Process Algebra and Stochastic Timed Activities 2003

# Towards compositional Model Checking of Continuous Time Markov Chains

Paolo Ballarini and Jane Hillston

12th June 2003 - Edinburgh

# OUTLINE

1. Aim
2. A brief survey on Model Checking
3. Compositionality and Model Checking
4. A compositional framework for CTMC: the *Boucherie* product process
5. Compositional CSL Model Checking over *Boucherie* processes
6. Conclusion and future work

# 1 - INTRODUCTION

- **<u>AIM OF THIS WORK</u>:** to investigate the possibility for a compositional approach to Continuous Time Markov Chain (CTMC) model checking.

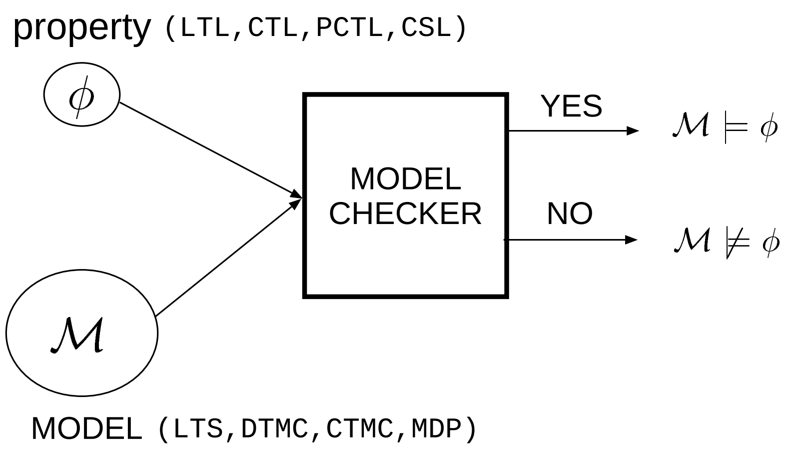
- we take in consideration a specific compositional framework for CTMCs, which is the *Boucherie* product-form process.

- we show that a (de)compositional model checking approach is possible for CSL model checking under certain restrictions (i.e. no nested *path formulae*).

## 2 - A BRIEF SURVEY ON MODEL CHECKING

· Model checking is a methodology for testing a system's model against properties expressed in terms of some temporal logic formulae.

· A MODEL CHECKER takes a model $\mathcal{M}$ and a formula $\phi$ and returns either $YES$ if $\phi$ is satisfied in $\mathcal{M}$ or $NO$ if it is not.



# MAIN TYPES OF TEMPORAL LOGICS

- **Linear Time Logic (LTL)**

- **Computational Tree Logic (CTL)**

- **Probabilistic Computational Tree Logic (PCTL)**

- **Continuous Stochastic Logic (CSL)**

# LTL MODEL CHECKING

- **Linear Time Logic [Pnueli et al.]:** a language to reason about the *future* considering the time as extending in a linear fashion.

- models are Labelled Transition Systems (LTS): $\mathcal{M} = \langle S, R, L \rangle$

$$\begin{array}{lcl} S & : & \text{set of possible } \textit{states} \\ R & \subseteq & S^2 \ \textit{linear} \text{ binary relation between } \textit{states} \\ L & : & S \rightarrow 2^{AP} \text{ is the } \textit{labelling function} \end{array}$$

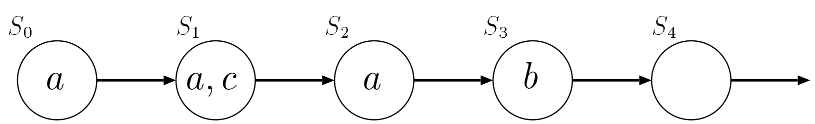
$AP = \{a, b, c \ldots\}$ is the atomic proposition set.

