Towards a Feature $\mu$-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
1. Context: SPL model checking

2. Towards family-based model checking with mCRL2
   - mCRL2: language and toolset
   - The (modal) $\mu$-calculus $\mu L$
   - FTS: feature transition systems
   - A feature $\mu$-calculus $\mu L_f$ over FTS
   - Main results of our paper

3. Conclusions and future work
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)
  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)
  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)
  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:

“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:

“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:

“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 $\mu$-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 $\mu$-calculus $\mu L$

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

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

$$
\varphi ::= = \bot \mid \top \mid \\
\neg \varphi \mid \varphi \lor \psi \mid \varphi \land \psi \mid \\
\langle a \rangle \varphi \mid [a] \varphi \mid \
X \mid \mu X.\varphi \mid \nu X.\varphi
$$

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 $\varphi$ are in the scope of an even number of negations (guarantees well-definedness fixpoint formulas)
Examples of $\mu L$-formulas

- $\langle a \rangle ([b] \perp \land \langle c \rangle \top)$
  
  “it is possible to execute action $a$, after which action $b$ cannot be executed whereas action $c$ can”

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

  “there exists a finite repetition of executing action $a$, followed by an execution of action $b$”

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

  “action $b$ is executed infinitely often on all infinite executions containing actions $a$ and $b$”
Examples of $\mu L$-formulas

- $\langle a \rangle (\lbrack b \rbrack \bot \land \langle c \rangle \top)$
  
  “it is possible to execute action $a$, after which action $b$ cannot be executed whereas action $c$ can”

- $\mu X. (\langle a \rangle X \lor \langle b \rangle \top)$
  
  “there exists a finite repetition of executing action $a$, followed by an execution of action $b$”

- $\nu X. (\mu Y. [a] Y \land [b] X)$
  
  “action $b$ is executed infinitely often on all infinite executions containing actions $a$ and $b$”

$\mu X$: finite looping
Examples of $\mu L$-formulas

- $\langle a \rangle ( [b] \bot \land \langle c \rangle \top )$
  
  “it is possible to execute action $a$, after which action $b$ cannot be executed whereas action $c$ can”

- $\mu X. (\langle a \rangle X \lor \langle b \rangle \top )$
  
  “there exists a finite repetition of executing action $a$, followed by an execution of action $b$”

- $\nu X. (\mu Y. [a] Y \land [b] X )$
  
  “action $b$ is executed infinitely often on all infinite executions containing actions $a$ and $b$”

$\mu X$: finite looping vs. $\nu X$: infinite looping
Formal semantics of $\mu L$

sets of states $U \in 2^S$, environments $\varepsilon \in \text{Env} = X \rightarrow 2^S$

semantic function $\llbracket \cdot \rrbracket_L : \mu L \rightarrow \text{Env} \rightarrow 2^S$

\[
\begin{align*}
\llbracket \bot \rrbracket_L(\varepsilon) &= \emptyset \\
\llbracket \top \rrbracket_L(\varepsilon) &= S \\
\llbracket \neg \varphi \rrbracket_L(\varepsilon) &= S \setminus \llbracket \varphi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \lor \psi) \rrbracket_L(\varepsilon) &= \llbracket \varphi \rrbracket_L(\varepsilon) \cup \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \land \psi) \rrbracket_L(\varepsilon) &= \llbracket \varphi \rrbracket_L(\varepsilon) \cap \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket \langle a \rangle \varphi \rrbracket_L(\varepsilon) &= \{ s \mid \exists t : s \xrightarrow{a} t \land t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket [a] \varphi \rrbracket_L(\varepsilon) &= \{ s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket X \rrbracket_L(\varepsilon) &= \varepsilon(X) \\
\llbracket \mu X. \varphi \rrbracket_L(\varepsilon) &= \text{lfp}( U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X])) \\
\llbracket \nu X. \varphi \rrbracket_L(\varepsilon) &= \text{gfp}( U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X]))
\end{align*}
\]

