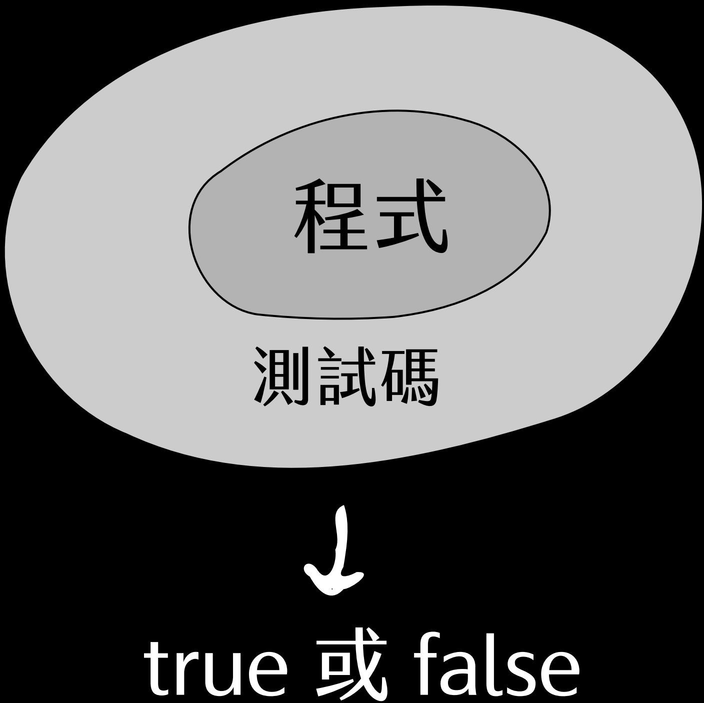
多型 polymorphism

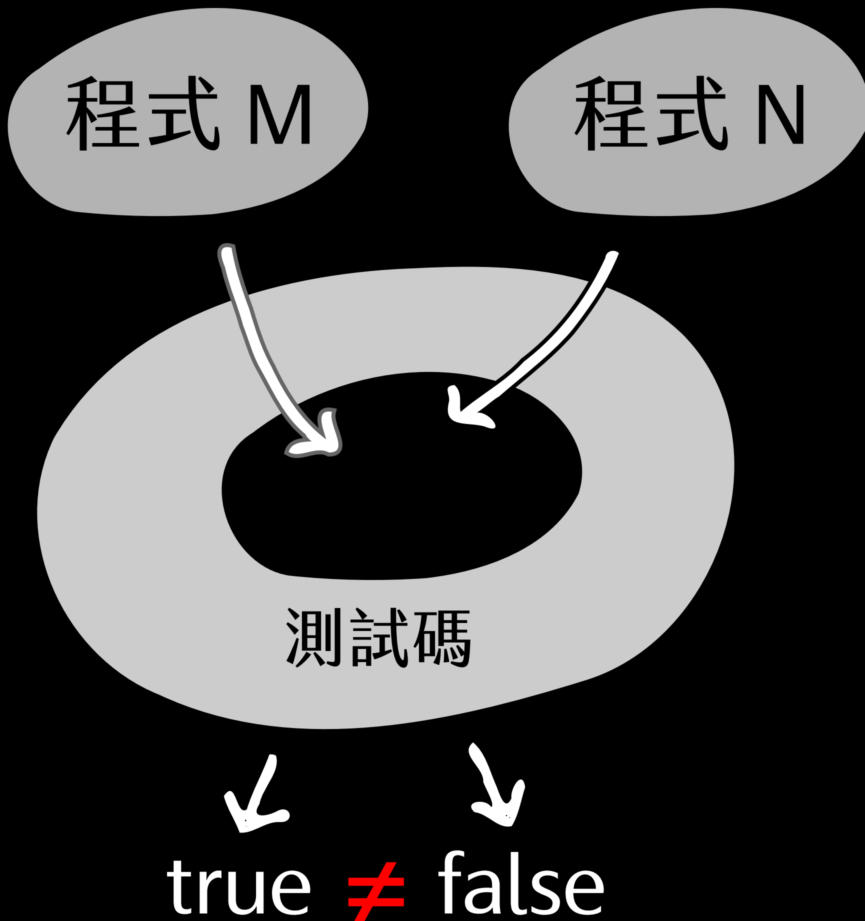
無知就是力量 Ignorance is Power

西風 Favonia
University of Minnesota
Twin Cities

程式 M = 程式 N

