A Modal BI Logic for Dynamic Resource Properties ⋆

J.R. Courtault and D. Galmiche

Université de Lorraine – LORIA UMR 7503
Campus Scientifique, BP 239
Vandœuvre-lès-Nancy, France

Abstract. The logic of Bunched implications (BI) and its variants or extensions provide a powerful framework to deal with resources having static properties. In this paper, we propose a modal extension of BI logic, called DBI, which allows us to deal with dynamic resource properties. After defining a Kripke semantics for DBI, we illustrate the interest of DBI for expressing some dynamic properties and then we propose a labelled tableaux calculus for this logic. This calculus is proved sound and complete w.r.t. the Kripke semantics. Moreover, we also give a method for countermodel generation in this logic.

1 Introduction

The notion of *resource* is an important notion in computer science. The location, ownership, access to and, indeed, consumption of, resources are central concerns in the design of systems, such as networks, and in the design of programs, which access memory and manipulate data structures like pointers. We are interested in studying such notions on resources through logics with an emphasis on usable semantics and proof-theory. In this context we can mention Linear Logic (LL) [5] that focuses on resource consumption and the logic of Bunched Implications (BI) [13] that mainly focuses on resource sharing and separation. The BI logic and its variants, like Boolean BI (BBI) [11,13], can be seen as the logical kernel of so-called separation logics, that provides a concrete way of understanding the connectives in the context of program verification [7,14]. Some recent results on BI and BBI concern new semantics [4], proof-search with labelled tableaux and resource graphs [3,4] and (un)decidability of these logics [4,9]. Some extensions or refinements have led to separation logics, like BI's pointer logic (PL) [7] that allows us to express properties on pointers or BiLoc [1] that is based on resource trees and captures the notion of place. In this context MBI logic [12] extends BI with modalities and a calculus à la Hennessy-Milner [10] dealing with processes and resources.
We can remark that two kinds of dynamic are captured by BI, BBI and their extensions. On the one hand, there are logics that transform resources into other resources, which is a first kind of dynamic. On the other hand, there are logics where properties of resources can change (called here dynamic properties) or not (called here static properties). For example, in BI logic the resource properties are static because if a resource satisfies a property, it will always satisfies this property. The dynamic, that corresponds

⋆ This work is supported by the ANR grant DynRes on Dynamic Resources and Separation and Update Logics (project no. ANR-11-BS02-011).

to the transformation of resources, is captured in LL by proofs and in PL by a calculus à la Hoare [6]. Moreover in MBI, the dynamic is also based on resource transformation because of a calculus à la Hennessy-Milner with judgements of the form $R, E \xrightarrow{a} R', E'$, which means that a process E performs an action a on a resource R in order to obtain a resource R' and then becomes a process E'. But the modalities à la Hennessy-Milner can only express properties on R' and E', directly at the next state, but not on any reachable resource and process (or state), knowing that reachable means after performing any action.
In this paper, we are interested in expressing some dynamic properties on resources directly at level of formulae, on future states (and not only on the next ones) and in dealing with interacting systems. Then we define a modal extension of BI, called DBI (Dynamic Bunched Implications logic), in order to model some dynamic properties of resources. We define a Kripke semantics for this logic, which is an extension of Kripke semantics for BI with state constraints (a set of states with a preorder) introduced in addition to resource constraints. We also give a labelled tableaux calculus in the spirit of works on BI logic [3,4] but dealing with both resource graphs and state graphs. This calculus is proved sound and complete w.r.t. this semantics, with generation of countermodels in case of non-validity in DBI.

2 The DBI logic

BI logic is a logic that expresses sharing and separation properties on resources [11,13]. We present here a modal extension of BI, called DBI, which allows us to express some dynamic properties on resources. The language $\mathcal{L}$ of DBI is obtained by adding two modalities $\Box$ and $\Diamond$ to the BI language [13].
Let *Prop* be a countable set of propositional symbols, the language $\mathcal{L}$ of DBI is defined as follows, where $p \in Prop$:

$$X ::= p \mid \top \mid \bot \mid \mathrm{I} \mid X \wedge X \mid X \vee X \mid X \rightarrow X \mid X * X \mid X \mathrel{-\!\!*} X \mid \Diamond X \mid \Box X$$

The negation is defined by: $\neg X \equiv X \rightarrow \bot$. We now define a Kripke semantics that can be seen as an extension of the Kripke semantics of BI [4] based on a resource monoid. In the case of DBI we consider a dynamic resource monoid with an explicit inconsistency, and also a preorder set of states with an accessibility relation between states.

