00 0c 03 01 00 02 11 03 11 00 3f 00 e6 1a | .....?...          |
| 24 | 0000170: | 72 0a 91 44 ce 3f 44 59 3e a5 79 2b a3 ba 84 9d | r..D.?DY>.y+....   |
| 25 | 0000180: | a5 a3 72 33 e8 99 24 00 e4 92 98 12 09 28 b1 b6 | ..r3..\$......(..  |
| 26 | 0000190: | 32 85 1a 04 cb 31 03 be 06 56 8d 34 32 b8 0d 2c | 2....1...V.42..,   |
| 27 | 00001a0: | 2b 2a 23 8e ea ed 3c a4 63 0e 3f aa 8b 2c db a5 | +*#...<.c.?...,..  |

86 行がフィルタ処理されました

1,1

先頭

# 人間側から見る



# マシン側から見る2(一部)

```
1 0000000: 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00 .ELF.....
2 0000010: 02 00 03 00 01 00 00 00 80 83 04 08 34 00 00 00 .....4...
3 0000020: 74 12 00 00 00 00 00 00 34 00 20 00 08 00 28 00 t.....4. ...(.
4 0000030: 1e 00 1b 00 06 00 00 00 34 00 00 00 34 80 04 08 .....4...4...
5 0000040: 34 80 04 08 00 01 00 00 00 01 00 00 05 00 00 00 4.....
6 0000050: 04 00 00 00 03 00 00 00 34 01 00 00 34 81 04 08 .....4...4...
7 0000060: 34 81 04 08 13 00 00 00 13 00 00 00 04 00 00 00 4.....
8 0000070: 01 00 00 00 01 00 00 00 00 00 00 00 00 80 04 08 .....
9 0000080: 00 80 04 08 94 0d 00 00 94 0d 00 00 05 00 00 00 .....
10 0000090: 00 10 00 00 01 00 00 00 00 10 00 00 00 90 04 08 .....
11 00000a0: 00 90 04 08 20 01 00 00 6c 35 f4 00 06 00 00 00 .... ...15.....
12 00000b0: 00 10 00 00 02 00 00 00 0c 10 00 00 0c 90 04 08 .....
13 00000c0: 0c 90 04 08 e8 00 00 00 e8 00 00 00 06 00 00 00 .....
14 00000d0: 04 00 00 00 04 00 00 00 48 01 00 00 48 81 04 08 .....H...H...
15 00000e0: 48 81 04 08 44 00 00 00 44 00 00 00 04 00 00 00 H...D...D.....
16 00000f0: 04 00 00 00 50 e5 74 64 74 0a 00 00 74 8a 04 08 ....P.tdt...t...
17 0001000: 74 8a 04 08 44 00 00 00 44 00 00 00 04 00 00 00 t...D...D.....
18 0001010: 04 00 00 00 51 e5 74 64 00 00 00 00 00 00 00 00 ....Q.td.....
19 0001020: 00 00 00 00 00 00 00 00 00 00 00 00 06 00 00 00 .....
20 0001030: 10 00 00 00 2f 6c 69 62 2f 6c 64 2d 6c 69 6e 75 .... /lib/ld-linu
21 0001040: 78 2e 73 6f 2e 32 00 00 04 00 00 00 10 00 00 00 x.so.2.....
22 0001050: 01 00 00 00 47 4e 55 00 00 00 00 00 02 00 00 00 ....GNU.....
23 0001060: 06 00 00 00 20 00 00 00 04 00 00 00 14 00 00 00 .....
24 0001070: 03 00 00 00 47 4e 55 00 b0 3a e3 cf 51 ac 90 4b ....GNU...:..Q..K
25 0001080: 8b 71 50 92 5d 51 a8 e1 cb 5a f5 29 02 00 00 00 .qP.]Q...Z.)....
26 0001090: 06 00 00 00 01 00 00 00 05 00 00 00 00 20 00 20 .....
27 00010a0: 00 00 00 00 06 00 00 00 ad 4b e3 c0 00 00 00 00 .....K.....
```

10 行がフィルタ処理されました

1,1

先頭

# 人間側から見る2

- 実は某東工大で行なわれる予定のプロコン予選問題回答プログラム
- つらみの塊と呼んでくれ



「画像もプログラムもマシンから見ると  
大差ないな」

つまり

コンピュータにものの判別はできない

それは

プログラムの中でももちろん一緒  
(データの区別はついていない)

