编程语言的设计原理
Design Principles of Programming Languages

Haiyan Zhao, Di Wang
赵海燕，王迪

Peking University, Spring Term 2023

Chap 21: Metatheory of Recursive Types

Finite & Infinite Types

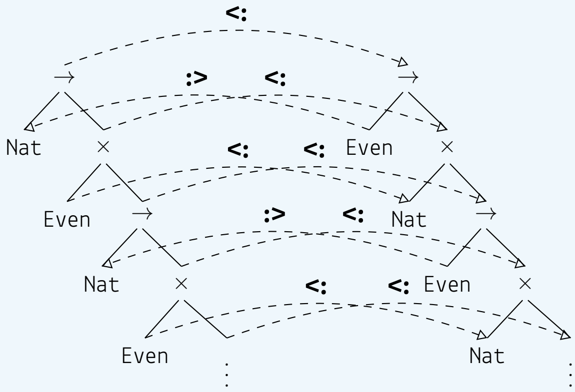
Induction & Coinduction

Subtyping

Membership Checking

Can we deduce the relation below, given that Even <: Nat?

$$\mu X.\mathsf{Nat} \to (\mathsf{Even} \times X) <: \mu X.\mathsf{Even} \to (\mathsf{Nat} \times X)$$



Principle

We need to develop a metatheory of subtyping on **infinite tree types**.

Finite & Infinite Types

Tree Types

For brevity, we only consider three type constructors: $\rightarrow$, $\times$, and Top.

$$\mathsf{T} ::= \mathsf{Top} \mid \mathsf{T} \rightarrow \mathsf{T} \mid \mathsf{T} \times \mathsf{T}$$

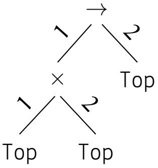
Definition

A **tree type** is a partial function $\mathsf{T} : \{1, 2\}^* \rightharpoonup \{\rightarrow, \times, \mathsf{Top}\}$ satisfying the following constraints:

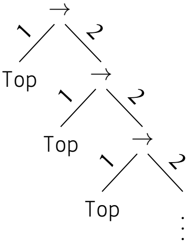
- $\mathsf{T}(\bullet)$ is defined;
- if $\mathsf{T}(\pi, \sigma)$ is defined then $\mathsf{T}(\pi)$ is defined;
- if $\mathsf{T}(\pi) = \rightarrow$ or $\mathsf{T}(\pi) = \times$ then $\mathsf{T}(\pi, 1)$ and $\mathsf{T}(\pi, 2)$ are defined;
- if $\mathsf{T}(\pi) = \mathsf{Top}$ then $\mathsf{T}(\pi, 1)$ and $\mathsf{T}(\pi, 2)$ are undefined.

Tree Types

$(\mathsf{Top} \times \mathsf{Top}) \to \mathsf{Top}$



$\mathsf{Top} \to (\mathsf{Top} \to (\mathsf{Top} \to \ldots))$



Definition

A tree type T is **finite** if $dom(\mathsf{T})$ is finite.
The set of all tree types is written $\mathcal{T}$; the subset of **all finite tree types** is written $\mathcal{T}_f$.

Question

How to characterize $\mathcal{T}$ and $\mathcal{T}_f$?

Induction & Coinduction

Review: Induction

By Inference Rules

$\mathcal{T}_f$ is the **least** set of tree types defined by the following rules:

$$\frac{}{\mathsf{Top} \in \mathcal{T}_f} \qquad \frac{T_1 \in \mathcal{T}_f \qquad T_2 \in \mathcal{T}_f}{T_1 \to T_2 \in \mathcal{T}_f} \qquad \frac{T_1 \in \mathcal{T}_f \qquad T_2 \in \mathcal{T}_f}{T_1 \times T_2 \in \mathcal{T}_f}$$

