Rank-2 Intersection Types for Cost Analysis of Functional Programs

Hugo R. Simões[1] Kevin Hammond[1] Mário Florido[2]
Pedro B. Vasconcelos[1]

[1]University of St Andrews

[2]Universidade do Porto

TYPES, 2006

Motivation

For doing cost analysis in general

- Compiler optimization
- Parallel computing
- Real-time systems
- ...

This work in particular

- Reducing size-aliasing
- Increasing the set of typable programs

Motivation

For doing cost analysis in general

- Compiler optimization
- Parallel computing
- Real-time systems
- ...

This work in particular

- Reducing size-aliasing
- Increasing the set of typable programs

Outline

Language $\mathcal{L}$

- Language:

$$
\begin{array}{rcl}
e & ::= & x \mid \lambda x.e \mid e_1\, e_2 \\
 & \mid & n \\
 & \mid & \mathit{true} \mid \mathit{false} \\
 & \mid & [\,] \mid e_1 :: e_2 \\
 & \mid & \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \\
 & \mid & p_1(e) \mid p_2(e_1, e_2)
\end{array}
$$

- β-reduction: $(\lambda x.e_1)e_2 \to_\beta e_1[e_2/x]$
- Evaluation order: call-by-value
- Weak normal forms: $(\lambda x.e) \not\to_\beta$

Sized-Time Rank-2 ITS

Judgements

$$A \vdash e : v$$

Types

$$\begin{aligned} u &::= \alpha \mid \text{Bool} \mid \text{Nat} \mid \text{List}\ u \mid u_1 \to u_2 \\ v &::= u \mid u_1 \wedge \ldots \wedge u_n \to v \end{aligned}$$

Rank-2 Intersection Types

Sized-Time Rank-2 ITS

Judgements

$$A \vdash e : v$$

Types

$$
\begin{aligned}
u &::= \alpha \mid \text{Bool} \mid \text{Nat}^{z} \mid \text{List}^{z}\, u \mid u_1 \rightarrow u_2 \\
v &::= u \mid u_1 \wedge \ldots \wedge u_n \rightarrow v \\
z &::= l \mid n \mid z_1 + z_2 \mid \omega
\end{aligned}
$$

Sized-Types

Sized-Time Rank-2 ITS

Judgements

$$A \vdash e : v$$

Types

$$\begin{array}{lcl} u & ::= & \alpha \mid \text{Bool} \mid \text{Nat}^{z} \mid \text{List}^{z}\,u \mid u_1 \rightarrow u_2 \\ v & ::= & u \mid u_1 \wedge \ldots \wedge u_n \rightarrow v \\ z & ::= & l \mid n \mid z_1 + z_2 \mid \omega \end{array}$$

$\text{List}^{5}\text{Nat}^{10}$

Sized-Types

Sized-Time Rank-2 ITS

Judgements

$$A \vdash e : v \;\&\; z$$

Types

$$\begin{aligned} u &::= \alpha \mid \mathrm{Bool} \mid \mathrm{Nat}^z \mid \mathrm{List}^z\, u \mid u_1 \xrightarrow{z} u_2 \\ v &::= u \mid u_1 \wedge \ldots \wedge u_n \xrightarrow{z} v \\ z &::= l \mid n \mid z_1 + z_2 \mid \omega \end{aligned}$$

Typing Rules

$$\frac{}{\{x : u\} \vdash x : u \;\&\; 0}\;[Var_{\wedge_{2st}}]$$

$$\frac{x \in \mathrm{FV}(e) \quad A, x : u_1 \wedge \ldots \wedge u_n \vdash e : v \;\&\; z}{A \vdash \lambda x.e : u_1 \wedge \ldots \wedge u_n \xrightarrow{z} v \;\&\; 0}\;[Abs_{\wedge_{2st}}]$$

$$\frac{x \notin \mathrm{FV}(e) \quad u \in \mathbf{T}_0 \quad A \vdash e : v \;\&\; z}{A \vdash \lambda x.e : u \xrightarrow{z} v \;\&\; 0}\;[AbsVac_{\wedge_{2st}}]$$

$$\frac{\begin{array}{c} A_0 \vdash e_1 : u_1 \wedge \ldots \wedge u_n \xrightarrow{z_3} v \;\&\; z_1 \\ (\forall i \in \{1, \ldots, n\})\; A_i \vdash e_2 : u_i \;\&\; z_2 \end{array}}{A_0 \wedge A_1 \wedge \ldots \wedge A_n \vdash e_1\, e_2 : v \;\&\; 1 + z_1 + z_2 + z_3}\;[App_{\wedge_{2st}}]$$

$$\frac{A_1 \vdash e : v_1 \;\&\; z_1 \quad A_2 \leq_1 A_1 \quad v_1 \leq_2 v_2 \quad z_1 \leq z_2}{A_2 \vdash e : v_2 \;\&\; z_2}\;[Sub_{\wedge_{2st}}]$$

