Undecidability of Model Checking in Brane Logic

Giorgio Bacci Marino Miculan

Department of Mathematics and Computer Science
University of Udine, ITALY

DCM 2007

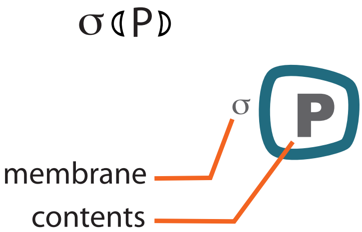
Talk Outline

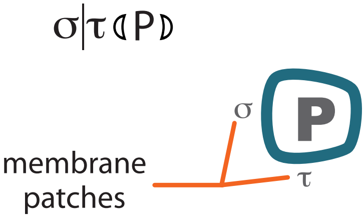
+ Summary of the Calculus and Logic
+ Proof of model checking undecidability
  - calculus with replication
  - logic with adjoints and quantifiers
+ Conclusions

(Basic) Brane Calculus [Cardelli '04]

Intended to be a model of biological membranes

systems	$P, Q ::=$	$\diamond \mid \sigma(\!|P|\!) \mid P \circ Q \mid !P$	nests of membranes
branes	$\sigma, \tau ::=$	$\mathbf{0} \mid \sigma\vert\tau \mid a.\sigma \mid !\sigma$	combination of actions
actions	$a, b ::=$	$\ldots$	(not now)



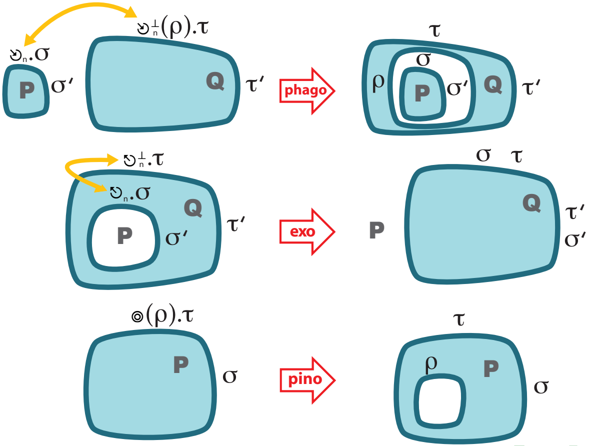


Structural Equivalence $\equiv$

	Systems	Membranes
Fluidity	$P \circ Q \equiv Q \circ P$ $P \circ (Q \circ R) \equiv (P \circ Q) \circ R$ $P \circ \diamond \equiv P$	$\sigma \vert \tau \equiv \tau \vert \sigma$ $\sigma \vert (\tau \vert \rho) \equiv (\sigma \vert \tau) \vert \rho$ $\sigma \vert \mathbf{0} \equiv \sigma$
Plenitude	$!P \equiv P \circ !P$ etc.	$!\sigma \equiv \sigma \vert !\sigma$ etc.
Congruence	$P \equiv Q \Rightarrow P \circ R \equiv Q \circ R$ $P \equiv Q \Rightarrow !P \equiv !Q$	$\sigma \equiv \tau \Rightarrow \sigma \vert \rho \equiv \tau \vert \rho$ $\sigma \equiv \tau \Rightarrow !\sigma \equiv !\tau$

Brane Reactions ➞ (PEP semantics)

actions $\ldots ⟲_n \mid ⟲_n^\perp(\sigma) \mid ⟳_n \mid ⟳_n^\perp \mid ◎(\sigma)$ phago ⟲, exo ⟳, pino ◎



Brane Logic [CMSB '06]: motivations

Logics allow to express formally the properties of biological systems, usually written in natural language.

- System specification and verification (possibly automatic): check whether a given system P satisfies a given property $\mathcal{A}$
- System synthesis: find a system which satisfies a given property $\mathcal{A}$ (synthetic biology)
- System characterization: find the formula which characterizes the behaviour of the system P
- Model validation: predict a property which should hold in a system and mount an experiment to verify it (predictive biology)

