Towards a Feature μ-Calculus Targeting SPL Verification

Maurice H. ter Beek, Erik P. de Vink & Tim A.C. Willemse

ISTI-CNR, Pisa, Italy & TU/e, Eindhoven, The Netherlands

FMSPLE 2016
ETAPS, Eindhoven, The Netherlands
Sunday 3 April 2016

Outline

1. Context: SPL model checking
2. Towards family-based model checking with mCRL2
   - mCRL2: language and toolset
   - The (modal) μ-calculus μL
   - FTS: feature transition systems
   - A feature μ-calculus μL_f over FTS
   - Main results of our paper
3. Conclusions and future work

Formal methods and tools in SPLE

Computer-aided analysis of feature models

- Traditionally: focus on modeling/analysing structural constraints
- But: software systems often embedded/distributed/safety-critical
- Important: model/analyze also behavior (e.g. quality assurance)

Goal: rigorously establish critical requirements of (software) systems

$\Rightarrow$ lift success stories from single product/system engineering to SPLE

Examples of behavioral SPL models with dedicated model checkers:

- Modal Transition Systems (MTS) with variability constraints

  Fantechi, Gnesi @ SPLC'08, Asirelli et al. @ iFM'10, SPLC'11, ter Beek et al. @ *JLAMP*, 2016

  Variability Model Checker VMC

  ter Beek et al. @ FM'12, SPLC'12, SPLat'14
- Featured Transition Systems (FTS)

  Classen et al. @ ICSE'10, *IEEE TSE*, 2013, *Sci. Comput. Program.*, 2014

  SNIP, ProVeLines, NuSMV extension

  Classen et al. @ ICSE'11, *Int. J. Softw. Tools Technol. Transf.*, 2012, Cordy et al. @ SPLC'13

Formal methods and tools in SPLE

Computer-aided analysis of feature models

- Traditionally: focus on modeling/analysing structural constraints
- But: software systems often embedded/distributed/safety-critical
- Important: model/analyze also behavior (e.g. quality assurance)

Goal: rigorously establish critical requirements of (software) systems

⇒ lift success stories from single product/system engineering to SPLE

Examples of behavioral SPL models with dedicated model checkers:

- Modal Transition Systems (MTS) with variability constraints

  Fantechi, Gnesi @ SPLC'08, Asirelli et al. @ iFM'10, SPLC'11, ter Beek et al. @ *JLAMP*, 2016

  Variability Model Checker VMC

  ter Beek et al. @ FM'12, SPLC'12, SPLat'14
- Featured Transition Systems (FTS)

  Classen et al. @ ICSE'10, *IEEE TSE*, 2013, *Sci. Comput. Program.*, 2014

  SNIP, ProVeLines, NuSMV extension

  Classen et al. @ ICSE'11, *Int. J. Softw. Tools Technol. Transf.*, 2012, Cordy et al. @ SPLC'13

Formal methods and tools in SPLE

Computer-aided analysis of feature models

- Traditionally: focus on modeling/analysing structural constraints
- But: software systems often embedded/distributed/safety-critical
- Important: model/analyze also behavior (e.g. quality assurance)

Goal: rigorously establish critical requirements of (software) systems

⇒ lift success stories from single product/system engineering to SPLE

Examples of behavioral SPL models with dedicated model checkers:

- Modal Transition Systems (MTS) with variability constraints

  Fantechi, Gnesi @ SPLC'08, Asirelli et al. @ iFM'10, SPLC'11, ter Beek et al. @ *JLAMP*, 2016

  Variability Model Checker VMC

  ter Beek et al. @ FM'12, SPLC'12, SPLat'14
- Featured Transition Systems (FTS)

  Classen et al. @ ICSE'10, *IEEE TSE*, 2013, *Sci. Comput. Program.*, 2014

  SNIP, ProVeLines, NuSMV extension

  Classen et al. @ ICSE'11, *Int. J. Softw. Tools Technol. Transf.*, 2012, Cordy et al. @ SPLC'13

