# Subtyping
# Intersection Types
# Union Types

Advanced Compiler Construction and Program Analysis

**Lecture 7**

Innopolis University, Spring 2022

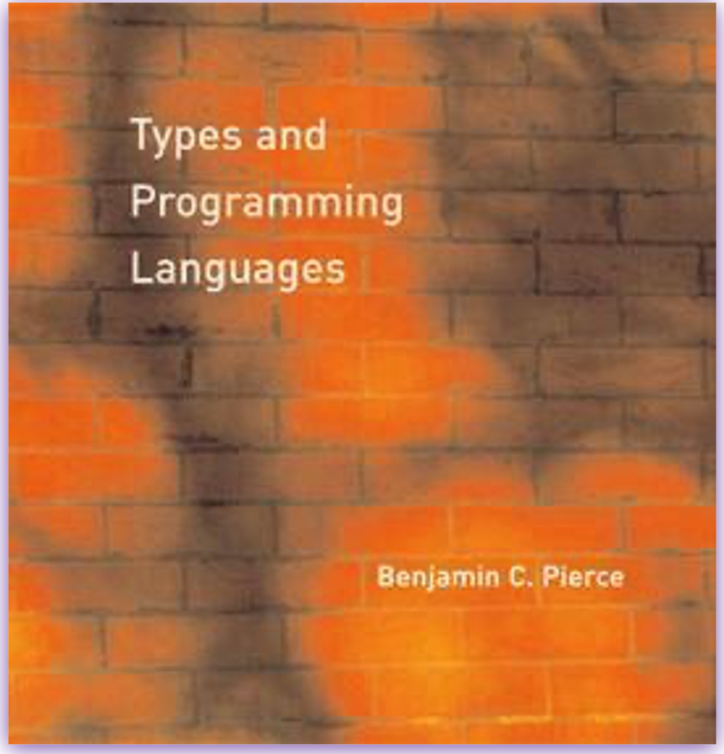
# The topics of this lecture are covered in detail in...

Benjamin C. Pierce.

**Types and Programming Languages**

MIT Press 2002





# Well-behaved but ill-typed

Consider the following term

```
(λr:{x:Nat}. r.x) {x=0,y=1}
```

# Well-behaved but ill-typed

Consider the following term

```
(λr:{x:Nat}. r.x) {x=0,y=1}
```

If we forget the types, it is well-behaved, but it is ill-typed since the actual argument has type `{x:Nat, y:Nat}`.

# Well-behaved but ill-typed

Consider the following term

```
(λr:{x:Nat}. r.x) {x=0,y=1}
```

If we forget the types, it is well-behaved, but it is ill-typed since the actual argument has type `{x:Nat, y:Nat}`.

Note that it is **always safe** to apply the function above to an argument of type `{x:Nat, y:Nat}`!

# Well-behaved but ill-typed

Consider the following term

```
(λr:{x:Nat}. r.x) {x=0,y=1}
```

If we forget the types, it is well-behaved, but it is ill-typed since the actual argument has type `{x:Nat, y:Nat}`.

Note that it is **always safe** to apply the function above to an argument of type `{x:Nat, y:Nat}`!

**Subtyping** offers one way to fix this kind of problems
by refining the typing rules.

# Subtyping: idea

**Principle of safe substitution.**
**S** is a subtype of **T** if any term `s:S` is safe to be used in any context where a term `t:T` is expected.

# Subtyping: idea

**Principle of safe substitution.**
**S** is a subtype of **T** if any term `s:S` is safe to be used in any context where a term `t:T` is expected.

**Intuition via subset semantics.**
**S** is a subtype of **T** if for any term **s∈S**, we also have **s∈T**.

# Subtyping: idea

**Principle of safe substitution.**
**S** is a subtype of **T** if any term **`s:S`** is safe to be used in any context where a term **`t:T`** is expected.

**Intuition via subset semantics.**
**S** is a subtype of **T** if for any term **s∈S**, we also have **s∈T**.

**Subsumption typing rule.**

$$\frac{\Gamma \vdash t : S \qquad S <: T}{\Gamma \vdash t : T}$$

# Subtyping: idea

**Principle of safe substitution.**
**S** is a subtype of **T** if any term **`s:S`** is safe to be used in any context where a term **`t:T`** is expected.