如何定義「相等」？


程式
測試碼
true 或 false


程式 M
程式 N
測試碼
true ≠ false

並無二致

observational equivalence

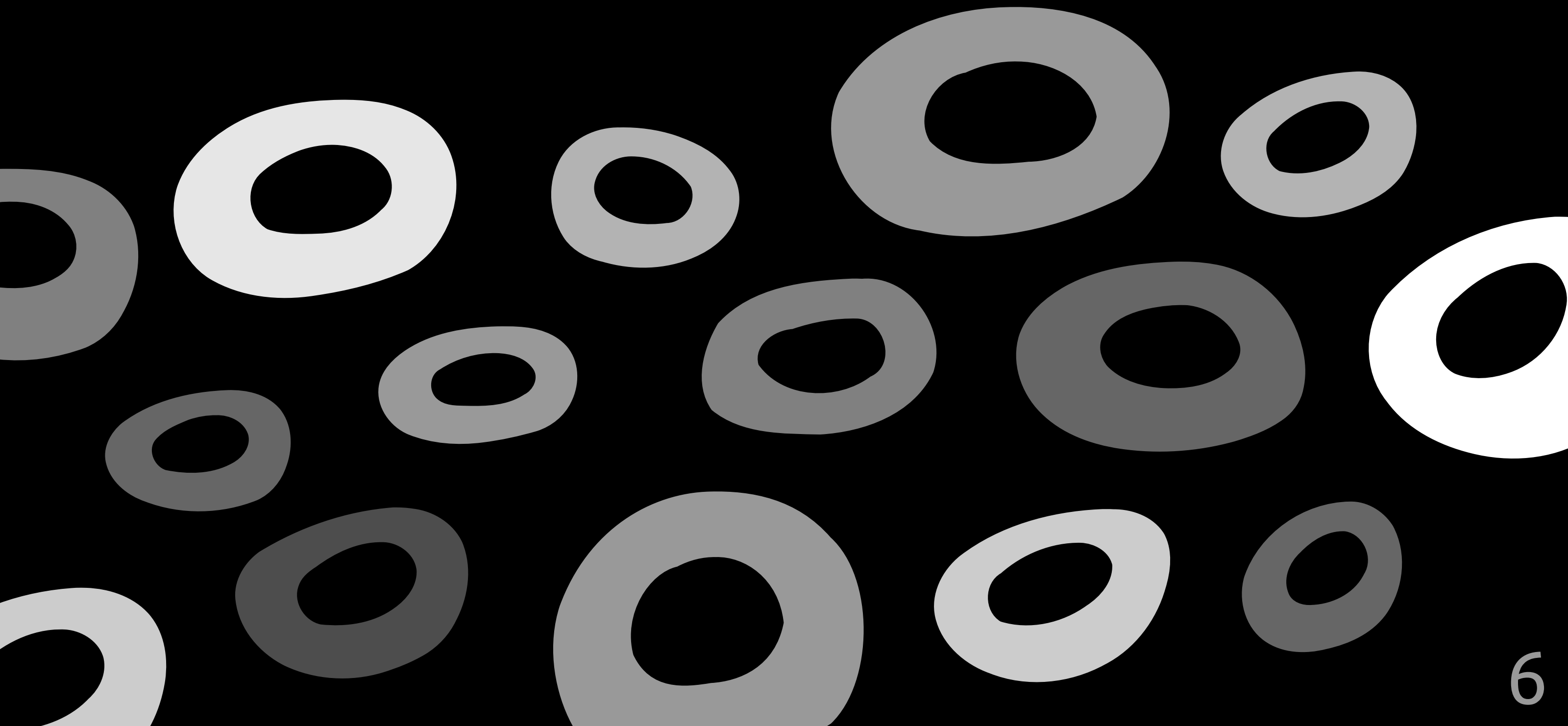
程式 M ≃ 程式 N

對於所有的測試

兩個一起得到 true 或是 false[註]