だから

その基準を与えてやろう

ということでは



型ですよ！！！！

型ですよ！！！！

型ですよ！！！！

「あれ？でも論理学がどうのって聞いたのは？」  
「なんか難しい話いっぱい聞くけど、それは？」



# アジェンダ

1、型とは

2、型と論理学

3、型と証明

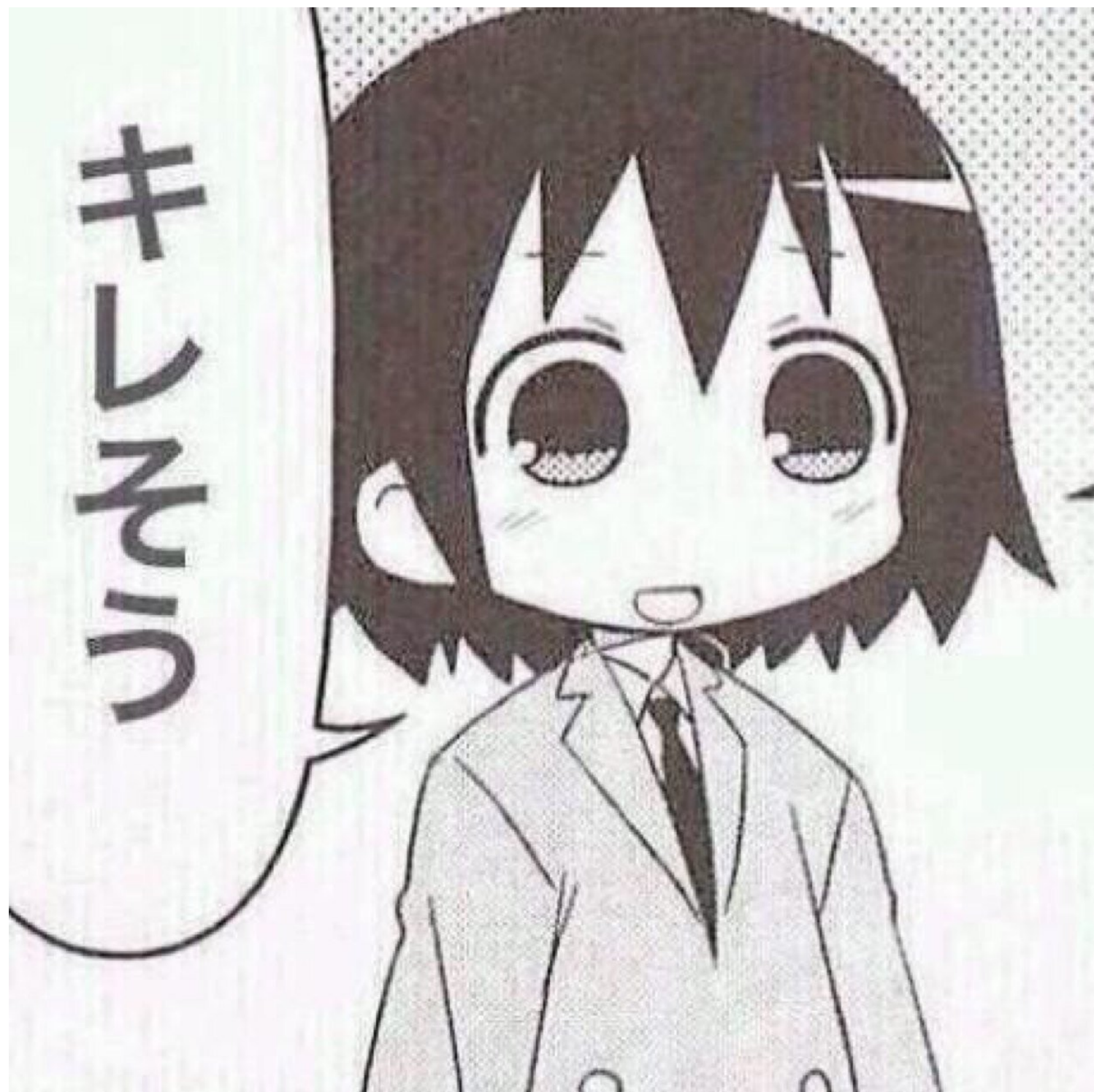
4、色々な型

5、型レベルプログラミング実演

6、質疑応答

# かたとろんりがくがどこにいるか

- カリー = ハワード 同型対応  
→ 型と証明の一対一対応
- 型推論  
→ ある値が何の型に属するのか？
- 意味論







# 型システム再考

型とは  
値に対するフォーマット

# 型システム再考



# 型システム再考

(いや、嘘ではないんだけど……)

# 型システム再考

<http://konn-san.com/prog/2013-advent-calendar.html>

# 型システム再考

目<

# 型システム再考

“型は不変条件である”

# 型システム再考





# 型システム再考

不変条件ならキャストとかできるのおかしいよね

# 型システム再考

つまり

# 型システム再考

(キャストができると)型安全でない

# 型システム再考

なぜならば

# 型システム再考

型は不変条件であるべき(前提)

# 型システム再考

不変条件でないなら型でない(対偶)

# 型システム再考

キャストすると不変条件にならない(補題)

# 型システム再考

キャストすると型でない！(結論)



# 型システム再考



論理学について少し話します

# 論理学とは

- 哲学の一分野
- 数理論理学、記号論理学と呼ばれる数学と合体したような分野が存在
- 論理学とは言ったものの今回は数理論理学について少しばかり説明します(型や証明に関係が深い)