Brane Logic: syntax

There are two interacting logics:

system formulas

$\mathcal{A},\mathcal{B} ::=$	$\mathbf{T} \mid \neg\mathcal{A} \mid \mathcal{A} \vee \mathcal{B}$	(classical propositional fragment)
	$\diamond$	(void system)
	$\mathcal{M}(\!|\mathcal{A}|\!) \mid \mathcal{A} @ \mathcal{M}$	(compartment, compartment adjoint)
	$\mathcal{A} \circ \mathcal{B} \mid \mathcal{A} \triangleright \mathcal{B}$	(spatial composition, composition adjoint)
	$\Diamond\mathcal{A} \mid \diamondsuit\mathcal{A}$	(eventually modality, somewhere modality)
	$\forall x.\mathcal{A}$	(quantification over names)

like Ambient Logic
but . . .

brane formulas

$\mathcal{M},\mathcal{N} ::=$	$\mathbf{T} \mid \neg\mathcal{M} \mid \mathcal{M} \vee \mathcal{N}$	(classical propositional fragment)
	$\mathbf{0}$	(void membrane)
	$\mathcal{M} \mid \mathcal{N} \mid \mathcal{M} \blacktriangleright \mathcal{N}$	(spatial composition, composition adjoint)
	$\langle\alpha\rangle\mathcal{M}$	(action modality)

a kind of Hennessy-Milner logic

Brane Logic: satisfaction $\models$

Spatial connectives and their adjoints...
(properly of spatial calculi)

$$P \models \mathcal{A} \circ \mathcal{B} \quad \triangleq \quad \exists P', P''. P \equiv P' \circ P'' \wedge P' \models \mathcal{A} \wedge P'' \models \mathcal{B}$$
$$P \models \mathcal{M}(\!|\mathcal{A}|\!) \quad \triangleq \quad \exists P', \sigma. P \equiv \sigma(\!|P'|\!) \wedge P' \models \mathcal{A} \wedge \sigma \models \mathcal{M}$$

$$P \models \mathcal{A} @ \mathcal{M} \quad \triangleq \quad \forall \sigma. \sigma \models \mathcal{M} \Rightarrow \sigma(\!|P|\!) \models \mathcal{A}$$
$$P \models \mathcal{A} \triangleright \mathcal{B} \quad \triangleq \quad \forall P'. P' \models \mathcal{A} \Rightarrow P \circ P' \models \mathcal{B} \qquad \text{(guarantee)}$$

...both temporal and spatial modalities (bi-modal logic)

$$P \models \Diamond \mathcal{A} \quad \triangleq \quad \exists P' : \Pi. P \longrightarrow^{*} P' \wedge P' \models \mathcal{A}$$
$$P \models \lozenge \mathcal{A} \quad \triangleq \quad \exists P' : \Pi. P \downarrow^{*} P' \wedge P' \models \mathcal{A}$$

Undecidability of model checking

Given P and $\mathcal{A}$, is $P \models \mathcal{A}$?

Two sources of undecidability:

1. if processes have unbound replication ($!P$), model checking is undecidable
   Solution:
   - consider only finite calculi (without replications)
   - or admit only guarded replications [Busi-Zavattaro '04]
2. if the logic contain guarantee ($\triangleright$) and quantifiers, model checking the finite state Brane Calulus is also undecidable.

In [CMSB '06] a model checking algorithm for finite calculus and $\triangleright$-free logic

Undecidability of model checking

Given P and $\mathcal{A}$, is $P \vDash \mathcal{A}$?

Two sources of undecidability:

1. if processes have unbound replication ($!P$), model checking is undecidable
   Solution:
   - consider only finite calculi (without replications)
   - or admit only guarded replications [Busi-Zavattaro '04]
2. if the logic contain guarantee ($\triangleright$) and quantifiers, model checking the finite state Brane Calulus is also undecidable.

In [CMSB '06] a model checking algorithm for finite calculus and $\triangleright$-free logic

Undecidability in presence of replication

The proof is done by reduction of a undecidable problem:

Proof Outline

- encode in Brane Calculus the Post Corrispondence Problem
- give a formula that holds iff PCP as a solution

Encoding PCP

Post Corrispondence Problem

Instance: a finite set of pairs of words $\{(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)\}$

Question: there exist a sequence $i_0, i_1, \ldots, i_k$ ($1 \leq i_j \leq n$ for all $0 \leq j \leq k$) such that $\alpha_{i_0} \cdot \ldots \cdot \alpha_{i_k} = \beta_{i_0} \cdot \ldots \cdot \beta_{i_k}$

Encoding idea:

