Verifying Featured Transition Systems using Variability Parity Games

Sjef van Loo

May 6, 2019

1 Definitions

1.1 Transition systems

Similar to [1].

Definition 1.1. *An LTS is a tuple $M = (S, Act, trans, s_0)$, 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 $s \xrightarrow{a} s'$,*
- *$s_0 \in S$ is the initial state.*

Definition 1.2. *An FTS is a tuple $M = (S, Act, trans, s_0, N, P, \gamma)$, where:*

- *$S, Act, trans, s_0$ 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 $M' = (S, Act, trans', s_0)$, where $trans' = \{t \in trans \mid p \models \gamma(t)\}$.*

Definition 1.4. *[3] A modal μ-calculus formula over the set of actions $\mathcal{A}$ and a set of variables $\mathcal{X}$ is defined by*

$$\varphi = \top \mid \bot \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

with $a \in \mathcal{A}$ and $X \in \mathcal{X}$.

No negations in the language because negations can be pushed inside to the propositions, ie. the $\top$ and $\bot$ elements..

A fixed point formula φ with variable X can be unfolded; the occurrences of X are replaced by φ. A fixed point formula is equivalent to its unfolding, ie. $\mu X.\varphi$ is equivalent to $\mu X.\varphi[X := \mu X.\varphi]$. [3]

2 Goal

Similar to [2].

Given an FTS $M = (S, Act, trans, s_0, N, P, \gamma)$ 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 $M_{|p} \not\models \varphi$.

A counterexample for every $p \in P \backslash P_s$ is preferred.

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

3 Parity Games

3.1 Parity games

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

- $V = V_0 \cup V_1$ *and* $V_0 \cap V_1 = \emptyset$,
- V_0 *is the set of vertices owned by player* 0,
- V_1 *is the set of vertices owned by player* 1,
- $E \subseteq V \times V$ *is the edge relation,*
- $\rho : V \to \mathbb{N}$ *is a priority assignment.*

3.2 Featured parity games

Definition 3.2. *A featured parity game (FPG) is a tuple $G = (V, V_0, V_1, E, \rho, N, P, \gamma, v_0)$, where:*

- $V = V_0 \cup V_1$ *and* $V_0 \cap V_1 = \emptyset$,
- V_0 *is the set of vertices owned by player* 0,
- V_1 *is the set of vertices owned by 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,*
- P *is a set of products*
- $\gamma : E \to \mathbb{B}(N)$ *is a total function, labelling each edge with a Boolean expression over the features,*
- $v_0 \in V$ *is a starting vertex.*

An FPG is played by player 0 and 1. A FPG is played for a specific product $p \in P$, the game starts by placing a token on a vertex $v \in V$. When the token is on vertex $v \in V$ that is owned by player $\alpha \in \{0, 1\}$, ie. $v \in V_\alpha$, then player α can move the token to another vertex w such that $(v, w) \in E$ and $p \models \gamma(v, w)$.

For a $p \in P$ we have winnings sets $W_0^p \subseteq V$ and $W_1^p \subseteq V$.

Definition 3.3. *The projection from FPG $G^F = (V^F, V_0^F, V_1^F, E^F, \rho^F, N, P, \gamma, v_0)$ to a product $p \in P$, noted $G^F_{|p}$, is the parity game (V, V_0, V_1, E, ρ) where:*

- (V, E) *is the subgraph of* (V^F, E') *that is reachable from* $v_0 \in V^F$. *With* $E' = \{e \in E^F | p \models \gamma(e)\}$,
- $V_0 = V_0^F \cap V$,
- $V_1 = V_1^F \cap V$,
- $\rho(v) = \rho^F(v)$ *for all* $v \in V$.

3.3 Variability parity games

Definition 3.4. *A variability parity game (VPG) is a tuple $G = (V, V_0, V_1, E, \rho, \mathfrak{C}, \theta)$, where:*

- $V = V_0 \cup V_1$ *and* $V_0 \cap V_1 = \emptyset$,
- V_0 *is the set of vertices owned by player* 0,
- V_1 *is the set of vertices owned by player* 1,
- $E \subseteq V \times V$ *is the edge relation; we assume that* E *is total, i.e. for all* $v \in V$ *there is some* $w \in V$ *such that* $(v, w) \in E$,
- $\rho : V \to \mathbb{N}$ *is a priority assignment,*
- $\mathfrak{C}$ *is a finite set of configurations,*
- $\theta : E \to \mathcal{P}(\mathfrak{C}) \setminus \{0\}$ *is the configuration mapping, satisfying for all* $v \in V$, $\bigcup\{\theta(v, w) | (v, w) \in E\} = \mathfrak{C}$.

A VPG is played by player 0 and 1. A VPG is played for a specific $c \in \mathfrak{C}$, the game starts by placing a token on a vertex $v \in V$. When the token is on vertex $v \in V$ that is owned by player $\alpha \in \{0, 1\}$, ie. $v \in V_\alpha$, then player α can move the token to another vertex w such that $(v, w) \in E$ and $c \in \theta(v, w)$.

For a $c \in \mathfrak{C}$ we have winnings sets $W_0^c \subseteq V$ and $W_1^c \subseteq V$.