# 数理論理学

- 命題論理
- 第一階述語論理
- 第二階述語論理  
証明論  
意味論
- 高階述語論理

# 命題論理

- やつは数理論理学の分野の中でも最弱...  
単純という意味です
- 公理をもたない(自然演繹も公理をもたないのでその体系を使うことも)
- 8つの推論規則

# 推論規則

- 二重否定の除去
- 論理積の導入
- 論理積の消去
- 論理和の導入
- 論理和の消去
- 前件肯定式
- 条件付き証明
- 背理法

具体例いきます★

# 二重否定の消去規則

$$\neg\neg\Phi$$

---

$$\Phi$$



# 論理積の導入規則

$A$

$B$

---

$A \wedge B$

# 論理積の除去規則 (1)

$$A \wedge B$$

---

$$A$$

## 論理積の除去規則 (2)

$$A \wedge B$$

---

$$B$$

以下省略

ね？簡単でしょう？

# 前提

- 型は小文字アルファベットで表記します
- $a \rightarrow b$  とは  $a$  を取って  $b$  を返す関数の型です
- $a \& b$  とは  $a$  と  $b$  の直積型です
- $a \mid b$  とは  $a$  と  $b$  の直和型です

# 直積型、直和型

- 直積型

e.g. (3, 4), (True, False, False)

Cで言う構造体です

- 直和型

e.g. Bool (= True | False)

Cで言う列挙型です

# 論理学復習

- 含意  $P \rightarrow Q$

$P$ の時 $Q$ であり、 $P$ でない時は常に真になる

- 論理積  $P \wedge Q$

$P$ と $Q$ が同時に真なら真

- 論理和  $P \vee Q$

$P$ もしくは $Q$ が真ならば真



# なんか対応してそう(小並感)

- 含意と  $a \rightarrow b$
- 論理積と  $a \& b$
- 論理和と  $a | b$
- And more ...

そして、証明へ...

# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答

# C-H同型対応(アバウトに)

- 型付き $\lambda$ 計算の圏  
デカルト閉圏
- 直観主義命題論理とその含意が成す圏  
これもデカルト閉圏
- 上二つの圏がうまく型と命題が対応してくれる！

# デカルト閉圏(CCC)

- 圏論において
- 終対象を持ち
- 任意の二対象に対してその直積を対象に持ち
- 任意の二対象に対してその冪対象を対象に持つ

○ ○ ○



# 定理証明系

- Coq

四色問題の証明等.有名.

- Agda

Haskellで実装された定理証明系

- Isabelle

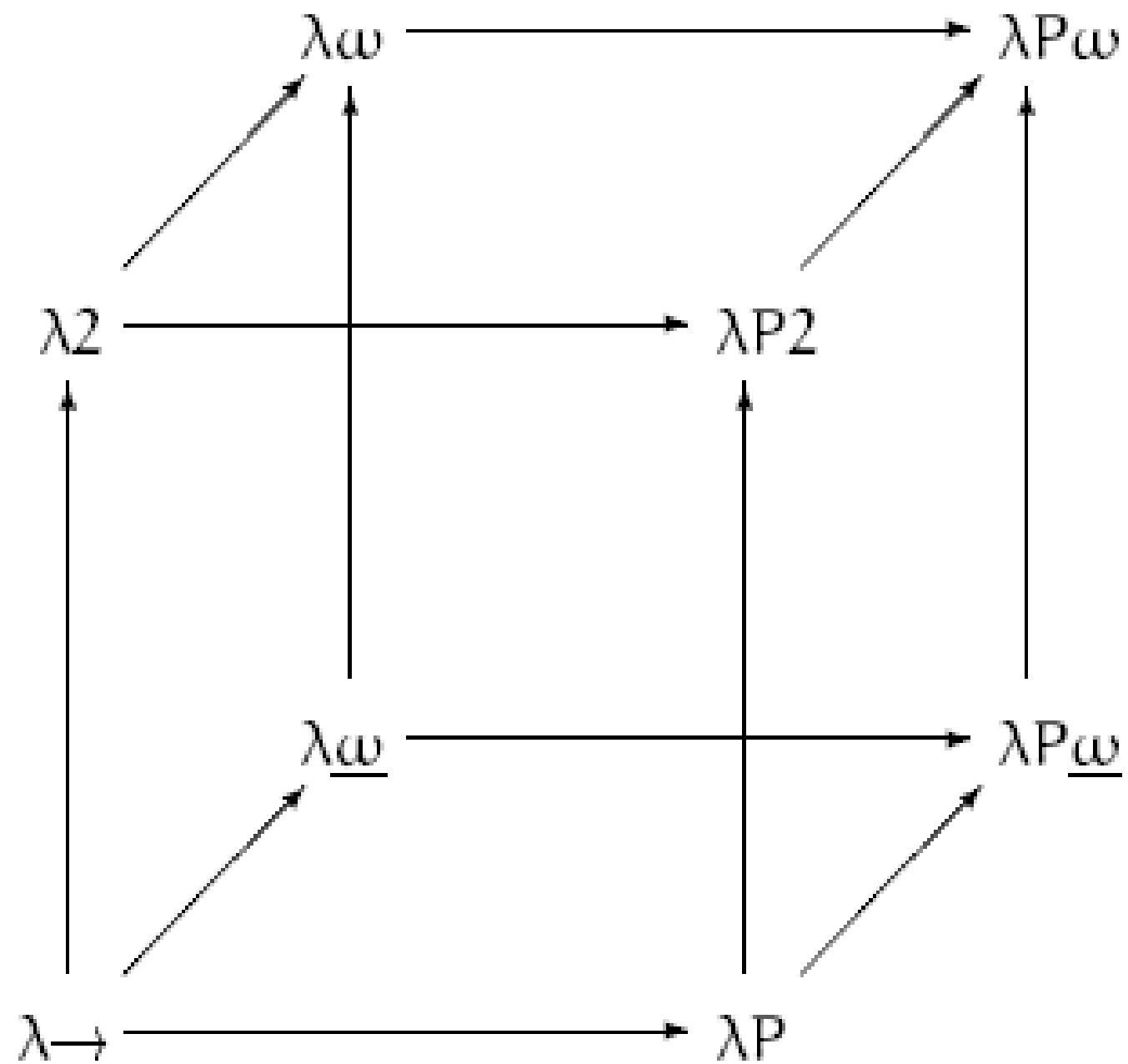
thorem advent calendarに怖い人がいたので  
そっち参照



次回までには証明ネタ仕入れるので  
Theorem advent calendarでも読んでてください...

# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答





11615

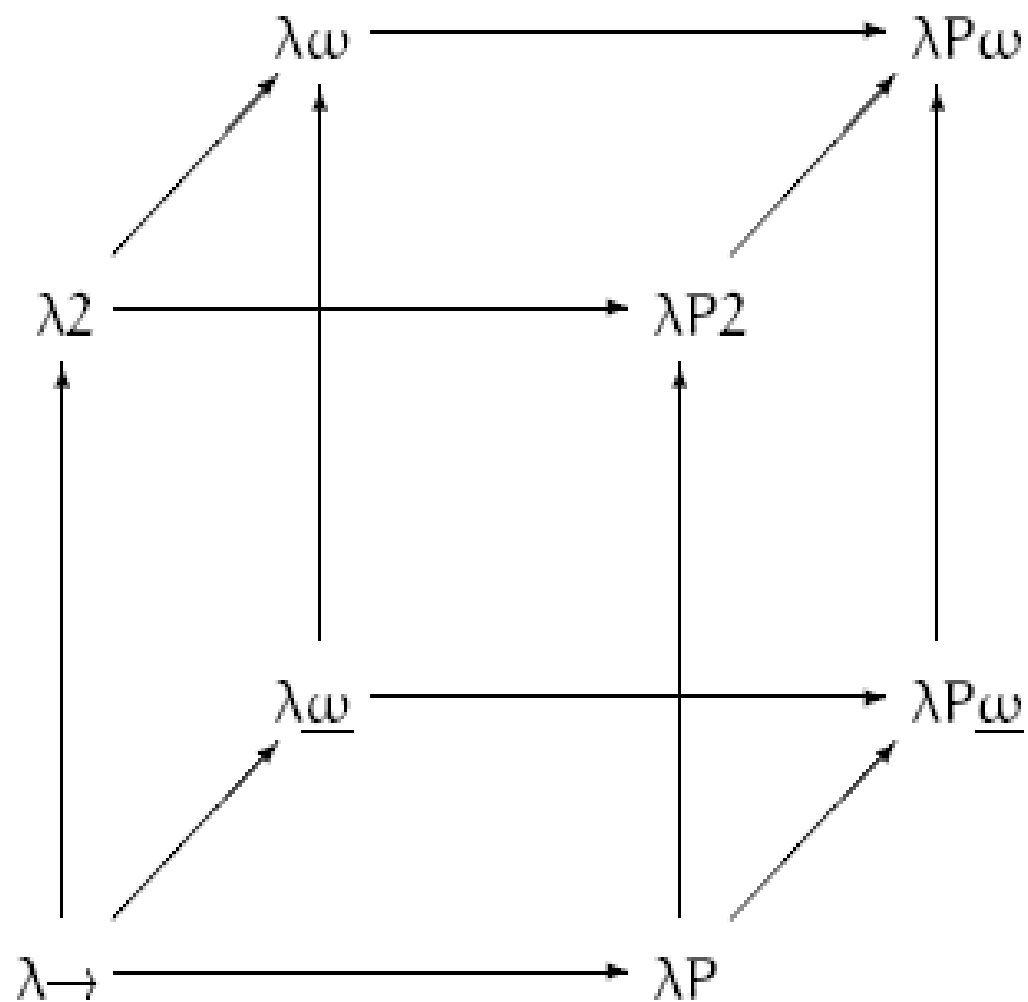


趣 100 150  
説明 2 1 1 1 1

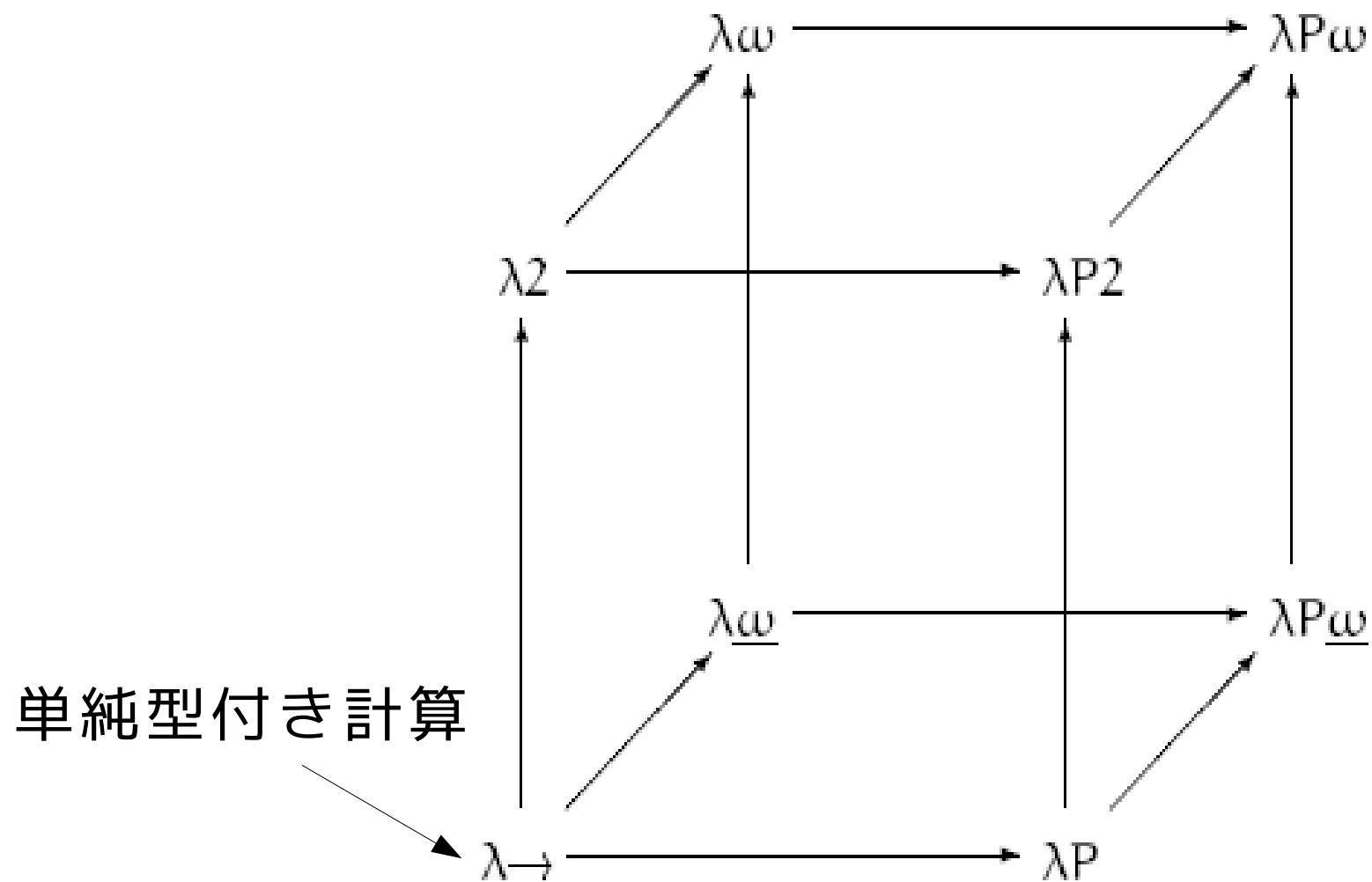


Allow me to tell you a tale.

