Algorithmus zur Erstellung der Funktion Υ

Gegeben sind die Definitionen 1 und 4 des Papers.

Die Funktion $\Upsilon : S_{exit} \to \mathbb{P}(\mathbb{P}(\mathbb{T}))$ ist gemäß Paper für alle $s \in S_{exit}$ definiert durch

$$\begin{aligned}\Upsilon(s) \quad = \quad & \{ \bigcup_{r \in dsr(s)} (\{f(r)\} \cup F(r)) \,|\, f : dsr(s) \to \mathbb{T} \wedge F : dsr(s) \to \mathbb{P}(\mathbb{T}) \wedge \\ & \forall r \in dsr(s) : \pi_{tar}(f(r)) = s \wedge regOf(\pi_{sor}(f(r))) = r \wedge F(r) \in \Upsilon(\pi_{sor}(f(r)))\}\end{aligned}$$

Da dsr nicht für Exitstates, sondern nur für CompositeStates definiert ist, ersetzen wir $dsr(s)$ durch $dsr(parent(s))$, erweitern den Definitionsbereich von dsr und setzen $dsr(s) = \emptyset$ für alle $s \in S_{final}$. Als korrigierte Definition ergibt sich für alle $s \in S_{exit}$

$$\begin{aligned}\Upsilon(s) \quad = \quad & \{ \bigcup_{r \in dsr(s)} (\{f(r)\} \cup F(r)) \,|\, f : dsr(stateOf(s)) \to \mathbb{T} \wedge F : dsr(stateOf(s)) \to \mathbb{P}(\mathbb{T}) \wedge \\ & \forall r \in dsr(stateOf(s)) : \pi_{tar}(f(r)) = s \wedge regOf(\pi_{sor}(f(r))) = r \wedge F(r) \in \Upsilon(\pi_{sor}(f(r)))\}\end{aligned}$$

Diese Definition ist so nicht verwendbar, da weder die Funktion dsr noch die Menge $\mathbb{T}$ aller Transitionen direkt implementiert ist.

Schritt 1: konkrete Mengen verwenden Sei $s_{com} \in S_{com} \cup S_{final}$. Seien $s_{exit} := \{s \in S_{exit} \,|\, parent(s) = s_{com}\}$, $R := dsr(s_{com})$. Diese Mengen können durch rekursives Durchlaufen des Komponentenbaumes leicht ermittelt werden.

Sei $T : s_{exit} \to \mathbb{P}(\mathbb{T})$ mit $T(s) = \{t \in \mathbb{T} \,|\, \pi_{tar}(t) = s\}$ die im Sourcecode TRANSITIONSPERExitstate genannte Funktion. Dann ist für alle $s \in s_{exit}$:

$$\begin{aligned}\Upsilon(s) \quad = \quad & \{ \bigcup_{r \in R} (\{f(r)\} \cup F(r)) \,|\, f : R \to T(s) \wedge F : R \to \mathbb{P}(\mathbb{T}) \wedge \\ & \forall r \in R : regOf(\pi_{sor}(f(r))) = r \wedge F(r) \in \Upsilon(\pi_{sor}(f(r)))\}\end{aligned}$$

Diese Definition ist noch nicht verwendbar, weil die Mengen aller Funktionen $f : R \to T(s)$ und $F : R \to \mathbb{P}(\mathbb{T})$ aus Effizienzgründen nicht kostruiert werden sollen.

Schritt 2: f und F aus der Definition herausziehen Seien für alle $s \in s_{exit}$, $f : R \to \mathbb{T}$:

$$\Phi(s) := \{f : R \to T(s) \,|\, \forall r \in R : regOf(\pi_{sor}(f(r))) = r\}$$

$$\Psi(f) := \{F : R \to \mathbb{P}(\mathbb{T}) \,|\, \forall r \in R : F(r) \in \Upsilon(\pi_{sor}(f(r)))\}$$

Dann ergibt sich für alle $s \in s_{exit}$:

$$\Upsilon(s) = \{ \bigcup_{r \in R} (\{f(r)\} \cup F(r)) \,|\, f \in \Phi(s) \wedge F \in \Psi(f)\}$$

Schritt 3: Rekursion über R Wir wählen ein festes $s \in s_{exit}$ und parametrisieren Φ und Ψ mit R. Das heißt, wir setzen $\Phi(s) = \Phi(s, R)$ und $\Psi(s) = \Psi(s, R)$. Jetzt können wir Φ und Ψ rekursiv formulieren:

$$
\begin{aligned}
\Phi(s, \emptyset) &= \{\emptyset\} \\
\Phi(s, R \uplus \{r\}) &= \{\phi \uplus (r, t) \mid \phi \in \Phi(s, R) \wedge t \in T(s) \wedge regOf(\pi_{sor}(t)) = r\} \\
\Psi(f, \emptyset) &= \{\emptyset\} \\
\Psi(f, R \uplus \{r\}) &= \{\psi \uplus (r, t) \mid \psi \in \Psi(f, R) \wedge t \in \Upsilon(\pi_{sor}(f(r)))\}
\end{aligned}
$$

Diese Rekursionen können durch Iterationen über R implementiert werden.

Beobachtung: Für alle $s \in S_{exit}$ mit $parent(s) \in S_{final}$ ist $\Phi(s) = \{\emptyset\}$, $\Psi(\Phi(s)) = \{\emptyset\}$ und damit $\Upsilon(s) = \{\emptyset\}$

Schritt 4: Algorithmus Setze zunächst $\Upsilon := \emptyset$. Traversiere dann den Komponentenbaum in post-order. Setze dabei für alle $s \in S_{exit}$ mit $parent(s) \in S_{final}$ $\Upsilon(s) = \{\emptyset\}$ und verwende für alle $s_{com} \in S_{com}$ den folgenden Algorithmus:

`Eingabe:` $s_{com} \in S_{com}$`, partielle Funktion` $\Upsilon : S_{exit} \rightarrow \mathbb{P}(\mathbb{P}(\mathbb{T}))$
s_{exit}`:=`$\{s \in S_{exit} \mid parent(s) = s_{com}\}$
R`:=`$dsr(s_{com})$`.`
`forall` $s \in s_{exit}$`:`
    $\Upsilon(s)$`:=`$\emptyset$
    Φ`:=`$\{\emptyset\}$
    `forall` $r \in R$`:`
        Φ`:=`$\{\phi \uplus (r, t) \mid \phi \in \Phi \wedge t \in T(s) \wedge regOf(\pi_{sor}(t)) = r\}$
    `forall` $f \in \Phi$`:`
        Ψ`:=`$\{\emptyset\}$
        `forall` $r \in R$`:`
            Ψ`:=`$\{\psi \uplus (r, t) \mid \psi \in \Psi \wedge t \in \Upsilon(\pi_{sor}(f(r)))\}$
        `forall` $F \in \Psi$`:`
            υ`:=`$\bigcup_{r \in R}(\{f(r)\} \cup F(r))$
            $\Upsilon(s) = \Upsilon(s) \cup \{\upsilon\}$
`Ausgabe:` Υ