A modal extension of BBI for resource transformation (work in progress)

J.R. Courtault - D. Galmiche

ANR DynRes Meeting - Nancy

February 2013

Introduction - resource logics

Resources

- Resource is a key notion in computer science:
  - Memory
  - Processes
  - Messages
- Different concerns about resources:
  - Location
  - Ownership
  - Access to
  - Consumption of

▶ **Study of resources and related notions through logics**

Introduction - resource logics

Bunched Implications (BI) logic (O'Hearn and Pym 1999, Pym 2002)

- $\textbf{BI} = \begin{cases} \wedge, \vee, \rightarrow, \top, \bot \text{ (additives)} \\ *, \mathrel{-\!\!*}, \mathrm{I} \text{ (multiplicatives)} \end{cases}$

  BI (intuitionistic additives) , **BBI** (classical additives)

- Sequents with bunches (trees of formulae where internal nodes are "," or ";"): $\dfrac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \mathrel{-\!\!*} \psi}$ $\dfrac{\Gamma; \phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi}$

- Bunches can be viewed as areas of a model:

  $$A, (B; C), A \quad \rightsquigarrow \quad \boxed{\; A \;\vdots\; BC \;\vdots\; A \;}$$

- Resources are areas and propositional symbols are properties of resources (areas)

- **BI** and **BBI** focus on separation (,) / sharing (;)

Introduction - resource logics

Separation logics

- **BI** and **BBI** logical kernels of separation logics
- Some separation logics:
  - **PL**: Pointer (Separation) Logic with $(x \mapsto a, b)$ (O'Hearn et al. 2001)
  - **BI-Loc**: Separation Logic with locations (Biri-Galmiche 2007)
  - **MBI**: Separation Logic with modalities for processes $(R, E \xrightarrow{a} R', E')$ (Pym-Toft 2006)
  - **DBI**: Separation Logic with modalities for dynamic properties of resources (Courtault-Galmiche 2013)

▶ **Study of dynamics in resource/separation logics**

Introduction - resource logics

Dynamics in resource logics

- What are systems with dynamic resources?
  - Systems that transform resources (producers / consumers)
  - Systems that modify resource properties (value of cells of a cellular automata): no resource production/consumption
- Resource logics and dynamics
  - **BI**: Properties on resources = no dynamics
  - **MBI** $(R, E \xrightarrow{a} R', E')$: Dynamics is resource transformation
  - **DBI** (**BI** $+ \Diamond, \Box$): Dynamic properties of resources

Introduction - MBI logic

MBI and **SCRP** (Pym-Tofte 2006)

- **SCRPr**: Synchronous Calculus of Resources and Processes
  - Processes: $E ::= 0 \mid X \mid a : E \mid E + E \mid E \times E \mid \nu R.E \mid fix_i X.E$
  - **SCRPr** transitions (some rules):

$$\frac{}{R, a : E \xrightarrow{a} \mu(a, R), E}\ (\mu(a, R)\downarrow) \qquad \frac{R, E \xrightarrow{a} R', E' \qquad S, F \xrightarrow{b} S', F'}{R \circ S, E \times F \xrightarrow{a\#b} R' \circ S', E' \times F'}\ (R \circ S \downarrow)$$

- **MBI**: **BBI** + modalities ($\langle a\rangle$, $[a]$, $\langle a\rangle_\nu$, $[a]_\nu$)
- Forcing relation:
  - $R, E \vDash \phi * \psi$ iff $\exists R_1, R_2, E_1, E_2 \cdot R = R_1 \circ R_2$ and $E \sim E_1 \times E_2$ and $R_1, E_1 \vDash \phi$ and $R_2, E_2 \vDash \psi$
  - $R, E \vDash \langle a\rangle\phi$ iff $\exists R', E' \cdot R, E \xrightarrow{a} R', E'$ and $R', E' \vDash \phi$
  - $R, E \vDash \langle a\rangle_\nu\phi$ iff $\exists T, R', E' \cdot R \circ T, E \xrightarrow{a} R', E'$ and $R', E' \vDash \phi$

An example: mutual exclusion

- Processes:

  $E \stackrel{def}{=} nc : E + critical : E_{critical}$
  $E_{critical} \stackrel{def}{=} critical : E_{critical} + critical : E$

- Minimum resources required for the action: $\rho(nc) = \{e\}$ and $\rho(critical) = \{R\}$

- The μ function: $\mu(a, R) = R$ for any a action

- The action *critical#critical* is never performed: $R, E \times E \models [critical\#critical]\perp$

- Problems:
  - Only a calculus with bunches and without completeness
  - $R, E \times E \models [critical\#critical]\perp$ does not mean that in any reachable state, couple (resource, process), it is impossible to execute two concurrent critical actions (**need of** $\Diamond$ **and** $\Box$)