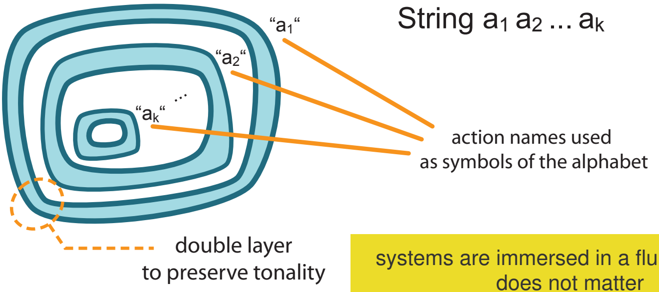
start from two empty words W_1, W_2

- non-deterministically choose a pair from the instace to concatenate to W_1 and W_2
- compare the two words

and repeat...

Encoding PCP: strings

. . . we use membranes as string constructors

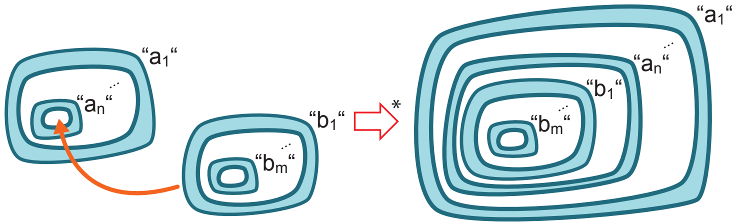


systems are immersed in a fluid, so order does not matter

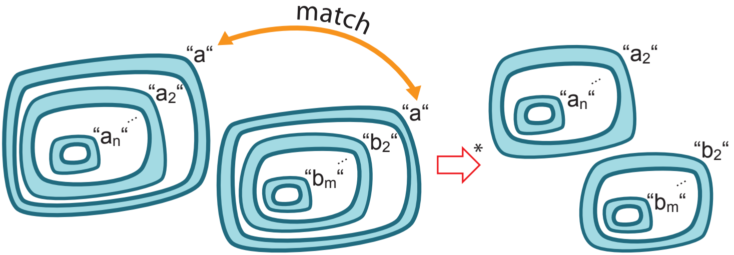
membrane nesting preserves the ordering

Encoding PCP: concatenation & comparison

concatenation



comparison



Undecidability in presence of replication

Two replication constructors:

- replication on systems $(!P \equiv P \circ !P)$
- replication on branes $(!\sigma \equiv \sigma | !\sigma)$

We have to treat them separately. . .

PCP$_S$ **PCP$_m$**

Encoding PCP on systems: first definition

$$\mathbf{PCP}_S \triangleq \mathbf{Word}_1(\epsilon) \circ \mathbf{Word}_2(\epsilon) \circ \mathbf{Concatenate} \circ \mathbf{Compare}$$

$$\mathbf{Concatenate} \triangleq\ !\mathbf{Concatenate}(\alpha_1, \beta_1) \circ \ldots \circ !\mathbf{Concatenate}(\alpha_n, \beta_n)$$

$$\mathbf{Compare} \triangleq\ !\mathbf{Consume}(a) \circ !\mathbf{Consume}(b)$$

Encoding PCP on systems: first definition

$$\textbf{PCP}_S \triangleq \textbf{Word}_1(\epsilon) \circ \textbf{Word}_2(\epsilon) \circ \textbf{Concatenate} \circ \textbf{Compare}$$

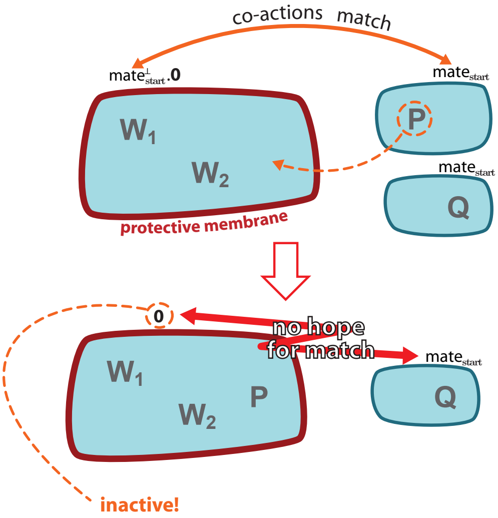
$$\textbf{Concatenate} \triangleq\ !\textbf{Concatenate}(\alpha_1, \beta_1) \circ \ldots \circ !\textbf{Concatenate}(\alpha_n, \beta_n)$$

$$\textbf{Compare} \triangleq\ !\textbf{Consume}(a) \circ !\textbf{Consume}(b)$$

WRONG!

if comparison is interleaved with concatenation?

