Constructing Set Constraints for ReScript

서울대학교 전기·정보공학부 2018-12602 이준협

1 Definition of set expressions

$$
\begin{array}{rcll}
se & ::= & \emptyset & \textit{empty set} \\
& | & \_ & \textit{maximum set} \\
& | & () & \textit{unit} \\
& | & n & \textit{integer} \\
& | & b & \textit{boolean} \\
& | & \lambda x.e & \textit{function} \\
& | & V_e & \textit{set variable corresponding to the possible values of } e \\
& | & P_e & \textit{set variable corresponding to the possible exn packets of } e \\
& | & \mathsf{body}_V(se) & \textit{values that } se \textit{ can spit out when } se \textit{ is applied to something} \\
& | & \mathsf{body}_P(se) & \textit{exn packets that } se \textit{ can spit out when } se \textit{ is applied to something} \\
& | & \mathsf{par}(se) & \textit{values that can be a parameter for } se \\
& | & \kappa & \textit{constructor} \\
& | & l & \textit{field of a record} \\
& | & \mathsf{con}(\kappa, se) & \textit{construct} \\
& | & \mathsf{exn}(\kappa, se) & \textit{exception} \\
& | & \mathsf{fld}(se, l) & \textit{contents of the field } l \textit{ of a record } se \\
& | & \mathsf{cnt}(se) & \textit{contents of a reference} \\
& | & \mathsf{bop}(se, se) & \textit{binary operators, where } \mathsf{bop} \in \{+, -, \times, \div, =, <, >\} \\
& | & f_{(i)}^{-1}(se) & \textit{projection onto the i-th argument of } f \\
& | & se \cup se & \textit{union} \\
& | & se \cap se & \textit{intersection} \\
& | & \overline{se} & \textit{complement}
\end{array}
$$

The definition of the conditional set expression needs clarification.

$$se_1 \Rightarrow se_2 := \begin{cases} \emptyset & (se_1 = \emptyset) \\ se_2 & (o.w) \end{cases}$$

The conditional set expression is a naive approximation for pattern matching. Consider the case when we want to match an expression against the record pattern with fields x and y. The constraints describing the record $r = \{x = 1, y = 2\}$ are $1 \subseteq \mathsf{fld}(V_r, x) \wedge 2 \subseteq \mathsf{fld}(V_r, y)$. To pattern-match r against $\{x, y\}$, we want the fields x and y of V_r to be nonempty. Thus, the value of "*match r with* $\{x, y\} \to e$" is $\mathsf{fld}(V_r, x) \Rightarrow (\mathsf{fld}(V_r, y) \Rightarrow V_e)$.

The conditional set expression can also be used to define conditional set constraints [1].

$$se \Rightarrow \bigwedge_{i=1}^{n} X_i \subseteq Y_i := \bigwedge_{i=1}^{n} (se \Rightarrow X_i) \subseteq Y_i$$

2 Constructing set constraints

Now we are in a position to define constraint construction rules for our ReScript-like language. Hopefully this would be reasonably fast when implemented and be accurate enough...

$$[\text{UNIT, INT, BOOL}] \ \frac{}{\rhd c : V_e \supseteq c} \ c = (), n, b$$

$$[\text{APP}] \ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2}{\rhd e_1\ e_2 : (V_e \supseteq \mathsf{body}_V(V_{e_1})) \wedge (P_e \supseteq (\mathsf{body}_P(P_{e_1}) \cup P_{e_1} \cup P_{e_2})) \wedge (\mathsf{par}(V_{e_1}) \supseteq V_{e_2}) \wedge C_1 \wedge C_2}$$

$$[\text{FN}] \ \frac{\rhd e' : C'}{\rhd \lambda x.e' : (V_e \supseteq \lambda x.e') \wedge (\mathsf{body}_V(V_e) \supseteq V_{e'}) \wedge (\mathsf{body}_P(V_e) \supseteq P_{e'}) \wedge (\mathsf{par}(V_e) \subseteq V_x) \wedge C'}$$

$$[\text{LET}]\ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2}{\rhd \mathit{let}\ x = e_1\ \mathit{in}\ e_2 : (V_x \supseteq V_{e_1}) \wedge (V_e \supseteq V_{e_2}) \wedge (P_e \supseteq P_{e_1} \cup P_{e_2}) \wedge C_1 \wedge C_2}$$

$$[\text{BOP}]\ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2}{\rhd e_1\ \mathsf{bop}\ e_2 : (V_e \supseteq \mathsf{bop}(V_{e_1}, V_{e_2})) \wedge (P_e \supseteq P_{e_1} \cup P_{e_2}) \wedge \underbrace{(P_e \supseteq (V_{e_2} \cap 0 \Rightarrow \mathsf{exn}(\mathsf{Divide\_by\_zero}, ())))}_{\text{when bop is } \div} \wedge C_1 \wedge C_2}$$

$$[\text{CON}]\ \frac{\rhd e' : C'}{\rhd \mathsf{con}\ \kappa\ e' : (V_e \supseteq \mathsf{con}(\kappa, V_{e'})) \wedge (P_e \supseteq P_{e'}) \wedge C'}$$

$$[\text{EXN}]\ \frac{\rhd e' : C'}{\rhd \mathsf{exn}\ \kappa\ e' : (V_e \supseteq \mathsf{exn}(\kappa, V_{e'})) \wedge (P_e \supseteq P_{e'}) \wedge C'}$$

