Judgmental Subtyping

서정봉　박성우
Pohang University of Science and Technology, Republic of Korea

하고 싶은 것

• 보다 유연한 언어

✓ 잡종 데이터 모음

```
type dyn = MyInt of int | MyString of string
let ex_list : dyn list = [MyInt 1; MyString “hello world”]
```

```
type dyn = int ∨ string
let ex_list : dyn list = [1; “hello world”]
```

✓ 타입 따라 '그때 그때 달라요' 함수

```
let inc : dyn → dyn = function
| MyInt x → MyInt (x + 1)
| MyString x → MyString (x ˆ “+”)

let result = List.map inc ex_list
(* val result : dyn list = [MyInt 2; MyString “hello world+”] *)
```

```
let inc : (int → int) ∧ (string → string) = function
| x:int → x + 1
| x:string → x ˆ “+”

let result = List.map inc ex_list
(* val result : dyn list = [2; “hello world+”] *)
```

• 보다 정교한 타입

두 문자열을 합치는 함수:

$$bitstring * bitstring \to bitstring$$

결과로 나오는 값이 특정 조건(parity condition)을 만족함을 표현:

$$(even * even \to even) \wedge (even * odd \to odd)$$
$$\wedge (odd * even \to odd) \wedge (odd * odd \to even)$$

재료1: Intersection Types

• Introduction Rule

$$\frac{\Gamma \vdash M : A_1 \qquad \Gamma \vdash M : A_2}{\Gamma \vdash M : A_1 \wedge A_2} \wedge I$$

• Elimination Rules

$$\frac{\Gamma \vdash M : A_1 \wedge A_2}{\Gamma \vdash M : A_i} \wedge E_i \quad \text{or} \quad \frac{\Gamma \vdash M : A_1 \wedge A_2 \qquad A_1 \wedge A_2 \leq A_1}{\Gamma \vdash M : A_1} \, sub$$

• Subtyping

$$\frac{}{A_1 \wedge A_2 \leq A_i} \qquad \frac{A \leq C_1 \qquad A \leq C_2}{A \leq C_1 \wedge C_2}$$

재료2: Union Types

• Introduction Rule

$$\frac{\Gamma \vdash M : A_i}{\Gamma \vdash M : A_1 \vee A_2} \vee I_i \quad \text{or} \quad \frac{\Gamma \vdash M : A_1 \qquad A_1 \leq A_1 \vee A_2}{\Gamma \vdash M : A_1 \vee A_2} \, sub$$

• Elimination Rules

$$\frac{\Gamma \vdash M : A_1 \vee A_2 \qquad \begin{array}{c}\Gamma, x_1 : A_1 \vdash \mathcal{E}[x_1] : C \\ \Gamma, x_2 : A_2 \vdash \mathcal{E}[x_2] : C\end{array}}{\Gamma \vdash \mathcal{E}[M] : C} \vee E$$

• Subtyping

$$\frac{}{A_i \leq A_1 \vee A_2} \qquad \frac{A_1 \leq C \qquad A_2 \leq C}{A_1 \vee A_2 \leq C}$$

문제점

• 복잡해지는 서브타입 관계들

$$(A_1 \to C) \wedge (A_2 \to C) \leq (A_1 \vee A_2) \to C$$
$$(A_1 \vee A_2) \to C \leq (A_1 \to C) \wedge (A_2 \to C)$$
$$A \vee (B \wedge C) \leq (A \vee B) \wedge (A \vee C)$$
$$\vdots$$

• 불명확한 논리적 해석
• 복잡해지는 타입검사 / 비결정적인 계산법 정의
• Effect와 언어의 안전성

현재 접근법1: Subtyping judgment

• 타입 생성자의 의미를 내포하는 Subtyping judgment

$A_1, \cdots, A_n \preceq B_1, \cdots, B_m$
:= $A_1 \wedge \cdots \wedge A_n$ 타입을 가지는 코드는 $B_1 \vee \cdots \vee B_m$ 타입도 가진다.