Typing Rules (cont.)

$$\frac{n \in \mathbb{N}}{\emptyset \vdash n : \mathrm{Nat}^{n} \;\&\; 0}\;[Nat_{\wedge_{2st}}] \qquad \frac{b \in \{true, false\}}{\emptyset \vdash b : \mathrm{Bool} \;\&\; 0}\;[Bool_{\wedge_{2st}}]$$

$$\frac{u \in \mathbf{T}_0}{\emptyset \vdash [\,] : \mathrm{List}^{0} u \;\&\; 0}\;[Nil_{\wedge_{2st}}]$$

$$\frac{A_1 \vdash e_1 : u \;\&\; z_1 \quad A_2 \vdash e_2 : \mathrm{List}^{z} u \;\&\; z_2}{A_1 \wedge A_2 \vdash e_1 :: e_2 : \mathrm{List}^{1+z} u \;\&\; z_1 + z_2}\;[Cons_{\wedge_{2st}}]$$

$$\frac{A_0 \vdash e_0 : Bool \;\&\; z_0 \quad A_1 \vdash e_1 : u \;\&\; z \quad A_2 \vdash e_2 : u \;\&\; z}{A_0 \wedge A_1 \wedge A_2 \vdash \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : u \;\&\; z_0 + z}\;[If_{\wedge_{2st}}]$$

Subtyping Relations

$(\leq_2)$

$$\frac{u \trianglelefteq u'}{u \leq_2 u'}\ [\text{simple}_{\leq_2}]$$

$$\frac{u'_1 \wedge \ldots \wedge u'_m \leq_1 u_1 \wedge \ldots \wedge u_n \qquad v \leq_2 v' \qquad z \leq z'}{u_1 \wedge \ldots \wedge u_n \xrightarrow{z} v \leq_2 u'_1 \wedge \ldots \wedge u'_m \xrightarrow{z'} v'}\ [\text{rank2}_{\leq_2}]$$

$(\leq_1)$

$$\frac{n \geq m \quad \exists i_1, \ldots, i_m \in \{1, \ldots, n\} : u_{i_1} \trianglelefteq u'_1, \ldots, u_{i_m} \trianglelefteq u'_m}{u_1 \wedge \ldots \wedge u_n \leq_1 u'_1 \wedge \ldots \wedge u'_m}\ [\text{rank1}_{\leq_1}]$$

$(\trianglelefteq)$

$$\frac{u = u'}{u \trianglelefteq u'}\ [\text{reflex}_{\trianglelefteq}] \qquad \frac{u'_1 \trianglelefteq u_1 \qquad u_2 \trianglelefteq u'_2 \qquad z \leq z'}{u_1 \xrightarrow{z} u_2 \trianglelefteq u'_1 \xrightarrow{z'} u'_2}\ [\text{abs}_{\trianglelefteq}]$$

$$\frac{z \leq z'}{\text{Nat}^{z} \trianglelefteq \text{Nat}^{z'}}\ [\text{nat}_{\trianglelefteq}] \qquad \frac{z \leq z' \quad u \trianglelefteq u'}{\text{List}^{z}\, u \trianglelefteq \text{List}^{z'}\, u'}\ [\text{list}_{\trianglelefteq}]$$

Example: reducing size-aliasing

...based on Hindley-Milner

$$twice \equiv \lambda f\, x. f\,(f\, x) \quad : (a \xrightarrow{l} a) \xrightarrow{0} a \xrightarrow{2+l+l} a$$

$$succ \equiv \lambda y. add(y, 1) : \mathrm{Nat}^m \xrightarrow{0} \mathrm{Nat}^{m+1}$$

$$\frac{A \vdash e_1 : u \xrightarrow{z_3} u' \;\&\; z_1 \qquad A \vdash e_2 : u \;\&\; z_2}{A \vdash e_1\, e_2 : u' \;\&\; 1 + z_1 + z_2 + z_3}\ [App_{HM_{st}}]$$

$twice\, succ\ :$?

