Refinement Type Inference via Multi-Objective Optimization Subject to Horn Clauses

Kodai Hashimoto and Hiroshi Unno (University of Tsukuba)

{kodai, uhiro}@logic.cs.tsukuba.ac.jp

Our proposal

❖ We propose a refinement type inference method that allows users to control "the quality" of inferred types.

Unknown predicates to be inferred

```
sum :: (x: {x:int | P(x)}) -> {y:int | Q(x,y)}
let rec sum n = if n = 0 then 0 else n + sum (n-1)

maximize(P). (* find the logically weakest P *)
minimize(Q). (* find the logically strongest Q *)
```

`prioritize(P,Q)`: $(x:\{x:\mathrm{int} \mid \top\}) \to \{y:\mathrm{int} \mid y \geq 0\}$

$(x:\{x:\mathrm{int} \mid x = 0\}) \to \{y:\mathrm{int} \mid y = x\}$

`prioritize(Q,P)`: $(x:\{x:\mathrm{int} \mid x < 0\}) \to \{y:\mathrm{int} \mid \bot\}$

Overall structure (cf. [Unno and Kobayashi 2009])

Program with Type template:

```
sum :: (x:{x:int|P(x)}) -> {y:int|false}
let rec sum n =
  if n=0 then 0 else n + sum (n-1)
```

↓ Constraint generation ↓

Horn clause constraint set:

$$\exists P.\forall x.\left(\begin{array}{l}\bot \Leftarrow P(x) \land x = 0\\ P(x-1) \Leftarrow P(x) \land x \neq 0\end{array}\right)$$

↓ Multi-objective optimization ← Additional constraints for optimization: `maximize(P)`

↓ Solution: $P(x) \equiv x < 0$

Applications

Precondition inference

Infer the (preferably weakest) precondition that satisfies a given postcondition.

```
sum :: (x: {x:int | P(x)}) -> {y:int | x = y}
let rec sum n = if n<=0 then 0 else n + sum (n-1)

maximize(P).
```

inferred type:

$$(x:\{x:\mathrm{int} \mid 0 \leq x \leq 1\}) \to \{y:\mathrm{int} \mid x = y\}$$

Non-termination analysis

Infer the (preferably weakest) precondition that leads to non-termination.

```
sum :: (x: {x:int | P(x)}) -> {y:int | false}
let rec sum n = if n=0 then 0 else n + sum (n-1)

maximize(P).
```

inferred type: $(x:\{x:\mathrm{int} \mid x < 0\}) \to \{y:\mathrm{int} \mid \bot\}$

Bounds analysis

Infer the upper bound on the number of recursive calls.

```
sum :: (x: {x:int | x>=0}) -> (i: int)
       -> (c : {c:int | P(x,i,c)} -> int
(* i: initial value of x, c: # of recursive calls *)
let rec sum x i c =
  if x = 0 then 0 else x + sum (x-1) i (c+1)
minimize(P).
P(x,i,c) :- x = i && c = 0.
```

inferred type: $(x:\{x:\mathrm{int} \mid x \geq 0\}) \to (i:\mathrm{int})$
$\to (c:\{c:\mathrm{int} \mid c + x = i \land c \geq 0\}) \to \mathrm{int}$

The upper bound of c is i because $x \geq 0 \land i = x + c$ is an invariant of sum.

Our optimization algorithm

repeatedly improve an approximate solution until convergence!

We extended [Gulwani et al. 2008] to support:

❖ Horn clause constraint sets,
❖ multiple objectives, and
❖ priority orders

We implemented a prototype type inference system (demo available)

maximize

$$\exists P.\forall x.\left(\begin{array}{l}\bot \Leftarrow P(x) \land x = 0\\ P(x-1) \Leftarrow P(x) \land x \neq 0\end{array}\right)$$

↓ substitute template $P(x) \equiv c_1 x + c_2 \geq 0$ (c_1 and c_2 are unknown coefficients)

$$\exists c_1, c_2.\forall x.\left(\begin{array}{l}\bot \Leftarrow c_1 x + c_2 \geq 0 \land x = 0\\ c_1(x-1) + c_2 \geq 0 \Leftarrow c_1 x + c_2 \geq 0 \land x \neq 0\end{array}\right)$$

↓ Farkas' lemma

$$\exists c_1, c_2.\exists \ell_1, \dots \ell_4.(\ \dots\ )$$

polynomial inequalities in $c_1, c_2, \ell_1, \dots \ell_4$

→ SMT solver → $c_1 = 0$, $c_2 = -1$

→ initial solution $P(x) \equiv \bot$

→ Constraints for finding a weaker solution:

$$\exists P.\left(\begin{array}{l}\forall x.(P(x) \Leftarrow \bot)\\ \exists x.(\bot \not\Leftarrow P(x))\\ \forall x.\left(\begin{array}{l}\bot \Leftarrow P(x) \land x = 0\\ P(x-1) \Leftarrow P(x) \land x \neq 0\end{array}\right)\end{array}\right)$$

↓ ⋮ ↓ repeat until convergence

improved solution $P(x) \equiv x < -1$ → …