$$[\text{DECON}]\ \frac{\rhd e' : C'}{\rhd \mathsf{decon}\ \kappa\ e' : (V_e \supseteq \mathsf{con}^{-1}_{(2)}(V_{e'} \cap \mathsf{con}(\kappa, \_)) \cup \mathsf{exn}^{-1}_{(2)}(V_{e'})) \wedge (P_e \supseteq P_{e'}) \wedge C'}\ \text{as in SML-NJ's Lambda}$$

$$[\text{RECORD}]\ \frac{\rhd e_i : C_i\ (1 \le i \le n)}{\rhd \{l_1 = e_i, ..., l_n = e_n\} : \bigwedge_{i=1}^n (\mathsf{fld}(V_e, l_i) \supseteq V_{e_i}) \wedge (P_e \supseteq \bigcup_{i=1}^n P_{e_i}) \wedge \bigwedge_{i=1}^n C_i}$$

$$[\text{FIELD}]\ \frac{\rhd e' : C'}{\rhd e'.l : (V_e \supseteq \mathsf{fld}(V_{e'}, l)) \wedge (P_e \supseteq P_{e'}) \wedge C'}$$

$$[\text{REF}]\ \frac{\rhd e' : C'}{\rhd \mathsf{ref}\ e' : (V_e \supseteq \mathsf{loc}) \wedge (\mathsf{cnt}(\mathsf{loc}) \supseteq V_{e'}) \wedge (P_e \supseteq P_{e'}) \wedge C'}\ \text{new loc}$$

$$[\text{UPDATE}]\ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2}{\rhd e_1 := e_2 : (\mathsf{cnt}(V_{e_1}) \supseteq V_{e_2}) \wedge (P_e \supseteq P_{e_1} \cup P_{e_2}) \wedge C_1 \wedge C_2}$$

$$[\text{BANG}]\ \frac{\rhd e' : C'}{\rhd !e' : (V_e \supseteq \mathsf{cnt}(V_{e'})) \wedge (P_e \supseteq P_{e'}) \wedge C'}$$

We define an auxilary function for generating constraints out of pattern matching. If we want to figure out the constraint for the value X of "match Y with $p \to e$":

$$\mathsf{case}(X, Y, p, e) := \begin{cases} (Y \subseteq V_x) \wedge (V_e \subseteq X) & (p = x) \\ (\mathsf{fld}(Y, l_1) \Rightarrow ... \Rightarrow \mathsf{fld}(Y, l_n) \Rightarrow V_e) \subseteq X & (p = \{l_i\}_{i=1}^n) \\ (\kappa \cap (\mathsf{con}^{-1}_{(1)}(Y) \cup \mathsf{exn}^{-1}_{(1)}(Y))) \Rightarrow V_e \subseteq X & (p = \kappa) \\ (Y \cap c) \Rightarrow V_e \subseteq X & (p = \mathsf{constant}) \\ V_e \subseteq X & (p = \_) \end{cases}$$

$$[\text{CASE}]\ \frac{\rhd e' : C' \qquad \rhd e_i : C_i\ (1 \le i \le n)}{\rhd \mathsf{case}\ e'\ (p_i \to e_i)_{i=1}^n : \bigwedge_{i=1}^n \mathsf{case}(V_e, V_{e'}, p_i, e_i) \wedge (P_e \supseteq \bigcup_{i=1}^n P_i \cup P_{e'}) \wedge C' \wedge \bigwedge_{i=1}^n C_i}$$

$$[\text{HANDLE}]\ \frac{\rhd e' : C' \qquad \rhd e_i : C_i\ (1 \le i \le n)}{\rhd \mathsf{handle}\ e'\ (p_i \to e_i)_{i=1}^n : (P_e \supseteq (P_{e'} \cap \bigcap_{i=1}^n \overline{\mathsf{exn}(p_i, \_)}) \cup \bigcup_{i=1}^n P_{e_i}) \wedge (V_e \supseteq V_{e'} \cup \bigcup_{i=1}^n V_{e_i}) \wedge C' \wedge \bigwedge_{i=1}^n C_i}$$

$$[\text{RAISE}]\ \frac{\rhd e' : C'}{\rhd \mathsf{raise}\ e' : (P_e \supseteq V_{e_1} \cup P_{e_1}) \wedge C'}$$

$$[\text{FOR}]\ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2 \qquad \rhd e_3 : C_3}{\rhd \mathsf{for}\ x\ e_1\ e_2\ e_3 : \begin{array}{l} (>(V_x, V_{e_1}) \cap \mathtt{true} \Rightarrow () \subseteq V_e) \\ \wedge (>(V_x, V_{e_1}) \cap \mathtt{false} \Rightarrow (C_3 \wedge (V_x \supseteq +(V_x, 1)) \wedge (P_e \supseteq P_{e_3}))) \\ \wedge (V_x \supseteq V_{e_1}) \wedge (P_e \supseteq P_{e_1} \cup P_{e_2}) \wedge C_1 \wedge C_2 \end{array}}$$

$$[\text{WHILE}]\ \frac{\rhd e_1 : C_1 \qquad \rhd e_2 : C_2}{\rhd \mathsf{while}\ e_1\ e_2 : \begin{array}{l} (V_{e_1} \cap \mathtt{true}) \Rightarrow (C_2 \wedge (P_e \supseteq P_{e_2})) \\ \wedge ((V_{e_1} \cap \mathtt{false}) \Rightarrow () \subseteq V_e) \wedge (P_e \supseteq P_{e_1}) \wedge C_1 \end{array}}$$

References

[1] Alexander Aiken. "Introduction to set constraint-based program analysis". In: *Science of Computer Programming* 35.2 (1999), pp. 79–111. ISSN: 0167-6423. DOI: `https://doi.org/10.1016/S0167-6423(99)00007-6`.