Polymorphism

A closer look at types....

Chap 8

polymorphism ≡ comes from Greek meaning 'many forms'

In programming:

Def: A function or operator is polymorphic if it has at least two possible types.

Polymorphism

i) Overloading

Def: An overloaded function name or operator is one that has at least two definitions, all of different types.

Example: In Java the ‘+’ operator is overloaded.

String s = “abc” + “def”;

+: String * String → String

int i = 3 + 5;

+: int * int → int

Polymorphism

Example: Java allows user defined polymorphism with overloaded function names.

```
bool f (char a, char b) {
        return a == b;
}
```

f : char * char → bool

```
bool f (int a, int b) {
        return a == b;
}
```

f : int * int → bool

Note: ML does not allow function overloading

Polymorphism

ii) Parameter Coercion

Def: An implicit type conversion is called a coercion.

Coercions usually exploit the type-subtype relationship because a widening type conversion from subtype to supertype is always deemed safe → a compiler can insert these automatically → type coercions.

Example: type coercion in Java

```
double x;
x = 2;
```

↖ the value 2 is coerced from int to double by the compiler

Polymorphism

Parameter coercion is an implicit type conversion on parameters. Parameter coercion makes writing programs easier – one function can be applied to many subtypes.

Example: Java

```
void f (double a) { ... }
```

$$\left.\begin{array}{l}\text{int} \subset \text{double}\\ \text{float} \subset \text{double}\\ \text{short} \subset \text{double}\\ \text{byte} \subset \text{double}\\ \text{char} \subset \text{double}\end{array}\right\} \text{all legal types that can be passed to function ‘f’.}$$

Note: ML does not perform type coercion (ML has no notion of subtype).

Polymorphism

iii) Parametric Polymorphism

Def: A function exhibits parametric polymorphism if it has a type that contains one or more type variables.

Example: ML

$$\begin{array}{l}\text{- fun f(x,y) = (x = y);}\\ \text{val f = fn : }\underbrace{\text{'a * 'a -> bool}}_{\substack{\text{polytype}\\ \text{(poly} \equiv \text{many)}}}\end{array}$$

Example: C++ and Java
C++ and Java have templates
that support parametric polymorphism.

Polymorphism

iv) Subtype Polymorphism

Def: A function or operator exhibits subtype polymorphism if one or more of its constructed types have subtypes.

Note: one way to think about this is that this is type coercion on constructed types.

Polymorphism

Example: Java

```
class Cup { ... };
class CoffeeCup extends Cup { ... };
class TeaCup extends Cup { ... };
```

```
TeaCup t = new TeaCup();
Cup c = t;
```

type coercion: TeaCup → Cup

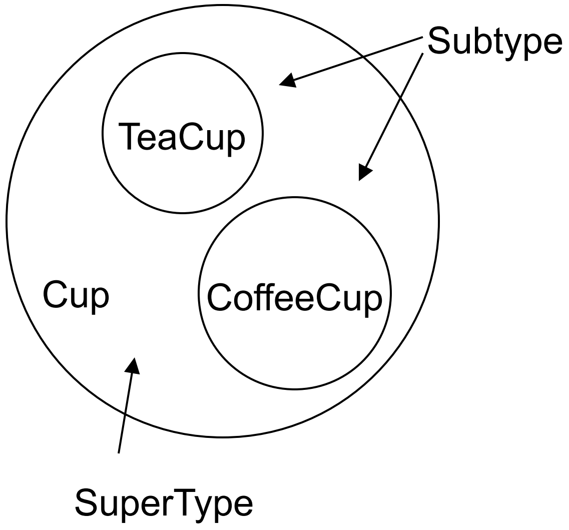
safe!

```
void fill (Cup c) {...}
```

```
TeaCup t = new TeaCup();
CoffeeCup k = new CoffeeCup();
```

```
fill(t);
fill(k);
```

subtype polymorphism


Subtype
TeaCup
Cup
CoffeeCup
SuperType