DBI logic

- Dynamic modal **BI**
  - **BI** with modalities $\Diamond$ and $\Box$
  - Dynamic resource properties
  - A calculus that is sound and complete
- **DBI** models:
  - resources (resource monoid)
  - states and a state relation
- Forcing relation:
  - $r, s \vDash \phi * \psi$ iff $\exists r', r'' \cdot r' \bullet r'' \sqsubseteq r$ and $r', s \vDash \phi$ and $r'', s \vDash \psi$ (remark: $*$ separates only the resource r)
  - $r, s \vDash \Diamond\phi$ iff $\exists s' \cdot s \preceq s'$ and $r, s' \vDash \phi$

Introduction - DBI logic

An example: properties on states of webservices

- A set of composed webservices $W = \{W_0, W_1, W_2, W_3, ...\}$
- A model:

	t_0	t_1	t_2	t_3	t_4	t_5	
W_0 :	id	ru	ru	ru	id	id	(id: idle)
W_1 :	id	id	ru	ru	ru	id	(ru: running)

....

- An interpretation $[\![.]\!]$:
  - $[\![P_{idle}]\!] = \{(S, t_i) \mid \exists W_i \in S \cdot W_i \text{ is } \textit{idle} \text{ at time } t_i\}$
  - $[\![P_{running}]\!] = \{(S, t_i) \mid \exists W_i \in S \cdot W_i \text{ is } \textit{running} \text{ at time } t_i\}$

  where $S \subseteq W$ is a set of webservices.

  For example: $S, t \models P_{idle}$ if there is at least a webservice in S that is idle at time t

Introduction - DBI logic

An example: properties on states of webservices

	t_0	t_1	t_2	t_3	t_4	t_5	
W_0 :	id	ru	ru	ru	id	id	(id: idle)
W_1 :	id	id	ru	ru	ru	id	(ru: running)

....

- Properties that can be expressed:
  - $\{W_0, W_1\}, t_1 \vDash P_{idle}$
  - $\{W_0, W_1\}, t_1 \vDash P_{idle} \wedge P_{idle}$ but $\{W_0, W_1\}, t_1 \nvDash P_{idle} * P_{idle}$
  - $\{W_0, W_1\}, t_0 \vDash P_{idle} * P_{idle}$
  - $\{W_0, W_1\}, t_0 \vDash (P_{idle} * P_{idle}) \wedge \Diamond(P_{idle} * P_{running})$
- Problem: resource transformation cannot be express in **DBI** (it is not possible to model the messages that are produced / exchanged by the webservices)

Some results

- **DMBI** logic
  - captures resource transformation ($\approx$ **MBI**)
  - includes modalities $\Diamond$ and $\Box$ ($\approx$ **DBI**)
  - Restriction to only one process ($\not\approx$ **MBI**)
- Semantics: μ-dynamic resource monoids
- Proof theory: a tableaux method that is sound and complete
- Counter-model extraction

Plan

Plan

DMBI Logic - Language

Language

- **DMBI** $=$ **BBI** $+ \langle a \rangle\ [a]\ \Diamond\ \Box$:

$$\phi ::= p \mid \bot \mid \mathrm{I} \mid \phi \to \phi \mid \phi * \phi \mid \phi \mathrel{-\!\!*} \phi \mid \langle a \rangle \phi \mid [a]\phi \mid \Diamond \phi \mid \Box \phi$$

- Syntactic sugar:

$$\neg\phi \equiv \phi \to \bot \qquad \top \equiv \neg\bot$$

$$\phi \vee \psi \equiv \neg\phi \to \psi \qquad \phi \wedge \psi \equiv \neg(\phi \to \neg\psi)$$

$$[a]\phi \equiv \neg\langle a \rangle \neg\phi \qquad \Box\phi \equiv \neg\Diamond\neg\phi$$

Semantics

- Resource monoid: $\mathcal{R} = (R, \bullet, e)$
  - R is a set of *resources*
  - $e \in R$ is the unit resource
  - $\bullet : R \times R \rightarrow R$ such that, for any $r_1, r_2, r_3 \in R$:
    - Neutral element: $r_1 \bullet e = e \bullet r_1 = r_1$
    - Commutativity: $r_1 \bullet r_2 = r_2 \bullet r_1$
    - Associativity: $r_1 \bullet (r_2 \bullet r_3) = (r_1 \bullet r_2) \bullet r_3$

  Remark: $\bullet$ is total because a resource is viewed as a multiset of atomic resources

Semantics

- Action monoid (non commutative): $\mathcal{A} = (Act, \odot, 1)$
  - *Act* is a set of *actions*
  - $1 \in Act$ is the unit action
  - $\odot : Act \times Act \to Act$ such that, for any $a_1, a_2, a_3 \in Act$:
    - Neutral element: $a_1 \odot 1 = 1 \odot a_1 = a_1$
    - Associativity: $a_1 \odot (a_2 \odot a_3) = (a_1 \odot a_2) \odot a_3$

