Recall that type systems are conservative predicates that filter out bad programs, but also some good programs.

Much of the research on more advanced type systems tries to allow more good programs, while still filtering out the bad ones.

Example of a good program that is filtered out by the type systems we’ve seen so far:

\[
(\lambda r: \{ x: \text{Nat} \}. r. x) \{ x=0,y=1 \}
\]
The **subtype** relation is a binary relation on types, usually written $T_1 <: T_2$, that is meant to capture when values of type $T_1$ can be safely used in contexts that expect values of type $T_2$.

It is often helpful to think of types as describing sets of values, and subtyping as the subset relation between these sets.

Different languages provide different definitions for which pairs of types are in the subtype relation.

The type system is extended with a **subsumption** rule that allows an expression of type $T_1$ to be treated as if it has type $T_2$, provided $T_1 <: T_2$.

$$
\Gamma \vdash e : T_1 \quad T_1 <: T_2 \quad \frac{\Gamma \vdash e : T_1}{\Gamma \vdash e : T_2}
$$
Defining the Subtype Relation

- For the moment, the setting is the simply typed lambda calculus with naturals, Booleans, and records.
- And of course, a value of type $\text{Bool}$ can be used as a $\text{Bool}$, and similarly for $\text{Nat}$.

\[
\begin{align*}
\text{Bool} & \:<: \text{Bool} \\
\text{Nat} & \:<: \text{Nat}
\end{align*}
\]

- We want to allow records with more fields to be used in contexts that expect fewer fields. This is called **width subtyping**. Common fields must have the same type. In the following, we view a record type as a partial function from labels to types.

\[
\begin{align*}
\text{dom}(R_2) & \subseteq \text{dom}(R_1) \\
\forall l \in \text{dom}(R_2). \ R_1(l) & = R_2(l)
\end{align*}
\]

\[
R_1 \:<: R_2
\]

For example, \{x : Nat, y : Nat\} \:<: \{x : Nat\}. 
We also want to allow subtyping between records whose fields may differ according to the subtype relation. This is called **depth subtyping**.

\[
\text{dom}(R_1) = \text{dom}(R_2) \quad \text{for } l \in \text{dom}(R_1). \quad R_1(l) <: R_2(l)
\]

\[
R_1 <: R_2
\]

For example,
\[
\{x : \{a : \text{Nat}, b : \text{Nat}\}, y : \text{Nat}\} <: \{x : \{a : \text{Nat}\}, y : \text{Nat}\}.
\]
Functions are **contravariant** in their parameter types and **covariant** in the return types.
Properties of the Subtype Relation

Proposition (Basic Properties)

1. *Subtyping is reflexive*: for any $T$, $T <: T$.

Proposition (Inversion)

1. *If* $S <: T_1 \rightarrow T_2$ *then* $S = S_1 \rightarrow S_2$, $T_1 <: S_1$, and $S_2 <: T_2$ *for some* $S_1$ *and* $S_2$.
2. *If* $S <: R$ *and* $R$ *is a record type*, then $S$ *is a record type and* $\text{dom}(R) \subseteq \text{dom}(S)$ *and for* $l \in \text{dom}(S)$, $S(l) <: R(l)$.
Properties of the Subtype Relation

Lemma (Lambda Typing)

If $\Gamma \vdash \lambda x : S_1. e : T_1 \rightarrow T_2$, then $T_1 \ll S_1$, and $\Gamma, x : S_1 \vdash e : T_2$.

Let $\rho$ range over record values and $R$ over record types. A record value is a partial function from labels to values.

Lemma (Record Typing)

If $\Gamma \vdash \rho : R$, then $\text{dom}(R) \subseteq \text{dom}(\rho)$ and for $l \in \text{dom}(R)$, $\Gamma \vdash \rho(l) : R(l)$. 
Lemma (Substitution)

If $\Gamma, x : S \vdash e : T$ and $\Gamma \vdash e' : S$, then $\Gamma \vdash [x \mapsto e']e : T$.

Proof.

By induction on $\Gamma, x : S \vdash e : T$.

Case $\Gamma \vdash e : T_1$ : $T_1 \ll T_2$ : $\Gamma \vdash e : T_2$

From the induction hypothesis, we have $\Gamma \vdash [x \mapsto e']e : T_1$. Then by the subsumption rule we have $\Gamma \vdash [x \mapsto e']e : T_2$. 

\[\square\]
Lemma (Subject Reduction)

If $\Gamma \vdash e : T$ and $e \rightarrow e'$, then $\Gamma \vdash e' : T$.

Proof. By cases on $e \rightarrow e'$.

Case $((\lambda x : S_1. e_b) \; v) \rightarrow [x \leftarrow v]e_b$

Because $e$ is well typed, we have $T = T_1 \rightarrow T_2$,
$\Gamma \vdash (\lambda x : S_1. e_b) : T_1 \rightarrow T_2$, and $\Gamma \vdash v : T_1$. From the lambda typing lemma, we have $T_1 <: S_1$ and $\Gamma, x : S_1 \vdash e_b : T_2$. Because $T_1 <: S_1$ we have $\Gamma \vdash v : S_1$. Then by the substitution lemma,
$\Gamma \vdash [x \leftarrow v]e_b : T_2$. 
Subject Reduction, continued

Case

\[
\text{for } l' \in \text{dom}(\rho). \ \rho(l') \text{ is a value} \quad l \in \text{dom}(\rho)
\]

\[
\rho.l \rightarrow \rho(l)
\]

Because e (which is \(\rho.l\)) is well typed, we have \(\Gamma \vdash \rho : R\), for some record type \(R\), and \(l \in \text{dom}(R)\) and \(T = R(l)\). Then by the record typing lemma, for every \(j \in \text{dom}(R)\), we have \(\Gamma \vdash \rho(j) : R(j)\). Therefore \(\Gamma \vdash \rho(l) : R(l)\).