I L P S ' 9 3

Incremental Equational Constraint Analyses

María Alpuente **Moreno Falaschi** **Germán Vidal**

Universidad Politécnica de Valencia Università di Padova

The Objective

Equational Logic Programming in a Constraint Setting

The Naive Solution

An instance of the CLP scheme with **narrowing** as **constraint solver**

(Basic) Conditional Narrowing

$$\frac{e \in g \quad \wedge \quad u \in \bar{O}(e) \quad \wedge \quad (\lambda \rightarrow \rho \Leftarrow \tilde{e}) \ll \mathcal{R} \ \wedge \quad \sigma = mgu(\{(e_{|u})\theta = \lambda\})}{\langle \Leftarrow g, \theta \rangle \quad \leadsto \quad \langle \Leftarrow (g \sim \{e\}) \cup \{e[u \leftarrow \rho]\} \cup \tilde{e}, \theta\sigma \rangle}$$

Concrete success set semantics

$$\boxed{\mathcal{O}(\Leftarrow g) = \{\theta \mid \ \langle \Leftarrow g, \epsilon \rangle \leadsto^* \langle \Leftarrow true, \theta \rangle\}}$$

Resolution Procedure

$$\frac{H \Leftarrow c_0 \ \Box \ \tilde{B} \ll \mathcal{P} \ \wedge \ \tilde{c} = c_0 \cup \{H = A_1\} \ \wedge \ c \stackrel{\tilde{c}}{\longrightarrow} c \cup \tilde{c}}{\Leftarrow c \ \Box \ A_1, \ldots, A_n \longrightarrow_{CLP} \ \Leftarrow c \cup \tilde{c} \ \Box \ \tilde{B}, A_2, \ldots, A_n}$$

Narrowing as Constraint Solver

$$\frac{\mathcal{O}(\Leftarrow c \cup \tilde{c}) \neq \emptyset}{c \quad \stackrel{\tilde{c}}{\longrightarrow} \quad c \cup \tilde{c}}$$

The Problem

Non-termination of (Basic) Conditional Narrowing

Non-incrementality of the Constraint Solver

The Idea

Incremental Lazy Resolution Procedure based on an Approximated Narrowing

Abstract Interpretation $\longrightarrow$ Constraint Analyzer

Abstract Conditional Narrowing

$$\frac{e \in g \quad \wedge \quad u \in \bar{O}(e) \quad \wedge \quad (\lambda \to \rho \Leftarrow \tilde{e}) \ll \mathcal{R}_{\mathcal{A}} \ \wedge \quad \sigma = mgu_{\mathcal{A}}(\{(e_{|u})\kappa = \lambda\})}{\langle \Leftarrow g, \kappa \rangle \quad \leadsto_{\mathcal{A}} \quad \langle \Leftarrow (g \sim \{e\}) \cup \{e[u \leftarrow \rho]\} \cup \tilde{e}, \kappa\sigma \rangle}$$

where

$mgu_{\mathcal{A}}$ is the abstract mgu

$\mathcal{R}_{\mathcal{A}}$ is a simplified abstract program which always terminates

Abstract success set semantics

$$\Delta(\Leftarrow g) = \{\kappa \in Sub_{\mathcal{A}} \mid \ \langle \Leftarrow g, \epsilon \rangle \leadsto^*_{\mathcal{A}} \langle \Leftarrow true, \kappa \rangle\}$$

unsatisfiability

$$\Delta(\Leftarrow g) = \emptyset \quad \to \quad g \ \ \textbf{unsatisfiable}$$

Abstract Narrowing as Constraint Analyzer

$$\frac{\Delta(\Leftarrow c \cup \tilde{c}) \neq \emptyset}{c \quad \stackrel{\tilde{c}}{\longrightarrow} \quad c \cup \tilde{c}}$$

Compositionality $\longrightarrow$ Incrementality

Concrete success set semantics is **compositional**

$$\boxed{\mathcal{O}(\Leftarrow g_1, g_2) = \mathcal{O}(\Leftarrow g_1) \Uparrow \mathcal{O}(\Leftarrow g_2)}$$

where

$$\theta_1 \Uparrow \theta_2 = mgu(\hat{\theta}_1 \cup \hat{\theta}_2)$$

$$\Theta_1 \Uparrow \Theta_2 = \begin{cases} \bigcup_{\theta_1 \in \Theta_1, \theta_2 \in \Theta_2} \theta_1 \Uparrow \theta_2 & \text{if different from } \{fail\} \\ \emptyset & \text{otherwise} \end{cases}$$

