Program Extraction from Large Proof Developments

Luís Cruz-Filipe[a,b]

(com Bas Spitters[a])

[a] University of Nijmegen, Netherlands

[b] Centro de Lógica e Computação, Portugal

Disclaimer

Por motivos alheios à responsabilidade dos autores, não foi possível executar nenhum dos programas que aqui serão discutidos. Por este motivo, todas as afirmações de tipo

$$\text{o programa A é } \left\{ \begin{array}{c} \text{mais} \\ \text{tão} \\ \text{menos} \end{array} \right\} \text{ eficiente } \left\{ \begin{array}{c} \text{que} \\ \text{como} \\ \text{que} \end{array} \right\} \text{ o programa B}$$

devem ser interpretadas de espírito aberto.

Dead code removal

a priori (labeling)

- Existem tipos distinguidos que nunca são extraídos
- Rápida, modular

a posteriori

- Os termos que efectivamente contribuem para a computação são marcados e extraídos
- Melhor optimização

Extracção

Externa

- Reutilização de software existente: interfaces, compiladores, debuggers
- Interacção com outro software

Interna

- Compilador para redução-β
- Independência das premissas: $\exists x[P \to A(x)] \to P \to \exists x[A(x)]$
- Axioma da escolha: $\forall x \exists y[A(x,y)] \to \exists f \forall x[A(x,f(x))]$

Limitação: os sistemas SN não permitem termos incompletos.

Conectivos

$$
\begin{array}{rcl}
\neg & : & s \to \mathsf{Prop} \\
\to & : & s_1 \to s_2 \to s_2 \\
\vee & : & s_1 \to s_2 \to \mathsf{Set} \\
\wedge & : & s_1 \to s_2 \to \begin{cases} \mathsf{Prop} & s_1 = s_2 = \mathsf{Prop} \\ \mathsf{Set} & s_1 = \mathsf{Set} \text{ ou } s_2 = \mathsf{Set} \end{cases} \\
\forall & : & \Pi(A : t_\forall).(A \to s) \to s \\
\exists & : & \Pi(A : t_\exists).(A \to s) \to \mathsf{Set}
\end{array}
$$

onde $\{s, s_1, s_2\}$ denotam Set ou Prop, $t_\forall$ é um tipo proposicional ou de dados e $t_\exists$ é um tipo de dados genérico

$$\dfrac{\dfrac{\dfrac{}{|(x_m - x_n)| \leqslant \frac{\varepsilon}{2}} \quad \dfrac{}{|(y_m - y_n)| \leqslant \frac{\varepsilon}{2}}}{|(x_m - x_n) + (y_m - y_n)| \leqslant \frac{\varepsilon}{2} + \frac{\varepsilon}{2}} \leqslant + \leqslant\text{-}|\cdot| \quad \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon}{|(x_m - x_n) + (y_m - y_n)| \leqslant \varepsilon} \leqslant\text{-wd} \quad (x_m - x_n) + (y_m - y_n) = (x_m + y_m) - (x_n + y_n)}{(x+y)_m - (x+y)_n =^{*}_{\beta\delta} (x_m + y_m) - (x_n + y_n) \quad |(x_m + y_m) - (x_n + y_n)| \leqslant \varepsilon}{|(x+y)_m - (x+y)_n| \leqslant \varepsilon} \leqslant\text{-wd}$$

conv

$$\dfrac{|(x+y)_m - (x+y)_n| \leq \frac{\varepsilon}{2} \quad \frac{\varepsilon}{2} < \varepsilon}{|(x+y)_m - (x+y)_n| < \varepsilon} \leq\text{-}<\text{-trans}$$

$$
\begin{aligned}
&<\text{-}<\text{-tr} && a < b \to b < c \to a < c \\
&<\text{-}\leq\text{-tr} && a < b \to b \leq c \to a < c \\
&\leq\text{-}<\text{-tr} && a \leq b \to b < c \to a < c \\
\\
&< + <\text{-tr} && a < a' \to b < b' \to a + b < a' + b' \\
&< + \leq\text{-tr} && a < a' \to b \leq b' \to a + b < a' + b' \\
&\leq + <\text{-tr} && a \leq a' \to b < b' \to a + b < a' + b' \\
\\
&\leq\text{-}\leq\text{-tr} && a \leq b \to b \leq c \to a \leq c \\
&\leq + \leq\text{-tr} && a \leq a' \to b \leq b' \to a + b \leq a' + b' \\
\\
&<\text{-}\leq && a < b \to a \leq b
\end{aligned}
$$

Lema de Kneser

Lema: Seja $n \geq 2$. Então existe um número real $q \in]0,1[$ tal que, para todo o polinómio da forma

$$f(x) = x^n + b_{n-1}x^{n-1} + \ldots + b_1 x + b_0,$$

se verifica a seguinte desigualdade:

$$\forall_{c>|b_0|} \exists_{z\in\mathbb{C}} \left[|z| < c^{\frac{1}{n}} \wedge |f(z)| < qc \right]$$