註：假如所有程式都會終止，沒有並行也不能存取記憶體

並無二致
observational equivalence



分類合符

程式 M ～ 程式 N

分類合符

程式 M ~ 程式 N

$M \sim_{bool} N$　同時 true 或 false

分類合符

程式 M ~ 程式 N

$M \sim_{\text{bool}} N$　同時 true 或 false

$M \sim_{A\times B} N$　第一個元素相同且第二個元素相同

$\pi_1(M) \sim_A \pi_1(N) \wedge \pi_2(M) \sim_B \pi_2(N)$

分類合符

程式 M ~ 程式 N

$M \sim_{\mathrm{bool}} N$　同時 true 或 false

$M \sim_{A\times B} N$　第一個元素相同且第二個元素相同

$\pi_1(M) \sim_A \pi_1(N) \land \pi_2(M) \sim_B \pi_2(N)$

$M \sim_{A\to B} N$　若輸入相同則輸出相同

$M' \sim_A N' \Rightarrow M(M') \sim_B N(N')$

分類合符

程式 M ~ 程式 N

$M \sim_{bool} N$　同時 true 或 false

$M \sim_{A\times B} N$　第一個元素相同且第二個元素相同

$\pi_1(M) \sim_A \pi_1(N) \land \pi_2(M) \sim_B \pi_2(N)$

$M \sim_{A\to B} N$　若輸入相同則輸出相同

$M' \sim_A N' \Rightarrow M(M') \sim_B N(N')$

Logical Equivalence

分類合符
logical equivalence

並無二致
observational equivalence

在所有程式語言中應該都要等價
某些麻煩的語言還在研究中

多型

$id : a \to a$

原封歸還

id 10 = 10

$map : (a \to b) \to list(a) \to list(b)$

各個擊破

map (+1) [1,2,3] = [2,3,4]

多型

$\mathrm{id} : \forall a.\ a \to a$

原封歸還

$\mathrm{id}\{\mathbb{N}\}\ 10 = 10$

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

各個擊破

$\mathrm{map}\{\mathbb{N}\}\{\mathbb{N}\}\ (+1)\ [1,2,3] = [2,3,4]$

多型合符

$M \sim_{bool} N$　同時 true 或 false

$\vdots$

$M \sim_{A \to B} N$　若輸入相同則輸出相同

多型合符

$M \sim_{\text{bool}} N$　同時 true 或 false

$\vdots$

$M \sim_{A \to B} N$　若輸入相同則輸出相同

$M \sim_{\forall a.A} N$　所有實例相同？

B closed type $\Rightarrow M\{B\} \sim_{A[B/a]} N\{B\}$?