By Union of Sets

$$\mathcal{T}_0 \stackrel{\text{def}}{=} \varnothing$$

$$\mathcal{T}_{i+1} \stackrel{\text{def}}{=} \{\mathsf{Top}\} \cup \{T_1 \to T_2 \mid T_1, T_2 \in \mathcal{T}_i\} \cup \{T_1 \times T_2 \mid T_1, T_2 \in \mathcal{T}_i\}$$

$$\mathcal{T}_f \stackrel{\text{def}}{=} \bigcup_i \mathcal{T}_i$$

Let $F(X) \stackrel{\text{def}}{=} \{\mathsf{Top}\} \cup \{T_1 \to T_2 \mid T_1, T_2 \in X\} \cup \{T_1 \times T_2 \mid T_1, T_2 \in X\}$. Then $\mathcal{T}_f = \bigcup_i F^i(\varnothing)$.

How to characterize the function F?

Generating Functions

A Universal Set $\mathcal{U}$

$\mathcal{U}$ represents "everything in the world."

Definition (Generating Functions)

A generating function is a function $F : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ that is **monotone**, i.e., $X \subseteq Y$ implies $F(X) \subseteq F(Y)$.

Let F be monotone, and X be a subset of $\mathcal{U}$.

- X is **F-closed** if $F(X) \subseteq X$.
- X is **F-consistent** if $X \subseteq F(X)$.
- X is a **fixed point** of F if $F(X) = X$.

Example

Recall $F(X) \stackrel{\text{def}}{=} \{\text{Top}\} \cup \{T_1 \to T_2 \mid T_1, T_2 \in X\} \cup \{T_1 \times T_2 \mid T_1, T_2 \in X\}$. Then $\mathcal{T}_f$ is a fixed point of F.

Is $\mathcal{T}_f$ the only fixed point?

Knaster-Tarski Theorem

Theorem

- The intersection of all F-closed sets is the **least** fixed point of F, written μF.
- The union of all F-consistent sets is the **greatest** fixed point of F, written νF.

Question

$$\frac{}{c} \qquad\qquad \frac{c}{b} \qquad\qquad \frac{b \quad c}{a}$$

What are μE_1 and νE_1?

Question

Recall $F(X) \stackrel{\text{def}}{=} \{\mathsf{Top}\} \cup \{T_1 \to T_2 \mid T_1, T_2 \in X\} \cup \{T_1 \times T_2 \mid T_1, T_2 \in X\}$.
What are the least and greatest fixed points of F?

$$\mu F = \mathcal{T}_f \qquad\qquad \nu F = \mathcal{T}$$

Principles of Induction & Coinduction

Principle (Induction)

If X is F-closed (i.e., $F(X) \subseteq X$), then $\mu F \subseteq X$.

Remark

Any property whose characteristic set **is closed under** F is true of all elements of the inductively defined set μF.

Principle (Coinduction)

If X is F-consistent (i.e., $X \subseteq F(X)$), then $X \subseteq \nu F$.

Remark

The coinduction principle is a method for establishing that **an element** x **is in** the coinductively defined set νF.

Subtyping

Finite Subtyping

By Inference Rules

$$\frac{}{T <: \mathsf{Top}} \qquad \frac{T_1 <: S_1 \qquad S_2 <: T_2}{S_1 \to S_2 <: T_1 \to T_2} \qquad \frac{S_1 <: T_1 \qquad S_2 <: T_2}{S_1 \times S_2 <: T_1 \times T_2}$$

By a Generating Function

Two finite tree types S and T are in the **subtype relation** ("S is a subtype of T") if $(S, T) \in \mu S_f$, where the monotone function

$$S_f \in \mathcal{P}(\mathcal{T}_f \times \mathcal{T}_f) \to \mathcal{P}(\mathcal{T}_f \times \mathcal{T}_f)$$

is defined by

$$\begin{aligned} S_f(R) \stackrel{\text{def}}{=} & \{(T, \mathsf{Top}) \mid T \in \mathcal{T}_f\} \\ & \cup \{(S_1 \to S_2, T_1 \to T_2) \mid (T_1, S_1), (S_2, T_2) \in R\} \\ & \cup \{(S_1 \times S_2, T_1 \times T_2) \mid (S_1, T_1), (S_2, T_2) \in R\}. \end{aligned}$$

Infinite Subtyping

By Inference Rules

$$\frac{}{T <: \mathsf{Top}} \qquad \frac{T_1 <: S_1 \qquad S_2 <: T_2}{S_1 \to S_2 <: T_1 \to T_2} \qquad \frac{S_1 <: T_1 \qquad S_2 <: T_2}{S_1 \times S_2 <: T_1 \times T_2}$$

The same set of rules, but interpreted coinductively!

