Equation Solving

- The MFP solution — "Maximum" (actually least) Fixed Point
  - Worklist algorithm for Monotone Frameworks

- The MOP solution — "Meet" (actually join) Over all Paths

The MFP Solution

– Idea: iterate until stabilisation.

Worklist Algorithm

Input: An instance $(L, \mathcal{F}, F, E, \iota, f_\cdot)$ of a Monotone Framework

Output: The MFP Solution: $\mathit{MFP}_\circ, \mathit{MFP}_\bullet$

Data structures:

- Analysis: the current analysis result for block entries (or exits)
- The worklist W: a list of pairs (ℓ, ℓ') indicating that the current analysis result has changed at the entry (or exit) to the block ℓ and hence the entry (or exit) information must be recomputed for ℓ'

Worklist Algorithm

Step 1 Initialisation (of W and Analysis)

W := nil;
for all (ℓ,ℓ') in F do W := cons((ℓ,ℓ'),W);
for all ℓ in F or E do
  if $\ell \in E$ then Analysis[ℓ] := ι else Analysis[ℓ] := $\bot_L$;

Step 2 Iteration (updating W and Analysis)

while W $\neq$ nil do
  ℓ := fst(head(W)); ℓ' = snd(head(W)); W := tail(W);
  if f_ℓ(Analysis[ℓ]) $\not\sqsubseteq$ Analysis[ℓ'] then
    Analysis[ℓ'] := Analysis[ℓ'] $\sqcup$ f_ℓ(Analysis[ℓ]);
    for all ℓ'' with (ℓ',ℓ'') in F do W := cons((ℓ',ℓ''),W);

Step 3 Presenting the result ($MFP_\circ$ and $MFP_\bullet$)

for all ℓ in F or E do
  $MFP_\circ(\ell)$:= Analysis[ℓ];
  $MFP_\bullet(\ell)$:= f_ℓ(Analysis[ℓ])

Correctness

The worklist algorithm always terminates and it computes the least (or MFP) solution to the instance given as input.

Complexity

Suppose that E and F contain at most $b \geq 1$ distinct labels, that F contains at most $e \geq b$ pairs, and that L has finite height at most $h \geq 1$.

Count as basic operations the applications of f_ℓ, applications of $\sqcup$, or updates of Analysis.

Then there will be at most $O(e \cdot h)$ basic operations.

Example: Reaching Definitions (assuming unique labels):

$O(b^2)$ where b is size of program: $O(h) = O(b)$ and $O(e) = O(b)$.

The MOP Solution

– Idea: propagate analysis information along paths.

Paths

The paths up to but not including ℓ:

$$\mathit{path}_{\circ}(\ell) = \{[\ell_1, \cdots, \ell_{n-1}] \mid n \geq 1 \land \forall i < n : (\ell_i, \ell_{i+1}) \in F \land \ell_n = \ell \land \ell_1 \in E\}$$

The paths up to and including ℓ:

$$\mathit{path}_{\bullet}(\ell) = \{[\ell_1, \cdots, \ell_n] \mid n \geq 1 \land \forall i < n : (\ell_i, \ell_{i+1}) \in F \land \ell_n = \ell \land \ell_1 \in E\}$$

Transfer functions for a path $\vec{\ell} = [\ell_1, \cdots, \ell_n]$:

$$f_{\vec{\ell}} = f_{\ell_n} \circ \cdots \circ f_{\ell_1} \circ id$$

The MOP Solution

The solution up to but not including ℓ:

$$MOP_\circ(\ell) = \bigsqcup\{f_{\vec{\ell}}(\iota) \mid \vec{\ell} \in path_\circ(\ell)\}$$

The solution up to and including ℓ:

$$MOP_\bullet(\ell) = \bigsqcup\{f_{\vec{\ell}}(\iota) \mid \vec{\ell} \in path_\bullet(\ell)\}$$

Precision of the MOP versus MFP solutions

The MFP solution safely approximates the MOP solution: $MFP \sqsupseteq MOP$ ("because" $f(x \sqcup y) \sqsupseteq f(x) \sqcup f(y)$ when f is monotone).

For Distributive Frameworks the MFP and MOP solutions are equal: $MFP = MOP$ ("because" $f(x \sqcup y) = f(x) \sqcup f(y)$ when f is distributive).

Lemma

Consider the MFP and MOP solutions to an instance $(L, \mathcal{F}, F, B, \iota, f.)$ of a Monotone Framework; then:

$$MFP_\circ \sqsupseteq MOP_\circ \text{ and } MFP_\bullet \sqsupseteq MOP_\bullet$$

If the framework is distributive and if $path_\circ(\ell) \neq \emptyset$ for all ℓ in E and F then:

$$MFP_\circ = MOP_\circ \text{ and } MFP_\bullet = MOP_\bullet$$

Decidability of MOP and MFP

The MFP solution is always computable (meaning that it is decidable) because of the Ascending Chain Condition.

The MOP solution is often uncomputable (meaning that it is undecidable): the existence of a general algorithm for the MOP solution would imply the decidability of the *Modified Post Correspondence Problem*, which is known to be undecidable.

Lemma

The MOP solution for Constant Propagation is undecidable.

Proof: Let $u_1, \cdots, u_n$ and $v_1, \cdots, v_n$ be strings over the alphabet $\{1,\cdots,9\}$; let $\mid u \mid$ denote the length of u; let $[\![u]\!]$ be the natural number denoted.

The Modified Post Correspondence Problem is to determine whether or not $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_n}$ for some sequence $i_1, \cdots, i_m$ with $i_1 = 1$.

```
x:=[[u1]]; y:=[[v1]];
while [···] do
   (if [···] then x:=x * 10^|u1| + [[u1]]; y:=y * 10^|v1| + [[v1]] else
    ⋮
    if [···] then x:=x * 10^|un| + [[un]]; y:=y * 10^|vn| + [[vn]] else skip)
[z:=abs((x-y)*(x-y))]^ℓ
```

Then $MOP_\bullet(\ell)$ will map z to 1 if and only if the Modified Post Correspondence Problem has no solution. This is undecidable.