Using mCRL2 for behavioral SPL analysis

Recommendations for Improving the Usability of Formal Methods for Product Lines: Atlee, Beidu, Day, Faghih & Shaker @ FormaliSE'13

"adopt and extend state-of-the-art analysis tools"

"examine[s] only valid product variants"

"visualize and (manually or automatically) analyze feature combinations corresponding to products of the product line"

"support (feature) modularity"

- We showed how to use the mCRL2 toolset for (product-based) SPL analysis in ter Beek & de Vink @ FormaliSE'14, SPLC'14
- We made modularization in a feature-oriented fashion concrete in ter Beek & de Vink @ FMSPLE'14
- We extended branching bisimulation for LTS to branching feature bisimulation for FTS in Belder, ter Beek & de Vink @ FMSPLE'15

Using mCRL2 for behavioral SPL analysis

Recommendations for Improving the Usability of Formal Methods for Product Lines: Atlee, Beidu, Day, Faghih & Shaker @ FormaliSE'13

"adopt and extend state-of-the-art analysis tools"

"examine[s] only valid product variants"

"visualize and (manually or automatically) analyze feature combinations corresponding to products of the product line"

"support (feature) modularity"

- We showed how to use the mCRL2 toolset for (product-based) SPL analysis in ter Beek & de Vink @ FormaliSE'14, SPLC'14
- We made modularization in a feature-oriented fashion concrete in ter Beek & de Vink @ FMSPLE'14
- We extended branching bisimulation for LTS to branching feature bisimulation for FTS in Belder, ter Beek & de Vink @ FMSPLE'15

Using mCRL2 for behavioral SPL analysis

Recommendations for Improving the Usability of Formal Methods for Product Lines: Atlee, Beidu, Day, Faghih & Shaker @ FormaliSE'13

"adopt and extend state-of-the-art analysis tools"

"examine[s] only valid product variants"

"visualize and (manually or automatically) analyze feature combinations corresponding to products of the product line"

"support (feature) modularity"

- We showed how to use the mCRL2 toolset for (product-based) SPL analysis in ter Beek & de Vink @ FormaliSE'14, SPLC'14
- We made modularization in a feature-oriented fashion concrete in ter Beek & de Vink @ FMSPLE'14
- We extended branching bisimulation for LTS to branching feature bisimulation for FTS in Belder, ter Beek & de Vink @ FMSPLE'15

Using mCRL2 for behavioral SPL analysis

Recommendations for Improving the Usability of Formal Methods for Product Lines:

Atlee, Beidu, Day, Faghih & Shaker @ FormaliSE'13

"adopt and extend state-of-the-art analysis tools"

"examine[s] only valid product variants"

"visualize and (manually or automatically) analyze feature combinations corresponding to products of the product line"

"support (feature) modularity"

- We showed how to use the mCRL2 toolset for (product-based) SPL analysis in ter Beek & de Vink @ FormaliSE'14, SPLC'14
- We made modularization in a feature-oriented fashion concrete in ter Beek & de Vink @ FMSPLE'14
- We extended branching bisimulation for LTS to branching feature bisimulation for FTS in Belder, ter Beek & de Vink @ FMSPLE'15

mCRL2: language and toolset

- Formal, process-algebraic specification of distributed and concurrent systems, associated industrial-strength toolset
- Exploration of 10^6 states/sec, state spaces up to 10^{12} states
- Built-in datatypes (Bool, Int, Real, Sets, Functions), user-defined abstract datatypes, parametrized actions
- Modal μ-calculus with data (incl. LTL, CTL, etc.)
- Visualization, behavioral reduction, model checking
- Highly optimized, actively maintained
- Intermediate artifacts user-accessible

`www.mcrl2.org`

The modal μ-calculus μL

set of actions $\mathcal{A}$ and set of variables $\mathcal{X}$

μ-calculus μL over $\mathcal{A}$ and $\mathcal{X}$, formula $\varphi \in \mu L$ given by

$$\begin{aligned}\varphi ::= {} & \bot \mid \top \mid \\ & \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \\ & \langle a \rangle \varphi \mid [a]\varphi \mid \\ & X \mid \mu X.\varphi \mid \nu X.\varphi\end{aligned}$$

duality $\langle a \rangle \varphi \equiv \neg[a]\neg\varphi$, positive normal form avoids negations

