

## 4.2 State-transition-system

---

---

---

---



前回やっていた State-transition system を具体的な例で考え方

Def A STS consists of

本とは違うSTSです。

$S$ : 状態の集合

$\rightarrow \subseteq S \times S$ : transition relation

$T$ : token の集合

$L$ :  $S \rightarrow P(T)$  protocol owned tokens

$\varphi$ :  $S \rightarrow iProp$  ?

$S = \{\text{pending}, \text{shot}(1), \text{shot}(2)\}$

$\rightarrow = \{\text{pending} \rightarrow \text{shot}(1), \text{pending} \rightarrow \text{shot}(2)\}$

$T = \{1, 2\}$

$L(\text{pending}) = \emptyset, L(\text{shot}(i)) = \{i\}$

?

Def token の増減を反映した transition relation

$$(s, T) \rightarrow (s', T') := s \rightarrow s' \wedge L(s) \uplus T = L(s') \uplus T'$$

Example

$$(\text{pending}, \{i\} \uplus T) \rightarrow (\text{shot}(i), \emptyset \uplus T)$$

Def  $T$  に含まれる token を貰わなくても可能な transition

$$s \xrightarrow{\emptyset} s' := \exists T_1, T_2. \quad T_1 \# (L(s) \cup T) \wedge \underset{T_1 \# T}{(s, T_1) \rightarrow (s', T_2)}$$

Example

$$\text{pending} \xrightarrow{\emptyset} \text{shot}(1). \quad \text{pending} \not\xrightarrow{\{1, 2\}} s.$$

$$\left( T' \subseteq T \wedge s \xrightarrow{\emptyset} s' \Rightarrow s \xrightarrow{\emptyset} s' \wedge, \right) \\ s \xrightarrow{\emptyset} s' \Leftrightarrow s \rightarrow s' \text{ が成立}.$$

Def  $T$  に含まれている token を ラベルに持たず、

貰わない 遷移は閉じて閉じていること。

$$\text{closed}(S, T) := \forall s \in S, L(s) \# T \wedge (\forall s', s \xrightarrow{*} s' \Rightarrow s' \in S)$$

$T$  に含まれている token を 貰わない遷移は閉じる閉包

$$\uparrow(s, T) := \{s' \in S \mid \exists s \in S. s \xrightarrow{*} s'\}$$

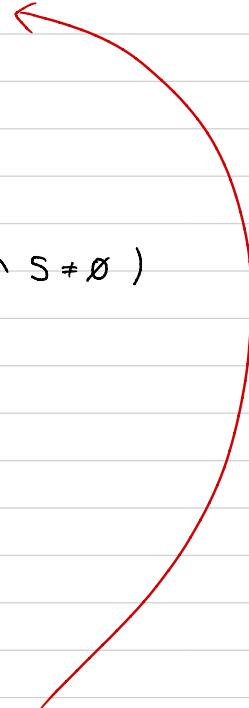
Claim  $S \models^* \forall s \in S, \lfloor s \rfloor \# T \Leftrightarrow \exists t \in T, \text{closed}(\uparrow(s, t), T)$ .

Example

$\text{closed}(\{\text{pending}\}, \{1, 2\})$ .

$\text{closed}(\{\text{shot}(i)\}, \emptyset)$ .

$\text{closed}(\{\text{pending}, \text{shot}(2)\}, \{1\})$ . etc.



Def (STS RA)

$M := \text{auth}(s : S, T : P(T) \mid \lfloor s \rfloor \# T)$

|  $\text{frag}(S : P(S), T : P(T) \mid \text{closed}(S, T) \wedge S \neq \emptyset)$

| {

$\mathcal{V}(\alpha) := \alpha \neq \emptyset$

Example

$$M = \left\{ \begin{array}{ll} \text{auth}(\text{pending}, T), & (T: \text{任意}) \\ \text{auth}(\text{shot}(i), T \mid i \notin T), & \\ \{ , \} \\ \text{frag}(\{\text{pending}\}, \{1, 2\}), & \\ \vdots & \\ \text{frag}(S, T) & (S \neq \emptyset, \text{closed}(S, T)) \end{array} \right.$$