多型合符

$M \sim_{\text{bool}} N$　同時 true 或 false

$\vdots$

$M \sim_{A \to B} N$　若輸入相同則輸出相同

$M \sim_{\forall a.A} N$　所有實例相同？

B closed type $\Rightarrow M\{B\} \sim_{A[B/a]} N\{B\}$？

$A[B/a]$ 可能比 $\forall a.A$ 還「大」！

考慮 $A = a$ 且 $B = \forall a.a$

多型合符

$M \sim_{\forall a.A} N\ [\eta]$ 所有候選實例相同

R good relation over B

$\Rightarrow M\{B\} \sim_A N\{B\}\ [\eta, a \mapsto R]$

多型合符

$M \sim_{\forall a.A} N\ [\eta]$ 所有候選實例相同

R good relation over B

$\Rightarrow M\{B\} \sim_A N\{B\}\ [\eta, a \mapsto R]$

參選條件

左右分別尊重 observational equivalence 即可

$M \simeq M' \land N \simeq N' \Rightarrow R(M,N) \Leftrightarrow R(M',N')$

本身不一定要是 equivalence

多型合符

$M \sim_{\forall a.A} N\ [\eta]$ 所有候選實例相同

R good relation over B

$\Rightarrow M\{B\} \sim_A N\{B\}\ [\eta, a \mapsto R]$

$M \sim_a N\ [\eta]$ 看 $\eta(a)(M, N)$ 怎麼說

參選條件

左右分別尊重 observational equivalence 即可

$M \simeq M' \wedge N \simeq N' \Rightarrow R(M,N) \Leftrightarrow R(M',N')$

本身不一定要是 equivalence

多型合符

$M \sim_{\forall a.A} N\ [\eta]$ 所有候選實例相同

R good relation between B and C

$\Rightarrow M\{B\} \sim_A N\{C\}\ [\eta, a \mapsto R]$

$M \sim_a N\ [\eta]$ 看 η(a)(M, N) 怎麼說

參選條件

左右分別尊重 observational equivalence 即可

$M \simeq M' \wedge N \simeq N' \Rightarrow R(M,N) \Leftrightarrow R(M',N')$

本身不一定是 equivalence，甚至可以跨類別

多型魔法

$X : \forall a.\ a \to a$

X 必然原封歸還

“ a → a” 代表 F 的輸出保留了輸入所有的性質，包括值本身

多型魔法

$X : \forall a.\ a \to a$

X 必然原封歸還

“ a → a” 代表 F 的輸出保留了輸入所有的性質，包括值本身

a M~M

X

a X(M)~X(M)

$R_a(O,P) \coloneqq O \simeq M$ 且 $P \simeq M$

多型魔法

$X : \forall a.\ a \to a$

X 必然原封歸還

定理：$X\{A\}\ M \simeq M$

多型魔法

$X : \forall a.\, a \to a$

X 必然原封歸還

定理：$X\{A\}\ M \simeq M$

首先 $X \sim_{\forall a.a \to a} X$

多型魔法

$X : \forall a.\, a \to a$

X 必然原封歸還

定理：$X\{A\}\ M \simeq M$

首先 $X \sim_{\forall a.a \to a} X$

選擇 $R(O,P) \coloneqq O \simeq P \simeq M$（只認 M）

得到 $X\{A\} \sim_{a \to a} X\{A\}\ [a \mapsto R]$

多型魔法

$X : \forall a.\, a \to a$

X 必然原封歸還

定理：$X\{A\}\ M \simeq M$

首先 $X \sim_{\forall a.a\to a} X$

選擇 $R(O,P) \coloneqq O{\simeq}P{\simeq}M$（只認 M）

得到 $X\{A\} \sim_{a\to a} X\{A\}\ [a \mapsto R]$

因為 $R(M,M)$ 所以 $M \sim_a M\ [a \mapsto R]$

所以 $X\{A\}\ M \sim_a X\{A\}\ M\ [a \mapsto R]$

多型魔法

$X : \forall a.\, a \to a$

X 必然原封歸還

定理：$X\{A\}\ M \simeq M$

首先 $X \sim_{\forall a.a\to a} X$

選擇 $R(O,P) \coloneqq O{\simeq}P{\simeq}M$（只認 M）

得到 $X\{A\} \sim_{a\to a} X\{A\}\ [a \mapsto R]$

因為 $R(M,M)$ 所以 $M \sim_a M\ [a \mapsto R]$

所以 $X\{A\}\ M \sim_a X\{A\}\ M\ [a \mapsto R]$

所以 $R(X\{A\}\ M, X\{A\}\ M)$

也就是 $X\{A\}\ M \simeq M$

多型魔法：測試

$F : \forall a.\ a \to a$

必然和 id 相等

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

如何確定是 map?

多型魔法：測試

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

考慮 id 和 [0,1,...,n] 就足夠

定理：如果 X id [0,...,n] ≃ [0,...,n]

那麼 X F $[E_0,\ldots,E_n] \simeq [F(E_0),\ldots,F(E_n)]$

多型魔法：測試

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

考慮 id 和 [0,1,...,n] 就足夠

定理：如果 $X\ \mathrm{id}\ [0,\ldots,n] \simeq [0,\ldots,n]$

那麼 $X\ F\ [E_0,\ldots,E_n] \simeq [F(E_0),\ldots,F(E_n)]$

a→b　　list(a)

list(b)