**Intuition via subset semantics.**
**S** is a subtype of **T** if for any term **s∈S**, we also have **s∈T**.

**Subsumption typing rule (example).**

$$\frac{\Gamma \vdash t : \{x{:}Nat, y{:}Nat\} \qquad \{x{:}Nat, y{:}Nat\} <: \{x{:}Nat\}}{\Gamma \vdash t : \{x{:}Nat\}}$$

# Subtyping relation



`S <: T`

Subtype

Supertype

# Subtyping relation

$\boxed{S <: T}$

$\boxed{S <: S}$

# Subtyping relation

$$S <: T$$

$$S <: S$$

$$\frac{S <: U \qquad U <: T}{S <: T}$$

# Subtyping relation: records

$S <: T$

$\{x : T_1, y : T_2\} <: \{x : T_3\}$

# Subtyping relation: records

$$\boxed{S <: T}$$

$$\boxed{\{x:T_1, y:T_2\} <: \{x:T_3\}}$$

$\{x:Nat, y:Nat\}$ $\{x:Nat\}$

# Subtyping relation: records

$S <: T$

$\{x:T_1, y:T_2\} <: \{x:T_3\}$

{x:Nat,y:Nat}

{x:Nat}

…
{x=1,y=2}
{x=1,y=2,z=false}
…

# Subtyping relation: records

$S <: T$

$\{x{:}T_1, y{:}T_2\} <: \{x{:}T_3\}$

{x:Nat,y:Nat}

{x:Nat}

…
{x=1,y=2}
{x=1,y=2,z=false}
…

∈

…
{x=1}
{x=1,y=2}
{x=1,y=2,z=false}
{x=1,a=false}
…

# Subtyping relation: records

$\boxed{\texttt{S <: T}}$

$$\boxed{\texttt{\{x:}T_1\texttt{,y:}T_2\texttt{\} <: \{x:}T_3\texttt{\}}}$$

| `{x:Nat,y:Nat}` | | `{x:Nat}` |
|---|---|---|
| … | | … |
| `{x=1,y=2}` | | `{x=1}` |
| `{x=1,y=2,z=false}` | ∈ | `{x=1,y=2}` |
| … | | `{x=1,y=2,z=false}` |
| | | `{x=1,a=false}` |
| | | … |
| **smaller type** | | **larger type** |

# Subtyping relation: records

$S <: T$

$\{x:T_1, y:T_2\} <: \{x:T_3\}$

| {x:Nat,y:Nat} | | {x:Nat} |
| --- | --- | --- |
| … | | … |
| | | {x=1} |
| {x=1,y=2} | | {x=1,y=2} |
| {x=1,y=2,z=false} | ∈ | {x=1,y=2,z=false} |
| | | {x=1,a=false} |
| … | | … |
| **smaller type** | | **larger type** |
| **more fields** | | **less fields** |

# Subtyping relation: records

`S <: T`

$$\{l_1{:}T_1,\ldots,l_{n+k}{:}T_{n+k}\} <: \{l_1{:}T_1,\ldots,l_n{:}T_n\}$$

| `{x:Nat,y:Nat}` | | `{x:Nat}` |
|---|---|---|
| … | | … |
| `{x=1,y=2}` | | `{x=1}` |
| `{x=1,y=2,z=false}` | ∈ | `{x=1,y=2}` |
| … | | `{x=1,y=2,z=false}` |
| | | `{x=1,a=false}` |
| | | … |
| **smaller type** | | **larger type** |
| **more fields** | | **less fields** |

# Subtyping relation: records

$S <: T$

$$\{l_1{:}T_1, \ldots, l_{n+k}{:}T_{n+k}\} <: \{l_1{:}T_1, \ldots, l_n{:}T_n\}$$

$$\frac{\forall (i \in 1 \ldots n)\ S_i <: T_i}{\{l_1{:}S_1, \ldots, l_n{:}S_n\} <: \{l_1{:}T_1, \ldots, l_n{:}T_n\}}$$

# Subtyping relation: records

$$S <: T$$

$$\{l_1{:}T_1,\ldots,l_{n+k}{:}T_{n+k}\} <: \{l_1{:}T_1,\ldots,l_n{:}T_n\}$$

$$\frac{\forall(i\in 1\ldots n)\ S_i <: T_i}{\{l_1{:}S_1,\ldots,l_n{:}S_n\} <: \{l_1{:}T_1,\ldots,l_n{:}T_n\}}$$

