對數偵錯

favonia
西風

id x = x

∀a.a→a

id 42 = 42

id "hi" = "hi"

id $\cancel{x = x}$

$\forall a.a \to a$

怎麼測試
它對不對？

id ~~x = x~~

怎麼測試
它對不對？

$\forall a. a \to a$

答案：測都不用測！

fst (x, y) = x

$\forall a. a \times a \to a$

怎麼測它對不對？

fst ~~(x, y) = x~~

$\forall a. a \times a \to a$

答：測 fst (0,1)！

怎麼測它對不對？

fst ~~(x, y) = x~~

$\forall a. a \times a \to a$

答：測 fst (0,1)！

怎麼測它對不對？

fst ~~(x, y) = x~~

$\forall a. a \times a \to a$

問題：為什麼
fst ("hi", "hi")
完全沒用？

再看一次...

$$\mathrm{fst}\ (x,\ y) = x$$

$$\forall a. a \times a \to a$$

```
fst (x, y) = x
  ∀a.∀b.a×b→a
```

怎麼測它對不對？

fst ~~(x, y) = x~~

$\forall a.\forall b.a \times b \to a$

答：不用測！

怎麼測它對不對？

fst ~~(x, y) = x~~

$\forall a. \forall b. a \times b \to a$

```
apply2 (f, x)
  = f (f x)
```

$\forall a.(a \to a) \times a \to a$

怎麼測它對不對？

$$\mathrm{apply2}\ (f, x)$$

$$\cancel{= f\ (f\ x)}$$

$$\forall a.(a \to a) \times a \to a$$

答：測 apply2 (+1, 0)

怎麼測它對不對？

apply2 (~~f, x~~)

~~= f (f x)~~

$\forall a.(a \to a) \times a \to a$

$$\forall a.\overbrace{a \times a \times \cdots \times a}^{n\text{個}} \to a$$

$$\mathrm{fst}\ (x_1,\ x_2,\ \ldots) = x_1$$

怎麼測它對不對？

$$\forall a.\overbrace{a \times a \times \cdots \times a}^{n個} \to a$$

fst (x [redacted]

怎麼測它對不對？

答：fst (0,1,...,n-1)

$$\forall a.\overbrace{a \times a \times \cdots \times a}^{n\text{個}} \to a$$

fst ([redacted]

怎麼測它對不對？

答：fst (0,1,...,n-1)

問題：改測
fst (0,1,...,1)
可不可以？

$$\forall a.\overbrace{a\times a\times\cdots\times a}^{n\text{個}}\to a$$

fst ([redacted]

$$\text{map } f\ [x_1,\ x_2,\ \ldots] = [f\ x_1,\ f\ x_2,\ \ldots]$$

$$\forall a. \forall b. (a \to b) \to \text{list}(a) \to \text{list}(b)$$

怎麼測它對不對？

$$\mathrm{map}\ f\ [x_1, x_2, \ldots]$$

$$= \cancel{[f\ x_1, f\ x_2, \ldots]}$$

$$\forall a. \forall b. (a \to b) \to \mathrm{list}(a) \to \mathrm{list}(b)$$

怎麼測它對不對？

map id []
map id [0]
map id [0,1]
map id [0,1,2]
map id [0,1,2,3]
map id [0,1,2,3,4]
...

$$\mathrm{map}\ f\ [x_1, x_2, \ldots]$$

$$= \cancel{[f\ x_1, f\ x_2, \ldots]}$$

$$\forall a. \forall b. (a \to b) \to \mathrm{list}(a) \to \mathrm{list}(b)$$

怎麼測它對不對？

```
map id []
map id [0]
map id [0,1]
map id [0,1,2]
map id [0,1,2,3]
map id [0,1,2,3,4]
...
```

$$\forall a.\forall b.(a\to b)\to list(a)\to list(b)$$

換位思考：可以如何惡搞？

怎麼測它對不對？

```
map id []
map id [0]
map id [0,1]
map id [0,1,2]
map id [0,1,2,3]
map id [0,1,2,3,4]
...
```

$$\forall a.\forall b.(a\to b)\to list(a)\to list(b)$$

換位思考：可以如何惡搞？

(1) 刪除某個元素 (2) 複製某個元素
(3) ???