Def

$\text{frag}(S_1, T_1) \cdot \text{frag}(S_2, T_2) := \text{frag}(S_1 \cap S_2, T_1 \cup T_2)$

if  $T_1 \# T_2$  and  $S_1 \cap S_2 \neq \emptyset$

$\text{frag}(S, T) \cdot \text{auth}(s, T') := \text{auth}(s, T') \cdot \text{frag}(S, T) := \text{auth}(s, T \cup T')$

if  $T \# T'$  and  $s \in S$

$|\text{frag}(S, T)| := \text{frag}(\uparrow(S, \emptyset), \emptyset)$

$|\text{auth}(s, T)| := \text{frag}(\uparrow(\{s\}, \emptyset), \emptyset)$

$T_1 \# T_2 \text{ の時しか合成立りません}$

$(1 \text{ resource } \Rightarrow \text{thread } N \text{ まで})$

Example  $\text{frag}(\{\text{pending}, \text{shot}(1)\}, \{2\}) \cdot \text{frag}(\{\text{pending}, \text{shot}(2)\}, \{1\})$

=  $\text{frag}(\{\text{pending}\}, \{1, 2\})$

$$\begin{aligned}
 \left| \text{auth}(\{\text{pending}\}, \{1\}) \right| &= \text{frag}(\uparrow(\{\text{pending}\}, \emptyset), \emptyset) \\
 \text{STS-STEP} \quad (s, T) \xrightarrow{*} (s', T') &= \text{frag}(\{\text{pending}, \text{shot}(1), \text{shot}(2)\}, \emptyset) \\
 \text{auth}(s, T) \rightsquigarrow \text{auth}(s', T')
 \end{aligned}$$

$\text{auth}(\{\text{pending}\}, \{1\}) \rightsquigarrow B$  は  $B$  の条件を満たす。

- $\text{auth}(\{\text{pending}\}, \{1\}) \cdot \text{frag}(S, T) \neq \emptyset$  は  $S$  と  $T$  を満たす  $B$  。
- $\{1\} \# T$  なら  $T = \emptyset, \{2\}$  。
  - $\text{pending} \in S$

（注目）

$$T = \emptyset \text{ なら } S \supseteq \uparrow(\{\text{pending}\}, \emptyset) = \{\text{pending}, \text{shot}(1), \text{shot}(2)\} \supseteq S.$$

$S$  の全体で  $T \neq \emptyset$  なら  $T$  は unit である意味。

$$T = \{2\} \text{ なら } S \supseteq \uparrow(\{\text{pending}\}, \emptyset) = \{\text{pending}, \text{shot}(1)\}.$$

$$\{(\text{shot}(2)) \cap T \neq \emptyset \text{ なら } S \neq \text{shot}(2). \text{ したがって } S = \{\text{pending}, \text{shot}(1)\}$$

$$\text{（この場合、} \text{auth}(\{\text{pending}\}, \{1\}) \cdot \text{frag}(\{\text{pending}, \text{shot}(1)\}, \{2\}) = \text{auth}(\{\text{pending}\}, \{1, 2\}) \text{ となる。} \text{ つまり、} B \text{ は } \text{IR} \text{ の } m \in M \text{ を含むことが必要十分}.$$

$$m \cdot \text{frag}(\{\text{pending}, \text{shot}(1)\}, \{2\}) \neq \emptyset.$$

これが  $\text{IR}$  のような場合しかない。

$$\begin{aligned}
 \text{auth}(\text{pending}, \{1\}) \rightsquigarrow &\text{ auth}(\text{pending}, \emptyset), \text{ auth}(\text{pending}, \{1\}), \text{ auth}(\text{shot}(1), \emptyset), \text{ auth}(\text{shot}(1), \{1\}), \\
 &\text{ frag}(S, \emptyset) \text{ where } S = \{\text{shot}(1), \{\text{shot}(1), \text{shot}(2)\}, \{\text{shot}(1)\}, \\
 &\text{ frag}(\{\text{pending}, \text{shot}(2)\}, \{1\}) \text{ ただし } (\Rightarrow \text{ これは単体で } B \text{ が } \emptyset \text{ です})
 \end{aligned}$$

Recall STS = 逆概念  $\Psi: S \rightarrow \text{Prop}$  で  $\forall$ .