# ラムダ・キューブ



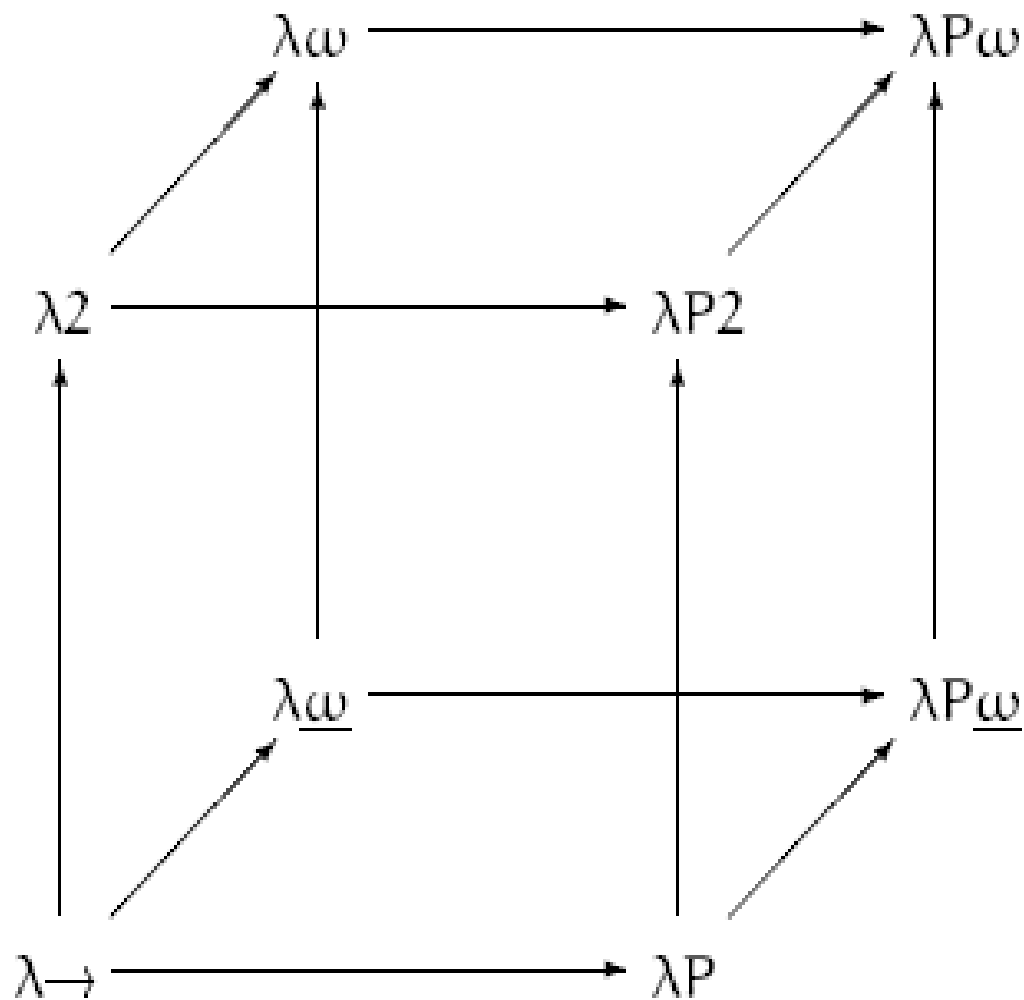
# ラムダ・キューブ



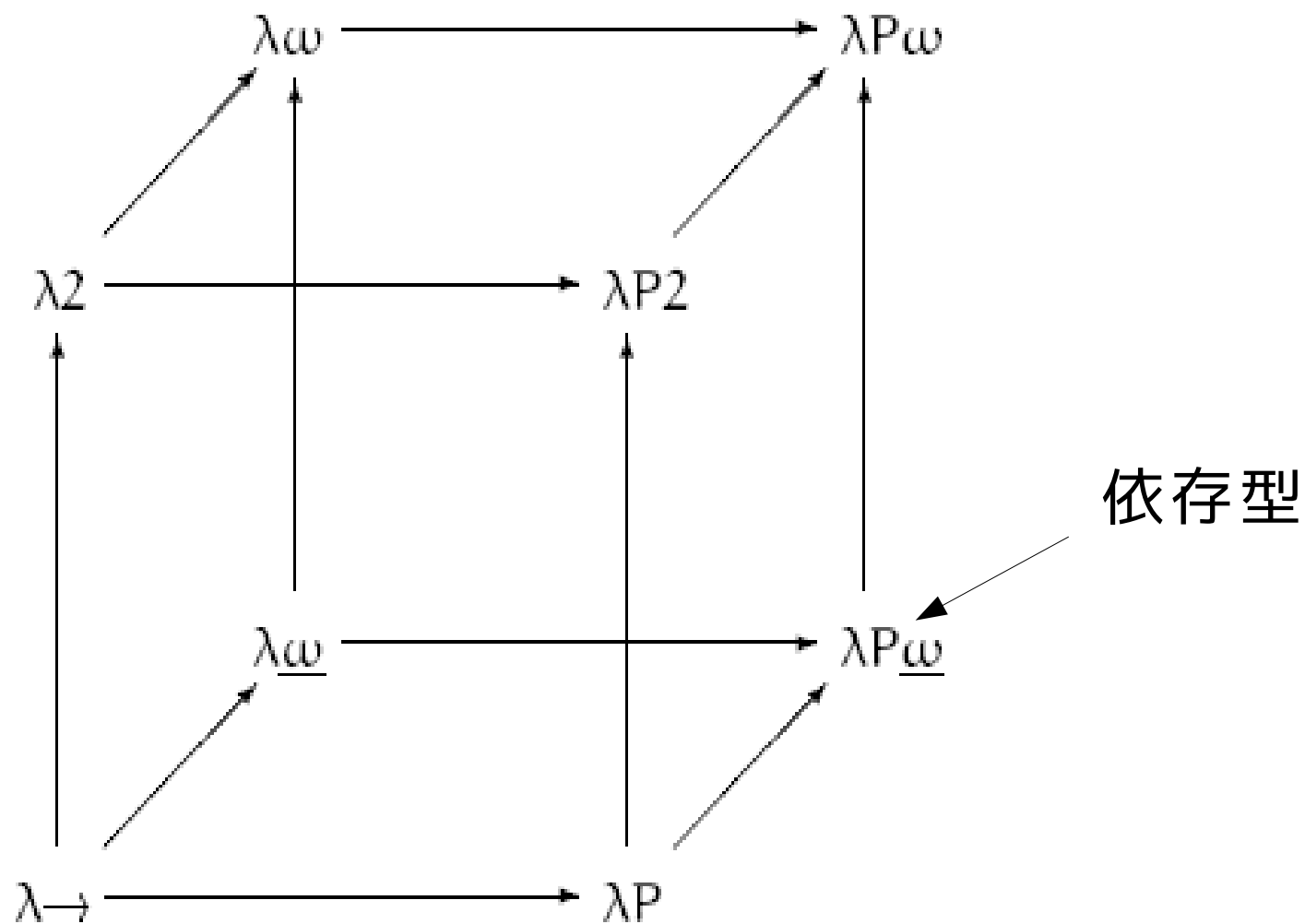


# ラムダ・キューブ

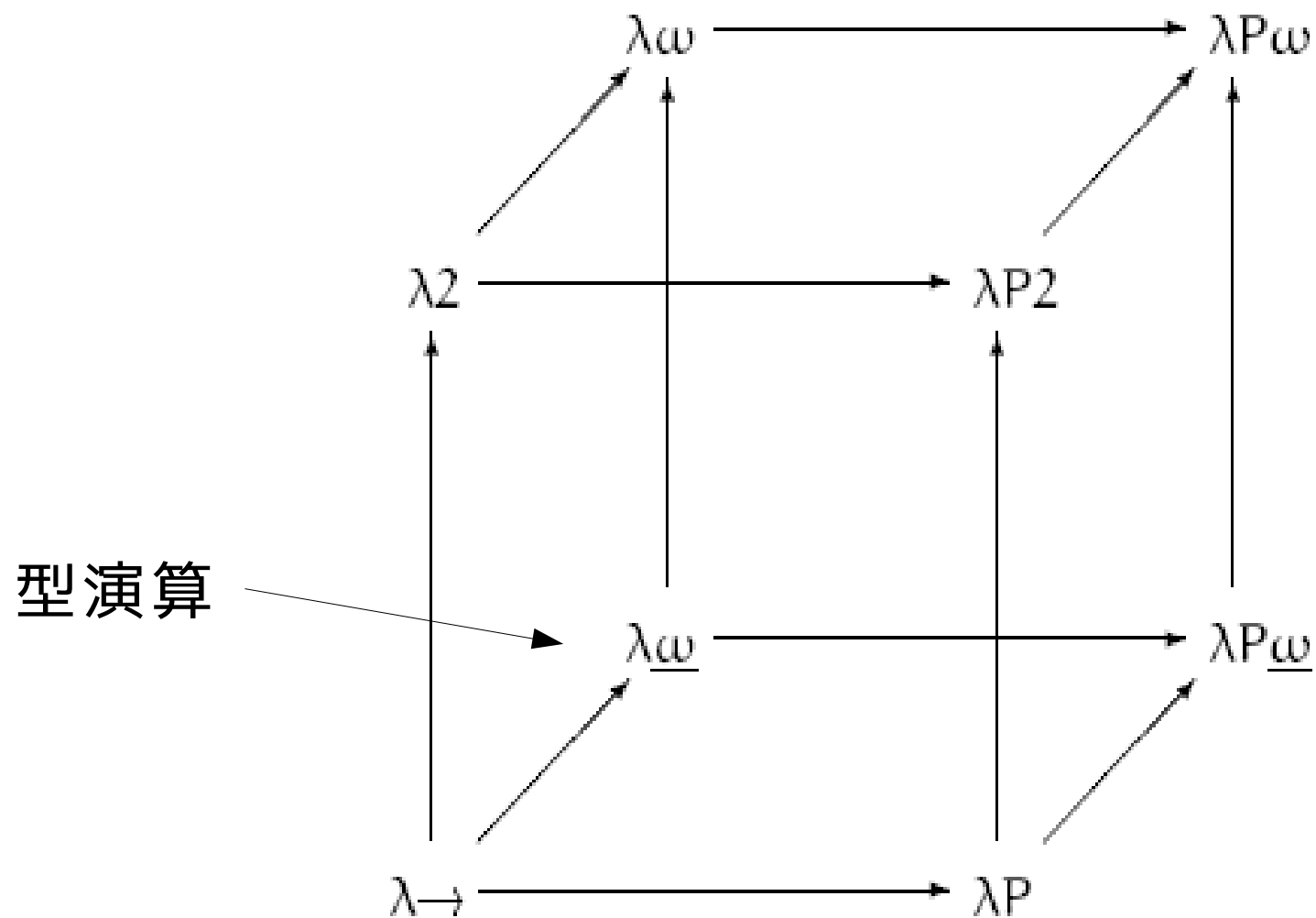
多相型  
(System F)



# ラムダ・キューブ



# ラムダ・キューブ



# 多相型<1>

- C++のテンプレート
  - C#のジェネリクス
  - Javaのポリモーフィズム
- 
- パラメータ多相
  - アドホック多相

# 多相型<2>

- パラメータ多相  
型の型を持つような変数を考えると

```
t = int;
```

```
t a;
```

- とかできそうだよね(感覚でok)
- これがパラメータ多相

# 多相型<3>

- アドホック多相
- 例えば型によってオーバーロードされているような関数はアドホック多相の恩恵を受けている
- 詳しくはHaskellの型クラスを学ぶとよいよ！

# 型クラス

- 数字(実数や整数)には同じ性質が幾つか存在
- 数字と文字も比較はできるし、同値関係も調べられる
- このように同じ関係のものを型クラスとしてインターフェイスを用意しておく機能をHaskellでは型クラスと呼びます

# 依存型

- 値に依存した型
- 例えばリスト

e.g. `List<n : int, int>` // int型で長さがnのリスト



## 依存型<2>

- なにが嬉しいのか
- headというリストの先頭を取る関数を考える
- 空リスト(要素のないリスト)があった時、headの返す値は未定義

# 依存型<3>

- ここで依存型を考えると
- [通常]  $\text{head} : \text{List}\langle a \rangle \rightarrow a$
- [依存型]  $\text{head} : \text{List}\langle n+1, a \rangle \rightarrow a$
- つまり、空リストを渡すと型レベルでエラーが出る