Example: reducing size-aliasing

...based on Hindley-Milner

$$twice \equiv \lambda f\, x. f\,(f\,x) \quad : (a \xrightarrow{l} a) \xrightarrow{0} a \xrightarrow{2+l+l} a$$

$$succ \equiv \lambda y. add(y, 1) : \mathrm{Nat}^{m} \xrightarrow{0} \mathrm{Nat}^{m+1}$$

$$\frac{A \vdash e_1 : u \xrightarrow{z_3} u' \;\&\; z_1 \qquad A \vdash e_2 : u \;\&\; z_2}{A \vdash e_1\, e_2 : u' \;\&\; 1 + z_1 + z_2 + z_3}\,[App_{HM_{st}}]$$

$$twice\, succ \;:\; \mathrm{Nat}^{\omega} \xrightarrow{2} \mathrm{Nat}^{\omega}$$

Example: reducing size-aliasing

...based on a rank-2 ITS

$$twice \equiv \lambda f\,x.f\,(f\,x) \quad : (a \xrightarrow{l_1} b) \wedge (b \xrightarrow{l_2} c) \xrightarrow{0} a \xrightarrow{2+l_1+l_2} c$$

$$succ \equiv \lambda y.add(y,1) : \mathrm{Nat}^m \xrightarrow{0} \mathrm{Nat}^{m+1}$$

$$\frac{\begin{array}{c} A_0 \vdash e_1 : u_1 \wedge \ldots \wedge u_n \xrightarrow{z_3} v \;\&\; z_1 \\ (\forall i \in \{1, \ldots, n\})\ A_i \vdash e_2 : u_i \;\&\; z_2 \end{array}}{A_0 \wedge A_1 \wedge \ldots \wedge A_n \vdash e_1\ e_2 : v \;\&\; 1 + z_1 + z_2 + z_3}\ [App_{\wedge_{2st}}]$$

$twice\; succ \;$: ?

Example: reducing size-aliasing

...based on a rank-2 ITS

$$twice \equiv \lambda f\, x. f\,(f\,x) \quad : (a \xrightarrow{l_1} b) \wedge (b \xrightarrow{l_2} c) \xrightarrow{0} a \xrightarrow{2+l_1+l_2} c$$

$$succ \equiv \lambda y. add(y, 1) : \text{Nat}^{m} \xrightarrow{0} \text{Nat}^{m+1}$$

$$\frac{A_0 \vdash e_1 : u_1 \wedge \ldots \wedge u_n \xrightarrow{z_3} v \;\&\; z_1 \qquad (\forall i \in \{1, \ldots, n\})\; A_i \vdash e_2 : u_i \;\&\; z_2}{A_0 \wedge A_1 \wedge \ldots \wedge A_n \vdash e_1\; e_2 : v \;\&\; 1 + z_1 + z_2 + z_3}\;[App_{\wedge_{2st}}]$$

$$twice\; succ \;:\; \text{Nat}^{m} \xrightarrow{2} \text{Nat}^{m+2}$$

Example: increasing the set of typable programs

...based on Hindley-Milner

$$twice \equiv \lambda f\, x. f\,(f\, x) : (a \xrightarrow{l} a) \xrightarrow{0} a \xrightarrow{2+l+l} a$$

$$tolist \equiv \lambda y.[y] \qquad : \alpha \xrightarrow{0} \mathrm{List}^1 \alpha$$

$$\frac{A \vdash e_1 : u \xrightarrow{z_3} u' \;\&\; z_1 \qquad A \vdash e_2 : u \;\&\; z_2}{A \vdash e_1\, e_2 : u' \;\&\; 1 + z_1 + z_2 + z_3}\ [App_{HM_{st}}]$$

$twice\ tolist$: ?