Definition 1 (Dynamic resource monoid). *A* dynamic resource monoid *is a structure* $\mathcal{M} = (R, \bullet, e, \pi, \sqsubseteq, S, \preceq)$ *such that:*

- *R is a set of* resources *and S is a set of* states
- *$e \in R$ and $\pi \in R$*
- *$\bullet : R \times R \rightarrow R$ such that:*
  - *Neutral element:* $\forall r \in R$, $r \bullet e = e \bullet r = r$
  - *Associativity:* $\forall r_1, r_2, r_3 \in R$, $r_1 \bullet (r_2 \bullet r_3) = (r_1 \bullet r_2) \bullet r_3$
  - *Commutativity:* $\forall r_1, r_2 \in R$, $r_1 \bullet r_2 = r_2 \bullet r_1$
- *$\sqsubseteq \subseteq R \times R$ is a preorder (on resources):*
  - *Reflexivity:* $\forall r \in R$, $r \sqsubseteq r$

- *Transitivity:* $\forall r_1, r_2, r_3 \in R$, *if* $r_1 \sqsubseteq r_2$ *and* $r_2 \sqsubseteq r_3$ *then* $r_1 \sqsubseteq r_3$

– $\pi \in R$ *is the greatest element:* $\forall r \in R$, $r \sqsubseteq \pi$ *and* $\forall r \in R$, $r \bullet \pi = \pi$.
– $\preceq \subseteq S \times S$ *is a preorder (on states)*
– *Compatibility (P):* $\forall r_1, r_2, r_3 \in R$, *if* $r_1 \sqsubseteq r_2$ *then* $r_1 \bullet r_3 \sqsubseteq r_2 \bullet r_3$

We note $\mathbb{P}(E)$ the powerset of the set E, namely the set of sets built from E. We call e the *unit resource* (empty resource), π the *inconsistent resource* and $\bullet$ the *resource composition*. A preordered set $(S, \preceq)$ is added to the Kripke's BI semantics with S that can be viewed as the states of a system and $\preceq$ as the accessibility (through transitions) of states of the system.

Definition 2 (Dynamic interpretation). *A* dynamic interpretation *is a function* $[\![\cdot]\!] : Prop \to \mathbb{P}(R \times S)$, *that verifies the following properties, for any* $s \in S$ *and* $p \in Prop$*:*

– *Monotonicity (K):* $\forall r, r' \in R$ *such that* $r \sqsubseteq r'$, *if* $(r, s) \in [\![p]\!]$ *then* $(r', s) \in [\![p]\!]$
– *Inconsistency (BC):* $\forall r \in R$ *such that* $\pi \sqsubseteq r$, $(r, s) \in [\![p]\!]$

As we see the dynamic interpretation makes the resource properties non static: the interpretation of a propositional symbol is not only a set of resources (as BI), but a set of pairs of resources and states.

Definition 3 (Dynamic resource model). *A* dynamic resource model *is a triple* $\mathcal{K} = (\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ *such that* $\mathcal{M}$ *is a dynamic resource monoid,* $[\![\cdot]\!]$ *is a dynamic interpretation and* $\vDash_{\mathcal{K}}$ *is a forcing relation on* $R \times S \times \mathcal{L}$ *defined as follows:*