- two basic temporal connectives: neXt ($\mathcal{X}$) and Until ($\mathcal{U}$)

$$\phi \quad := \quad a \mid tt \mid \neg\phi \mid \ \phi \wedge \phi \mid \mathcal{X}\phi \mid \phi\, \mathcal{U}\, \phi$$

- formulae are evaluated over paths

$\sigma =$




$$\sigma \quad \models \quad a\, \mathcal{U}\ b$$

$$\sigma \quad \models \quad \mathcal{X}\ c$$

# CTL MODEL CHECKING

- **Computational Tree Logic** [**Clarke, Emerson**] a language to deal with *non-deterministic* behaviours (e.g. "there is at least one computation at which the printer is not busy")

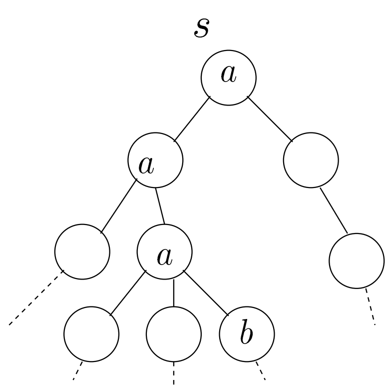
- models are still LTS: $\mathcal{M} = \langle S, R, L \rangle$.

- formulae are evaluated over states $s \in S$.

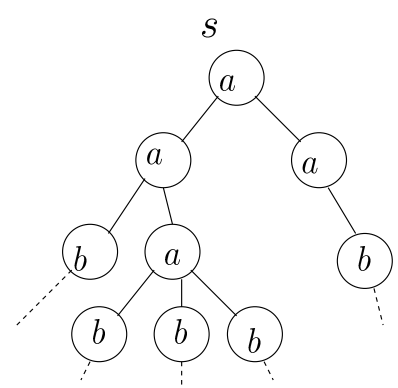
- temporal connectives are *existentially* ($E$) and *universally* ($A$) path quantified:

$$\phi \quad := \quad a \mid tt \mid \neg\phi \mid \; \phi \wedge \phi \mid E\mathcal{X}\phi \mid A\mathcal{X}\phi \mid E(\phi \; \mathcal{U} \; \phi) \mid A(\phi \; \mathcal{U} \; \phi)$$

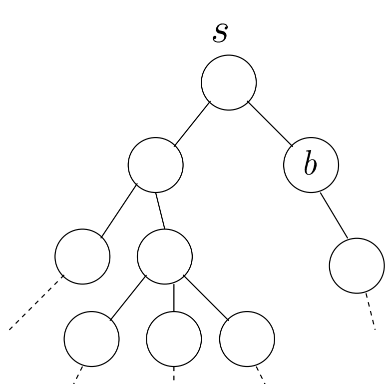
# CTL MODEL CHECKING (cont.)



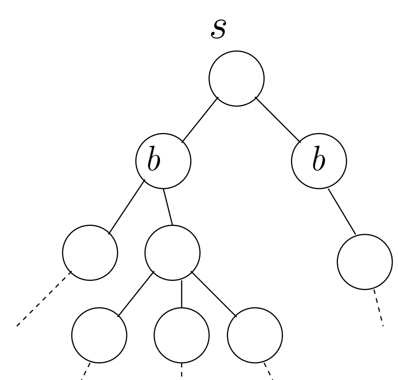

$s \models E(\ a\ \mathcal{U}\ b\ )$




$s \models A(\ a\ \mathcal{U}\ b\ )$



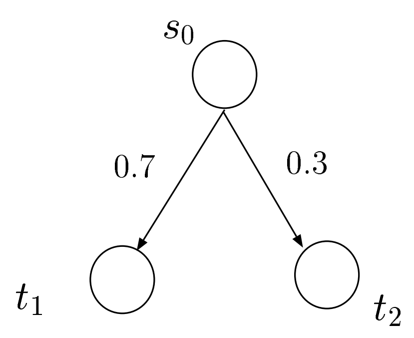

$s \models E\mathcal{X}\ (b)$




$s \models A\mathcal{X}\ (b)$

# PCTL MODEL CHECKING

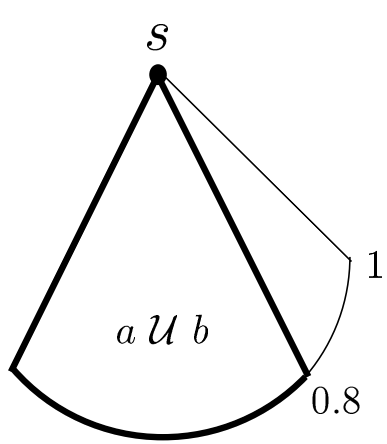
- **Probabilistic CTL** [**Hansson, Jonsson**] a language to deal with *probabilistic* behaviours.
- models are labelled Discrete Time Markov Chains (DTMC):