By a Generating Function

Two **(finite or infinite)** tree types S and T are in the **subtype relation** ("S is a subtype of T") if $(S, T) \in \nu S$, where the monotone function

$$S \in \mathcal{P}(\mathcal{T} \times \mathcal{T}) \to \mathcal{P}(\mathcal{T} \times \mathcal{T})$$

is defined by

$$\begin{aligned} S(R) \stackrel{\text{def}}{=} & \{(T, \mathsf{Top}) \mid T \in \mathcal{T}\} \\ & \cup \{(S_1 \to S_2, T_1 \to T_2) \mid (T_1, S_1), (S_2, T_2) \in R\} \\ & \cup \{(S_1 \times S_2, T_1 \times T_2) \mid (S_1, T_1), (S_2, T_2) \in R\}. \end{aligned}$$

Exercises

Question (Exercise 21.3.3)

Check that νS is not the whole of $\mathcal{T} \times \mathcal{T}$ by exhibiting a pair (S, T) that is not in νS.

Question (Exercise 21.3.4)

Is there a pair of types (S, T) that is related by νS, but not by μS?
What about a pair of types (S, T) that is related by νS_f, but not by μS_f?

Transitivity

Definition

A relation $R \subseteq \mathcal{U} \times \mathcal{U}$ is **transitive** if R is closed under the monotone function

$$TR(R) \stackrel{\text{def}}{=} \{(x, y) \mid \exists z \in \mathcal{U}.\ (x, z), (z, y) \in R\},$$

i.e., if $TR(R) \subseteq R$.

Theorem

νS is transitive.

Lemma

Let $F \in \mathcal{P}(\mathcal{U} \times \mathcal{U}) \to \mathcal{P}(\mathcal{U} \times \mathcal{U})$ be a monotone function.
If $TR(F(R)) \subseteq F(TR(R))$ for any $R \subseteq \mathcal{U} \times \mathcal{U}$, then νF is transitive.

Membership Checking

How to check $S <: T$ algorithmically?

μ-Types

Definition

Let X range over a fixed countable set $\{X_1, X_2, \ldots\}$ of type variables. The set of **raw μ-types** is the set of expressions defined by the following grammar (inductively):

$$T ::= X \mid \mathsf{Top} \mid T \to T \mid T \times T \mid \mu X.T$$

Definition

A raw μ-type T is **contractive** (and called a **μ-type**) if, for any subexpression of T of the form $\mu X_1.\mu X_2.\ldots.\mu X_n.S$, the body S is not X.

The set of μ-types is written $\mathcal{T}_m$.

Question

What is the relation between $\mathcal{T}_m$ and $\mathcal{T}$ (the set of tree types)?

Finite Notation for (Some) (Possibly-)Infinite Tree Types

The function *treeof*, mapping closed μ-types to tree types, is defined inductively as follows:

$$treeof(\mathsf{Top})(\bullet) \stackrel{\text{def}}{=} \mathsf{Top}$$

$$treeof(T_1 \to T_2)(\bullet) \stackrel{\text{def}}{=} \to \qquad treeof(T_1 \to T_2)(i, \pi) \stackrel{\text{def}}{=} treeof(T_i)(\pi)$$

$$treeof(T_1 \times T_2)(\bullet) \stackrel{\text{def}}{=} \times \qquad treeof(T_1 \times T_2)(i, \pi) \stackrel{\text{def}}{=} treeof(T_i)(\pi)$$

$$treeof(\mu X.T)(\pi) \stackrel{\text{def}}{=} treeof([X \mapsto \mu X.T]T)(\pi)$$