• 새로운 judgment를 이용한 서브타입 시스템 설계 (미완성)

$$\frac{}{\Sigma, P \preceq \Delta, P}\,\mathsf{Id}{\preceq} \qquad \frac{\Sigma, A, B \preceq \Delta}{\Sigma, A \wedge B \preceq \Delta}\,\wedge\mathsf{L}{\preceq} \qquad \frac{\Sigma \preceq \Delta, A \qquad \Sigma \preceq \Delta, B}{\Sigma \preceq \Delta, A \wedge B}\,\wedge\mathsf{R}{\preceq}$$

$$\frac{\Sigma, A \preceq \Delta \qquad \Sigma, B \preceq \Delta}{\Sigma, A \vee B \preceq \Delta}\,\vee\mathsf{L}{\preceq} \qquad \frac{\Sigma \preceq \Delta, A, B}{\Sigma \preceq \Delta, A \vee B}\,\vee\mathsf{R}{\preceq}$$

$$\frac{\Sigma \preceq \Delta, A}{\Sigma, \neg A \preceq \Delta}\,\neg\mathsf{L}{\preceq} \qquad \frac{\Sigma, A \preceq \Delta}{\Sigma \preceq \Delta, \neg A}\,\neg\mathsf{R}{\preceq} \qquad \frac{}{\Sigma \preceq \Delta, \top}\,\top\mathsf{L}{\preceq} \qquad \frac{}{\Sigma, \bot \preceq \Delta}\,\bot\mathsf{R}{\preceq}$$

$$\boxed{\frac{C \preceq \{A_i\} \quad \cdots \quad C \preceq \{B_j\} \quad A', \cdots, B' \preceq C'}{\Sigma, \{A_i \to A'\}, \cdots, \{B_j \to B'\} \preceq \Delta, C \to C'}\,{\to}{\preceq}}$$

$$\frac{\begin{array}{c}[A_1, \cdots, A_m \preceq \bigcup_{i \in I} B_i \text{ and } A'_1, \cdots, A'_m \preceq \bigcup_{j \in J} B'_j] \\ \text{for each pair of } I \text{ and } J \text{ such that } I \cup J = \{1, \cdots, n\} \text{ and } I \cap J = \varnothing\end{array}}{\Sigma, A_1 \times A'_1, \cdots, A_m \times A'_m \preceq \Delta, B_1 \times B'_1, \cdots, B_n \times B'_n}\,\times{\preceq}$$

$$\frac{S}{[S \text{ and } S']]}\,[]\mathsf{L} \qquad \frac{S'}{[S \text{ and } S']}\,[]\mathsf{R}$$

• 증명해야 하는 것 (서브타입의 transitivity에 해당)

Lemma(Cut elimination) If $\Gamma \preceq \Delta, C$ and $\Gamma', C \preceq \Delta'$, then $\Gamma, \Gamma' \preceq \Delta, \Delta'$.

현재 접근법2: 타입 시스템과 서브타입 분리

Subsumption 룰
• 타입 시스템과 서브타입 시스템이 상호 의존적이 됨.
• 사용되는 경우를 제약할 수 있다면 문제가 쉬워질 것으로 기대.

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\,\mathsf{Var} \qquad \frac{x : A \leq \{A_1, \cdots, A_n\} \in \Delta}{\Delta \vdash_{\leq} x : A_i}\,\mathsf{Var}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \to B}\,{\to}\mathsf{I} \qquad \frac{\Delta, x : A \leq \{A_1, \cdots, A_n\} \vdash_{\leq} M : B}{\Delta \vdash_{\leq} \lambda x. M : A \to B}\,{\to}\mathsf{I}$$

?

<Source language>　　*<Target language>*

앞으로 할 일

• Cut-elimination을 만족하도록 서브 타입 시스템 수정하기,
  혹은 타입시스템과 연동하여 그 보다 약한 합의점(weak-cut) 찾기.

• 타입 시스템과 서브타입 시스템 분리법 완성하기.
• 여기에 Intersection type과 union type 추가하며 확장해보기.

• 두 접근법을 합쳐 타입 시스템 완성하기.