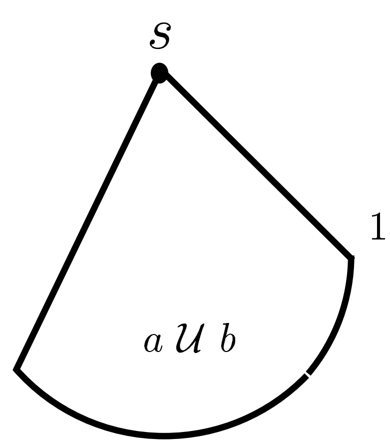

- *existential* and *universal* path quantifier are replaced by a *continuous* path quantifier ($\mathcal{P}_{\lhd p}$):

$$\phi \quad := \quad a \mid tt \mid \neg\phi \mid \ \phi \wedge \phi \mid \mathcal{P}_{\lhd p}(\mathcal{X}\ \phi) \mid \mathcal{P}_{\lhd p}(\phi\ \mathcal{U}\ \phi)$$

- the $\mathcal{P}_{\triangleleft p}$ quantifier allows to refer to the "proportion" of paths satisfying its argument.




$s \models \mathcal{P}_{\leq 0.8}(a\ \mathcal{U}\ b)$




$s \models \mathcal{P}_{\geq 1}(a\ \mathcal{U}\ b) \equiv A(a\ \mathcal{U}\ b)$

## Continuous Stochastic Logic [Aziz + Katoen & al.]

- a temporal logic for CTMC analysis
- a *steady-state* operator ($\mathcal{S}_{\lhd p}$) for steady-state analysis.
- *time-bounded* path operators ($\mathcal{X}^{\leq t}, \mathcal{U}^{\leq t}$) for transient-state analysis.
- **CSL syntax:**

$$\Phi \quad ::= \quad tt \;\mid\; a \;\mid\; \phi \wedge \phi \;\mid\; \neg\phi \;\mid\; \mathcal{S}_{\lhd p}(\phi) \;\mid\; \mathcal{P}_{\lhd p}(\varphi) \qquad (\textit{state-formulae})$$

$$\varphi \quad ::= \quad \mathcal{X}(\phi) \;\mid\; \phi\,\mathcal{U}\,\phi \;\mid\; \mathcal{X}^{\leq t}(\phi) \;\mid\; \phi\,\mathcal{U}^{\leq t}\,\phi \qquad (\textit{path-formulae})$$

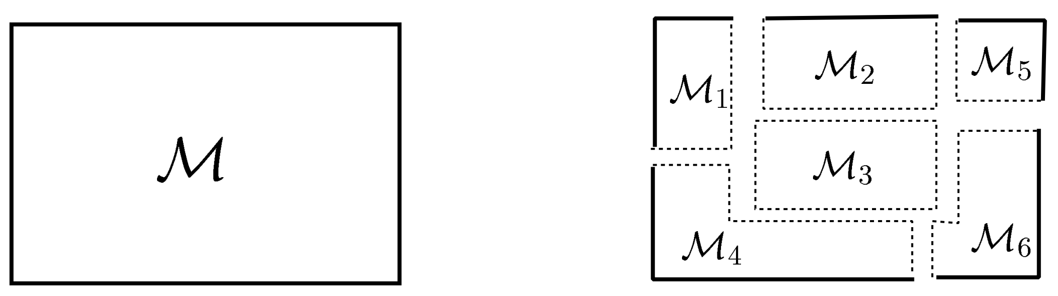
where: $a \in AP$ (*Atomic Proposition* set) with restricted syntax

# 3 - COMPOSITIONALITY AND MODEL CHECKING

- **STATE-SPACE EXPLOSION:** high-level modelling formalisms (i.e. Process Algebras, Petri Nets) easily lead to huge, hence intractable, models.