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set $U$ for $X$
Formal semantics of $\mu L$

sets of states $U \in 2^S$, environments $\varepsilon \in Env = X \rightarrow 2^S$

semantic function $\llbracket \cdot \rrbracket_L : \mu L \rightarrow Env \rightarrow 2^S$

\[
\begin{align*}
\llbracket \bot \rrbracket_L(\varepsilon) &= \emptyset \\
\llbracket \top \rrbracket_L(\varepsilon) &= S \\
\llbracket \neg \varphi \rrbracket_L(\varepsilon) &= S \setminus \llbracket \varphi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \lor \psi) \rrbracket_L(\varepsilon) &= \llbracket \varphi \rrbracket_L(\varepsilon) \cup \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \land \psi) \rrbracket_L(\varepsilon) &= \llbracket \varphi \rrbracket_L(\varepsilon) \cap \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket (a) \varphi \rrbracket_L(\varepsilon) &= \{ s \mid \exists t : s \xrightarrow{a} t \land t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket [a] \varphi \rrbracket_L(\varepsilon) &= \{ s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket X \rrbracket_L(\varepsilon) &= \varepsilon(X) \\
\llbracket \mu X. \varphi \rrbracket_L(\varepsilon) &= \text{lfp}(U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X])) \\
\llbracket \nu X. \varphi \rrbracket_L(\varepsilon) &= \text{gfp}(U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X]))
\end{align*}
\]

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set $U$ for $X$
Formal semantics of $\mu L$

sets of states $U \in 2^S$, environments $\varepsilon \in Env = X \to 2^S$

semantic function $\left[ \cdot \right]_L : \mu L \to Env \to 2^S$

$$\left[ \bot \right]_L(\varepsilon) = \varnothing$$

$$\left[ \top \right]_L(\varepsilon) = S$$

$$\left[ \neg \varphi \right]_L(\varepsilon) = S \setminus \left[ \varphi \right]_L(\varepsilon)$$

$$\left[ (\varphi \lor \psi) \right]_L(\varepsilon) = \left[ \varphi \right]_L(\varepsilon) \cup \left[ \psi \right]_L(\varepsilon)$$

$$\left[ (\varphi \land \psi) \right]_L(\varepsilon) = \left[ \varphi \right]_L(\varepsilon) \cap \left[ \psi \right]_L(\varepsilon)$$

$$\left[ \langle a \rangle \varphi \right]_L(\varepsilon) = \{ s \mid \exists t : s \xrightarrow{a} t \land t \in \left[ \varphi \right]_L(\varepsilon) \}$$

$$\left[ [a] \varphi \right]_L(\varepsilon) = \{ s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in \left[ \varphi \right]_L(\varepsilon) \}$$

$$\left[ X \right]_L(\varepsilon) = \varepsilon(X)$$

$$\left[ \mu X . \varphi \right]_L(\varepsilon) = \text{lfp}( U \mapsto \left[ \varphi \right]_L(\varepsilon[U/X]))$$

$$\left[ \nu X . \varphi \right]_L(\varepsilon) = \text{gfp}( U \mapsto \left[ \varphi \right]_L(\varepsilon[U/X]))$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set $U$ for $X$
Formal semantics of $\mu L$

sets of states $U \in 2^S$, environments $\varepsilon \in Env = X \rightarrow 2^S$

semantic function $[\cdot]_L : \mu L \rightarrow Env \rightarrow 2^S$

$$[\bot]_L(\varepsilon) = \emptyset$$
$$[\top]_L(\varepsilon) = S$$
$$[\neg \phi]_L(\varepsilon) = S \setminus [\phi]_L(\varepsilon)$$
$$[(\phi \lor \psi)]_L(\varepsilon) = [\phi]_L(\varepsilon) \cup [\psi]_L(\varepsilon)$$
$$[(\phi \land \psi)]_L(\varepsilon) = [\phi]_L(\varepsilon) \cap [\psi]_L(\varepsilon)$$

$$[[\langle a \rangle \phi]]_L(\varepsilon) = \{ s | \exists t: s \xrightarrow{a} t \land t \in [\phi]_L(\varepsilon) \}$$

$$[[a] \phi]_L(\varepsilon) = \{ s | \forall t: s \xrightarrow{a} t \Rightarrow t \in [\phi]_L(\varepsilon) \}$$

$$[[X]]_L(\varepsilon) = \varepsilon(X)$$

$$[[\mu X. \phi]]_L(\varepsilon) = \text{lfp}( U \mapsto [\phi]_L(\varepsilon[U/X]))$$

$$[[\nu X. \phi]]_L(\varepsilon) = \text{gfp}( U \mapsto [\phi]_L(\varepsilon[U/X]))$$

variant environment $\varepsilon[U/X]$: $\varepsilon(Y)$ for $Y \neq X$, the set $U$ for $X$
Formal semantics of $\mu L$

sets of states $U \in 2^S$, environments $\varepsilon \in \text{Env} = X \rightarrow 2^S$

semantic function $\llbracket \cdot \rrbracket_L : \mu L \rightarrow \text{Env} \rightarrow 2^S$