Remark: actions are viewed as lists of atomic actions

DMBI Logic - Semantics

Semantics

- A μ-dynamic resource monoid: $\mathcal{M} = (\mathcal{R}, \mathcal{A}, S, \|\cdot\rangle\!\rangle, \mu)$
  - S is a set of *states*
  - $\|\cdot\rangle\!\rangle \subseteq S \times Act \times S$, such that:
    - $\|\cdot\rangle\!\rangle$-unit: $s_1 \|1\rangle\!\rangle\, s_1$
    - $\|\cdot\rangle\!\rangle$-composition: if $s_1 \|a_1\rangle\!\rangle\, s_2$ and $s_2 \|a_2\rangle\!\rangle\, s_3$ then $s_1 \|a_1 \odot a_2\rangle\!\rangle\, s_3$
  - $\mu : Act \times R \rightharpoonup R$, such that:
    - μ-unit: $\mu(1, r)\downarrow$ and $\mu(1, r) = r$
    - μ-composition: if $\mu(a_1, r)\downarrow$ and $\mu(a_2, \mu(a_1, r))\downarrow$ then $\mu(a_1 \odot a_2, r)\downarrow$ and $\mu(a_1 \odot a_2, r) = \mu(a_2, \mu(a_1, r))$
- Denotations:
  - $r, s \xrightarrow{a} r', s'$ iff $\mu(a, r)\downarrow$, $\mu(a, r) = r'$ and $s \|a\rangle\!\rangle\, s'$
  - $r, s \rightsquigarrow r', s'$ iff $r, s \xrightarrow{a_0} r_1, s_1 \xrightarrow{a_1} \dots \xrightarrow{a_{n-1}} r_n, s_n \xrightarrow{a_n} r', s'$

DMBI Logic - Semantics

Semantics

- μ-Model: $\mathcal{K} = (\mathcal{M}, [\![\cdot]\!], |\cdot|, \vDash_{\mathcal{K}})$
  - $r, s \vDash_{\mathcal{K}} p$ iff $(r, s) \in [\![p]\!]$
  - $r, s \vDash_{\mathcal{K}} \bot$ never
  - $r, s \vDash_{\mathcal{K}} \mathrm{I}$ iff $r = e$
  - $r, s \vDash_{\mathcal{K}} \phi \to \psi$ iff $r, s \vDash_{\mathcal{K}} \phi \Rightarrow r, s \vDash_{\mathcal{K}} \psi$
  - $r, s \vDash_{\mathcal{K}} \phi * \psi$ iff $\exists r_1, r_2 \in R \cdot r = r_1 \bullet r_2$ and $r_1, s \vDash_{\mathcal{K}} \phi$ and $r_2, s \vDash_{\mathcal{K}} \psi$
  - $r, s \vDash_{\mathcal{K}} \phi \mathrel{-\!\!*} \psi$ iff $\forall r' \in R \cdot r', s \vDash_{\mathcal{K}} \phi \Rightarrow r \bullet r', s \vDash_{\mathcal{K}} \psi$
  - $r, s \vDash_{\mathcal{K}} \langle a \rangle \phi$ iff $\exists r' \in R \cdot \exists s' \in S \cdot r, s \xrightarrow{|a|} r', s'$ and $r', s' \vDash_{\mathcal{K}} \phi$
  - $r, s \vDash_{\mathcal{K}} \Diamond \phi$ iff $\exists r' \in R \cdot \exists s' \in S \cdot r, s \rightsquigarrow r', s'$ and $r', s' \vDash_{\mathcal{K}} \phi$
- Validity: ϕ is *valid* iff $r, s \vDash_{\mathcal{K}} \phi$ for any $\mathcal{K}$, r and s

Plan

DMBI Logic - Expressiveness

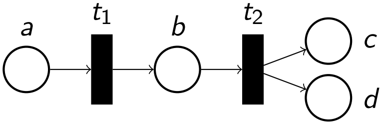
Example 1 - Petri nets

- A Petri net $\mathcal{P} = (P, T, pre, post)$
- We show that $\mathcal{P}$ is a μ-DRM $\mathcal{M} = (\mathcal{R}, \mathcal{A}, S, \|\cdot\rangle\rangle, \mu)$:
  - $\mathcal{R} = (R, \bullet, e)$ where:
    - R is the set of all multisets over P
    - $\bullet$ is the addition over multisets
    - e is the empty multiset
  - $\mathcal{A} = (Act, \odot, 1)$ where:
    - Act is the set of lists over T
    - $\odot$ is the concatenation of lists
    - 1 is the empty list
  - $S = \{s_1\}$
  - $s_1 \, \| t_1 \odot \ldots \odot t_n \rangle\rangle \, s_1$ for any $t_1, \ldots, t_n \in T$
  - $\mu(t_1 \odot \ldots \odot t_n, M) = \begin{cases} \uparrow & \text{if } \nexists M_1, \ldots, M_n \in R \text{ such that} \\ & M \, [t_1\rangle \, M_1 \, [t_2\rangle \ldots [t_n\rangle \, M_n \\ M_n & \text{otherwise} \end{cases}$