- **COMPOSITIONAL PERFORMANCE MODELLING**

  · to decompose a model $\mathcal{M}$ in a number of submodels $\mathcal{M}_i$.
  · to solve (smaller) submodels $\mathcal{M}_i$ in order to deduce properties concerning $\mathcal{M}$.



- what if we want to apply model checking to test a property $\phi$ over a decomposed model $\mathcal{M} = compose(\mathcal{M}_1, \ldots, \mathcal{M}_n)$ ?

- issue: is it possible to derive a set of *transformed formulae* $f_t(\phi) = \{\phi^1, \ldots, \phi^n\}$ such that a *satisfiability equivalence* like:

$$\mathcal{M} \models \phi \Longleftrightarrow \bigwedge_{i=1}^{n} \mathcal{M}_i \models_i \phi^i$$

  does hold ?

- benefit: checking properties ($\phi^i$) against smaller (sub)models ($\mathcal{M}_i$) is "cheaper" than checking properties ($\phi$) against bigger models ($\mathcal{M}$)

# 4 - A COMPOSITIONAL FRAMEWORK FOR CTMC: THE BOUCHERIE PRODUCT PROCESS

- Boucherie establishes a multi-dimensional CTMC which is suscetible to product-form solution

- in the general case the Boucherie CTMC models a number of processes competing over a number of shared resources

- the Boucherie CTMC is characterised by 2 restriciting conditions:

  **MUTEX:** when a process holds a resource $R_i$ no other process can access $R_i$

  **STRONG BLOCKING:** when a process holds a resource $R_i$ no other process competing over $R_i$ can evolve

- in this work we refer to the simplest case of *Boucherie* process where the resulting CTMC $\mathcal{M}$ consists of 2 sub-processes $\mathcal{M}_1$ and $\mathcal{M}_2$ which compete on a resource $R$

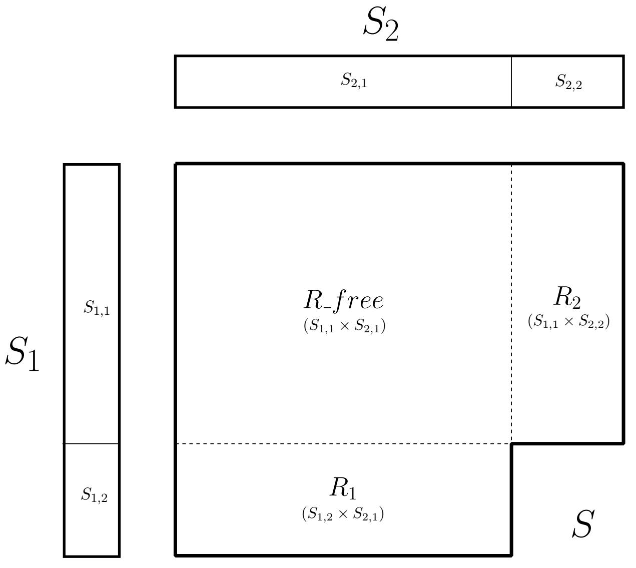
- the shared resource $R$ identifies a partition of the components' state-space

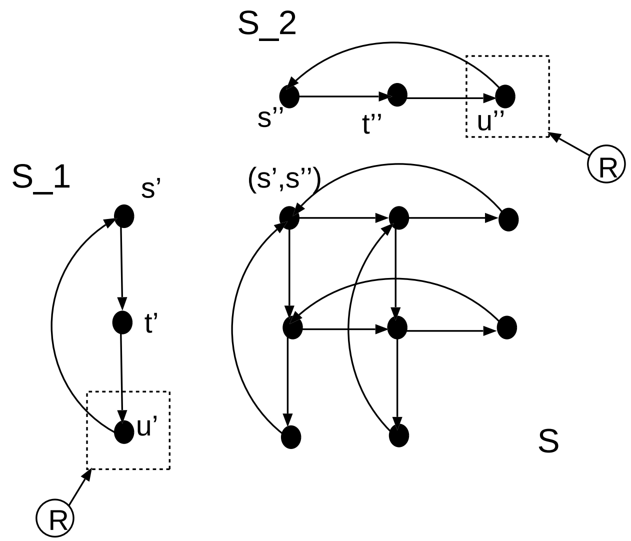
$$S_i = S_{i,1} \cup S_{i,2} \qquad\qquad i \in \{1, 2\}$$

  where:
  - $S_{i,1}$: $\mathcal{M}_i$ does not hold $R$
  - $S_{i,2}$: $\mathcal{M}_i$ does hold $R$.