---

for $\mu X.\varphi$ and $\nu X.\varphi$, all free occurrences of X in φ are in the scope of an even number of negations (guarantees well-definedness fixpoint formulas)

Examples of μL-formulas

- $\langle a \rangle ([b] \bot \wedge \langle c \rangle \top)$

  "it is possible to execute action a, after which action b cannot be executed whereas action c can"

- $\mu X . (\langle a \rangle X \vee \langle b \rangle \top)$

  "there exists a finite repetition of executing action a, followed by an execution of action b"

- $\nu X . (\mu Y . [a] Y \wedge [b] X)$

  "action b is executed infinitely often on all infinite executions containing actions a and b"

Examples of μL-formulas

- $\langle a \rangle ([b] \bot \wedge \langle c \rangle \top)$

  "it is possible to execute action a, after which action b cannot be executed whereas action c can"

- $\mu X . (\langle a \rangle X \vee \langle b \rangle \top)$

  "there exists a finite repetition of executing action a, followed by an execution of action b"

- $\nu X . (\mu Y . [a] Y \wedge [b] X)$

  "action b is executed infinitely often on all infinite executions containing actions a and b"

μX: finite looping

Examples of μL-formulas

- $\langle a \rangle ([b]\bot \wedge \langle c \rangle \top)$

  "it is possible to execute action a, after which action b cannot be executed whereas action c can"

- $\mu X.(\langle a \rangle X \vee \langle b \rangle \top)$

  "there exists a finite repetition of executing action a, followed by an execution of action b"

- $\nu X.(\mu Y.[a]Y \wedge [b]X)$

  "action b is executed infinitely often on all infinite executions containing actions a and b"

μX: finite looping vs. νX: infinite looping

Formal semantics of μL

sets of states $U \in 2^S$, environments $\varepsilon \in Env = \mathcal{X} \to 2^S$
semantic function $[\![\cdot]\!]_L : \mu L \to Env \to 2^S$

$$
\begin{aligned}
[\![\bot]\!]_L(\varepsilon) &= \varnothing \\
[\![\top]\!]_L(\varepsilon) &= S \\
[\![\neg\varphi]\!]_L(\varepsilon) &= S \setminus [\![\varphi]\!]_L(\varepsilon) \\
[\![(\varphi \vee \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cup [\![\psi]\!]_L(\varepsilon) \\
[\![(\varphi \wedge \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cap [\![\psi]\!]_L(\varepsilon) \\
[\![\langle a\rangle\varphi]\!]_L(\varepsilon) &= \{\, s \mid \exists t\colon s \xrightarrow{a} t \wedge t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![[a]\varphi]\!]_L(\varepsilon) &= \{\, s \mid \forall t\colon s \xrightarrow{a} t \Rightarrow t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![X]\!]_L(\varepsilon) &= \varepsilon(X) \\
[\![\mu X.\varphi]\!]_L(\varepsilon) &= \mathit{lfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,) \\
[\![\nu X.\varphi]\!]_L(\varepsilon) &= \mathit{gfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,)
\end{aligned}
$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set U for X

Formal semantics of μL

sets of states $U \in 2^S$, environments $\varepsilon \in Env = \mathcal{X} \to 2^S$
semantic function $[\![\cdot]\!]_L : \mu L \to Env \to 2^S$