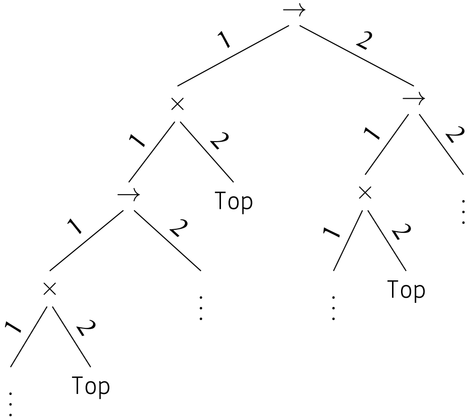
Question

Why is *treeof* well-defined?

Answer

Every recursive use of *treeof* on the right-hand side reduces the lexicographic size of the pair $(|\pi|, \mu\text{-}height(T))$.

$\mathit{treeof}(\mu X.((X \times \mathsf{Top}) \to X))$



Subtyping on μ-Types

μ-Folding Rules

$$\frac{S <: [X \mapsto \mu X.T]T}{S <: \mu X.T} \qquad \frac{[X \mapsto \mu X.S]S <: T}{\mu X.S <: T}$$

Inductive or coinductive?

By a Generating Function

Two μ-types S and T are said to be in the subtype relation if $(S, T) \in \nu S_d$, where the monotone function $S_d \in \mathcal{P}(\mathcal{T}_m \times \mathcal{T}_m) \to \mathcal{P}(\mathcal{T}_m \times \mathcal{T}_m)$ is defined by

$$\begin{aligned} S_d(R) \stackrel{\text{def}}{=} & \{(S, Top) \mid T \in \mathcal{T}_m\} \\ & \cup \{(S_1 \to S_2, T_1 \to T_2) \mid (T_1, S_1), (S_2, T_2) \in R\} \\ & \cup \{(S_1 \times S_2, T_1 \times T_2) \mid (S_1, T_1), (S_2, T_2) \in R\} \\ & \cup \{(S, \mu X.T) \mid (S, [X \mapsto \mu X.T]T) \in R\} \\ & \cup \{(\mu X.S, T) \mid ([X \mapsto \mu X.S]S, T) \in R\}. \end{aligned}$$

Subtyping Correspondence: μ-Types and Tree Types

Theorem

Let $(S, T) \in \mathcal{T}_m \times \mathcal{T}_m$. Then $(S, T) \in \nu S_d$ **if and only if** $(\mathit{treeof}(S), \mathit{treeof}(T)) \in \nu S$.

Question

How to characterize the subset $\mathit{treeof}(\mathcal{T}_m) \subseteq \mathcal{T}$?

Definition (Regular Tree Types)

A tree type S is a **subtree** of a tree type T if $S = \lambda\sigma.\ T(\pi, \sigma)$ for some π.
A tree type T is **regular** if $\mathit{subtrees}(T)$ is finite.

Lemma

Every μ-type $T \in \mathcal{T}_m$ corresponds to a regular tree type $\mathit{treeof}(T)$.

Regularity

Example $(\mu X.\mathsf{Top} \times X <: \mu X.\mathsf{Top} \times (\mathsf{Top} \times X))$

Let $S \stackrel{\text{def}}{=} \mu X.\mathsf{Top} \times X$ and $T \stackrel{\text{def}}{=} \mu X.\mathsf{Top} \times (\mathsf{Top} \times X)$.

$$\dfrac{\dfrac{\dfrac{}{\mathsf{Top} <: \mathsf{Top}} \qquad \dfrac{\dfrac{\dfrac{}{\mathsf{Top} <: \mathsf{Top}} \qquad \dfrac{\vdots}{\color{red}{S <: T}}}{\mathsf{Top} \times S <: \mathsf{Top} \times T}}{S <: \mathsf{Top} \times T}}{\mathsf{Top} \times S <: \mathsf{Top} \times (\mathsf{Top} \times T)}}{\dfrac{\mathsf{Top} \times S <: T}{S <: T}}$$

Observation (Finite-State)

To check the subtype relation $S <: T$ between μ-types S and T, the set of reachable states $S' <: T'$ is finite.