DMBI Logic - Expressiveness

Example 1 - Petri nets

- A μ-Model $\mathcal{K} = (\mathcal{M}, [\![\cdot]\!], |\cdot|, \vDash_{\mathcal{K}})$:
  - $[\![p]\!] = \{([p], s_1)\}$ for any $p \in P$
  - $|t| = t$ for any $t \in T$



- Examples: $[a], s_1 \vDash_{\mathcal{K}} \Diamond(c * d)$ and $[a], s_1 \vDash_{\mathcal{K}} \langle t_1 \rangle b$
- As opposed to PN semantics for **BI** (O'Hearn-Yang 1999):
  - No monotonicity that encodes reachability: $[a], s_1 \nvDash_{\mathcal{K}} b$ but $[a] \vDash_{BI} b$ and $[a], s_1 \vDash_{\mathcal{K}} \Diamond b$
  - $\rightarrow$ and $\neg$ are classical: $[b] \nvDash_{BI} \neg a$ but $[b] \vDash_{\mathcal{K}} \neg a$
- Remark: this example uses only one state s_1

DMBI Logic - Expressiveness

Example 2 - Concurrent process simulation

- In **DMBI** there is only one process:
  How models concurrent systems like webservices or protocol?

- **Idea:** let A_1 and A_2 two automata. The automaton $A_1 \times A_2$ is the automaton that simulates the concurrent execution of A_1 and A_2.

- **Objective:** let $\mathcal{M}_i = (\mathcal{R}, \bullet, e, \mathcal{A}, S_i, \| \cdot \rangle\!\rangle_i, \mu_i)$ such that $1 \leqslant i \leqslant n$.
  We want to construct $\mathcal{M} = (\mathcal{R}, \bullet, e, \mathcal{A}, S, \| \cdot \rangle\!\rangle, \mu)$ such that:

$$r_i, s_i \xrightarrow{a_i} r_i', s_i' \text{ for } 1 \leqslant i \leqslant n \text{ iff}$$

$$r_1 \bullet \ldots \bullet r_n, s_1/\ldots/s_n \xrightarrow{a_1 \# \ldots \# a_n} r_1' \bullet \ldots \bullet r_n', s_1'/\ldots/s_n'$$

- **Question:** what hypothesis on μ ?
  $(\mu(a, R)\downarrow \quad \Rightarrow \quad \mu(a, R \bullet S)\downarrow \text{ and } \mu(a, R \bullet S) = \mu(a, R) \bullet S)$

DMBI Logic - Expressiveness

Some questions

- **MBI**:
  - $R, E \models \mathrm{I}$ iff $R = e$ and $E \sim 1$
  - $R, E \models \phi * \psi$ iff $\exists R_1, R_2, E_1, E_3 \cdot R = R_1 \circ R_2$ and $E \sim E_1 \times E_2$ and $R_1, E_1 \models \phi$ and $R_2, E_2 \models \psi$
- **DMBI**:
  - $r, s \models_{\mathcal{K}} \mathrm{I}$ iff $r = e$
  - $r, s \models_{\mathcal{K}} \phi * \psi$ iff $\exists r_1, r_2 \in R \cdot r = r_1 \bullet r_2$ and $r_1, s \models_{\mathcal{K}} \phi$ and $r_2, s \models_{\mathcal{K}} \psi$

▶ What is the meaning of the formula I ?

"no resource" or "no resource and unit process"

$\Rightarrow$ it should mean "no resource": $\mathrm{I} \rightarrow [a]\bot$
(without resource the action a cannot be performed)

▶ What kind of decomposition with $*$?

decomposition of resources only or of resources and processes

$\Rightarrow$ case studies (protocols or Web services)

Plan

DMBI Proof theory - Tableaux method

An extension of **BI** calculus (Galmiche-Méry-Pym 2005) based on constrained set of statements (CSS in Larchey 2012)

- Resource labels (R), action labels (Act) and state labels (S)
- Resource constraints ($=$), μ-constraints (μ) and transition constraints ($\|.\rangle\!\rangle$)
- Signed formulae: $\mathbb{S}\phi : (x, u)$
- Branches are denoted $\langle \mathcal{F}, \mathcal{C} \rangle$ where $\mathcal{C}$ is a set of resource, transition and μ constraints
- Assertions/requirements

Labels

- **Resource labels** (L_r):

$$X ::= 1_r \mid c_i \mid X \circ X$$

where $c_i \in \gamma_r = \{c_1, c_2, ...\}$ and $\circ$ is a function on L_r that is associative, commutative and 1_r is its unit. $x \circ y$ is denoted xy.

- **Action labels** (L_a):

$$X ::= 1_a \mid a_i \mid d_i \mid X \centerdot X$$

where $a_i \in S_{Act}$, $d_i \in \gamma_a = \{d_1, d_2, ...\}$, $S_{Act} \cap \gamma_a = \emptyset$ and $\centerdot$ is a function on L_a that is associative (not commutative) and 1_a is its unit. $f \centerdot g$ is denoted fg.

- **State labels** (L_s): $L_s = \{l_1, l_2, ...\}$.

