경로 동일성이 추가된 경로 기반 타입의 형식화

Formalization of Path Dependent Types with Path-Equality

홍재민 류석영 – PLRG@KAIST

1 서론

- 정적 분석기: 프로그램의 안전성을 검증하기 위한 도구
- 정적 분석기를 구현하기 위해서는 요약 도메인이 필요
- 스칼라에서는 객체가 타입 멤버를 가짐
- 타입 멤버를 통해서 요약 도메인을 안전하게 구현 가능
- 스칼라 타입 체계의 한계로 인해 구현할 수 없는 기능 존재
- 타입 체계에 경로 동일성을 추가해야 구현 가능
- DOT: 타입 멤버를 가지는 객체를 형식화한 언어
- 경로 동일성을 추가한 언어인 πDOT를 형식화 함

2 배경

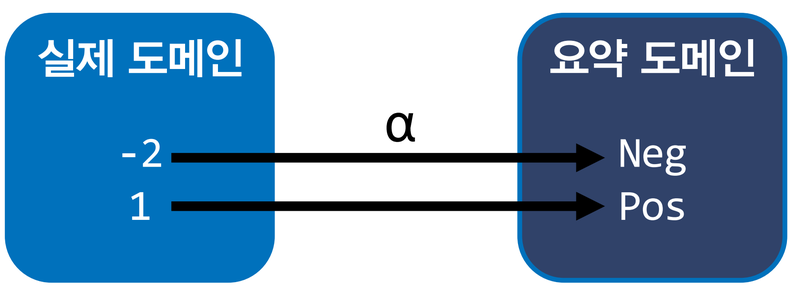
- 스칼라의 객체는 타입 멤버를 가짐
- 경로에 기반한 타입을 사용할 수 있음

```
class A {
  type T
  def get(): this.T = …
}

val a0: A = new A
val t0: a0.T = a0.get()

val a1: A = new A
val t1: a1.T = t0 // type error: a1.T != a0.T
```

- 정적 분석기에서는 실제 값을 요약한 요약 원소 사용



- V라는 실제 도메인에 대응되는 AbstractDomain 정의
- AbstractDomain의 요약 원소는 Elem을 타입으로 가짐

```
trait AbstractDomain[V] { type Elem }
```

- 같은 실제 도메인에 대해서 여러 방법으로 요약 도메인 구현 가능
- AbstractDomain을 상속하여 특정 도메인에 대한 요약 도메인 구현

```
object A extends AbstractDomain[Num] { … }
object B extends AbstractDomain[Num] { … }

val elem0: A.Elem = A.alpha(0)
val elem1: A.Elem = A.alpha(1)
val elem2: B.Elem = B.alpha(2)

elem0 + elem1
elem1 + elem2 // type error: A.Elem != B.Elem
```

- 둘 이상의 도메인을 조합하여 새로운 도메인 생성
- 타입 체계가 경로 동일성을 고려하지 않아 도메인 사용이 어려움

```
abstract class PairDomain[V0, V1](
  val d0: AbstractDomain[V0],
  val d1: AbstractDomain[V1]
) { … }
object AB extends PairDomain[Num, Num](A, B) …

val elem: AB.Elem = AB.alpha((0, 1))
val elemF: AB.d0.Elem = elem.fst
elemF: A.Elem // type error: A.Elem != AB.d0.elem
```

3 타입 체계 형식화

3.1 문법 (Syntax)

- 레코드 타입과 타입 선택을 타입으로 가짐

$$
\begin{array}{lcl}
p, q & ::= & x \mid p.f \\
D & ::= & \mathsf{type}\ L = T .. T \mid \mathsf{type}\ L <: T \mid \mathsf{val}\ f : T \\
S, T, U & ::= & \{\overline{D}\} \mid p.L \\
I & ::= & \mathsf{type}\ L = T .. T \mid \mathsf{type}\ L <: T \mid \mathsf{val}\ f = t \\
s, t, u & ::= & x \mid t.f \mid \mathsf{new}\ (\overline{I}) \mid \mathsf{let}\ x : T = t\ \mathsf{in}\ t
\end{array}
$$

- 경로 환경: 경로들의 분할(동일 경로는 같은 경로 집합에 포함)
- 의사 경로, 널–가능 경로: 경로 동일성 정보를 모으기 위해 사용

$$
\begin{array}{lcll}
\psi & ::= & \{\overline{p}\} & \text{Path Set} \\
\Psi & ::= & \{\overline{\psi}\} & \text{Path Environemnt} \\
\rho & ::= & \cdot \mid f, \rho & \text{Pseudo Path} \\
\pi & ::= & p \mid \mathsf{null} & \text{Nullable Path}
\end{array}
$$

3.2 정적 의미 (Static Semantics)

- let 바인딩의 타입 – 변수 바인딩 시 경로 환경을 갱신

$$
\frac{\Gamma; \Psi \vdash t : T \quad t\ \mathsf{implies}\ \langle \overline{\rho \equiv \pi} \rangle \quad expand(\Psi, x, \langle \overline{\rho \equiv \pi} \rangle) = \Psi' \quad \Gamma, x : T; \Psi' \vdash s : U}{\Gamma; \Psi \vdash \mathsf{let}\ x : T = t\ \mathsf{in}\ s : U}
$$

- 서브타입 규칙 – 경로 환경에서 동일한 경로를 찾아 사용

$$
\frac{\psi \in \Psi \quad p, q \in \psi}{\Gamma; \Psi \vdash p.L <: q.L}
$$

$$
\frac{\psi \in \Psi \quad p, q \in \psi \quad \Gamma; \Psi \vdash q : \{\mathsf{type}\ L = S .. U\} \quad \Gamma; \Psi \vdash U <: T}{\Gamma; \Psi \vdash p.L <: T}
$$

- implies 함수 – 주어진 항으로부터 동일 경로 탐색

$$
p\ \mathsf{implies}\ \langle \cdot \equiv p \rangle
$$

$$
\frac{\overline{I\ \mathsf{implies}\ \langle \overline{\rho \equiv \pi} \rangle}}{\mathsf{new}\ (\overline{I})\ \mathsf{implies}\ \langle \overline{\overline{\rho \equiv \pi}}, \cdot \equiv \mathsf{null} \rangle}
$$

- expand 함수 – 동일한 경로들을 경로 환경에 추가

$$
\frac{\begin{array}{c} makepath(x; \rho') = p \quad \Psi = \{\overline{\psi}\} \\ \forall \psi_i \left(\begin{array}{ll} \text{if} & \exists \rho''(makepath(q; \rho'') = q' \wedge q' \in \psi_i) \\ \text{then} & \psi_i' = \psi_i \cup \{p'\}\ \text{where}\ makepath(p; \rho'') = p' \\ \text{else} & \psi_i' = \psi_i \end{array} \right) \\ \Psi' = \{\overline{\psi'}\} \quad expand(\Psi', x, \langle \overline{\rho \equiv \pi} \rangle) = \Psi'' \end{array}}{expand(\Psi, x, \langle \overline{\rho \equiv \pi}, \rho' \equiv q \rangle) = \Psi''}
$$

KAIST

이 논문은 정부(과학기술정보통신부 및 정보통신기술진흥센터)의 재원으로 한국연구재단–차세대정보·컴퓨팅기술개발사업(No. 2017M3C4A7068177) 및 SW중심대학사업(2016–0–00018)의 지원을 받아 수행된 연구임.

PLRG Programming Language Research Group