\[
\llbracket \bot \rrbracket_L(\varepsilon) = \emptyset \\
\llbracket \top \rrbracket_L(\varepsilon) = S \\
\llbracket \neg \varphi \rrbracket_L(\varepsilon) = S \setminus \llbracket \varphi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \lor \psi) \rrbracket_L(\varepsilon) = \llbracket \varphi \rrbracket_L(\varepsilon) \cup \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket (\varphi \land \psi) \rrbracket_L(\varepsilon) = \llbracket \varphi \rrbracket_L(\varepsilon) \cap \llbracket \psi \rrbracket_L(\varepsilon) \\
\llbracket \langle a \rangle \varphi \rrbracket_L(\varepsilon) = \{ s \mid \exists t : s \xrightarrow{a} t \land t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket [ a ] \varphi \rrbracket_L(\varepsilon) = \{ s \mid \forall t : s \xrightarrow{a} t \Rightarrow t \in \llbracket \varphi \rrbracket_L(\varepsilon) \} \\
\llbracket X \rrbracket_L(\varepsilon) = \varepsilon(X) \\
\llbracket \mu X . \varphi \rrbracket_L(\varepsilon) = \text{lfp}( U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X])) \\
\llbracket \nu X . \varphi \rrbracket_L(\varepsilon) = \text{gfp}( U \mapsto \llbracket \varphi \rrbracket_L(\varepsilon[U/X]))
\]

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
- $\theta : S \times \mathcal{A} \times S \rightarrow B[\mathcal{F}]$ the transition constraint function
- $s_* \in S$ the initial state

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

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

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

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

$\mathcal{P} \subseteq 2^\mathcal{F}$ set of products $p, \ldots$
FTS of example SPL

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

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 $\mu$-calculus $\mu L_f$ over FTS

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

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

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

$$\varphi_f ::= \bot \mid \top \mid$$
$$\neg \varphi_f \mid \varphi_f \lor \psi_f \mid \varphi_f \land \psi_f \mid$$
$$\langle a \mid \chi \rangle \varphi_f \mid [a \mid \chi] \varphi_f \mid$$
$$\mathcal{X} \mid \mu \mathcal{X}.\varphi_f \mid \nu \mathcal{X}.\varphi_f$$

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

$\mu L_f$, with an FTS semantics over sets of products, is $\mu L'_f$ in the paper, where a $\mu L_f$ is defined with an FTS semantics over individual products.
An FTS semantics of $\mu L_f$ (1/2)

state-family pairs $(s, P) \in sPSet = 2^{S \times 2^P}$

state-family environments $\zeta \in sPEnv = X \rightarrow sPSet$

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

\[
\begin{align*}
[\bot]_F(\zeta) &= \emptyset \\
[\top]_F(\zeta) &= S \times 2^P \\
[\neg \varphi_f]_F(\zeta) &= (S \times 2^P) \setminus [\varphi_f]_F(\zeta) \\
[(\varphi_f \lor \psi_f)]_F(\zeta) &= [\varphi_f]_F(\zeta) \cup [\psi_f]_F(\zeta) \\
[(\varphi_f \land \psi_f)]_F(\zeta) &= [\varphi_f]_F(\zeta) \cap [\psi_f]_F(\zeta) \\
[\langle a | X \rangle \varphi_f]_F(\zeta) &= \ldots \\
[[a | X] \varphi_f]_F(\zeta) &= \ldots \\
[X]_F(\zeta) &= \zeta(X) \\
[\mu X. \varphi_f]_F(\zeta) &= \text{lfp}(W \mapsto [\varphi_f]_F(\zeta[W/X])) \\
[\nu X. \varphi_f]_F(\zeta) &= \text{gfp}(W \mapsto [\varphi_f]_F(\zeta[W/X]))
\end{align*}
\]
An FTS semantics of $\mu L_f$ (2/2)

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

$$
[\llangle a|\chi \rrangle \varphi_f]_F(\zeta) = \\
\{ (s, P) \mid \exists \gamma, t : s \xrightarrow{a|\gamma}_F t \land P \subseteq \chi \cap \gamma \land \\
(t, P \cap \chi \cap \gamma) \in [\varphi_f]_F(\zeta) \}
$$

$$
[[a|\chi] \varphi_f]_F(\zeta) = \\
\{ (s, P) \mid \forall \gamma, t : s \xrightarrow{a|\gamma}_F t \land P \cap \chi \cap \gamma \neq \emptyset \Rightarrow \\
(t, P \cap \chi \cap \gamma) \in [\varphi_f]_F(\zeta) \}
$$
Example $\mu L_f$ formula: duality lost

$s, P \models_F \varphi_f$ iff $(s, P) \in \llbracket \varphi_f \rrbracket_F(\zeta_0)$

