Verifying Featured Transition Systems using Variability Parity Games

Sjef van Loo

May 1, 2019

1 Definitions

1.1 Transition systems

Similar to [1].

Definition 1.1. *An LTS is a tuple $M = (S, Act, trans, I)$, where:*

- *S is a set of states,*
- *Act a set of actions,*
- *$trans \subseteq S \times Act \times S$ is the transition relation with $(s, a, s') \in trans$ denoted by $s \xrightarrow{a} s'$,*
- *$I \subseteq S$ is a set of initial states.*

Definition 1.2. *An FTS is a tuple $M = (S, Act, trans, I, N, P, \gamma)$, where:*

- *$S, Act, trans, I$ are defined as in an LTS,*
- *N is a set of features,*
- *$P \subseteq \mathcal{P}(N)$ is a set of products, ie. feature assignments, that are valid,*
- *$\gamma : trans \to \mathbb{B}(N)$ is a total function, labelling each transition with a Boolean expression over the features. A product $p \in \mathcal{P}(N)$ satisfying the Boolean expression of transition t is denoted by $p \models \gamma(t)$, $\gamma(t)(p) = 1$ or $p \in [\![\gamma(t)]\!]$.*

  A transition $s \xrightarrow{a} s'$ and $\gamma((s, a, s')) = f$ is denoted by $s \xrightarrow{a/f} s'$.

Definition 1.3. *The projection of an FTS M to a product $p \in P$, noted $M_{|p}$, is the LTS $M' = (S, Act, trans', I)$, where $trans' = \{t \in trans \mid p \models \gamma(t)\}$.*

2 Goal

Similar to [2].

Given an FTS $M = (S, Act, trans, I, N, P, \gamma)$ and a modal μ-calculus formula φ we want to find the set $P_s \subseteq P$ such that:

- for every $p \in P_s$ we have $M_{|p} \models \varphi$,
- for every $p \in P \backslash P_s$ we have $M_{|p} \not\models \varphi$.

Furthermore for every $p \in P \backslash P_s$ we want a counter example.

If $P_s = P$, ie. all products satisfy φ, we write $M \models \varphi$.

3 Variability Parity Games

Definition 3.1. *A variability parity game is a tuple $G = (V, V_0, V_1, E, \rho, N, A, \gamma)$, where:*

- *$V = V_0 \cup V_1$ and $V_0 \cap V_1 = \emptyset$,*
- *V_0 is the set of vertices for player 0,*
- *V_1 is the set of vertices for player 1,*
- *$E \subseteq V \times V$ is the edge relation,*

- *$\rho : V \to \mathbb{N}$ is a priority assignment,*
- *N is a set of features,*
- *$A \subseteq \mathcal{P}(N)$ is a set of feature assignments for which the game can be played,*
- *$\gamma : E \to \mathbb{B}(N)$ is a total function, labelling each edge with a Boolean expression over the features.*

A VPG is played for a specific $a \in A$. A path π is valid iff for all pairs π_i and π_{i+1} in π we have $(\pi_i, \pi_{i+1}) \in E$ and $a \models \gamma((\pi_i, \pi_{i+1}))$.

Not deadlock free, so player $\alpha \in \{0, 1\}$ wins iff $\overline{\alpha}$ can't make a move or if the highest priority occurring infinitely often has the same parity as α.

For an $a \in A$ we have winning sets W_0^a and W_1^a.

Definition 3.2. *The projection of a VPG $G = (V, V_0, V_1, E, \rho, N, A, \gamma)$ to an assignment $a \in A$, noted $G_{|a}$, is the PG $G' = (V, V_0, V_1, E', \rho)$, where $E' = \{e \in E \mid a \models \gamma(e)\}$.*

Definition 3.3. *LTS2PG(M, φ) converts LTS M and formula φ to a PG.*
Use some existing (proven) method.
...

Definition 3.4. *FTS2VPG(M, φ) converts FTS M and formula φ to a VPG. Very similar to LTS2PG, guard edges created by diamond or box operators. The set of features in the VPG is equal to the set of features in the FTS, similarly the set of feature assignments in the VPG is equal to the set of valid products in the FTS.*
...

Theorem 3.1. *W_α^a for VPG $G = (V, V_0, V_1, E, \rho, N, A, \gamma)$ is equal to W_α in $G_{|a}$ for any $a \in A$ and $\alpha \in \{0, 1\}$.*

From this theorem it follows that the VPG is positionally determined.

Lemma 3.2. *Given*

- *FTS $M = (S, Act, trans, I, N, P, \gamma)$,*
- *formula φ and*
- *$p \in P$*

FTS2VPG(M, φ)$_{|p}$ is equal to LTS2PG($M_{|p}, \varphi$) .

Theorem 3.3. *Given*

- *FTS $M = (S, Act, trans, I, N, P, \gamma)$,*
- *formula φ,*
- *$p \in P$ and*
- *state $s \in S$*

it holds that M satisfies φ for product p in state s iff $s \in W_0^p$ in FTS2VPG(M, φ).

Proof. Winning set W_0^p in FTS2VPG(M, φ) is equal to winning set W_0 in FTS2VPG$(M, \varphi)_{|p}$ (using theorem 3.1). Furthermore FTS2VPG$(M, \varphi)_{|p}$ is equal to LTS2PG$(M_{|p}, \varphi)$ (using lemma 3.2).

So winning set W_0^p in FTS2VPG(M, φ) is equal to winning set W_0 in LTS2PG$(M_{|p}, \varphi)$. Since $M_{|p}$ satisfies φ in state s iff $s \in W_0$ in LTS2PG$(M_{|p}, \varphi)$ (existing LTS verification theory) the theorem holds. □

References

[1] A. Classen, M. Cordy, P.-Y. Schobbens, P. Heymans, A. Legay, and J.-F. Raskin, "Featured transition systems: Foundations for verifying variability-intensive systems and their application to ltl model checking," *IEEE Transactions on Software Engineering*, vol. 39, pp. 1069–1089, 2013.

[2] A. Classen, P. Heymans, P. Y. Schobbens, A. Legay, and J.-P. Raskin, "Model checking lots of systems: Efficient verification of temporal properties in software product lines," vol. 1, 01 2010.