Programming Paradigms

Type Systems (Part 2)

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

Strongly Typed PLs

PL implementation enforces:

Operations only on values of proper type

- Most PLs since 1970s
- C is mostly strongly typed
  - Exceptions, e.g.,:
    - Subroutines with variable number of parameters
    - Interoperability of pointers and arrays

Statically Typed PLs

Strongly typed and checked at compile-time

- Strictly speaking, practically no PL is statically typed
  - E.g., Java: Upcasts and reflection allow for runtime type errors
- In practice, means ”mostly statically typed”

Demo: Casts.java

Dynamically Typed PLs

Type checking is delayed until runtime

- Type errors found only later in development process
- Common in “scripting languages”, e.g., JavaScript and Python
- Note: Every value has a type and type errors manifest as runtime errors

Gradual Typing

Middleground between statically and dynamically typed PLs

- Annotating types is optional
  - Can quickly write code and add types later
- Static type checker warns about errors obvious from the available types
  - No guarantee to find all type errors

Demo: gradual_typing.py

Quiz: * Typed PLs

What's the outcome of compiling and running this code in

- a strongly typed language?
- a statically typed language?
- a dynamically typed language?

```
a = 23;
b = true;
c = a + a;
d = c - b;
print(d);
```

Please vote in Ilias.

Quiz: * Typed PLs

What's the outcome of compiling and running this code in

- a strongly typed language?
- a statically typed language?
- a dynamically typed language?

```
a = 23;
b = true;
c = a + a;
d = c - b;
print(d);
```

Strongly typed language: Type error

Please vote in Ilias.

Quiz: * Typed PLs

What's the outcome of compiling and running this code in

- a strongly typed language?
- a statically typed language?
- a dynamically typed language?

```
a = 23;
b = true;
c = a + a;
d = c - b;
print(d);
```

Statically typed language: Compile-time type error

Please vote in Ilias.

Quiz: * Typed PLs

What's the outcome of compiling and running this code in

- a strongly typed language?
- a statically typed language?
- a dynamically typed language?

```
a = 23;
b = true;
c = a + a;
d = c - b;
print(d);
```

Dynamically typed language: Runtime type error

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.