**Exercise 7.1.** Show that

$$\{x{:}\{a{:}Nat,b{:}Nat\},y{:}\{m{:}Nat\}\} <: \{x{:}\{a{:}Nat\}\}$$

# Subtyping relation: records

$S <: T$

$$\{l_1:T_1,\ldots,l_{n+k}:T_{n+k}\} <: \{l_1:T_1,\ldots,l_n:T_n\}$$

$$\frac{\forall(i\in 1\ldots n)\ \ S_i<:T_i}{\{l_1:S_1,\ldots,l_n:S_n\} <: \{l_1:T_1,\ldots,l_n:T_n\}}$$

$$\frac{\{l_1:S_1,\ldots,l_n:S_n\}\ \text{is permutation of}\ \{l_1:T_1,\ldots,l_k:T_k\}}{\{l_1:S_1,\ldots,l_n:S_n\} <: \{l_1:T_1,\ldots,l_k:T_k\}}$$

# Subtyping relation: records

$$S \mathrel{<:} T$$

$$\{l_1{:}T_1,\ldots,l_{n+k}{:}T_{n+k}\} \mathrel{<:} \{l_1{:}T_1,\ldots,l_n{:}T_n\}$$

$$\frac{\forall(i \in 1\ldots n)\quad S_i \mathrel{<:} T_i}{\{l_1{:}S_1,\ldots,l_n{:}S_n\} \mathrel{<:} \{l_1{:}T_1,\ldots,l_n{:}T_n\}}$$

$$\frac{\{l_1{:}S_1,\ldots,l_n{:}S_n\} \text{ is permutation of } \{l_1{:}T_1,\ldots,l_k{:}T_k\}}{\{l_1{:}S_1,\ldots,l_n{:}S_n\} \mathrel{<:} \{l_1{:}T_1,\ldots,l_k{:}T_k\}}$$

**Exercise 7.2.** Show that

$$\{x{:}\mathrm{Nat}, y{:}\mathrm{Nat}, z{:}\mathrm{Nat}\} \mathrel{<:} \{y{:}\mathrm{Nat}\}$$

# Subtyping relation: functions

$S <: T$

$$\frac{T_1 <: S_1 \qquad S_2 <: T_2}{S_1 \to S_2 <: T_1 \to T_2}$$

# Subtyping relation: functions

$S \;<:\; T$

$$\frac{T_1 <: S_1 \qquad \boxed{S_2} <: \boxed{T_2}}{S_1 \rightarrow \boxed{S_2} \;<:\; T_1 \rightarrow \boxed{T_2}}$$

**Covariant**

# Subtyping relation: functions

$S <: T$

$$\frac{\boxed{T_1} <: \boxed{S_1} \qquad S_2 <: T_2}{\boxed{S_1} \to S_2 <: \boxed{T_1} \to T_2}$$

**Contravariant**

# Subtyping relation: Top

$$S <: T$$

$$\frac{T_1 <: S_1 \qquad S_2 <: T_2}{S_1 \to S_2 <: T_1 \to T_2}$$

$$S <: \mathbf{Top}$$

# Subtyping relation: exercises

**Exercise 7.3.** How many supertypes exist for this type?

`{a:Top,b:Top}`

**Exercise 7.4.** Is there a type that is a subtype of every type?
Is there a function type that is supertype of all function types?

# Subtyping: type safety (1 of 6)

**Lemma 7.5** [Inversion of subtyping relation].

1. If $S <: T_1 \to T_2$, then
   $S = S_1 \to S_2$ where $T_1 <: S_1$ and $S_2 <: T_2$

2. If $S <: \{l_1 : T_1, \ldots, l_k : T_k\}$, then
   $S = \{f_1 : S_1, \ldots, f_n : S_n\}$ where
   $\{f_1, \ldots, f_n\}$ is a subset of $\{l_1, \ldots, l_k\}$ and
   $S_i <: T_j$ for all matching labels $f_i <: l_j$

# Subtyping: type safety (2 of 6)

**Lemma 7.6** [Inversion of typing relation].

1. If $\Gamma \vdash \lambda x{:}S_1.s : T_1 \to T_2$, then
    1. $T_1 <: S_1$
    2. $\Gamma, x{:}S_1 \vdash s : T_1 \to T_2$

2. If $\Gamma \vdash \{l_1{=}s_1, \ldots, l_k{=}s_k\} : \{k_1{:}T_1, \ldots, k_n{:}T_n\}$, then
    1. $\{l_1, \ldots, l_k\} \subseteq \{k_1, \ldots, k_n\}$
    2. $\Gamma \vdash s_i : T_j$ for each $l_i{=}k_j$

# Subtyping: type safety (3 of 6)

**Lemma 7.7** [Substitution].

If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then

$$\Gamma \vdash [x \mapsto s]t : T$$

# Subtyping: type safety (4 of 6)

**Theorem 7.8** [Preservation].

If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{t} \longrightarrow \mathtt{t'}$, then $\Gamma \vdash \mathtt{t'} : \mathtt{T}$.