$$
\begin{aligned}
[\![\bot]\!]_L(\varepsilon) &= \emptyset \\
[\![\top]\!]_L(\varepsilon) &= S \\
[\![\neg\varphi]\!]_L(\varepsilon) &= S \setminus [\![\varphi]\!]_L(\varepsilon) \\
[\![(\varphi \vee \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cup [\![\psi]\!]_L(\varepsilon) \\
[\![(\varphi \wedge \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cap [\![\psi]\!]_L(\varepsilon) \\
[\![\langle a \rangle\varphi]\!]_L(\varepsilon) &= \{\, s \mid \exists t : s \xrightarrow{a} t \wedge t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![[a]\varphi]\!]_L(\varepsilon) &= \{\, s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![X]\!]_L(\varepsilon) &= \varepsilon(X) \\
[\![\mu X.\varphi]\!]_L(\varepsilon) &= lfp(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X]) \,) \\
[\![\nu X.\varphi]\!]_L(\varepsilon) &= gfp(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X]) \,)
\end{aligned}
$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set U for X

Formal semantics of μL

sets of states $U \in 2^S$, environments $\varepsilon \in Env = \mathcal{X} \to 2^S$
semantic function $[\![\cdot]\!]_L : \mu L \to Env \to 2^S$

$$[\![\bot]\!]_L(\varepsilon) = \emptyset$$
$$[\![\top]\!]_L(\varepsilon) = S$$
$$[\![\neg\varphi]\!]_L(\varepsilon) = S \setminus [\![\varphi]\!]_L(\varepsilon)$$
$$[\![(\varphi \vee \psi)]\!]_L(\varepsilon) = [\![\varphi]\!]_L(\varepsilon) \cup [\![\psi]\!]_L(\varepsilon)$$
$$[\![(\varphi \wedge \psi)]\!]_L(\varepsilon) = [\![\varphi]\!]_L(\varepsilon) \cap [\![\psi]\!]_L(\varepsilon)$$
$$[\![\langle a \rangle \varphi]\!]_L(\varepsilon) = \{ s \mid \exists t : s \xrightarrow{a} t \wedge t \in [\![\varphi]\!]_L(\varepsilon) \}$$
$$[\![[a] \varphi]\!]_L(\varepsilon) = \{ s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in [\![\varphi]\!]_L(\varepsilon) \}$$
$$[\![X]\!]_L(\varepsilon) = \varepsilon(X)$$
$$[\![\mu X . \varphi]\!]_L(\varepsilon) = lfp(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,)$$
$$[\![\nu X . \varphi]\!]_L(\varepsilon) = gfp(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,)$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set U for X

Formal semantics of μL

sets of states $U \in 2^S$, environments $\varepsilon \in \mathit{Env} = \mathcal{X} \to 2^S$
semantic function $[\![\cdot]\!]_L : \mu L \to \mathit{Env} \to 2^S$

$$
\begin{aligned}
[\![\bot]\!]_L(\varepsilon) &= \varnothing \\
[\![\top]\!]_L(\varepsilon) &= S \\
[\![\neg\varphi]\!]_L(\varepsilon) &= S \setminus [\![\varphi]\!]_L(\varepsilon) \\
[\![(\varphi \vee \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cup [\![\psi]\!]_L(\varepsilon) \\
[\![(\varphi \wedge \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cap [\![\psi]\!]_L(\varepsilon) \\
[\![\langle a\rangle\varphi]\!]_L(\varepsilon) &= \{\, s \mid \exists t \colon s \xrightarrow{a} t \wedge t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![[a]\varphi]\!]_L(\varepsilon) &= \{\, s \mid \forall t \colon s \xrightarrow{a} t \Rightarrow t \in [\![\varphi]\!]_L(\varepsilon) \,\} \\
[\![X]\!]_L(\varepsilon) &= \varepsilon(X) \\
[\![\mu X.\varphi]\!]_L(\varepsilon) &= \mathit{lfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X]) \,) \\
[\![\nu X.\varphi]\!]_L(\varepsilon) &= \mathit{gfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X]) \,)
\end{aligned}
$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set U for X