Prova: Sejam $r = |z|$, $a_i = |b_i|$ e $q = 1 - 3^{-2n^2-n}$; então existem a_0, η, ε e k tais que a seguinte sequência de desigualdades se verifica:

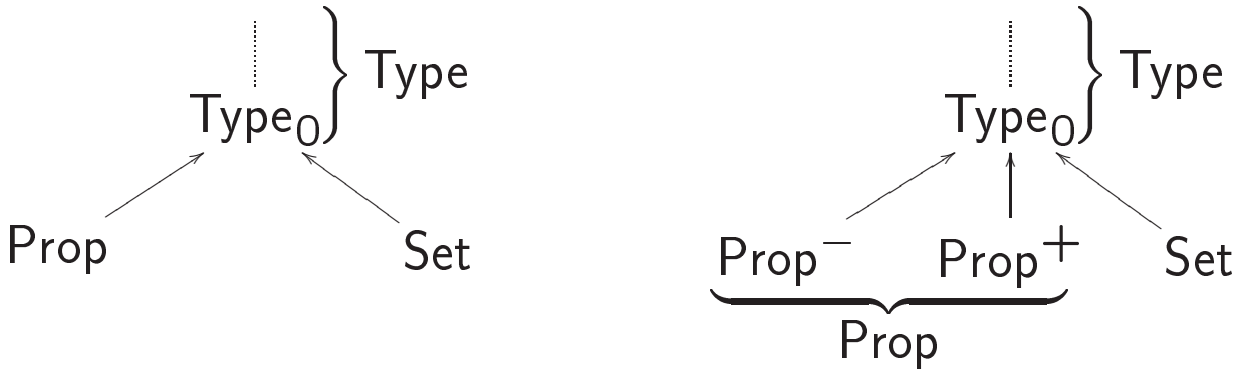
$$
\begin{aligned}
\left|\sum_{i=0}^{n} b_i z^i\right| &\leqslant \left|b_0 + b_k z^k\right| + \sum_{i\neq 0,k} a_i r^i \\
&\leqslant \left(a_0 - a_k r^k + \eta\right) + \left(\left(1 - 3^{-n}\right) a_k r^k + 3^n \varepsilon\right) \\
&= a_0 - 3^{-n} a_k r^k + 3^n \varepsilon + \eta \\
&\leqslant a_0 - 3^{-n}\left(3^{-2n^2} a_0 - 2\varepsilon\right) + 3^n \varepsilon + \eta \\
&= \left(1 - 3^{-2n^2-n}\right) a_0 + 3^n \varepsilon + 3^{-n} 2\varepsilon + \eta \\
&\leqslant \left(1 - 3^{-2n^2-n}\right) a_0 + 3^n \varepsilon + \varepsilon + \eta \\
&= q a_0 + 3^n \varepsilon + \varepsilon + \eta \\
&< qc
\end{aligned}
$$

$$\frac{|f(z)| \leq qa_0 + 3^n\varepsilon + \varepsilon + \eta \quad qa_0 + 3^n\varepsilon + \varepsilon + \eta < qc}{|f(z)| < qc} \leq\text{-}<\text{-tr}$$

Alteração	Reais (Mb)	fta (Mb)	Total (Mb)	Δ(%)
Original	7.5	7.5	15	
Def. seq. Cauchy	1.5	6.5	8	47
Lema de Kneser	1.5	5.0	6.5	19
Divisão	1.4	2.0	3.4	48
Diversos	1.4	1.6	3.0	12

Descrição	Tamanho (kb)	% do total
Código "relevante"	110	6.5
inlining de $\mathbb{C}$	1050	62.5
inlining de polinómios ($R[x]$)	330	19.5
Coerções	190	11.5
Total	1680	100

Uma possível solução. . .



Hierarquia de tipos em Coq: actual (à esquerda) e proposta (à direita)

Conectivos II

$$
\begin{array}{rcl}
\neg & : & \mathsf{Prop} \to \mathsf{Prop}^- \\
\to & : & \mathsf{Prop} \to s \to s \\
\vee & : & \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop}^+ \\
\underline{\vee} & : & \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop}^- \\
\wedge & : & s_1 \to s_2 \to \begin{cases} \mathsf{Prop}^- & s_1 = s_2 = \mathsf{Prop}^- \\ \mathsf{Prop}^+ & s_1 = \mathsf{Prop}^+ \text{ ou } s_2 = \mathsf{Prop}^+ \end{cases} \\
\forall & : & \Pi(A : t_\forall).(A \to s) \to s \\
\exists & : & \Pi(A : t_\exists).(A \to \mathsf{Prop}) \to \mathsf{Prop}^+ \\
\underline{\exists} & : & \Pi(A : t_\exists).(A \to \mathsf{Prop}) \to \mathsf{Prop}^-
\end{array}
$$

onde $\{s, s_1, s_2\}$ denotam Prop^+ ou Prop^-, $t_\forall$ é um tipo proposicional ou de dados e $t_\exists$ é um tipo de dados genérico

Conclusões

- Motivação proveniente de exemplos
- Dicas referentes à escrita de provas
- Melhoramento do mecanismo de extracção