# Subtyping: type safety (5 of 6)

**Lemma 7.9** [Canonical forms].

1. If **v** is a closed value of type $\mathtt{T_1 \to T_2}$, then **v** has the form $\mathtt{\lambda x{:}S_1.t}$

2. If **v** is a closed value of type $\mathtt{\{k_1{:}T_1, \ldots, k_n{:}T_n\}}$, then **v** has the form $\mathtt{\{l_1{=}s_1, \ldots, l_k{=}s_k\}}$ with $\mathtt{\{l_1, \ldots, l_k\}} \subseteq \mathtt{\{k_1, \ldots, k_n\}}$

# Subtyping: type safety (6 of 6)

**Theorem 7.10.** Suppose $\Gamma \vdash \texttt{t} : T$ then either

1. `t` is a ***value***, or
2. there exists `t'`, such that $\texttt{t} \longrightarrow \texttt{t'}$

# Subtyping: Top and Bot types

```
S <: Top
```

```
Bot <: T
```

# Subtyping: Top and Bot types

$$\boxed{\texttt{S <: Top}}$$

$$\boxed{\texttt{Bot <: T}}$$

**Exercise 7.11.** Show that no value can have type Bot.

**Exercise 7.12.** Assuming **error** : Bot, show that ...

# Subtyping: ascription and casting

$$\frac{\Gamma \vdash \mathtt{t} : \mathtt{T}}{\Gamma \vdash \mathtt{t}\ \mathbf{as}\ \mathtt{T} : \mathtt{T}}$$

$$\mathtt{v}\ \mathbf{as}\ \mathtt{T} \longrightarrow \mathtt{v}$$

# Subtyping: ascription and casting

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash t \textbf{ as } T : T}$$

$$\frac{\Gamma \vdash t : S}{\Gamma \vdash t \textbf{ cast-as } T : T}$$

$$v \textbf{ as } T \longrightarrow v$$

# Subtyping: ascription and casting

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash t \texttt{ as } T : T}$$

$$v \texttt{ as } T \longrightarrow v$$

$$\frac{\Gamma \vdash t : S}{\Gamma \vdash t \texttt{ cast-as } T : T}$$

$$\frac{\vdash v : T}{v \texttt{ cast-as } T \longrightarrow v}$$

# Subtyping: ascription and casting

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash t \ \mathbf{as}\ T : T}$$

$$v \ \mathbf{as}\ T \longrightarrow v$$

$$\frac{\Gamma \vdash t : S}{\Gamma \vdash t \ \mathbf{cast\text{-}as}\ T : T}$$

$$\frac{\vdash v : T}{v \ \mathbf{cast\text{-}as}\ T \longrightarrow v}$$

**Exercise 7.13.** Show that runtime check for casting is required for the type preservation property.

# Casting: dynamic type test

$$\frac{\Gamma \vdash t_1 : S \qquad \Gamma, x{:}T \vdash t_2 : U \qquad \Gamma \vdash t_3 : U}{\Gamma \vdash \mathbf{if}\ (t_1\ \mathbf{in}\ T)\ \mathbf{then}\ x \Rightarrow t_2\ \mathbf{else}\ t_3 : T}$$

# Casting: dynamic type test

$$\frac{\Gamma \vdash t_1 : S \qquad \Gamma, x{:}T \vdash t_2 : U \qquad \Gamma \vdash t_3 : U}{\Gamma \vdash \textbf{if}\ (t_1\ \textbf{in}\ T)\ \textbf{then}\ x \Rightarrow t_2\ \textbf{else}\ t_3 : T}$$

$$\frac{\vdash v_1 : T}{\textbf{if}\ (v_1\ \textbf{in}\ T)\ \textbf{then}\ x \Rightarrow t_2\ \textbf{else}\ t_3 \longrightarrow [x \mapsto v_1]t_2}$$