Formal semantics of μL

sets of states $U \in 2^S$, environments $\varepsilon \in Env = \mathcal{X} \to 2^S$
semantic function $[\![\cdot]\!]_L : \mu L \to Env \to 2^S$

$$
\begin{aligned}
[\![\bot]\!]_L(\varepsilon) &= \emptyset \\
[\![\top]\!]_L(\varepsilon) &= S \\
[\![\neg\varphi]\!]_L(\varepsilon) &= S \setminus [\![\varphi]\!]_L(\varepsilon) \\
[\![(\varphi \vee \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cup [\![\psi]\!]_L(\varepsilon) \\
[\![(\varphi \wedge \psi)]\!]_L(\varepsilon) &= [\![\varphi]\!]_L(\varepsilon) \cap [\![\psi]\!]_L(\varepsilon) \\
[\![\langle a\rangle\varphi]\!]_L(\varepsilon) &= \{\, s \mid \exists t\colon s \xrightarrow{a} t \wedge t \in [\![\varphi]\!]_L(\varepsilon)\,\} \\
[\![[a]\varphi]\!]_L(\varepsilon) &= \{\, s \mid \forall t\colon s \xrightarrow{a} t \Rightarrow t \in [\![\varphi]\!]_L(\varepsilon)\,\} \\
[\![X]\!]_L(\varepsilon) &= \varepsilon(X) \\
[\![\mu X.\varphi]\!]_L(\varepsilon) &= \mathit{lfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,) \\
[\![\nu X.\varphi]\!]_L(\varepsilon) &= \mathit{gfp}(\, U \mapsto [\![\varphi]\!]_L(\varepsilon[U/X])\,)
\end{aligned}
$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set U for X

FTS: feature transition systems

FTS $F = (S, \theta, s_*)$ over actions $\mathcal{A}$ and features $\mathcal{F}$

- S a finite set of states
- $\theta : S \times \mathcal{A} \times S \to \mathbb{B}[\mathcal{F}]$ the transition constraint function
- $s_* \in S$ the initial state

LTS $L = (S, \to, s_*)$ over actions $\mathcal{A}$

- S a finite set of states
- $\to \subseteq S \times \mathcal{A} \times S$ the transition relation
- $s_* \in S$ the initial state

FTS: feature transition systems

FTS $F = (S, \theta, s_*)$ over actions $\mathcal{A}$ and features $\mathcal{F}$

- S a finite set of states
- $\theta : S \times \mathcal{A} \times S \to \mathbb{B}[\mathcal{F}]$ the transition constraint function
- $s_* \in S$ the initial state

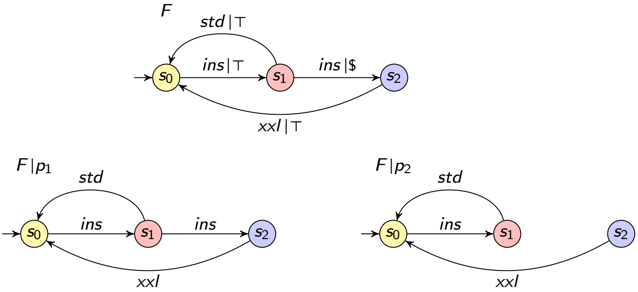
LTS $F|p = (S, \to_{F|p}, s_*)$ projection of F with respect to product p

$$\to_{F|p} \subseteq S \times \mathcal{A} \times S \text{ such that } s \xrightarrow{a}_{F|p} t \text{ iff } p \models \theta(s, a, t)$$

$\mathcal{P} \subseteq \mathbf{2}^{\mathcal{F}}$ set of products p, ...

FTS of example SPL

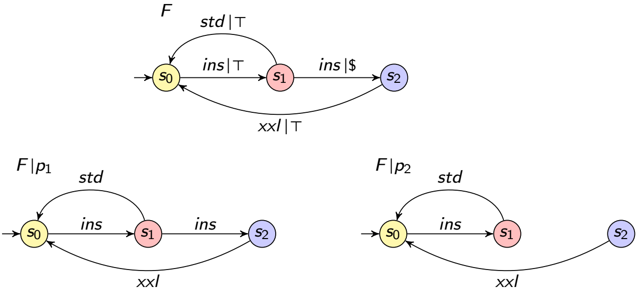
Product line of (four) coffee machines with independent features $\{\$, €\}$



Products with feature \$ can obtain an *xxl* coffee upon coin insertion, but products without cannot: how to express this?

