Programming Paradigms

Type Systems (Part 3)

Prof. Dr. Michael Pradel
Software Lab, University of Stuttgart
Summer 2020
Overview

- Introduction
- Types in Programming Languages
- Polymorphism
- Type Equivalence
- Type Compatibility
- Formally Defined Type Systems
Meaning of “Type”

Three interpretations

- **Denotational**: Set of values
- **Structural**: Built-in, primitive type or composite type created from simpler types
- **Abstraction-based**: Interface that provides a set of operations

In practice: Combination of all three
Polymorphism

- Greek origin: “Having multiple forms”
- Two kinds
  - Parametric polymorphism: Code takes (set of) type(s) as parameter
    - E.g., generics in Java, containers in C++
  - Subtype polymorphism: Extending or refining a supertype
    - E.g., subclasses in Java or C++
Demo

ParametricPolymorphism.java
Demo

SubtypePolymorphism.java
Polymorphic Variables

In some PLs, a single variable may refer to objects of completely different types.

Example (pseudo language):

```python
a = "abc"
b = 42
a = b
a = "def"
```
Polymorphic Variables

In some PLs, a **single variable** may refer to **objects of completely different types**

Example (pseudo language):

```plaintext
a = "abc"  // a holds a string
b = 42     // b holds an int
a = b      // a holds an int
a = "def"  // a holds a string (again)
```
Polymorphic Variables

In some PLs, a single variable may refer to objects of completely different types

Example (pseudo language):

```plaintext
a = "abc"     // a holds a string
b = 42        // b holds an int
a = b         // a holds an int
a = "def"     // a holds a string (again)
```

Type-correct in most dynamically typed (and even some statically typed) PLs
Special Types and Values

- **void type**: Indicates the absence of a type and has only one (trivial) value.

- **null value**: Means “does not hold a value of its type”.

- **Option types**: Indicates that the value may or may not hold a value of a specific type.

  - E.g., `Option[int]` in Python means `int` or `None`.
Quiz

Which of the following statements is true?

- A type system checks whether all types in a program are equivalent.
- PLs with dynamic scoping may be statically typed.
- Subclasses are a form of polymorphic typing.
- Option types cannot exist in strongly typed PLs.

Please vote in Ilias.
Quiz

Which of the following statements is true?

- A type system checks whether all types in a program are equivalent.
- PLs with dynamic scoping may be statically typed.
- Subclasses are a form of polymorphic typing.
- Option types cannot exist in strongly typed PLs.

Please vote in Ilias.
Quiz: Types

Which of the following statements is true?

- Types are compatible if and only if they are equal
- Coercions mean that a programmer casts a value from one type to another type
- Type conversions are guaranteed to preserve the meaning of a value
- PLs with type inference may provide static type guarantees

Please vote in Ilias.
Quiz: Types

Which of the following statements is true?

- Types are compatible if and only if they are equal
- Coercions mean that a programmer casts a value from one type to another type
- Type conversions are guaranteed to preserve the meaning of a value
- PLs with type inference may provide static type guarantees

Please vote in Ilias.