Example: increasing the set of typable programs

...based on Hindley-Milner

$$twice \equiv \lambda f\, x. f\,(f\, x) : (a \xrightarrow{l} a) \xrightarrow{0} a \xrightarrow{2+l+l} a$$

$$tolist \equiv \lambda y.[y] \qquad : \alpha \xrightarrow{0} \mathrm{List}^1 \alpha$$

$$\frac{A \vdash e_1 : u \xrightarrow{z_3} u' \ \&\ z_1 \qquad A \vdash e_2 : u \ \&\ z_2}{A \vdash e_1\, e_2 : u' \ \&\ 1 + z_1 + z_2 + z_3}\ [App_{HM_{st}}]$$

twice tolist : not typable

Example: increasing the set of typable programs

...based on a rank-2 ITS

$$twice \equiv \lambda f\, x. f\,(f\,x) : (a \xrightarrow{l_1} b) \wedge (b \xrightarrow{l_2} c) \xrightarrow{0} a \xrightarrow{2+l_1+l_2} c$$

$$tolist \equiv \lambda y.[y] \qquad : \alpha \xrightarrow{0} \mathrm{List}^1 \alpha$$

$$\frac{\begin{array}{c} A_0 \vdash e_1 : u_1 \wedge \ldots \wedge u_n \xrightarrow{z_3} v \ \&\ z_1 \\ (\forall i \in \{1, \ldots, n\})\ A_i \vdash e_2 : u_i \ \&\ z_2 \end{array}}{A_0 \wedge A_1 \wedge \ldots \wedge A_n \vdash e_1\, e_2 : v \ \&\ 1 + z_1 + z_2 + z_3}\ [App_{\wedge_{2st}}]$$

$twice\ tolist$: ?

Example: increasing the set of typable programs

...based on a rank-2 ITS

$$twice \equiv \lambda f\, x. f\,(f\,x) : (a \xrightarrow{l_1} b) \wedge (b \xrightarrow{l_2} c) \xrightarrow{0} a \xrightarrow{2+l_1+l_2} c$$

$$tolist \equiv \lambda y.[y] \qquad : \alpha \xrightarrow{0} \mathrm{List}^1 \alpha$$

$$\frac{\begin{array}{c} A_0 \vdash e_1 : u_1 \wedge \ldots \wedge u_n \xrightarrow{z_3} v \;\&\; z_1 \\ (\forall i \in \{1, \ldots, n\})\; A_i \vdash e_2 : u_i \;\&\; z_2 \end{array}}{A_0 \wedge A_1 \wedge \ldots \wedge A_n \vdash e_1\, e_2 : v \;\&\; 1 + z_1 + z_2 + z_3} \; [App_{\wedge_{2st}}]$$

$$twice\; tolist \; : \; \alpha \xrightarrow{2} \mathrm{List}^1 \mathrm{List}^1 \alpha$$

Correctness Results

- **Theorem**
  (Conservative extension) If $A \vdash_{\wedge_2} e : v$ *then there exists* $A' \vdash e : v' \,\&\, z'$ *for some* A', v' *and* z'.

- **Theorem**
  (Subject reduction) If $e \to e'$ *and* $A \vdash e : v \,\&\, z$, *then there exists a judgement* $A' \vdash e' : v \,\&\, z$ *where* $A' \subseteq A$.

- **Theorem**
  (Cost correctness) If $e \to e'$ *is a* β*-reduction and* $A \vdash e : v \,\&\, z$ *then there exists* z' *such that* $1 + z' \leq z$ *and* $A' \vdash e' : v \,\&\, z'$ *where* $A' \subseteq A$.

- **Theorem**
  (At least the same results as in the Hindley-Milner approach) If $A \vdash_{HM_{st}} e : u \,\&\, z$ *then there exists* $A' \vdash e^* : u \,\&\, z$ *for some* A' *where* $e^* =$ *replace let by app in e.*

Correctness Results

- Theorem
  (Conservative extension) If $A \vdash_{\wedge_2} e : v$ *then there exists* $A' \vdash e : v' \;\&\; z'$ *for some* A', v' *and* z'.

- Theorem
  (Subject reduction) If $e \to e'$ *and* $A \vdash e : v \;\&\; z$*, then there exists a judgement* $A' \vdash e' : v \;\&\; z$ *where* $A' \subseteq A$.

- Theorem
  (Cost correctness) If $e \to e'$ *is a* β*-reduction and* $A \vdash e : v \;\&\; z$ *then there exists* z' *such that* $1 + z' \leq z$ *and* $A' \vdash e' : v \;\&\; z'$ *where* $A' \subseteq A$.

- Theorem
  (At least the same results as in the Hindley-Milner approach) If $A \vdash_{HM_{st}} e : u \;\&\; z$ *then there exists* $A' \vdash e^* : u \;\&\; z$ *for some* A' *where* $e^* =$ *replace let by app in e.*

Conclusions and Future Work

Conclusions

- More precise size information
- Cost analysis: good application domain for intersection types

Future Work

- Cost inference algorithm
- Recursion