FTS of example SPL

Product line of (four) coffee machines with independent features $\{\$, €\}$



Products with feature $\$$ can obtain an *xxl* coffee upon coin insertion, but products without cannot: how to express this?

A feature μ-calculus μL_f over FTS

$\alpha_p : \mathcal{F} \to \mathbb{B}$ with $\alpha_p(f) = \textbf{true}$ iff $f \in p$

notation $p \in \chi$ for $\alpha_p \models \chi$

Feature μ-calculus μL_f over $\mathcal{A}$, $\mathcal{F}$ and $\mathcal{X}$, formula $\varphi_f \in \mu L_f$ given by

$$\begin{aligned} \varphi_f ::= {} & \bot \mid \top \mid \\ & \neg\varphi_f \mid \varphi_f \vee \psi_f \mid \varphi_f \wedge \psi_f \mid \\ & \langle\langle a|\chi \rangle\rangle \varphi_f \mid [a|\chi]\varphi_f \mid \\ & X \mid \mu X.\varphi_f \mid \nu X.\varphi_f \end{aligned}$$

for $\mu X.\varphi_f$ and $\nu X.\varphi_f$ an even number of negations as before

---

μL_f, with an FTS semantics over sets of products, is $\mu L'_f$ in the paper, where a μL_f is defined with an FTS semantics over individual products

An FTS semantics of μL_f (1/2)

state-family pairs $(s, P) \in sPSet = 2^{S \times 2^{\mathcal{P}}}$
state-family environments $\zeta \in sPEnv = \mathcal{X} \to sPSet$

semantic function $[\![\cdot]\!]_F : \mu L_f \to sPEnv \to sPSet$

$$
\begin{aligned}
[\![\bot]\!]_F(\zeta) &= \emptyset \\
[\![\top]\!]_F(\zeta) &= S \times 2^{\mathcal{P}} \\
[\![\neg\varphi_f]\!]_F(\zeta) &= (S \times 2^{\mathcal{P}}) \setminus [\![\varphi_f]\!]_F(\zeta) \\
[\![(\varphi_f \vee \psi_f)]\!]_F(\zeta) &= [\![\varphi_f]\!]_F(\zeta) \cup [\![\psi_f]\!]_F(\zeta) \\
[\![(\varphi_f \wedge \psi_f)]\!]_F(\zeta) &= [\![\varphi_f]\!]_F(\zeta) \cap [\![\psi_f]\!]_F(\zeta) \\
[\![\langle\!\langle a|\chi \rangle\!\rangle \varphi_f]\!]_F(\zeta) &= \ldots \\
[\![[a|\chi] \varphi_f]\!]_F(\zeta) &= \ldots \\
[\![X]\!]_F(\zeta) &= \zeta(X) \\
[\![\mu X.\varphi_f]\!]_F(\zeta) &= lfp(\, W \mapsto [\![\varphi_f]\!]_F(\zeta[W/X])\,) \\
[\![\nu X.\varphi_f]\!]_F(\zeta) &= gfp(\, W \mapsto [\![\varphi_f]\!]_F(\zeta[W/X])\,)
\end{aligned}
$$

An FTS semantics of μL_f (2/2)

semantic function $[\![\cdot]\!]_F : \mu L_f \to sPEnv \to sPSet$

$$[\![\langle\!\langle a|\chi\rangle\!\rangle\varphi_f]\!]_F(\zeta) =$$

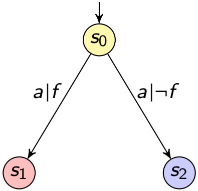
$$\{\,(s,P) \mid \exists\gamma, t\colon s \xrightarrow{a|\gamma}_F t \wedge P \subseteq \chi \cap \gamma \wedge (t, P \cap \chi \cap \gamma) \in [\![\varphi_f]\!]_F(\zeta)\,\}$$

$$[\![[a|\chi]\varphi_f]\!]_F(\zeta) =$$

$$\{\,(s,P) \mid \forall\gamma, t\colon s \xrightarrow{a|\gamma}_F t \wedge P \cap \chi \cap \gamma \neq \emptyset \Rightarrow (t, P \cap \chi \cap \gamma) \in [\![\varphi_f]\!]_F(\zeta)\,\}$$