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

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

$\{g\} \models_{F|p_2} \langle a \rangle_T$

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

Hence, since neither $\{p_1, p_2\} \models_F \langle a|\top \rangle_T$ nor $\{p_1, p_2\} \models_F [a|\top] \perp$,

$\langle a|\chi \rangle$ and $[a|\chi]$ are not each other's dual
Example $\mu L_f$ formula: duality lost

$s, P \models_F \varphi_f$ iff $(s, P) \in \llbracket \varphi_f \rrbracket_F(\zeta_0)$

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

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

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

but... $\{p_1, p_2\} \not\models_F \langle a \mid \chi \rangle T$

Hence, since neither $\{p_1, p_2\} \models_F \langle a \mid \chi \rangle T$ nor $\{p_1, p_2\} \models_F [a \mid \chi] \perp$, $\langle a \mid \chi \rangle$ and $[a \mid \chi]$ are not each other’s dual
Example $\mu L_f$ formula: duality lost

$$s, P \models_F \varphi_f \iff (s, P) \in \llbracket \varphi_f \rrbracket_F(\zeta_0)$$

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

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

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

but... $\{p_1, p_2\} \not\models_{F} \langle a|T \rangle T$

Hence, since neither $\{p_1, p_2\} \models_{F} \langle a|T \rangle T$ nor $\{p_1, p_2\} \models_{F} [a|T] \perp$, $\langle a|\chi \rangle$ and $[a|\chi]$ are not each other’s dual
Example multi-feature $\mu L_f$ 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:

$$[\text{true}^* | T] (([\text{ins} | \$$]<true*.xxl | T> T) \land [\text{xxl} | \neg\$$] \perp)$$
Towards family-based model checking

Model checking a $\mu L_f$-formula over an FTS for an individual product reduces to model checking a $\mu L$-formula over the corresponding LTS translation function $sm : \mu L_f \times \mathcal{P} \rightarrow \mu L$

$$sm(\bot, p) = \bot$$
$$sm(\top, p) = \top$$
$$sm(\varphi_f \lor \psi_f, p) = sm(\varphi_f) \lor sm(\psi_f)$$
$$sm(\varphi_f \land \psi_f, p) = sm(\varphi_f) \land sm(\psi_f)$$
$$sm(\langle a | \chi \rangle \varphi_f, p) = \text{if } p \in \chi \text{ then } \langle a \rangle sm(\varphi_f, p) \text{ else } \bot \text{ end}$$
$$sm([a | \chi] \varphi_f, p) = \text{if } p \in \chi \text{ then } [a] sm(\varphi_f, p) \text{ else } \top \text{ end}$$
$$sm(X, p) = X$$
$$sm(\mu X . \varphi_f, p) = \mu X . sm(\varphi_f, p)$$
$$sm(\nu X . \varphi_f, p) = \nu X . sm(\varphi_f, p)$$
Main results of our paper

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

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

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

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

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

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

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**  \[ s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f) \]
closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

**Theorem 2**  \[ s, P \models_F \varphi_f \iff \forall p \in P: s \models_{F|p} sm(\varphi_f) \]
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**  \[ s, P \models_F \varphi_f \implies \forall p \in P: s \models_{F|p} sm(\varphi_f) \]
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** $s, \{p\} \models_F \varphi_f \iff s \models_{F|p} sm(\varphi_f)$
closed $\varphi_f \in \mu L_f$, $s \in S$, product $p \in \mathcal{P}$

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

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

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

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**  
$s, P \models_F \varphi_f \implies \forall p \in P: s \models_{F|_p} sm(\varphi_f)$

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 $\mu$-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive.

Embedding in $\mu$-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 $\mu$-calculus formulas?
- perform and evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)
Conclusions and future work

Introduced and compared two $\mu$-calculus variants with FTS semantics.
Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive.
Embedding in $\mu$-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 $\mu$-calculus formulas?
- Perform and evaluate family-based model checking with mCRL2
  on an exemplary SPL model (minepump case study).
Conclusions and future work

Introduced and compared two $\mu$-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive

Embedding in $\mu$-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 $\mu$-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 $\mu$-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive

Embedding in $\mu$-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 $\mu$-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 $\mu$-calculus variants with FTS semantics.
Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive.
Embedding in $\mu$-calculus with data allows family-based model checking of 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 $\mu$-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 $\mu$-calculus variants with FTS semantics. Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive. Embedding in $\mu$-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 $\mu$-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 $\mu$-calculus variants with FTS semantics

Resembles fLTL and fCTL by Classen et al., but $\mu L_f$ is more expressive

Embedding in $\mu$-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 $\mu$-calculus formulas?
- perform + evaluate family-based model checking with mCRL2 on an exemplary SPL model (minepump case study)