DMBI Proof theory - Tableaux method

Constraints

- Resource constraints:
  - encode equality on resources.
  - $x \sim y$ where x and y are resource labels.
- μ-constraints:
  - encode the function μ.
  - $x \overset{f}{\twoheadrightarrow} y$ where x and y are resource labels and f is an action label.
- Transition constraints:
  - Encode the function $\|\cdot\rangle\!\rangle$.
  - $u \overset{f}{\rightarrowtail} v$ where u and v are state labels and f is an action label.

Constraint closure

- Rules that product resource constraints:

$$\frac{}{1_r \sim 1_r}\ \langle 1_r \rangle \qquad \frac{x \sim y}{y \sim x}\ \langle s_r \rangle \qquad \frac{xy \sim xy}{x \sim x}\ \langle d_r \rangle$$

$$\frac{x \sim y \qquad y \sim z}{x \sim z}\ \langle t_r \rangle \qquad \frac{x \sim x' \qquad y \sim y'}{xy \sim x'y'}\ \langle g_r \rangle$$

$$\frac{x \stackrel{f}{\twoheadrightarrow} y \qquad x \stackrel{f}{\twoheadrightarrow} z}{y \sim z}\ \langle k_r \rangle \qquad \frac{x \stackrel{f}{\twoheadrightarrow} y}{x \sim x}\ \langle a_{r_1} \rangle$$

DMBI Proof theory - Tableaux method

Constraint closure

- Rules that product μ-constraints:

$$\frac{x \sim x}{x \overset{1_a}{\twoheadrightarrow} x}\ \langle 1_\mu \rangle \qquad \frac{x \overset{f}{\twoheadrightarrow} y \qquad y \overset{g}{\twoheadrightarrow} z}{x \overset{fg}{\twoheadrightarrow} z}\ \langle t_\mu \rangle$$

$$\frac{x \overset{f}{\twoheadrightarrow} y \qquad x \sim x'}{x' \overset{f}{\twoheadrightarrow} y}\ \langle k_{\mu_1} \rangle \qquad \frac{x \overset{f}{\twoheadrightarrow} y \qquad y \sim y'}{x \overset{f}{\twoheadrightarrow} y'}\ \langle k_{\mu_2} \rangle$$

- Rules that product transition constraints:

$$\frac{u \overset{f}{\rightarrowtail} v}{u \overset{1_a}{\rightarrowtail} u}\ \langle 1_{t_1} \rangle \qquad \frac{u \overset{f}{\rightarrowtail} v}{v \overset{1_a}{\rightarrowtail} v}\ \langle 1_{t_2} \rangle \qquad \frac{u \overset{f}{\rightarrowtail} v \qquad v \overset{g}{\rightarrowtail} w}{u \overset{fg}{\rightarrowtail} w}\ \langle t_t \rangle$$

DMBI Tableaux method

Modal rules

- **Assertion** rules:
  Introduction of new labels and assertions (or constraints)

$$\frac{\mathbb{T}\langle f\rangle\phi : (x,u) \in \mathcal{F}}{\langle\{\mathbb{T}\phi : (c_i, l_i)\}, \{x \overset{f}{\twoheadrightarrow} c_i, u \overset{f}{\rightarrowtail} l_i\}\rangle} \ \langle\mathbb{T}\langle - \rangle\rangle$$

$$\frac{\mathbb{T}\Diamond\phi : (x,u) \in \mathcal{F}}{\langle\{\mathbb{T}\phi : (c_i, l_i)\}, \{x \overset{d_i}{\twoheadrightarrow} c_i, u \overset{d_i}{\rightarrowtail} l_i\}\rangle} \ \langle\mathbb{T}\Diamond\rangle$$

- **Requirement** rules:
  Conditions that must be verified in the closure of constraints

$$\frac{\mathbb{F}\langle f\rangle\phi : (x,u) \in \mathcal{F} \text{ and } x \overset{f}{\twoheadrightarrow} y \in \overline{\mathcal{C}} \text{ and } u \overset{f}{\rightarrowtail} v \in \overline{\mathcal{C}}}{\langle\mathbb{F}\phi : (y,v), \emptyset\rangle} \ \langle\mathbb{F}\langle - \rangle\rangle$$

$$\frac{\mathbb{F}\Diamond\phi : (x,u) \in \mathcal{F} \text{ and } x \overset{f}{\twoheadrightarrow} y \in \overline{\mathcal{C}} \text{ and } u \overset{f}{\rightarrowtail} v \in \overline{\mathcal{C}}}{\langle\{\mathbb{F}\phi : (y,v)\}, \emptyset\rangle} \ \langle\mathbb{F}\Diamond\rangle$$

DMBI Tableaux method

Definition: closed branch

A CSS (branch) $\langle \mathcal{F}, \mathcal{C} \rangle$ is **closed** iff one of these conditions holds:

- $\mathbb{T}\phi : (x, u) \in \mathcal{F}$, $\mathbb{F}\phi : (y, u) \in \mathcal{F}$ and $x \sim y \in \overline{\mathcal{C}}$
- $\mathbb{F}\mathrm{I} : (x, u) \in \mathcal{F}$ and $1_r \sim x \in \overline{\mathcal{C}}$
- $\mathbb{T}\bot : (x, u) \in \mathcal{F}$

Definition: μ-proof

A *μ-proof* for a formula ϕ is a μ-tableau for ϕ which is closed.