Example μL_f formula: duality lost

$s,P \models_F \varphi_f$ iff $(s,P) \in [\![\varphi_f]\!]_F(\zeta_0)$



Products $p_1 = \{f, g\}$ and $p_2 = \{g\}$

Clearly: $\{f, g\} \models_{F|p_1} \langle a \rangle \top$

$\{g\} \models_{F|p_2} \langle a \rangle \top$

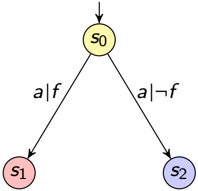
but... $\{p_1, p_2\} \not\models_F \langle\!\langle a|\top \rangle\!\rangle \top$

Hence, since neither $\{p_1, p_2\} \models_F \langle\!\langle a|\top \rangle\!\rangle \top$ nor $\{p_1, p_2\} \models_F [a|\top]\bot$,

$\langle\!\langle a|\chi \rangle\!\rangle$ and $[a|\chi]$ are not each other's dual

Example μL_f formula: duality lost

$s,P \models_F \varphi_f$ iff $(s,P) \in [\![\varphi_f]\!]_F(\zeta_0)$



Products $p_1 = \{f, g\}$ and $p_2 = \{g\}$

Clearly: $\{f, g\} \models_{F|p_1} \langle a \rangle \top$

$\{g\} \models_{F|p_2} \langle a \rangle \top$

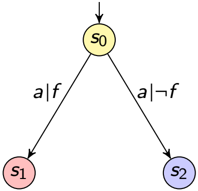
but... $\{p_1, p_2\} \not\models_F \langle\!\langle a|\top \rangle\!\rangle \top$

Hence, since neither $\{p_1, p_2\} \models_F \langle\!\langle a|\top \rangle\!\rangle \top$ nor $\{p_1, p_2\} \models_F [a|\top]\bot$,

$\langle\!\langle a|\chi \rangle\!\rangle$ and $[a|\chi]$ are not each other's dual

Example μL_f formula: duality lost

$s,P \models_F \varphi_f$ iff $(s,P) \in [\![\varphi_f]\!]_F(\zeta_0)$



Products $p_1 = \{f, g\}$ and $p_2 = \{g\}$

Clearly: $\{f, g\} \models_{F|p_1} \langle a \rangle \top$

$\{g\} \models_{F|p_2} \langle a \rangle \top$

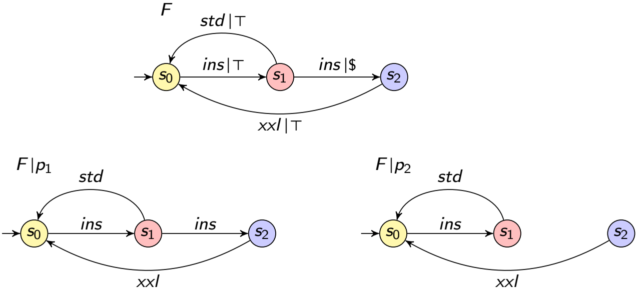
but... $\{p_1, p_2\} \not\models_F \langle\!\langle a|\top \rangle\!\rangle \top$

Hence, since neither $\{p_1, p_2\} \models_F \langle\!\langle a|\top \rangle\!\rangle \top$ nor $\{p_1, p_2\} \models_F [a|\top]\bot$,

$\langle\!\langle a|\chi \rangle\!\rangle$ and $[a|\chi]$ are not each other's dual