– $r, s \vDash_{\mathcal{K}} p$ *iff* $(r, s) \in [\![p]\!]$
– $r, s \vDash_{\mathcal{K}} \mathrm{I}$ *iff* $e \sqsubseteq r$
– $r, s \vDash_{\mathcal{K}} \top$ *always*
– $r, s \vDash_{\mathcal{K}} \bot$ *iff* $\pi \sqsubseteq r$
– $r, s \vDash_{\mathcal{K}} \phi \wedge \psi$ *iff* $r, s \vDash_{\mathcal{K}} \phi$ *and* $r, s \vDash_{\mathcal{K}} \psi$
– $r, s \vDash_{\mathcal{K}} \phi \vee \psi$ *iff* $r, s \vDash_{\mathcal{K}} \phi$ *or* $r, s \vDash_{\mathcal{K}} \psi$
– $r, s \vDash_{\mathcal{K}} \phi \to \psi$ *iff* $\forall r' \in R \cdot (r \sqsubseteq r'$ *and* $r', s \vDash_{\mathcal{K}} \phi) \Rightarrow r', s \vDash_{\mathcal{K}} \psi$
– $r, s \vDash_{\mathcal{K}} \phi * \psi$ *iff* $\exists r', r'' \in R \cdot r' \bullet r'' \sqsubseteq r$ *and* $r', s \vDash_{\mathcal{K}} \phi$ *and* $r'', s \vDash_{\mathcal{K}} \psi$
– $r, s \vDash_{\mathcal{K}} \phi \mathrel{-\!\!*} \psi$ *iff* $\forall r' \in R \cdot r', s \vDash_{\mathcal{K}} \phi \Rightarrow r \bullet r', s \vDash_{\mathcal{K}} \psi$
– $r, s \vDash_{\mathcal{K}} \Diamond\phi$ *iff* $\exists s' \in S \cdot s \preceq s'$ *and* $r, s' \vDash_{\mathcal{K}} \phi$
– $r, s \vDash_{\mathcal{K}} \Box\phi$ *iff* $\forall s' \in S \cdot s \preceq s' \Rightarrow r, s' \vDash_{\mathcal{K}} \phi$

The definition of the forcing relation is an extension of the BI forcing realtion with the cases for $\Box$ and $\Diamond$. For instance $r, s \vDash_{\mathcal{K}} \Diamond\phi$ means that a resource r at state s satisfies $\Diamond\phi$ if a state s' can be reached from the state s ($s \preceq s'$) such that r in state s' satisfies ϕ ($r, s' \vDash_{\mathcal{K}} \phi$). Now we define the notion of validity.

Definition 4 (Validity). *A formula* ϕ *is* valid, *denoted* $\vDash \phi$, *if and only if* $e, s \vDash_{\mathcal{K}} \phi$ *for all dynamic resource models* $\mathcal{K}$ *(and all states* $s \in S$*).*
The notation $\phi \vDash \psi$ *means that for all resources* r *and all states* s *of any dynamic resource model* $\mathcal{K}$, *if* $r, s \vDash_{\mathcal{K}} \phi$ *then* $r, s \vDash_{\mathcal{K}} \psi$.

We give two lemmas that hold for all dynamic resource models $\mathcal{K}$, all formulae ϕ, all resources $r, r' \in R$ and all states $s' \in S$.

Lemma 1 (Monotonicity). *If* $r, s \vDash_{\mathcal{K}} \phi$ *and* $r \sqsubseteq r'$ *then* $r', s \vDash_{\mathcal{K}} \phi$.

Lemma 2 (Inconsistency). *We have* $\pi, s \vDash_{\mathcal{K}} \phi$.

3 Expressiveness of DBI

We have previously introduced a semantics for modelling resources having dynamic properties. In this section we emphasize the interest of this modal extension of BI by illustrating it through some simple examples.
The first example deals with the management of resources with dynamic properties. In BI logic the propositional symbols are considered as static descriptions/properties of resources. But, we know that resource properties are not always static. For example, if we consider the price of gold and silver, it is a dynamic property depending not only on the resource. Let us denote r_g the resource "one ounce of gold" and r_s the resource "one ounce of silver". Propositional symbols P_{g_y} and P_{s_y} are prices of r_g and r_s on January 1st of the year y. Moreover, s_y denotes the state of the market on January 1st of the year y. With DBI we are able to express the evolution of the silver and gold price:

$$r_g \bullet r_s, s_{1970} \vDash_{\mathcal{K}} (P_{g_{1970}} * P_{s_{1970}}) \wedge \Diamond(P_{g_{2012}} * P_{s_{2012}})$$

It means that on January 1st of the year 1970 (s_{1970}), a resource composed by one ounce of gold and one ounce of silver ($r_g \bullet r_s$) has two properties: it could be decomposed into two resources respectively satisfying the properties $P_{g_{1970}}$ and $P_{s_{1970}}$ ($P_{g_{1970}} * P_{s_{1970}}$) and, in a future state, it could be decomposed into two resources respectively satisfying the properties $P_{g_{2012}}$ and $P_{s_{2012}}$ ($P_{g_{2012}} * P_{s_{2012}}$).