Theorem: soundness

If there exists a μ-proof for a formula ϕ then ϕ is valid.

Theorem: completeness

If a formula ϕ is valid then there is a μ-proof for ϕ.

DMBI Tableaux method - an example

- How to prove $\phi \equiv (\mathrm{I} \mathrel{-\!\!*} \langle a \rangle \langle b \rangle P) \to \Diamond P$?

Step 1: Initialization

$[\mathcal{F}]$	$[\mathcal{C}]$	
$\mathbb{F}(\mathrm{I} \mathrel{-\!\!*} \langle a \rangle \langle b \rangle P) \to \Diamond P : (c_1, l_1)$	$c_1 \sim c_1$	$l_1 \overset{1_a}{\rightarrowtail} l_1$

DMBI Tableaux method - an example

$$[\mathcal{F}]$$

$$\mathbb{F}(\mathrm{I} \twoheadrightarrow \langle a \rangle \langle b \rangle P) \to \Diamond P : (c_1, l_1)$$

$$[\mathcal{C}]$$

$$c_1 \sim c_1 \qquad l_1 \xrightarrowtail{1_a} l_1$$

DMBI Tableaux method - an example

$[\mathcal{F}]$

$$\surd_1\ \mathbb{F}(\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P) \to \Diamond P : (c_1, l_1)$$

$$\mathbb{T}\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P : (c_1, l_1)$$
$$\mathbb{F}\Diamond P : (c_1, l_1)$$

$[\mathcal{C}]$

$$c_1 \sim c_1 \qquad l_1 \xrightarrow{1_a} l_1$$

$$\frac{\mathbb{F}\phi \to \psi : (x, u) \in \mathcal{F}}{\langle\{\mathbb{T}\phi : (x, u), \mathbb{F}\psi : (x, u)\}, \emptyset\rangle}\ \langle\mathbb{F} \to\rangle$$

DMBI Tableaux method - an example

$[\mathcal{F}]$

$\surd_1\ \mathbb{F}(\mathrm{I} \twoheadrightarrow \langle a\rangle\langle b\rangle P) \to \Diamond P : (c_1, l_1)$

$\surd_2\ \mathbb{T}\mathrm{I} \twoheadrightarrow \langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\mathbb{F}\Diamond P : (c_1, l_1)$

$\mathbb{F}\mathrm{I} : (1_r, l_1) \qquad \mathbb{T}\langle a\rangle\langle b\rangle P : (c_1, l_1)$

$[\mathcal{C}]$

$c_1 \sim c_1 \qquad l_1 \xrightarrow{1_a} l_1$

$$\frac{\mathbb{T}\phi \twoheadrightarrow \psi : (x, u) \in \mathcal{F} \text{ and } xy \sim xy \in \overline{\mathcal{C}}}{\langle\{\mathbb{F}\phi : (y, u)\}, \emptyset\rangle \ \mid\ \langle\{\mathbb{T}\psi : (xy, u)\}, \emptyset\rangle}\ \langle\mathbb{T}\twoheadrightarrow\rangle$$

Remark: $c_1 \circ 1_r = c_1$

DMBI Tableaux method - an example

$[\mathcal{F}]$

$\surd_1\ \mathbb{F}(\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P) \to \Diamond P : (c_1, l_1)$

$\surd_2\ \mathbb{T}\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\mathbb{F}\Diamond P : (c_1, l_1)$

$\mathbb{F}\mathrm{I} : (1_r, l_1)$ $\quad$ $\surd_3\ \mathbb{T}\langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\mathbb{T}\langle b\rangle P : (c_2, l_2)$

$[\mathcal{C}]$

$c_1 \sim c_1 \qquad l_1 \overset{1_a}{\rightarrowtail} l_1$

$c_1 \overset{a}{\twoheadrightarrow} c_2 \qquad l_1 \overset{a}{\rightarrowtail} l_2$

$$\frac{\mathbb{T}\langle f\rangle\phi : (x, u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (c_i, l_i)\}, \{x \overset{f}{\twoheadrightarrow} c_i, u \overset{f}{\rightarrowtail} l_i\}\rangle}\ \langle \mathbb{T}\langle - \rangle \rangle$$