# Casting: dynamic type test

$$\frac{\Gamma \vdash t_1 : S \qquad \Gamma, x{:}T \vdash t_2 : U \qquad \Gamma \vdash t_3 : U}{\Gamma \vdash \textbf{if}\ (t_1\ \textbf{in}\ T)\ \textbf{then}\ x \Rightarrow t_2\ \textbf{else}\ t_3 : T}$$

$$\frac{\vdash v_1 : T}{\textbf{if}\ (v_1\ \textbf{in}\ T)\ \textbf{then}\ x \Rightarrow t_2\ \textbf{else}\ t_3 \longrightarrow [x \mapsto v_1]t_2}$$

$$\frac{\nvdash v_1 : T}{\textbf{if}\ (v_1\ \textbf{in}\ T)\ \textbf{then}\ x \Rightarrow t_2\ \textbf{else}\ t_3 \longrightarrow t_3}$$

# Subtyping: variants

$$\overline{\langle l_1 : T_1, \ldots, l_{n+k} : T_{n+k} \rangle <: \langle l_1 : T_1, \ldots, l_n : T_n \rangle}$$

$$\frac{\forall (i \in 1 \ldots n) \quad S_i <: T_i}{\langle l_1 : S_1, \ldots, l_n : S_n \rangle <: \langle l_1 : T_1, \ldots, l_n : T_n \rangle}$$

$$\frac{\langle l_1 : S_1, \ldots, l_n : S_n \rangle \text{ is permutation of } \langle l_1 : T_1, \ldots, l_k : T_k \rangle}{\langle l_1 : S_1, \ldots, l_n : S_n \rangle <: \langle l_1 : T_1, \ldots, l_k : T_k \rangle}$$

# Subtyping: variants

$$\langle l_1{:}T_1, \ldots, l_{n+k}{:}T_{n+k} \rangle <: \langle l_1{:}T_1, \ldots, l_n{:}T_n \rangle$$

$$\frac{\forall (i \in 1 \ldots n)\ \ S_i <: T_i}{\langle l_1{:}S_1, \ldots, l_n{:}S_n \rangle <: \langle l_1{:}T_1, \ldots, l_n{:}T_n \rangle}$$

$$\frac{\langle l_1{:}S_1, \ldots, l_n{:}S_n \rangle \text{ is permutation of } \langle l_1{:}T_1, \ldots, l_k{:}T_k \rangle}{\langle l_1{:}S_1, \ldots, l_n{:}S_n \rangle <: \langle l_1{:}T_1, \ldots, l_k{:}T_k \rangle}$$

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash \langle l{=}t \rangle : \langle l{:}T \rangle}$$

# Subtyping: lists, references

$$\frac{\texttt{S <: T}}{\texttt{List[S] <: List[T]}}$$

# Subtyping: lists, references

$$\frac{\texttt{S <: T}}{\texttt{List[S] <: List[T]}}$$

$$\frac{\texttt{S <: T} \qquad \texttt{T <: S}}{\texttt{Ref S <: Ref T}}$$

# Intersection Types

$$T_1 \wedge T_2 <: T_1$$

$$T_2 \wedge T_2 <: T_2$$

# Intersection Types

$$T_1 \land T_2 <: T_1$$

$$T_1 \land T_2 <: T_2$$

$$\frac{S <: T_1 \qquad S <: T_2}{S <: T_1 \land T_2}$$

# Intersection Types

$$T_1 \wedge T_2 <: T_1$$

$$T_1 \wedge T_2 <: T_2$$

$$\frac{S <: T_1 \qquad S <: T_2}{S <: T_1 \wedge T_2}$$

$$S \to T_1 \wedge S \to T_2 <: S \to (T_1 \wedge T_2)$$

# Intersection Types

$$T_1 \wedge T_2 <: T_1$$

$$T_1 \wedge T_2 <: T_2$$

$$\frac{S <: T_1 \qquad S <: T_2}{S <: T_1 \wedge T_2}$$

$$S \to T_1 \wedge S \to T_2 <: S \to (T_1 \wedge T_2)$$

**Remark 7.14.** Untyped lambda terms that can be typed using simple and intersection types are **exactly** the normalizing terms.

# Union Types

$$T_1 <: T_1 \lor T_2$$

$$T_2 <: T_1 \lor T_2$$

$$(T_1 \lor T_2) \to S <: T_1 \to S \lor T_2 \to S$$

## Summary

- ❑ Subtyping relation
- ❑ Properties of subtyping
- ❑ Downcasting
- ❑ Intersection and Union Types

## Summary

- ❏ Subtyping relation
- ❏ Properties of subtyping
- ❏ Downcasting
- ❏ Intersection and Union Types

# See you next time!