

Remarks:

- A binary relation R over some set \mathbb{D} is called **total** if for every $d_1 \in \mathbb{D}$ there is $d_2 \in \mathbb{D}$ with $(d_1, d_2) \in R$.
- To indicate the basis of an STS Γ we will also write $\Gamma(SIG, S)$. Furthermore, we will often write $SIG_\Gamma, \mathbf{S}_\Gamma, \mathbf{F}_\Gamma, \mathbf{P}_\Gamma, S_\Gamma, X_\Gamma, V_\Gamma, T_\Gamma, W_\Gamma$ for the single ingredients of a given $\Gamma(SIG, S)$ and depict an execution sequence $(\eta_0, \eta_1, \eta_2, \dots)$ by $\eta_0 \longrightarrow \eta_1 \longrightarrow \eta_2 \longrightarrow \dots$
- An STS is called first-order in the definition because the system variables of the set X range over arbitrary sorts of individuals. If $X = \emptyset$ then the STS is called **propositional**. (In this case, SIG and S are irrelevant and can be omitted from the definition. The corresponding theory is a **propositional theory**.)

Specifications

Definition(Language $\mathcal{L}_{T\Gamma}$):

Let $\Gamma = (X, V, S, T)$ be an STS over SIG and S . A language $\mathcal{L}_T(TSIG_\Gamma)$ with $TSIG_\Gamma = (SIG, X, V)$ is called **language of linear temporal logic of Γ** and denoted by $\mathcal{L}_{T\Gamma}$. A formula of $ker(\mathcal{L}_{T\Gamma})$ is called **state formula** of Γ .

Note: For every execution sequence W of Γ the pair $K = (S, W)$ is a temporal structure for $TSIG_\Gamma$.

Definition(Γ -Validity):

For a transition system Γ (over SIG and S_Γ) let

$$\mathcal{C}_\Gamma = \{K = (S_\Gamma, W_\Gamma) \mid W_\Gamma \text{ is an execution sequence of } \Gamma\}.$$

A formula A of $\mathcal{L}_{T\Gamma}$ is called **Γ -valid** (denoted by $\models_\Gamma A$) if A is valid in every $K \in \mathcal{C}_\Gamma$.

A **\mathcal{C}_Γ -FOLTL-theory** (briefly: **Γ -theory**) is called **FOLTL-Specification** of Γ .

Rooted Transition System

Definition(Rooted Transition System):

A **rooted (state) transition system** (briefly: rSTS)

$\Gamma = (X, V, S, T, start)$ (over some SIG and S) is an STS $\Gamma'(SIG, S) = (X, V, S, T)$

together with a closed formula $start$ of $ker(\mathcal{L}_{T\Gamma'})$ called **initial condition**. An

execution sequence of Γ is an execution sequence $(\eta_0, \eta_1, \eta_2, \dots)$ of Γ' with

$S^{(\eta_0)}(start) = tt$.

(A formula is called closed if it does not contain free variable occurrences. For

closed formulas A we also write $S^{(\eta_i)}(A)$ instead of $S^{(\xi, \eta_i)}(A)$ since this value does not depend on ξ .)

Theorem:

For every rSTS Γ the following axiom is Γ -valid:

$(root_\Gamma) \quad \mathbf{init} \rightarrow start_\Gamma.$

Example(Towers of Hanoi):

$$SIG = (\{STONE, PILE\}, \{TOWER, EMPTY, PUSH, POP, TOP\}, \{<, DECR\})$$

$$S : \quad |S|_{STONE} = \{1, \dots, n\} \quad |S|_{PILE} = \{1, \dots, n\}^*$$

$$S(TOWER) = (n, n - 1, \dots, 2, 1),$$

$$S(EMPTY) = \varepsilon,$$

$$S(PUSH) = push,$$

$$S(POP) = pop,$$

$$S(TOP) = top,$$

$$S(<)(i, j) = \mathbf{tt} \Leftrightarrow i < j,$$

$$S(DECR)(i_1, \dots, i_m) = \mathbf{tt} \Leftrightarrow i_m < i_{m-1} < \dots < i_1$$