怎麼測它對不對？

```
map id []
map id [0]
map id [0,1]
map id [0,1,2]
map id [0,1,2,3]
map id [0,1,2,3,4]
...
```

$$\forall a.\forall b.(a \to b) \to list(a) \to list(b)$$

換位思考：可以如何惡搞？

(1) 刪除某個元素 (2) 複製某個元素
(3) 改元素順序

以上都很容易一個一個去證
但還是要人工一個一個去想

以上都很容易一個一個去證
但還是要人工一個一個去想

不少人想自動找測資

Testing Polymorphic Properties
by Bernardy, Jansson, Claessen

可以處理 type 長得像 $\forall a.(F(a)\to a)\times(G(a)\to K)\to H(a)$ 的函式

metamorph (Haskell library) by Li-yao Xia

以上都很容易一個一個去證
但還是要人工一個一個去想

去年初我和我的學生王竹阳
發表了全自動找測資的方法

Logarithm and Program Testing @ POPL 2022
by Favonia and Zhuyang Wang

$$\forall a.\overbrace{a \times a \times \cdots \times a}^{n個} \to a$$

測 fst (0,1,...,n-1) 需要 n 個不同的值

$$\forall a.\underbrace{a\times a\times\cdots\times a}^{n\text{個}}\to a$$

測 fst (0,1,...,n-1) 需要 n 個不同的值

$$\log_a(a\times a\times\cdots\times a) = \log_a(a^n) = n$$

數有幾個「a」

$$\mathrm{apply2}\ (f,\ x) = f\ (f\ x)$$

$$\forall a.(a \to a) \times a \to a$$

$$\log_a((a \to a) \times a)$$

apply2 (f, x) = f (f x)

$$\forall a.(a \to a) \times a \to a$$

$$\log_a((a \to a) \times a)$$
$$= \log_a(a \to a) + \log_a(a)$$

apply2 (f, x) = f (f x)

$$\forall a.(a \to a) \times a \to a$$

$$
\begin{aligned}
&\log_a((a \to a) \times a) \\
&= \log_a(a \to a) + \log_a(a) \\
&= \log_a(a^a) + 1
\end{aligned}
$$

apply2 (f, x) = f (f x)

$$\forall a.(a \to a) \times a \to a$$

$$
\begin{aligned}
&\log_a((a \to a) \times a) \\
&= \log_a(a \to a) + \log_a(a) \\
&= \log_a(a^a) + 1 \\
&= a + 1
\end{aligned}
$$

apply2 (f, x) = f (f x)

$$\forall a.(a \to a) \times a \to a$$

$$\begin{aligned}
&\log_a((a \to a) \times a) \\
&= \log_a(a \to a) + \log_a(a) \\
&= \log_a(a^a) + 1 \\
&= a + 1
\end{aligned}$$

對數有 a 代表可遞迴產生 a 元素

$$\forall a.(a \to a) \times a \to a$$

$$\log_a((a \to a) \times a) = a + 1$$

對數有 a 代表可遞迴產生 a 元素

$$\forall a.(a \to a) \times a \to a$$

$$\log_a((a \to a) \times a) = a + 1$$

對數有 a 代表可遞迴產生 a 元素

可能產生 a 元素的方法：

x

f x

f (f x)

f (f (f x))

f (f (f (f x)))

…

$$\forall a.(a \to a) \times a \to a$$

$$\log_a((a \to a) \times a) = a + 1$$

對數有 a 代表可遞迴產生 a 元素

可能產生 a 元素的方法：

```
x        f (f x)          f (f (f (f x)))
f x      f (f (f x))      ...
```

```
data Elem = F Elem | X
```

$\forall a.(a \to a) \times a \to a$

$\log_a((a \to a) \times a) = a + 1$

對數有 a 代表可遞迴產生 a 元素

可能產生 a 元素的方法：

x	f (f x)	f (f (f (f x)))
f x	f (f (f x))	…

data Elem = F Elem | X

也就是 $\mu a.\log_a((a \to a) \times a) = \mu a.a+1$ = 自然數

定理

假設要測試一個多型函式，它的 type 是

$$\forall a. A(a) \to B(a)$$

如果 B(-) 是 functor 而且 $\mu a.\log_a(A(a))$ 有定義，那麼只需要考慮 a = 0 (空) 以及 $a = \mu a.\log_a(A(a))$ 兩種特例即可！