# 依存型：余談

- Haskellにはない
- CoqやAgda,Isabelleにはある
- Haskellにない理由
  - 型推論が決定不能になってしまうのです……

# 型演算

- 調べても出てこなかったので正直直感
- 例えばHaskellのkindは型演算(?)  
kindとは型の型

## 型演算<2>

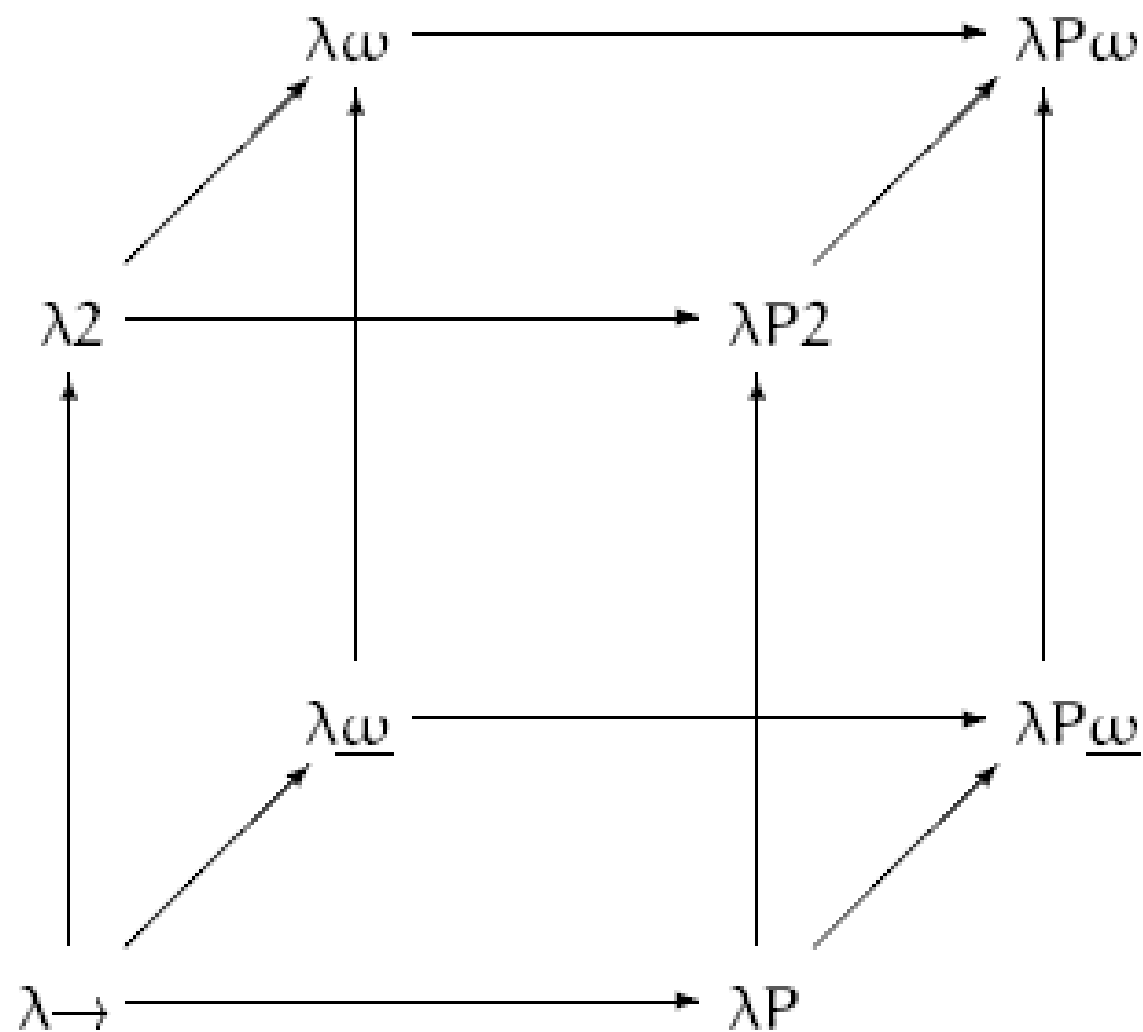
- 例えばInt型のkindは\*で示される  
\*とは具体型
- 関数型は $* \rightarrow *$ で示される
- Haskellは一旦忘れて  
直積型は $* \times *$ になるだろうし、  
直和型は $* + *$ になると思う

# 型演算<3>

型演算、一体何者なんだ……  
うごご……



これら三つを組み合わせる



# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答



アイエエエ！型  
レベルプログラ  
ミング！？型レ  
ベルプログラミ  
ングナンデ！？

# 今回、作るもの

- 定番中の定番、自然数  
(但し、この場合の自然数は0を含む)
- 0を表すZ
- ある自然数nに対して次を表わすS
- Add 足し算
- Mul かけ算

# 宣伝



トトリのアトリエ: アールランドの錬金術士2  
PS3, PSVitaにて好評発売中

# アジェンダ

- 1、型とは
- 2、型と論理学
- 3、型と証明
- 4、色々な型
- 5、型レベルプログラミング実演
- 6、質疑応答

# 参考資料

- Mr.konn氏のツイート

<http://togetter.com/li/608>

- Category theory – Steve Awodey
- プログラム意味論とかTaPLとか……



## 質疑応答