Example multi-feature μL_f formula

Product line of (four) coffee machines with independent features $\{\$, €\}$



Products with feature $\$$ can obtain an *xxl* coffee upon coin insertion, but products without cannot:

$$[\,\mathtt{true}^* \mid \top\,]\,(\,(\,[\,ins \mid \$\,]\,\langle\,\mathtt{true}^*.\,xxl \mid \top\,\rangle\,\top\,) \wedge [\,xxl \mid \neg\$\,]\,\bot\,)$$

Towards family-based model checking

Model checking a μL_f-formula over an FTS for an individual product reduces to model checking a μL-formula over the corresponding LTS

translation function $sm : \mu L_f \times \mathcal{P} \rightarrow \mu L$

$$
\begin{aligned}
sm(\bot, p) &= \bot \\
sm(\top, p) &= \top \\
sm(\varphi_f \vee \psi_f, p) &= sm(\varphi_f) \vee sm(\psi_f) \\
sm(\varphi_f \wedge \psi_f, p) &= sm(\varphi_f) \wedge sm(\psi_f) \\
sm(\langle\!\langle a|\chi \rangle\!\rangle \varphi_f, p) &= \textbf{if } p \in \chi \textbf{ then } \langle a \rangle sm(\varphi_f, p) \textbf{ else } \bot \textbf{ end} \\
sm([a|\chi]\,\varphi_f, p) &= \textbf{if } p \in \chi \textbf{ then } [a]\, sm(\varphi_f, p) \textbf{ else } \top \textbf{ end} \\
sm(X, p) &= X \\
sm(\mu X . \varphi_f, p) &= \mu X . sm(\varphi_f, p) \\
sm(\nu X . \varphi_f, p) &= \nu X . sm(\varphi_f, p)
\end{aligned}
$$

Main results of our paper

Given an FTS F and a set of products $\mathcal{P}$

Theorem 1 $s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f)$

closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

Theorem 2 $s, P \models_F \varphi_f \iff \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$ without $\langle\langle a|\chi \rangle\rangle$, $s \in S$, family $P \subseteq \mathcal{P}$

Theorem 3 $s, P \models_F \varphi_f \implies \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$, $s \in S$, family $P \subseteq \mathcal{P}$

Main results of our paper

Given an FTS F and a set of products $\mathcal{P}$

Theorem 1 $\quad s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f)$

closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

Theorem 2 $\quad s, P \models_F \varphi_f \iff \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$ without $\langle\!\langle a|\chi \rangle\!\rangle$, $s \in S$, family $P \subseteq \mathcal{P}$

Theorem 3 $\quad s, P \models_F \varphi_f \implies \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$, $s \in S$, family $P \subseteq \mathcal{P}$

Main results of our paper

Given an FTS F and a set of products $\mathcal{P}$

Theorem 1 $\quad s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f)$

closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

Theorem 2 $\quad s, P \models_F \varphi_f \iff \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$ without $\langle\!\langle a|\chi \rangle\!\rangle$, $s \in S$, family $P \subseteq \mathcal{P}$

Theorem 3 $\quad s, P \models_F \varphi_f \implies \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$, $s \in S$, family $P \subseteq \mathcal{P}$

Main results of our paper

Given an FTS F and a set of products $\mathcal{P}$

Theorem 1 $\quad s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f)$

closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

Theorem 2 $\quad s, P \models_F \varphi_f \iff \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$ without $\langle\!\langle a|\chi \rangle\!\rangle$, $s \in S$, family $P \subseteq \mathcal{P}$

Theorem 3 $\quad s, P \models_F \varphi_f \implies \forall p \in P \colon s \models_{F|p} sm(\varphi_f)$

closed, negation-free $\varphi_f \in \mu L_f$, $s \in S$, family $P \subseteq \mathcal{P}$

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)

Conclusions and future work

Introduced and compared two μ-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but μL_f is more expressive

Embedding in μ-calculus with data allows family-based model checking multi-feature properties of SPL models with the mCRL2 toolset as is

Future work (extending/applying Theorems 1–3):

- what about the preservation of the invalidity of a formula for a product family by the family's individual products?
- under which conditions can the equivalence of Theorem 2 be obtained for a larger set of feature μ-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)