Synchronizing jobs. . .



the two words are enveloped in a protective membrane

Encoding PCP on systems: final definition

formally...

$$\mathbf{PCP}_S \triangleq \text{mate}^{\perp}_{start} \llparenthesis \mathbf{Word}_1(\epsilon) \circ \mathbf{Word}_2(\epsilon) \circ \mathbf{End} \rrparenthesis \circ$$
$$\mathbf{Concatenate} \circ \mathbf{Compare}$$

$$\mathbf{Concatenate} \triangleq !\mathbf{Concatenate}(\alpha_1, \beta_1) \circ \ldots \circ !\mathbf{Concatenate}(\alpha_n, \beta_n)$$

$$\mathbf{Compare} \triangleq !\mathbf{Consume}(a) \circ !\mathbf{Consume}(b)$$

Undecidability (systems replication)

if $\mathbf{PCP}_S$ satisfy the the formula $\mathcal{A}$ the PCP as a solution

$$\mathcal{A} \triangleq \Diamond(\mathit{nonempty}(w_1) \wedge \Diamond(\mathit{empty}(w_1) \wedge \mathit{empty}(w_2)))$$

$\mathcal{A}$ contains only propositional connectives, temporal and spatial modalities and the compartment connective.

No need of quantifiers or adjoint connectives

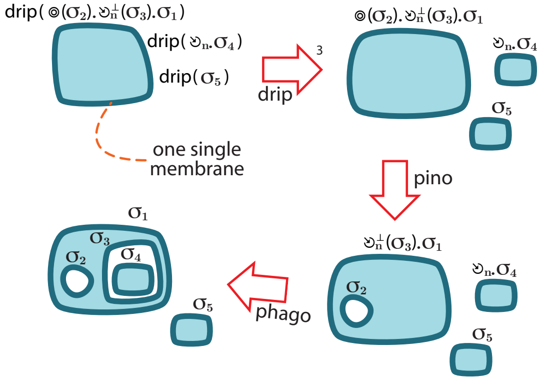
Theorem

The model checking problem for Brane Calculi with replication on systems against the Brane Logic is undecidable.

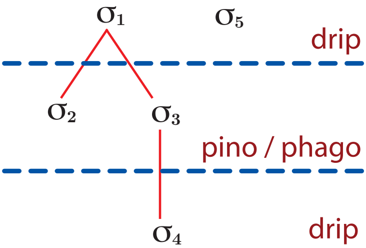
Reducing membranes to systems

we do not directly define a system **PCP**$_m$...

...instead we use a little trick



membrane language is expressive enough to produce all possible systems



Generate(P): definition & properties

formally…

$$
\begin{array}{lcl}
\textbf{Generate}_\phi(\diamond) & \triangleq & \mathbf{0} \\
\textbf{Generate}_\phi(\sigma \llparenthesis P \rrparenthesis) & \triangleq & \text{drip}(\textbf{Endo}^\perp_\phi(P,\sigma)) | \textbf{Endo}_\phi(P) \\
\textbf{Generate}_\phi(P \circ Q) & \triangleq & \text{drip}(\textbf{Generate}_\phi(P)) | \text{drip}(\textbf{Generate}_\phi(Q))
\end{array}
$$

$$
\begin{array}{lcl}
\textbf{Endo}_\phi(\diamond) & \triangleq & \mathbf{0} \\
\textbf{Endo}_\phi(\tau \llparenthesis Q \rrparenthesis) & \triangleq & \begin{cases} \mathbf{0} & \text{if } Q \equiv \diamond \\ \text{drip}(\circlearrowleft_{\phi(\tau \llparenthesis Q \rrparenthesis)}.\textbf{Generate}_\phi(Q)) & \text{otherwise} \end{cases} \\
\textbf{Endo}_\phi(P \circ Q) & \triangleq & \textbf{Endo}_\phi(P) | \textbf{Endo}_\phi(Q) \\
\textbf{Endo}^\perp_\phi(\diamond, \sigma) & \triangleq & \sigma \\
\textbf{Endo}^\perp_\phi(\tau \llparenthesis Q \rrparenthesis, \sigma) & \triangleq & \begin{cases} \circledcirc(\tau).\sigma & \text{if } Q \equiv \diamond \\ \circlearrowleft^\perp_{\phi(\tau \llparenthesis Q \rrparenthesis)}(\tau).\sigma & \text{otherwise} \end{cases} \\
\textbf{Endo}^\perp_\phi(P \circ Q, \sigma) & \triangleq & \textbf{Endo}^\perp_\phi(P, \textbf{Endo}^\perp_\phi(Q, \sigma))
\end{array}
$$