- the $\mathcal{M}$ state-space, is naturally obtained by excluding $S_{1,2} \times S_{2,2}$ (mutex) from the cartesian product of the components' state-space

$$S = S_1 \times S_2 / S_{1,2} \times S_{2,2}$$






**product-form distribution:** the steady-state distribution $\pi$ of any state $(s', s'')$ of the Boucherie product process is given by:

$$\pi(s', s'') = G \cdot \pi_1(s')\pi_2(s'')$$

where $G$ is a normalising constant determined by the exact form of $S$ and $\pi_i$ is the steady-state distribution of process $i$.

# 5 - COMPOSITIONAL CSL MODEL CHECKING

- we refer to a 2 components Boucherie process.
- we consider only CSL *state formulae*:

$$\Phi \quad ::= \quad tt \;|\; a \;|\; \phi \wedge \phi \;|\; \neg\phi \;|\; \mathcal{S}_{\lhd p}(\phi) \;|\; \mathcal{P}_{\lhd p}(\varphi) \quad \text{(\textit{state-formulae})}$$

$$\varphi \quad ::= \quad \mathcal{X}(\phi) \;|\; \phi\,\mathcal{U}\,\phi$$

- partitioning $AP$ $\Longrightarrow$ partitioning $\Phi$ :

$$AP = AP_1 \cup AP_2 \qquad \Longrightarrow \qquad \Phi = \Phi_1 \cup \Phi_2 \cup \Phi_{12}$$

$$\Phi_i ::= tt \;|\; a_i \;|\; \phi_i \wedge \phi_i \;|\; \neg\phi_i \;|\; \mathcal{S}_{\lhd p}(\phi_i)$$

$$\Phi_{12} ::= \phi_1 \wedge \phi_2 \;|\; \phi_2 \wedge \phi_1 \;|\; \phi_i \wedge \phi'_{12} \;|\; \phi'_{12} \wedge \phi_i \;|\; \phi'_{12} \wedge \phi''_{12}$$
$$\;|\; \neg\phi'_{12} \;|\; \mathcal{S}_{\lhd p}(\phi'_{12})$$

***Theorem 1:*** a *single component* formula $\phi_1$ is satisfied in a state $(s', s'')$ of the Boucherie process *if and only if* its *transformation* $f_t(\phi_1)$ is satisfied in the state $s'$ of component $\mathcal{M}_1$

$$(s', s'') \models \phi_1 \Longleftrightarrow s' \models_1 f_t(\phi_1)$$

where:

$$f_t(\phi_1) = \begin{cases} a_1 & \underline{\textbf{if}} \;\; \phi_1 = a_1 \\ \neg f_t(\phi_1') & \underline{\textbf{if}} \;\; \phi_1 = \neg\phi_1' \\ f_t(\phi_1') \wedge f_t(\phi_1'') & \underline{\textbf{if}} \;\; \phi_1 = \phi_1' \wedge \phi_1'' \\ \mathcal{S}_{g(p,\phi_1',\mathcal{M}_1)}(f_t(\phi_1')) & \underline{\textbf{if}} \;\; \phi_1 = \mathcal{S}_{\lhd p}\phi_1' \end{cases}$$

the *transformation* of a *single component* formula, recursively modifies the probability argument $p$ of each steady-state formula $(\mathcal{S}_{\lhd p}(\phi_1))$