id~F　　$[0,\ldots,n] \sim [E_0,\ldots,E_n]$

X

$[0,\ldots,n] \sim [F(E_0),\ldots,F(E_n)]$

特例 ~ 一般

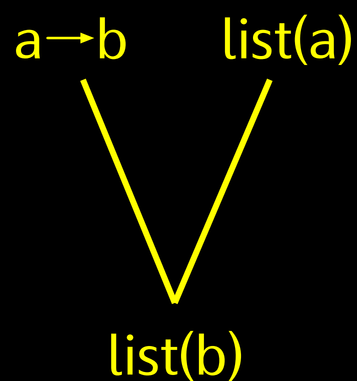
多型魔法：測試

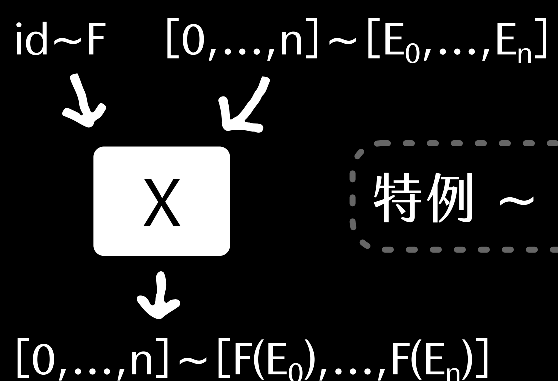
$X : \forall a.\forall b.\ (a \to b) \to list(a) \to list(b)$

考慮 id 和 [0,1,...,n] 就足夠

定理：如果 $X\ id\ [0,\ldots,n] \simeq [0,\ldots,n]$

那麼 $X\ F\ [E_0,\ldots,E_n] \simeq [F(E_0),\ldots,F(E_n)]$





特例 ~ 一般

$R_a(N,O) \coloneqq O \simeq E_N$ 且 $R_b(N,O) \coloneqq O \simeq F(E_N)$

多型魔法：測試

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

考慮 id 和 [0,1,...,n] 就足夠

定理：如果 $X\ id\ [0,\ldots,n] \simeq [0,\ldots,n]$

那麼 $X\ F\ [E_0,\ldots,E_n] \simeq [F(E_0),\ldots,F(E_n)]$

只考慮 id{bool} 以及所有的 list(bool) 也可

（還有很多種組合）

多型魔法：測試

$X : \forall a.\ (a \to \mathrm{bool}) \to \mathrm{list}(a) \to \mathrm{list}(a)$

多型魔法：測試

$X : \forall a.\ (a \to \text{bool}) \to \text{list}(a) \to \text{list}(a)$

怎麼確定 X 和 filter 一樣？

多型魔法：測試

$X : \forall a.\ (a \to \text{bool}) \to \text{list}(a) \to \text{list}(a)$

怎麼確定 X 和 filter 一樣？

考慮 id{bool} 以及所有的 list(bool) 即可

多型魔法：測試

$X : \forall a.\ (a \to bool) \to list(a) \to list(a)$

怎麼確定 X 和 filter 一樣？

考慮 id{bool} 以及所有的 list(bool) 即可

$X : \forall a. \forall b. \forall c.\ (a \to b \to c)$
$\to list(a) \to list(b) \to list(c)$

多型魔法：測試

$X : \forall a.\ (a \to \text{bool}) \to \text{list}(a) \to \text{list}(a)$

怎麼確定 X 和 filter 一樣？

考慮 id{bool} 以及所有的 list(bool) 即可

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

怎麼確定 X 和 zipWith 一樣？

多型魔法：測試

$X : \forall a.\ (a \to \mathrm{bool}) \to \mathrm{list}(a) \to \mathrm{list}(a)$

怎麼確定 X 和 filter 一樣？

考慮 id{bool} 以及所有的 list(bool) 即可

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

怎麼確定 X 和 zipWith 一樣？

考慮 $a = b = \mathbb{N}$, $c = \mathbb{N} \times \mathbb{N}$ 以及 $\lambda x. \lambda y. \langle x, y \rangle$

多型魔法：測試

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

多型魔法：測試

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

怎麼確定 X 和 fold 一樣？

多型魔法：測試

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

怎麼確定 X 和 fold 一樣？

考慮 $a = \mathbb{N}$, $b = \mathrm{list}(\mathbb{N})$ 以及 (::) 和 [] 即可

Testing Polymorphic Properties
[Bernardy, Jansson and Claessen]

多型魔法：測試

愈多型
愈少可能
愈不需要測

$\forall a.\ a \to a$ 比 $\mathbb{N} \to \mathbb{N}$ 還好測

多型為什麼會失敗

```
type(x).__name__
x instanceof A
dynamic_cast<A*>(x)
typeof x
```

功能不是愈多愈好

無知就是幸福

多型魔法：替換

一個元素型態為 A 的佇列資料結構

```
Q type 佇列的型態
empty : Q
insert : Q × A → Q
deque : Q → Q × A
```