「B(-) 是 functor」代表 a 在 B(a) 裡面的「極性」是正的
某些奇怪函式需要多考慮 a = 0 不過再來我會假設函式都很 nice

對數

$\log_a(a) = 1$

$\log_a(1) = 0$

$\log_a(s \times t)$

$= \log_a s + \log_a t$

$\log_a(s \rightarrow t)$ 想成 t^s

$= s \times \log_a t$

對數

$\log_a(a) = 1$

$\log_a(1) = 0$

$\log_a(s \times t)$

$= \log_a s + \log_a t$

$\log_a(s \to t)$ 想成 t^s

$= s \times \log_a t$

對數++

$\log_a(b) = 0$

對數

$\log_a(a) = 1$

$\log_a(1) = 0$

$\log_a(s \times t)$

$= \log_a s + \log_a t$

$\log_a(s \to t)$ 想成 t^s

$= s \times \log_a t$

對數++

$\log_a(b) = 0$ 高估

$\log_a(0) = 0$

對數

$\log_a(a) = 1$

$\log_a(1) = 0$

$\log_a(s \times t)$

$= \log_a s + \log_a t$

$\log_a(s \to t)$ 想成 t^s

$= s \times \log_a t$

對數++

$\log_a(b) = 0$

$\log_a(0) = 0$

高估

$\log_a(s + t)$

$= \log_a s + \log_a t$

對數#

$\log_a(\mu b.X(b)) = ...$

對數#

$$\log_a(\mu b.X(b)) = ...$$

$\mu b.X(b)$ 是 $b \simeq X(b)$ 的解

如果兩邊同時取 $\log_a$

會得到 $\log_a b \simeq \log_a(X(b))$

而 $\mu b.X(b)$ 的 $\log_a$ 就是

$\log_a b \simeq \log_a(X(b))$ 的解

解也會長得像 $\mu b'....$

對數#

$\log_a(\mu b.X(b)) = \ldots.$

$\mu b.X(b)$ 是 $b \simeq X(b)$ 的解
如果兩邊同時取 $\log_a$
會得到 $\log_a b \simeq \log_a(X(b))$
而 $\mu b.X(b)$ 的 $\log_a$ 就是
$\log_a b \simeq \log_a(X(b))$ 的解
解也會長得像 $\mu b'\ldots.$

$$
\begin{aligned}
& \log_a(\mathrm{list}(a)) \\
= \; & \log_a(\mu X.1 + a \times X) \\
= \; & \mu X'.\log_a(1 + a \times X) \\
= \; & \mu X'.\log_a(1) + \log_a(a \times X) \\
= \; & \mu X'.0 + (1 + \log_a(X)) \\
\simeq \; & \mu X'.1 + X' \\
= \; & \text{自然數}
\end{aligned}
$$

詳情請
洽論文

定理

假設要測試一個多型函式，它的 type 是

$$\forall a.A(a) \to B(a)$$

如果 B(-) 是 functor 而且 $\mu a.\log_a(A(a))$ 有定義，那麼只需要測試 a = 0 (空) 以及 $a = \mu a.\log_a(A(a))$ 兩種特例即可！

等一下，就算知道要用什麼 type,
怎麼知道要用什麼當測資？

例如 $f : \forall a.a\times a\to a$ 雖然因為 $\log_a(a\times a)=2$ 知道只需要找一個至少有兩個元素的 type, 怎麼知道可以只測試 f (0,1) 而不用測 f (0,0), f (1,0), 以及 f (1,1)?

f(0,1) 喔啦　f(0,0) 沒用
f(1,0) 喔啦　f(1,1) 沒用

等一下，就算知道要用什麼 type,
怎麼知道要用什麼當測資？

例如 f : $\forall a. a \times a \to a$ 雖然因為 $\log_a(a \times a)=2$ 知道只需要找一個至少有兩個元素的 type, 怎麼知道可以只測試 f (0,1) 而不用測 f (0,0), f (1,0), 以及 f (1,1)?

答：可以盡量擺不同的元素，例如
(0,1) 或 (1,0) 就比 (0,0) 和 (1,1) 有用

論文中有定理 2.0, 不僅固定了 type 還指出有用的測資

map id []
map id [0]
map id [0,1]

map id [0,1,2]
map id [0,1,2,3]
map id [0,1,2,3,4]
...

定理 2.0 說這樣就夠了

答：可以盡量擺不同的元素，例如
(0,1) 或 (1,0) 就比 (0,0) 和 (1,1) 有用

論文中有定理 2.0, 不僅固定了 type 還指出有用的測資

遺珠

我們的理論在某些狀況下不會給出「最佳」解

例如 length : ∀a.list(a)→int 雖然 $\log_a(\mathrm{list}(a))$ = 自然數
一個只有元素的 type 就夠了

length []
length [0]
length [0,1]
…
不太酷

length []
length [★]
length [★,★]
…
酷

我們目前的定理
說自然數夠用，
是也沒錯，但是
可以做得更好！

遺珠

我們的理論還無法整合 typing 以外的資訊

例如 sortBy : $\forall a.(a \to a \to 2) \to \mathrm{list}(a) \to \mathrm{list}(a)$
一開始的 comparator 必須是 partial order
不需要考慮亂搞的 comparator

遺珠

我們的理論還無法整合 typing 以外的資訊

例如 sortBy : $\forall a.(a \to a \to 2) \to \mathrm{list}(a) \to \mathrm{list}(a)$
一開始的 comparator 必須是 partial order
不需要考慮亂搞的 comparator

我們的理論也還無法整合 fuzzing 等工具

實作

github.com/hawnzug/polycheck

好玩

用對數可以「還原」Church encoding

$\mu a.a+1$

$\updownarrow$

$\forall a.(a \to a) \times a \to a$

完

μa.log$_a$(input)