rSTS $\Gamma(SIG, S)$:

$$X = X_{PILE} = \{p_1, p_2, p_3\}$$

$$V = \emptyset$$

$$S = \{\eta \mid \eta : X_{PILE} \rightarrow \{1, \dots, n\}^*\}$$

$$T = \left\{ \begin{array}{l} (\eta, \eta') \in S \times S \mid \eta(p_i) \neq \varepsilon, \text{top}(\eta(p_i)) < \text{top}(\eta(p_j)) \text{ if } \eta(p_j) \neq \varepsilon, \\ \eta'(p_i) = \text{pop}(\eta(p_i)), \\ \eta'(p_j) = \text{push}(\eta(p_j), \text{top}(\eta(p_i))), \\ \eta'(p_k) = \eta(p_k), \\ i, j, k \in \{1, 2, 3\} \quad (\text{pairwise distinct}) \end{array} \right\} \Bigg\} T'$$

$$\cup \{(\eta, \eta) \in S \times S \mid \text{there does not exist an } \eta' \in S \text{ with } (\eta, \eta') \in T'\}$$

Notation: $T = \text{tot}(T')$: **total closure** of T' .

$$\text{start} \equiv p_1 = \text{TOWER} \wedge p_2 = \text{EMPTY} \wedge p_3 = \text{EMPTY}$$

Specification and Verification of State Systems: Actions and Fairness

Labeled STS

Definition(Labeled STS):

A **labeled (state) transition system** (briefly: lSTS) $\Gamma = (X, V, S, T, Act)$ is given by

- a finite set Act of **actions**
- an STS $\Gamma' = (X, V, S, T)$ with V containing elements $exec\ \lambda$ for every $\lambda \in Act$ and such that, if $(\eta, \eta') \in T$ and $\eta(exec\ \lambda) = \text{ff}$ for every $\lambda \in Act$, then $\eta' = \eta$.

An **execution sequence** of Γ is an execution sequence of Γ' .

Abbreviation: For any lSTS Γ with $Act = \{\lambda_1, \dots, \lambda_n\}$ let

$$nil_{\Gamma} = \neg exec\ \lambda_1 \wedge \dots \wedge \neg exec\ \lambda_n.$$

Remark: Rooted labeled STS (rlSTS) are defined analogously.

Theorem:

For every (r)ISTS Γ the following axiom is Γ -valid:

$$(\text{nil}_\Gamma) \quad \text{nil}_\Gamma \wedge A \rightarrow \circ A \quad \text{if } A \text{ is a state formula of } \Gamma$$

Conclusion:

Let Γ be an (r)ISTS. The following derived rule is Γ -valid:

$$\text{exec } \lambda \wedge A \rightarrow \circ B \quad \text{for all } \lambda \in \text{Act}_\Gamma$$

$$(\text{trans}) \quad \text{nil}_\Gamma \wedge A \rightarrow B$$

$$\vdash A \rightarrow \circ B \quad \text{if } A \text{ and } B \text{ are state formulas of } \Gamma$$

Fair STS

Definition(Fair STS):

A **fair STS** (briefly: fSTS) is an ISTS $\Gamma = (X, V, S, T, Act)$ with a formula $enabled_\lambda$ of $ker(\mathcal{L}_\Gamma)$ for every $\lambda \in Act$ (enabling condition). An **execution sequence** of Γ is a sequence (η_0, η_1, \dots) (in the previous sense) with the following additional properties:

- For all $\lambda \in Act$ and $i \in \mathbb{N}_0$: If $S_\Gamma^{(\eta_i)}(exec\ \lambda) = tt$ then $S_\Gamma^{(\eta_i)}(enabled_\lambda) = tt$.
- For all $\lambda \in Act$: If $S_\Gamma^{(\eta_k)}(enabled_\lambda) = tt$ for infinitely many k then $S_\Gamma^{(\eta_k)}(exec\ \lambda) = tt$ for infinitely many k (**fair execution sequence**).

Remark:

Rooted fair STS (rfSTS) are defined analogously.