Narrowing as Incremental Constraint Solver

$$\frac{\Theta' = \Theta \Uparrow \mathcal{O}(\Leftarrow \tilde{c}) \ \wedge \ \Theta' \neq \emptyset}{\langle c, \Theta \rangle \quad \xrightarrow{\tilde{c}} \quad \langle c \cup \tilde{c}, \Theta' \rangle}$$

The Proposed Solution

Abstract success set semantics is **compositional**

$$\boxed{\Delta(\Leftarrow g_1, g_2) = \Delta(\Leftarrow g_1) \Uparrow_{\mathcal{A}} \Delta(\Leftarrow g_2)}$$

where $\kappa_1 \Uparrow_{\mathcal{A}} \kappa_2 = mgu_{\mathcal{A}}(\hat{\kappa}_1 \cup \hat{\kappa}_2)$

Lazy Resolution Procedure

$$\frac{H \Leftarrow c_0 \,\square\, \tilde{B} \ll \mathcal{P} \;\wedge\; \tilde{c} = c_0 \cup \{H = A_1\} \;\wedge\; \langle c, \Theta\rangle \xrightarrow{\tilde{c}} \langle c \cup \tilde{c}, \Theta'\rangle}{\langle c, \Theta\rangle \,\square\, A_1, \ldots, A_n \longrightarrow_{CLP} \langle c \cup \tilde{c}, \Theta'\rangle \,\square\, \tilde{B}, A_2, \ldots, A_n}$$

Abstract Narrowing as Incremental Constraint Analyzer

$$\frac{\Theta' = \Theta \Uparrow_{\mathcal{A}} \Delta(\Leftarrow \tilde{c}) \;\wedge\; \Theta' \neq \emptyset}{\langle c, \Theta\rangle \quad \xrightarrow{\tilde{c}} \quad \langle c \cup \tilde{c}, \Theta'\rangle}$$

The Program

```
knapsack(M,L,W)  ⇐ addweight(M) = W  □
                   sublist(M,L).
sublist([ ],Z).
sublist([X|Y],[X|Z])  ⇐  □  sublist(Y,Z).
sublist(Y,[X|Z])  ⇐  □  sublist(Y,Z).

addweight([a|R])  →  s(addweight(R))  ⇐.
addweight([b|R])  →  s²(addweight(R))  ⇐.
addweight([c|R])  →  s⁴(addweight(R))  ⇐.
addweight([d|R])  →  s⁶(addweight(R))  ⇐.
addweight([ ])  →  0  ⇐.
```

The Abstract Program

```
knapsack(M,L,W)  ⇐ addweight(M) = W  □
                   sublist(M,L).
sublist([ ],Z).
sublist([X|Y],[X|Z])  ⇐  □  sublist(Y,Z).
sublist(Y,[X|Z])  ⇐  □  sublist(Y,Z).

addweight([a|R])  →  s(⊥)  ⇐.
addweight([b|R])  →  s²(⊥)  ⇐.
addweight([c|R])  →  s⁴(⊥)  ⇐.
addweight([d|R])  →  s⁶(⊥)  ⇐.
addweight([ ])  →  0  ⇐.
```

The Benchmarks

constraint	CAn	ICAn	AbNar	APCom
addweight([X]) = s(0), addweight([d]) = W	0.74	0.76	0.74	0.00
addweight(Y) = $s^3(0)$	6.20	0.82	0.34	0.38
addweight([X|Y]) = $s^3(0)$, W = $s^3(0)$	**24.96**	**1.7**	**fail**	
addweight(X) = 0	0.26	0.26	0.26	0.00
addweight(Y) = $s^2(0)$	2.58	0.42	0.32	0.06
addweight(Z) = $s^3(0)$	20.82	0.80	0.36	0.32
addweight(W) = 0	**155.38**	**0.68**	0.28	0.30
addweight(X) = Z	0.32	0.44	0.32	0.00
addweight(Y) = s(0)	3.00	0.98	0.32	0.50
X = Y	**3.54**	**0.40**	0.02	0.36
addweight([a,b,X]) = Y1, addweight([Z1,Z2]) = Y2	1.36	1.80	1.38	0.00
addweight([Z1]) = $s^2(0)$, Y1 = s(Y2)	4.58	2.74	0.42	2.18
addweight([X]) = $s^2(0)$, Z1 = Z2	**32.50**	**1.96**	0.38	1.14
addweight(X) = Z	0.34	0.46	0.34	0.00
addweight(Y) = s(0)	2.98	0.98	0.32	0.52
addweight(X) = 0, X = Y	**11.96**	**0.80**	**fail**	

CAn Constraint Analyzer
ICAn Incremental Constraint Analyzer
AbNar Abstract Narrowing
APCom Abstract Parallel Composition