$$
\begin{array}{lcl}
\textbf{Generate}_\phi(P) \llparenthesis \diamond \rrparenthesis & \rightarrow^* & P \\
!\textbf{Generate}_\phi(P) \llparenthesis \diamond \rrparenthesis & \rightarrow^* & !\textbf{Generate}_\phi(P) \llparenthesis \diamond \rrparenthesis \circ P
\end{array}
$$

Undecidability (membrane replication)

$$\cancel{!P \equiv !P \circ P}$$

instead...

$$!\textbf{Generate}_{\phi}(P)(\!|\diamond|\!) \;\Rightarrow^{*}\; !\textbf{Generate}_{\phi}(P)(\!|\diamond|\!) \circ P$$

Theorem

The model checking problem for Brane Calculi with replication on membranes against the Brane Logic is undecidable.

Guarantee ($\triangleright$) can express satisfiability

$$P \models \mathcal{A} \triangleright \mathbf{F} \iff \forall P'.(P' \models \mathcal{A} \Rightarrow P' \circ P \models \mathbf{F})$$
$$\iff \forall P'.P' \not\models \mathcal{A}$$
$$\iff \mathcal{A} \text{ is not satisfiable}$$

so...

$$P \models \neg(\mathcal{A} \triangleright \mathbf{F}) \iff \mathcal{A} \text{ is satisfiable}$$

Brane Logic is an extension of **FOL**

we can encode First Order Logic in Brane Logic...

from structures to systems

no need of replication

$$a \in \mathcal{D} \iff \exists P'.P \equiv \circlearrowleft_d (\!| \circlearrowright_a (\!| \diamond |\!) |\!) \circ P$$

$$R_i(a_1, \ldots, a_k) \in \mathcal{S} \iff \exists P''.P \equiv \circlearrowleft_{r_i} (\!| \circlearrowright_{a_1} (\!| \ldots \circlearrowright_{a_k} (\!| \diamond |\!) \ldots |\!) |\!) \circ P''$$

$$\begin{array}{lcl}
[\![R_i(x_1, \ldots, x_k)]\!] & \triangleq & \langle \circlearrowleft_{r_i} \rangle (\!| \langle \circlearrowright_{x_1} \rangle (\!| \langle \circlearrowright_{x_2} \rangle (\!| \ldots \langle \circlearrowright_{x_k} \rangle (\!| \diamond |\!) \ldots |\!) |\!) |\!) \circ \mathbf{T} \\
[\![\varphi \wedge \psi]\!] & \triangleq & [\![\varphi]\!] \wedge [\![\psi]\!] \\
[\![\neg\varphi]\!] & \triangleq & \neg[\![\varphi]\!] \\
[\![\exists x.\varphi]\!] & \triangleq & \exists x.((\langle \circlearrowleft_d \rangle (\!| \langle \circlearrowright_x \rangle (\!| \diamond |\!) |\!) \circ \mathbf{T}) \wedge [\![\varphi]\!])
\end{array}$$

Undecidability (guarantee + quantifiers)

Lemma

A closed first-order formula φ of **FO** *admits a finite model iff there exists a finite state Brane Calculus system P such that $P \vDash [\![\varphi]\!]$.*

Theorem (Trakhtenbrot)

Given a first-order formula φ, it is undecidable to know whether φ admits a finite model.

Lemma

Brane Logic satisfiability is undecidable

Theorem

The model checking problem of finite states Brane Calculus against formulas with guarantee is undecidable.

Conclusions

We have shown

- Undecidability of model checking without quantifiers and adjoints, in presence of replication
- Undecidability of model checking with quantifiers and adjoints, in absence of replication

Future works

- look for some weaker logical connectives in place of adjoints
- look for subsets of the calculus for which satisfaction is decidable (Mate-Bud-Drip calculus)

Thanks.