Hypothetical Subtyping

$\Sigma \vdash S <: T$: "one can derive $S <: T$ by assuming the subtype relations in Σ"

$$\frac{(S <: T) \in \Sigma}{\Sigma \vdash S <: T} \qquad \frac{}{\Sigma \vdash S <: \mathsf{Top}} \qquad \frac{\Sigma \vdash T_1 <: S_1 \qquad \Sigma \vdash S_2 <: T_2}{\Sigma \vdash S_1 \to S_2 <: T_1 \to T_2} \qquad \frac{\Sigma \vdash S_1 <: T_1 \qquad \Sigma \vdash S_2 <: T_2}{\Sigma \vdash S_1 \times S_2 <: T_1 \times T_2}$$

$$\frac{\Sigma, S <: \mu X.T \vdash S <: [X \mapsto \mu X.T]T}{\Sigma \vdash S <: \mu X.T} \qquad \frac{\Sigma, \mu X.S <: T \vdash [X \mapsto \mu X.S]S <: T}{\Sigma \vdash \mu X.S <: T}$$

Let $S \stackrel{\text{def}}{=} \mu X.\mathsf{Top} \times X$ and $T \stackrel{\text{def}}{=} \mu X.\mathsf{Top} \times (\mathsf{Top} \times X)$.

$$\dfrac{\dfrac{\dfrac{}{\ldots \vdash \mathsf{Top} <: \mathsf{Top}} \qquad \dfrac{\dfrac{\dfrac{}{\ldots \vdash \mathsf{Top} <: \mathsf{Top}} \qquad \dfrac{}{S <: T, \ldots \vdash S <: T}}{S <: T, \ldots \vdash \mathsf{Top} \times S <: \mathsf{Top} \times T}}{S <: T, \ldots \vdash S <: \mathsf{Top} \times T}}{S <: T, \ldots \vdash \mathsf{Top} \times S <: \mathsf{Top} \times (\mathsf{Top} \times T)}}{\dfrac{S <: T \vdash \mathsf{Top} \times S <: T}{\varnothing \vdash S <: T}}$$

Hypothetical Subtyping

Remark

The hypothetical subtyping $\Sigma \vdash S <: T$ corresponds to the $subtype^{ac}$ algorithm presented in Chapter 21.10.

Theorem

Let $(S, T) \in \mathcal{T}_m \times \mathcal{T}_m$. Then $\varnothing \vdash S <: T$ **if and only if** $(S, T) \in \nu S_d$.

Proof Sketch

- To prove "$\varnothing \vdash S <: T$ imples $(S, T) \in \nu S_d$," we can apply Lemma 21.6.5(2).
- To prove "$(S, T) \in \nu S_d$ implies $\varnothing \vdash S <: T$," we first turn to prove "$\Sigma \vdash S \not<: T$ implies $(S, T) \notin \nu S_d$." We can apply Lemma 21.5.8 in this part.
- It suffices to show either $\Sigma \vdash S <: T$ or $\Sigma \vdash S \not<: T$. This part is related to **regularity** (or finite-state) and is discussed in Chapter 21.9.

A Little More Details

Remark

We have been using S_d, but the textbook mostly uses S_m defined as follows:

$$
\begin{aligned}
S_m(R) \stackrel{\text{def}}{=} & \{(S, \mathsf{Top}) \mid T \in \mathcal{T}_m\} \\
& \cup \{(S_1 \to S_2, T_1 \to T_2) \mid (T_1, S_1), (S_2, T_2) \in R\} \\
& \cup \{(S_1 \times S_2, T_1 \times T_2) \mid (S_1, T_1), (S_2, T_2) \in R\} \\
& \cup \{(S, \mu X.T) \mid (S, [X \mapsto \mu X.T]T) \in R\} \\
& \cup \{(\mu X.S, T) \mid ([X \mapsto \mu X.S]S, T) \in R, T \neq \mathsf{Top}, T \neq \mu Y.T_1\}.
\end{aligned}
$$

Observation

If we think S_m in terms of inference rules, it is **algorithmic** but S_d is not.