註：先不管 deque 遇到空佇列的問題

多型魔法：替換

一個元素型態為 A 的佇列資料結構

```
Q type 佇列的型態
empty : Q
insert : Q × A → Q
deque : Q → Q × A
```

完整實作的型態 $\exists q.q \times (q \times A \to q) \times (q \to q \times A)$

∃ 可以由 ∀ 定義：$\exists a.A \coloneqq \forall b.(\forall a.A \to b) \to b$

註：先不管 deque 遇到空佇列的問題

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q. q \times (q \times A \to q) \times (q \to q \times A)$

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q. q \times (q \times A \to q) \times (q \to q \times A)$

`Q type`

R_q : Q_X 和 Q_Y 之間的對應關係

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q.q \times (q \times A \to q) \times (q \to q \times A)$

`empty : Q`

$\mathrm{empty}_X \sim_q \mathrm{empty}_Y\ [q \mapsto R_q]$

$R_q(\mathrm{empty}_X, \mathrm{empty}_Y)$

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q. q \times (q \times A \to q) \times (q \to q \times A)$

```
insert : Q × A → Q
```

$\text{insert}_X \sim_{q \times A \to q} \text{insert}_Y\ [q \mapsto R_q]$

若 $R_q(M_X, M_Y)$ 且 $E_X \simeq E_Y$ 則 $R_q(\text{insert}_X\langle M_X, E_X\rangle, \text{insert}_Y\langle M_Y, E_Y\rangle)$

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q.q \times (q \times A \to q) \times (q \to q \times A)$

`deque : Q → Q × A`

$\mathrm{deque}_X \sim_{q\to q\times A} \mathrm{deque}_Y\ [q \mapsto R_q]$

若 $R_q(M_X,M_Y)$ 則 $R_q(\pi_1(\mathrm{deque}_X(M_X)), \pi_1(\mathrm{deque}_Y(M_Y)))$

且 $\pi_2(\mathrm{deque}_X(M_X))\simeq\pi_2(\mathrm{deque}_Y(M_Y))$

註：先不管 deque 遇到空佇列的問題

多型魔法：替換

實作 X 和 Y 何時相等

$\exists q. q \times (q \times A \to q) \times (q \to q \times A)$

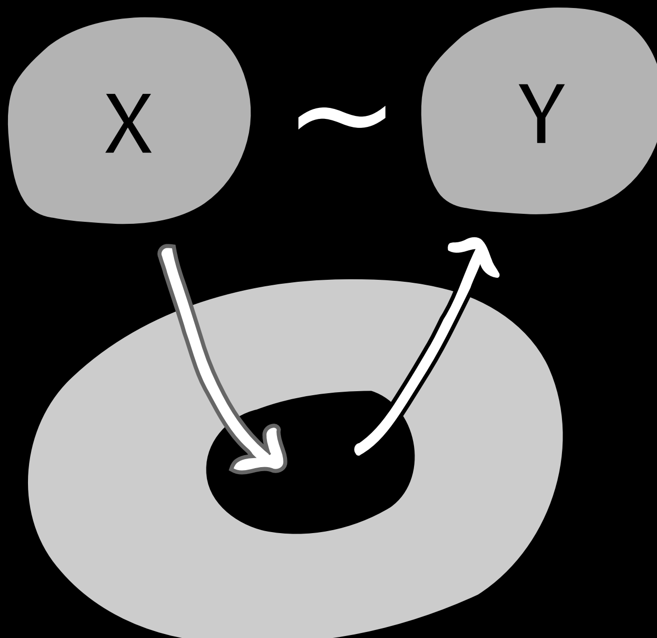
$\mathrm{empty}_X \sim_q \mathrm{empty}_Y \ [q \mapsto R_q]$

$\mathrm{insert}_X \sim_{q \times A \to q} \mathrm{insert}_Y \ [q \mapsto R_q]$

$\mathrm{deque}_X \sim_{q \to q \times A} \mathrm{deque}_Y \ [q \mapsto R_q]$

任何程式都無法區辨 X 和 Y！

多型魔法：替換



換內部完全不同（但等價）的實作來除錯

多型為什麼會失敗

```
type(x).__name__
x instanceof A
dynamic_cast<A*>(x)
typeof x
```

功能不是愈多愈好

語言「不能做什麼」也很重要