DMBI Tableaux method - an example

$$[\mathcal{F}]$$

$$\surd_1 \; \mathbb{F}(\mathrm{I} \twoheadrightarrow \langle a \rangle \langle b \rangle P) \to \Diamond P : (c_1, l_1)$$

$$\surd_2 \; \mathbb{T}\mathrm{I} \twoheadrightarrow \langle a \rangle \langle b \rangle P : (c_1, l_1)$$

$$\mathbb{F}\Diamond P : (c_1, l_1)$$

$$\mathbb{F}\mathrm{I} : (1_r, l_1) \qquad \surd_3 \; \mathbb{T}\langle a \rangle \langle b \rangle P : (c_1, l_1)$$

$$\surd_4 \; \mathbb{T}\langle b \rangle P : (c_2, l_2)$$

$$\mathbb{T}P : (c_3, l_3)$$

$$[\mathcal{C}]$$

$$c_1 \sim c_1 \qquad l_1 \overset{1_a}{\rightarrowtail} l_1$$

$$c_1 \overset{a}{\twoheadrightarrow} c_2 \qquad l_1 \overset{a}{\rightarrowtail} l_2$$

$$c_2 \overset{b}{\twoheadrightarrow} c_3 \qquad l_2 \overset{b}{\rightarrowtail} l_3$$

$$\frac{\mathbb{T}\langle f \rangle \phi : (x, u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (c_i, l_i)\}, \{x \overset{f}{\twoheadrightarrow} c_i, u \overset{f}{\rightarrowtail} l_i\} \rangle} \; \langle \mathbb{T}\langle - \rangle \rangle$$

DMBI Tableaux method - an example

$[\mathcal{F}]$

$\surd_1\ \mathbb{F}(\mathrm{I} \twoheadrightarrow\!\!* \langle a\rangle\langle b\rangle P) \to \Diamond P : (c_1, l_1)$

$\surd_2\ \mathbb{T}\mathrm{I} \twoheadrightarrow\!\!* \langle a\rangle\langle b\rangle P : (c_1, l_1)$

${\color{red}\surd_5}\ \mathbb{F}\Diamond P : (c_1, l_1)$

$\mathbb{F}\mathrm{I} : (1_r, l_1)$ | $\surd_3\ \mathbb{T}\langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\surd_4\ \mathbb{T}\langle b\rangle P : (c_2, l_2)$

$\mathbb{T}P : (c_3, l_3)$

$\mathbb{F}P : (c_3, l_3)$

$[\mathcal{C}]$

$c_1 \sim c_1 \qquad l_1 \xrightarrow{1_a} l_1$

$c_1 \overset{a}{\twoheadrightarrow} c_2 \qquad l_1 \overset{a}{\rightarrowtail} l_2$

$c_2 \overset{b}{\twoheadrightarrow} c_3 \qquad l_2 \overset{b}{\rightarrowtail} l_3$

$$\frac{\mathbb{F}\Diamond\phi : (x, u) \in \mathcal{F} \text{ and } x \overset{f}{\twoheadrightarrow} y \in \overline{\mathcal{C}} \text{ and } u \overset{f}{\rightarrowtail} v \in \overline{\mathcal{C}}}{\langle\{\mathbb{F}\phi : (y, v)\}, \emptyset\rangle}\ \langle\mathbb{F}\Diamond\rangle$$

$$\frac{c_1 \overset{a}{\twoheadrightarrow} c_2 \qquad c_2 \overset{b}{\twoheadrightarrow} c_3}{c_1 \overset{ab}{\twoheadrightarrow} c_3}\ \langle t_\mu\rangle \qquad \frac{l_1 \overset{a}{\rightarrowtail} l_2 \qquad l_2 \overset{b}{\rightarrowtail} l_3}{l_1 \overset{ab}{\rightarrowtail} l_3}\ \langle t_t\rangle$$

DMBI Tableaux method - an example

Step 2: Application of rules

$[\mathcal{F}]$

$\surd_1\ \mathbb{F}(\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P) \rightarrow \Diamond P : (c_1, l_1)$

$\surd_2\ \mathbb{T}\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\surd_5\ \mathbb{F}\Diamond P : (c_1, l_1)$

$\mathbb{F}\mathrm{I} : (1_r, l_1)$ | $\surd_3\ \mathbb{T}\langle a\rangle\langle b\rangle P : (c_1, l_1)$

$\times$ | $\surd_4\ \mathbb{T}\langle b\rangle P : (c_2, l_2)$

$\mathbb{T}P : (c_3, l_3)$

$\mathbb{F}P : (c_3, l_3)$

$\times$

$[\mathcal{C}]$

$c_1 \sim c_1 \qquad l_1 \overset{1_a}{\rightarrowtail} l_1$

$c_1 \overset{a}{\twoheadrightarrow} c_2 \qquad l_1 \overset{a}{\rightarrowtail} l_2$

$c_2 \overset{b}{\twoheadrightarrow} c_3 \qquad l_2 \overset{b}{\rightarrowtail} l_3$

The formula $(\mathrm{I} \mathrel{-\!\!*} \langle a\rangle\langle b\rangle P) \rightarrow \Diamond P$ is valid

Plan

Counter-model extraction

Definition: Hintikka CSS

A Hintikka CSS $\langle \mathcal{F}, \mathcal{C}_r \rangle \mathcal{C}_s$ is a unclosed branch such that "all information has been extracted":

1 $\mathbb{T}\phi : (x, u) \notin \mathcal{F}$ or $\mathbb{F}\phi : (y, u) \notin \mathcal{F}$ or $x \sim y \notin \overline{\mathcal{C}}$

2-12 ...