3.4 Relations

$$\begin{tikzcd}
\text{FTS} \arrow[r, "\varphi"] \arrow[d, Rightarrow, "\Pi"'] & \text{FPG} \arrow[r, dashed] \arrow[d, Rightarrow, "\Pi"] & \text{VPG} \\
\text{LTS} \arrow[r, "\varphi"] & \text{PG} &
\end{tikzcd}$$

3.5 Creating parity games

Definition 3.5. *[3] LTS2PG(M,φ) converts LTS $M = (S, Act, trans, s_0)$ and formula φ to a PG (V, V_0, V_1, E, ρ).*

A vertex in the parity game is represented by a pair (s, ψ) where $s \in S$ and ψ is a modal μ-calculus formula.

We create the parity game with the smallest sets V, V_0, V_1, E such that:

- $V = V_0 \cup V_1$,
- $V_0 \cap V_1 = \emptyset$,
- $(s_0, \varphi) \in V$,
- *for every $v = (s, \psi) \in V$ we have:*
  - *If $\psi = \top$ then $v \in V_1$.*
  - *If $\psi = \bot$ then $v \in V_0$.*
  - *If $\psi = \psi_1 \vee \psi_2$ then:*
    $v \in V_0$,
    $(s, \psi_1) \in V$,
    $(s, \psi_2) \in V$,
    $(v, (s, \psi_1)) \in E$ *and*
    $(v, (s, \psi_2)) \in E$.
  - *If $\psi = \psi_1 \wedge \psi_2$ then:*
    $v \in V_1$,
    $(s, \psi_1) \in V$,
    $(s, \psi_2) \in V$,
    $(v, (s, \psi_1)) \in E$ *and*
    $(v, (s, \psi_2)) \in E$.
  - *If $\psi = \langle a \rangle \psi_1$ then $v \in V_0$ and for every $s \xrightarrow{a} s'$ we have $(s', \psi_1) \in V$ and $(v, (s', \psi_1)) \in E$.*
  - *If $\psi = [a] \psi_1$ then $v \in V_1$ and for every $s \xrightarrow{a} s'$ we have $(s', \psi_1) \in V$ and $(v, (s', \psi_1)) \in E$.*
  - *If $\psi = \mu X.\psi_1$ then $(v, (s, \psi_1(\mu X.\psi_1[X := \mu X.\psi_1]))) \in E$.*
  - *If $\psi = \nu X.\psi_1$ then $(v, (s, \psi_1(\nu X.\psi_1[X := \mu X.\psi_1]))) \in E$.*

Finally we have $\rho(s, \psi) = \begin{cases} 2\lfloor adepth(X)/2 \rfloor & \text{if } \psi = \nu X.\psi' \\ 2\lfloor adepth(X)/2 \rfloor + 1 & \text{if } \psi = \mu X.\psi' \\ 0 & \text{otherwise} \end{cases}$

Definition 3.6. *FTS2FPG(M,φ) converts FTS $M = (S, Act, trans, s_0, N, P, \gamma)$ and formula φ to FPG $(V, V_0, V_1, E, \rho, N, P, \gamma', v_0)$.*

We have (V, V_0, V_1, E, ρ) = LTS2PG($(S, Act, trans, s_0), \varphi$), $v_0 = (s_0, \varphi)$ and

$$\gamma'((s, \psi), (s', \psi')) = \begin{cases} \gamma(s, a, s') & \text{if } \psi = \langle a \rangle \psi' \text{ or } \psi = [a]\psi' \\ \top & \text{otherwise} \end{cases}$$

Definition 3.7. *FPG2VPG(G^F) converts FPG $G^F = (V^F, V_0^F, V_1^F, E^F, \rho^F, N, P, \gamma, v_0)$ to VPG $G = (V, V_0, V_1, E, \rho, \mathfrak{C}, \theta)$.*

Let P be defined as $\{p_0, p_1, \ldots, p_m\}$, we define $\mathfrak{C} = \{c_0, c_1, \ldots, c_m\}$.

We create vertices l_0 and l_1 and define $V_0 = V_0^F \cup \{l_0\}$, $V_1 = V_1^F \cup \{l_1\}$ and $V = V_0 \cup V_1$.

We construct E by first making $E = E^F$ and adding edges (l_0, l_0) and (l_1, l_1) to E. Simultaneously we construct θ by first making $\theta(e) = \bigcup\{c_i \in \mathfrak{C} | p_i \models \gamma(e)\}$ for every $e \in E^F$. Furthermore $\theta(l_0, l_0) = \theta(l_1, l_1) = \mathfrak{C}$.

Next, for every vertex $v \in V_\alpha$ with $\alpha = \{0,1\}$, we have $C = \mathfrak{C} \setminus \bigcup \{\theta(v,w) | (v,w) \in E\}$. If $C \neq \emptyset$ then we add (v, l_α) to E and make $\theta(v, l_\alpha) = C$. Finally we have

$$\rho(v) = \begin{cases} m_o & \text{if } v = l_0 \\ m_e & \text{if } v = l_1 \\ \rho^F(v) & \text{otherwise} \end{cases}$$

where m_o is the highest odd priority given by ρ^F or 1 if no odd priorities occur. And m_e is the highest even priority given by ρ^F or 0 if no even priorities occur.

Theorem 3.1. *Given:*

- *VPG $G = (V, V_0, V_1, E, \rho, \mathfrak{C}, \theta)$,*
- *$c \in \mathfrak{C}$,*
- *$\alpha \in \{0,1\}$*

it holds for winning sets W_α^c in G and W_α in $G_{|c}$ that $W_\alpha \subseteq W_\alpha^c$.

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$(M, \varphi)_{|f(p)}$ 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.

[3] J. Bradfield and I. Walukiewicz, *The mu-calculus and Model Checking*, pp. 871–919. Cham: Springer International Publishing, 2018.