Def.  $\text{StsInv}_\Psi := \exists S. [\text{auth}(S, \emptyset)]^* * \Psi(S).$

Example.  $\Psi(\text{pending}) := x \mapsto \text{inl}(0)$

$\Psi(\text{shot}(i)) := x \mapsto \text{inr}(i)$  定義。

$$\text{前回例: } I := (x \mapsto \text{inl}(0) * [\text{pending}(x)]) \vee (\exists n. x \mapsto \text{inr}(n) * [\text{shot}(n)])$$

$$\begin{aligned}
 &= (\underline{x \mapsto \text{inl}(0)} \ * \ \underline{\text{pending}(\underline{z})}) \\
 &\vee (\underline{x \mapsto \text{inr}(1)} \ * \ \underline{\text{shot}(\underline{z})}) \\
 &\vee (\underline{x \mapsto \text{inr}(2)} \ * \ \underline{\text{shot}(\underline{z})})
 \end{aligned}$$

は、 $\text{StsInv}_r \in \text{PC}$ .

Def. at least in some state  $s$ , owning some tokens  $T$ .

$$\text{StsStr}(s, T) := \text{frag}(\underline{\uparrow(s, T)}, T)$$

Example

$$\begin{aligned}
 \text{StsStr}(\text{pending}, \{1, 2\}) &= \text{frag}(\underline{\text{pending}}, \underline{\{1, 2\}}) \\
 \text{StsStr}(\text{pending}, \{2\}) &= \text{frag}(\underline{\text{pending}}, \underline{\text{shot}(1)}), \underline{\{2\}})
 \end{aligned}$$

Thm.  $\vdash R$  が derivable.

atomic ( $e$ ),  $\mathcal{N} \subseteq E$ , timeless ( $\varphi$ ),

$$V_S. \quad s_1 \xrightarrow{T_1} s \Rightarrow \{ \varphi(s) * Q_1 \} \in \{ v. \exists s_2 \exists T_2, (s, T_1) \rightarrow (s_2, T_2) * \varphi(s_2) * Q_2 \}_{E \setminus N}$$

$$\{ \boxed{\text{StsInv}_r} \}^* * \text{StsStr}(s_1, T_1) * Q_1 \} \in \{ v. \exists s_2 \exists T_2. \boxed{\text{StsInv}_r}^* * \text{StsStr}(s_2, T_2) * Q_2 \}_{E \setminus N}$$

•  $Q_1, Q_2$  は無視OK.

• これまでの例だと、 $s_1 \xrightarrow{T_1} s \wedge (s, T_1) \rightarrow (s_2, T_2)$  を同時に成立させないので使えない。  
 $\text{pending} \rightarrow \text{dummy} \xrightarrow{\text{shot 1}} \text{shot 2}$  はなぜいい?

結論は、"  $s_1$  に居て (移動したかもしないが)  $T_1$  を持ったままでいる" & invariant ならば、 $e$  の実行後に " 何らかの state  $s_2$  に居て  $T_2$  を持つ" & invariant.

これを示すために、 $S_1$  から  $T_1$  を体得してから任意の  $s$  に到達し、

$T_1$  を使って別の移動可能な state にちゃんと到達する

(STS とアロケーションが整合的である、という意味か)

## Example

pending  $\xrightarrow{\text{st}} \text{dummy}$   $\xrightarrow[\text{set}(2)]{\text{set}(1)} \text{shot}(1)$   $\vdash t_1, t_2 \vdash 3.$

$e = \text{set}()$ ,  $T_1 = \{1, 2\}$   $\vdash 3 \vdash$ ,

$$\frac{\left\{ \begin{array}{l} x \mapsto \text{in}(0) \\ \text{set}() \end{array} \right\} \text{ set}() \quad \left\{ \begin{array}{l} \exists S_2, T_2. \quad (\text{dummy } \{1, 2\}) \rightarrow (S_2, T_2) * \wp(S_2) \\ \text{shot}(i) \\ \{1, 2\} \setminus \{i\} \end{array} \right\}}{\left\{ \begin{array}{l} I * \text{frag}(\{p, d\}, \{1, 2\}) \\ \text{set}() \quad \left\{ \begin{array}{l} \exists S_2, T_2. \quad I * \text{frag}(\{S_2\}, T_2) \end{array} \right\} \end{array} \right\}}$$