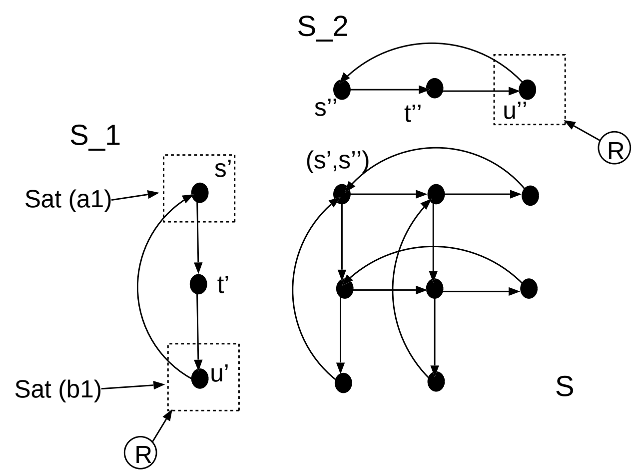
the *transformed* probability argument of a steady-state formula $\mathcal{S}_{\triangleleft p}(\phi_1)$ depends on the satisfiability of $f_t(\phi_1)$ on component $\mathcal{M}_1$

$$g(p, \phi_1, \mathcal{M}_1) = \begin{cases} \frac{p}{G} & \underline{\textbf{if}} \;\; Sat_1(f_t(\phi_1)) \subseteq S_{1,1} \\ \frac{p}{G \cdot K} & \underline{\textbf{if}} \;\; Sat_1(f_t(\phi_1)) \subseteq S_{1,2} \\ \frac{p - G \cdot J}{G \cdot K} & \underline{\textbf{if}} \;\; (Sat_{1_1}(f_t(\phi_1)) \neq \emptyset) \wedge \\ & \qquad\quad (Sat_{1_2}(f_t(\phi_1)) \neq \emptyset) \end{cases}$$

where $G$ is the *normalising* constant and:

$$\begin{aligned} J &= \sum_{s' \in Sat_{1_1}(f_t(\phi_1))} \pi_1(s') \sum_{s'' \in S_{2,2}} \pi_2(s'') \\ K &= \sum_{t'' \in S_{2,1}} \pi_2(t'') \end{aligned}$$

EXAMPLES (single component formuale)




- since $Sat_1(a_1) = \{s'\} \subseteq S_{1,1}$ hence:

$$(s', s'') \models \mathcal{S}_{\triangleleft p}(a_1) \iff s' \models_1 \mathcal{S}_{\triangleleft \frac{p}{G}}(a_1)$$

- since $Sat_1(b_1) = \{u'\} \subseteq S_{1,2}$ hence:

$$(s', s'') \models \mathcal{S}_{\triangleleft p}(b_1) \iff s' \models_1 \mathcal{S}_{\triangleleft \frac{p}{G[\pi_2(s'')+\pi_2(t'')]}}(b_1)$$

***Theorem 2:*** a *mixed* (state) formula $\phi_{12}$ is satisfied in a state $(s', s'')$ of the Boucherie process *if and only if* the boolean combination of satisfiability conditions given by $cond((s', s''), \phi_{12})$ holds

$$(s', s'') \models \phi_{12} \Longleftrightarrow cond((s', s''), \phi_{12})$$

$$cond\big((s', s''), \phi_{12}\big) = \begin{cases} s' \models_1 f_t(\phi_1) \ \ \textbf{and} \ \ s'' \models_2 f_t(\phi_2) & \textbf{if} \ \ \phi_{12} \equiv \phi_1 \wedge \phi_2 \\ s' \models_1 f_t(\phi_1) \ \ \textbf{and} \ \ cond\big((s', s''), \phi'_{12}\big) & \textbf{if} \ \ \phi_{12} \equiv \phi_1 \wedge \phi'_{12} \\ cond((s', s''), \phi'_{12}) \ \ \textbf{and} \ \ cond\big((s', s''), \phi''_{12}\big) & \textbf{if} \ \ \phi_{12} \equiv \phi'_{12} \wedge \phi''_{12} \\ \textbf{not} \ \ cond\big((s', s''), \phi'_{12}\big) & \textbf{if} \ \ \phi_{12} \equiv \neg\phi'_{12} \\ & \\ s' \models_1 \mathcal{S}_{\triangleleft \frac{p - G \cdot J((\alpha_1, \beta_2), \phi'_{12})}{G \cdot K(\beta_2)}}(\alpha_1) & \textbf{if} \ \ \phi_{12} \equiv \mathcal{S}_{\triangleleft p}(\phi'_{12}) \ \wedge \\ & (\alpha_1, \beta_2) \in dec(\phi'_{12}) \end{cases}$$