13 If $\mathbb{T}\Diamond\phi : (x, u) \in \mathcal{F}$ then $\exists y \in L_r$, $\exists f \in L_a$, $\exists v \in L_s$, $x \overset{f}{\twoheadrightarrow} y \in \overline{\mathcal{C}}$ and $u \overset{f}{\rightarrowtail} v \in \overline{\mathcal{C}}$ and $\mathbb{T}\phi : (y, v) \in \mathcal{F}$

14 If $\mathbb{F}\Diamond\phi : (x, u) \in \mathcal{F}$ then $\forall y \in L_r$, $\forall f \in L_a$, $\forall v \in L_s$, $(x \overset{f}{\twoheadrightarrow} y \in \overline{\mathcal{C}}$ and $u \overset{f}{\rightarrowtail} v \in \overline{\mathcal{C}}) \Rightarrow \mathbb{F}\phi : (y, v) \in \mathcal{F}$

Lemma: counter-model extraction

A counter-model can be extracted from a Hintikka branch.

Counter-model extraction

Function Ω

Let $\langle \mathcal{F}, \mathcal{C} \rangle$ be a Hintikka CSS. $\Omega(\langle \mathcal{F}, \mathcal{C} \rangle) = (\mathcal{M}, [\![\cdot]\!], |\cdot|, \vDash_{\mathcal{K}})$, such that:

- $R = \mathcal{D}_r(\overline{\mathcal{C}}) / \sim \qquad S = \mathcal{A}_s(\mathcal{C}) \qquad Act = \mathcal{D}_a(\overline{\mathcal{C}}) \cup \{\alpha\}$ (where $\alpha \notin \mathcal{D}_a(\overline{\mathcal{C}})$)
- $e = [1_r]$
- $1 = 1_a$
- $[x] \bullet [y] = [x \circ y]$
- $\mu(a, [x]) = \begin{cases} \uparrow & \text{if } \{y \mid x \overset{a}{\twoheadrightarrow} y \in \overline{\mathcal{C}}\} = \emptyset \\ \{y \mid x \overset{a}{\twoheadrightarrow} y \in \overline{\mathcal{C}}\} & \text{otherwise} \end{cases}$
- $s_1 \,\|f\rangle\!\rangle\, s_2$ iff $s_1 \overset{f}{\rightarrowtail} s_2 \in \overline{\mathcal{C}}$
- For all $a_1, a_2 \in Act$, $a_1 \odot a_2 = \begin{cases} a_1 \centerdot a_2 & \text{if } a_1 \centerdot a_2 \in \mathcal{D}_a(\overline{\mathcal{C}}) \\ \alpha & \text{otherwise} \end{cases}$
- For all $a \in S_{Act}$, $|a| = \begin{cases} a & \text{if } a \in \mathcal{D}_a(\overline{\mathcal{C}}) \\ \alpha & \text{otherwise} \end{cases}$
- $([x], s) \in [\![P]\!]$ iff $\exists y \in L_r, x \in [y]$ and $\mathbb{T}P : (y, s) \in \mathcal{F}$

Plan

Conclusions

A modal extension of **BBI** for resource transformations

- That captures resource transformations ($\approx$ **MBI**)
- That includes modalities $\Diamond$ and $\Box$ ($\approx$ **DBI**)
- That has a sound and complete calculus with a countermodel extraction method
- Some Questions:
  - How to model concurrent processes (protocols or Web services)?
  - Will the concurrent process simulation allow us to model it?
  - Should $*$ separate only resources or resources and processes?

Future works

- Our goals:
  - To study concurrent process simulation in **DMBI**
  - To define a language L to model systems, like Demos2k (HP Labs 2008) or Core Gnosis (HP Labs 2010), which does only simulation
  - To study satisfiability in **DMBI** $\Rightarrow$ by using the tableau method
  - To provide a decision procedure (bounds on number of resources, fragments of **DMBI**)
  - To model protocol or web service problems: are there new properties that we can express with **DMBI**?

Future works

Example 1: mutual exclusion

```
AtomicResources = {J}

AtomicAction aC  = e -> e;
AtomicAction aNC = e -> e;
AtomicAction aP  = J -> e;
AtomicAction aV  = e -> J;

Process p {
  s1 = aNC:s1 + aP:s2;
  s2 = aC:s2 + aV:s1;
}

init = (J, p.s1 # p.s1);

check [] [aC#aC] F;   // F = bottom
check ! <> (J*J*T);   // T = top
```

Example 2: producer / consumer

```
AtomicResources = {R}

AtomicAction p  = e -> R;
AtomicAction nP = e -> e;
AtomicAction c  = R -> e;
AtomicAction nC = e -> e;

Proc producer {
  s1 = p:s1 + nP:s1;
}

Proc consummer {
  s1 = c:s1 + nC:s1;
}

init = (e, producer.s1 # consummer.s1);

check [](I -> !<nP#c>T);
```