A Little More Details

Definition (Invertible Generating Functions)

A generating function F is said to be **invertible** if, for all $x \in \mathcal{U}$, the collection of sets $G_x = \{X \subseteq \mathcal{U} \mid x \in F(X)\}$ either is empty or contains a unique member that is a subset of all the others. When F is invertible, we define:

$$support_F(x) \stackrel{\text{def}}{=} \begin{cases} X & \text{if } X \in G_x \text{ and } \forall X' \in G_x.\ X \subseteq X' \\ \uparrow & \text{if } G_x = \varnothing \end{cases}$$

Example

$support_F(x)$ essentially **inverts** the unique inference rule for x under F.

$$support_{S_m}(S,T) \stackrel{\text{def}}{=} \begin{cases} \varnothing & \text{if } T = Top \\ \{(T_1,S_1),(S_2,T_2)\} & \text{if } S = S_1 \to S_2 \text{ and } T = T_1 \to T_2 \\ \{(S_1,T_1),(S_2,T_2)\} & \text{if } S = S_1 \times S_2 \text{ and } T = T_1 \times T_2 \\ \{(S,[X \mapsto \mu X.T]T)\} & \text{if } T = \mu X.T \\ \{([X \mapsto \mu X.S]S,T)\} & \text{if } S = \mu X.S \text{ and } T \neq Top, T \neq \mu Y.T_1 \\ \uparrow & \text{otherwise} \end{cases}$$

A Little More Details

A General Algorithm that Corresponds to $\Sigma \vdash S <: T$

Given an invertible generating function F, define the function gfp^{ac}_F as follows:

$$
\begin{aligned}
gfp^{ac}_F(A, x) \quad = \quad & \text{if } x \in A \text{, then } \mathit{true} \\
& \text{else if } \mathit{support}_F(x)\uparrow\text{, then } \mathit{false} \\
& \text{else} \\
& \quad \text{let } \{x_1, \ldots, x_n\} = \mathit{support}_F(x) \text{ in} \\
& \quad \text{let } A_0 = A \cup \{x\} \text{ in} \\
& \quad gfp^{ac}_F(A_0, x_1) \text{ and} \\
& \quad gfp^{ac}_F(A_0, x_2) \text{ and} \\
& \quad \ldots \\
& \quad gfp^{ac}_F(A_0, x_n).
\end{aligned}
$$

The $\Sigma \vdash S <: T$ system corresponds to $gfp^{ac}_{S_m}(\Sigma, (S, T))$.

A Little More Details

See Chapter 21.10 for an example that shows $gfp^{ac}_{S_m}$ is an exponential algorithm.

A Better Algorithm

Given an invertible generating function F, define the function gfp^t_F as follows:

$$
\begin{aligned}
gfp^t_F(A, x) \quad = \quad & \text{if } x \in A\text{, then } A \\
& \text{else if } support_F(x)\uparrow\text{, then } fail \\
& \text{else} \\
& \quad \text{let } \{x_1, \ldots, x_n\} = support_F(x) \text{ in} \\
& \quad \text{let } A_0 = A \cup \{x\} \text{ in} \\
& \quad \text{let } A_1 = gfp^t_F(A_0, x_1) \text{ in} \\
& \quad \ldots \\
& \quad \text{let } A_n = gfp^t_F(A_{n-1}, x_n) \text{ in} \\
& \quad A_n.
\end{aligned}
$$

Summary

Metatheory of Recursive Types

We have studied the theoretical foundation of type checkers (subtyping) for equi-recursive types.

- Finite & infinite types
- Induction & coinduction & their proof principles
- Subtyping
- Membership-checking algorithm

Homework

Question (Exercise 21.5.2)

Verify that S_f and S, the generating functions for the subtyping relations from Definitions 21.3.1 and 21.3.2, are invertible, and give their support functions.