$$J((\alpha_1,\beta_2),\phi_{12}) = \sum_{(\delta_1,\gamma_2)\in dec(\phi_{12})(\delta_1,\gamma_2)\neq(\alpha_1,\beta_2)} \Big[ \sum_{t'\in Sat_1(\delta_1)} \pi_1(t') \sum_{t''\in Sat_2(\gamma_2)} \pi_2(t'') \Big]$$

$$K(\beta_2) = \sum_{t''\in Sat_2(\beta_2)} \pi_2(t'')$$

## EXAMPLES (*mixed* formuale)

- $(s', s'') \models a_1 \wedge a_2 \qquad \Longleftrightarrow \qquad s' \models_1 a_1$ and $s'' \models_2 a_2$

- $(s', s'') \models \neg(a_1 \wedge a_2) \qquad \Longleftrightarrow \qquad$ not $(s' \models_1 a_1$ and $s'' \models_2 a_2)$

- $(s', s'') \models \mathcal{S}_{\lhd p}(a_1 \wedge a_2) \qquad \Longleftrightarrow \qquad s' \models_1 \mathcal{S}_{\lhd ??}(??)$

## DECOMPOSING THE SATISFIABILITY SET OF A *mixed* FORMULA

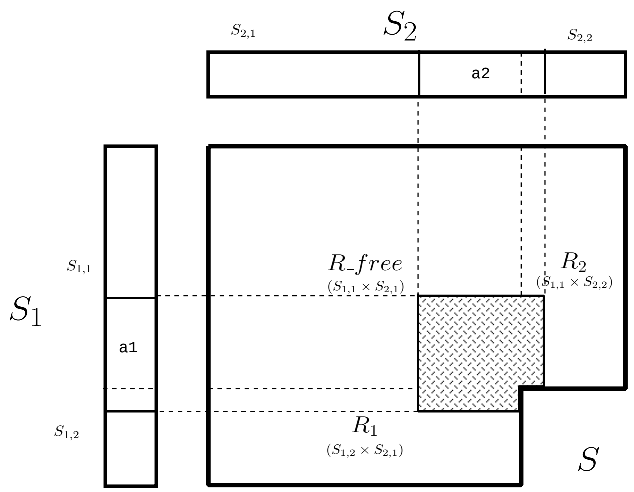
***lemma:*** the satisfiability set of any *mixed* formula $Sat(\phi_{12})$ is partitionable in a number of (disjoint) subsets each of which is characterized by a pair $(\alpha_1, \beta_2)$ of *single component* formulae

$$Sat(\phi_{12}) = \bigcup_{(\alpha_1,\beta_2)\in dec(\phi_{12})} [Sat_1(\alpha_1) \times Sat_2(\beta_2)]/S_{1,2} \times S_{2,2}$$

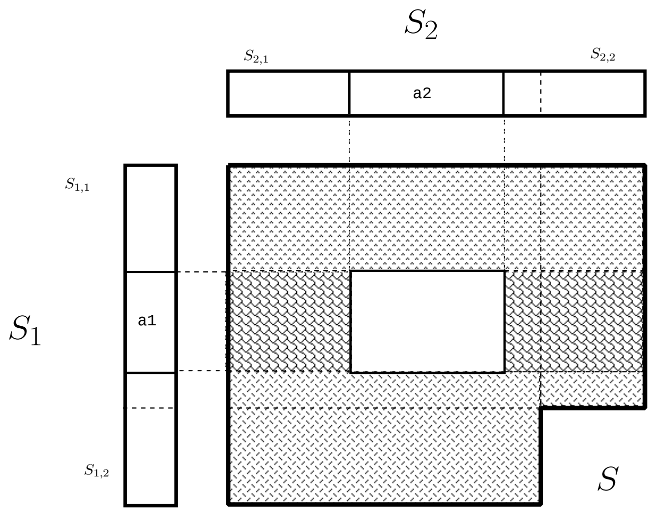
$$dec(\phi_{12}) = \begin{cases} \{(f_t(\phi_1), f_t(\phi_2))\} & \textbf{if} \quad \phi_{12} \equiv \phi_1 \wedge \phi_2 \\ f_t(\phi_1) \;\; AND_1 \;\; dec(\phi'_{12}) & \textbf{if} \quad \phi_{12} \equiv \phi_1 \wedge \phi'_{12} \\ f_t(\phi_2) \;\; AND_2 \;\; dec(\phi'_{12}) & \textbf{if} \quad \phi_{12} \equiv \phi_2 \wedge \phi'_{12} \\ dec(\phi'_{12}) \;\; AND \;\; dec(\phi''_{12}) & \textbf{if} \quad \phi_{12} \equiv \phi'_{12} \wedge \phi''_{12} \\ AND_{(\alpha_1,\beta_2)\in dec(\phi'_{12})}\{(\neg\alpha_1, true), (\alpha_1, \neg\beta_2)\} & \textbf{if} \quad \phi_{12} \equiv \neg\phi'_{12} \\ \{(true, true)\} & \textbf{if} \quad \phi_{12} \equiv \mathcal{S}_{\triangleleft p}\phi'_{12} \wedge \\ & s' \models_1 \mathcal{S}_{\triangleleft p'} dec_1(\alpha_1) \\ & \quad where \;\; (\alpha_1, \beta_2) \in dec(\phi'_{12}) \\ \{(false, false)\} & \textbf{if} \quad \phi_{12} \equiv \mathcal{S}_{\triangleleft p}\phi'_{12} \wedge \\ & s' \not\models_1 \mathcal{S}_{\triangleleft p'} dec_1(\alpha_1) \end{cases}$$

$p' = (p - G \cdot J((\alpha_1, \beta_2), \phi')12)/(G \cdot K(\beta_2))$

EXAMPLES (decomposing $Sat(\phi_{12})$)




(i)




(ii)

1- $\phi_{12} \equiv (a_1 \wedge a_2) \Longrightarrow dec(\phi_{12}) = \{(a_1, a_2)\} \Longrightarrow Sat(\phi_{12}) = Sat_1(a_1) \times Sat_2(a_2) / S_{1,2} \times S_{2,2}$.

$$(s', s'') \models \mathcal{S}_{\lhd p}(a_1 \wedge a_2) \Longleftrightarrow s' \models_1 \mathcal{S}_{\lhd \frac{p}{G\pi_2(s'')}}(a_1)$$

EXAMPLES (decomposing $Sat(\phi_{12})$)

2- $\phi_{12} \equiv \neg(a_1 \wedge a_2) \quad \Longrightarrow \quad dec(\phi_{12}) = \{(\neg a_1, true), (a_1, \neg a_2)\}$

$$\Longrightarrow \quad Sat(\phi_{12}) = (Sat_1(\neg a_1) \times S_2) \cup (Sat_1(a_1) \times Sat_2(\neg b_2))/S_{1,2} \times S_{2,2}.$$

$$(s', s'') \models \mathcal{S}_{\lhd p} \neg(a_1 \wedge a_2) \quad \Longleftrightarrow \quad s' \models_1 \mathcal{S}_{\lhd \frac{p - G\pi_1(s')}{G[\pi_2(t'')+\pi_2(u'')]}}(a_1)$$

given that $Sat_1(a_1) = \{s'\}$ and $Sat_2(a_2) = \{s''\}$

## 6 - CONCLUSION AND FUTURE WORK

- **results:** given a 2 component Boucherie process, a decompositional satisfiability equivalence exists for (restricted) CSL formulae (no *path formulae* have been considered).

- for single component formulae ($\phi_i$) the equivalence is achieved through transformation of the probability value of each steady-state formula

- for mixed formulae ($\phi_{12}$) the equivalence is given by a finite boolean combination of satisfiability conditions for *single component* formulae (which relies on the *decomposability* of $Sat(\phi_{12})$).

## FUTURE WORK

- ***path* formulae:** analysis of the more compicated probabilistic *path* formulae (issue: finding a way to relate the path measure between the product process and its components seems to be an hard task)

- **complexity:** a complexity analysis of the decompositional approach needs to be perfomed (computational costs of functions $f_t()$, $cond()$ and $dec()$ ).

- Extension to multi-dimensional Boucherie process

- Extension to other type of product-form