The second example illustrates how with DBI and a dynamic resource monoid we can deal with properties on interacting systems. A dynamic resource monoid can be viewed as two interacting systems. Indeed a resource monoid can model a first system, where resources are states of this system and the preoder on resources is the state reachability of this system [2]. Furthermore, the dynamic part of a dynamic resource monoid (set of states with a preorder), can be viewed as an automaton and easily models a second system. Moreover, the dynamic interpretation can be viewed as the result of the interaction of these systems. For example, $(r,s) \in [\![p]\!]$ can express that, if a first system is in state r and a second system is in state s then their interaction satisfies the property p. Here the word *interaction* does not mean that one of these systems influences the second one: the preorder on resources does not depend on states and the preorder on states does not depend on resources. Then the interaction $(r,s) \in [\![p]\!]$ means that there are two free (non influencing) systems which can perform together an action, which satisfies the property p if the first system is in state r and the second system is in state s.
Let us consider a message sent in a network and modelled with a resource monoid. We consider only five states (resources) $R = \{e, m_{sent}, m_{passing}, m_{delivered}, \pi\}$, where e is the state with no message, π is the state with an error that occurs in the system, m_{sent} is the state where the message is sent, $m_{passing}$ is the state where the message is passing in transit and $m_{delivered}$ is the state where the message is delivered. The relation $\sqsubseteq$, where reflexivity and transitivity are not represented, is:

In a first step, there is no message (e). Then the message is created and sent (m_{sent}). In a third step, it is passing in transit ($m_{passing}$) and then, in a fourth step, it is delivered ($m_{delivered}$). As we can remark, $m_{passing} \sqsubseteq m_{sent}$, but $m_{passing}$ is the next state of m_{sent} and it is not a mistake. As m_{sent} can reach $m_{passing}$ then we aim the properties of $m_{passing}$ to be satisfied by the resource m_{sent}. In other words, if a resource r satisfies a property p, then all resources that can reach r satisfy p. This is the property (K) of Definition 2. In this example, we only consider one message and then we define $\bullet$ by $(e \bullet r = r)$ and $(r \bullet r' = \pi$ if $r \neq e$ and $r' \neq e)$, but it is possible to consider states composed by more than one message. We remark that π is the biggest resource (by definition of dynamic resource monoid), so when an error occurs (π), all states are reachable: it is considered that when an error occurs, it is impossible to predict the behavior of the system.
Now we define the following service as a second system, where reflexivity and transitivity of $\preceq$ are not represented. It contains four states $S = \{s_0, s_1, s_2, s_3\}$ with s_0 as initial state and in the state s_3 our service reads the delivered messages.

Having defined a dynamic resource monoid we are able to express that when the message is sent, it is possible that our service read this message, that is: $m_{sent}, s_0 \vDash_{\mathcal{K}} \Diamond P_{m_{read}}$, where $P_{m_{read}}$ is the propositional symbol "message read" that occurs when m is delivered and the service is in state s_3: $[\![P_{m_{read}}]\!] = \{(r, s_3) \mid m_{delivered} \sqsubseteq r\}$.
We have $m_{delivered}, s_3 \vDash_{\mathcal{K}} P_{m_{read}}$. As $s_0 \preceq s_3$ then $m_{delivered}, s_0 \vDash_{\mathcal{K}} \Diamond P_{m_{read}}$ (the DBI modalities encode the reachability of states). As m_{sent} can reach $m_{delivered}$ ($m_{delivered} \sqsubseteq m_{sent}$) then $m_{sent}, s_0 \vDash_{\mathcal{K}} \Diamond P_{m_{read}}$ (DBI monotonicity encodes the resource reachability).

4 A proof system for DBI

In this section, we propose a proof system for DBI, in the spirit of previous works on labelled proof system for BI with resource graphs [4]. We introduce some rules to deal with modalities and also the notions of state labels and constraints, in order to capture some dynamic aspects.

4.1 Labels for resources and states

In labelled tableaux method for BI [4], there are *labels* and *constraints* in order to capture some semantic information inside the proof system. Labels are related to the resource set (R), a label composition is related to the resource composition ($\bullet$) and relations on labels named *label constraints* are related to $\sqsubseteq$. In DBI, the resource monoids are dynamic and then there are two sets (for resources and states) and two relations (on resources and states). Thus we introduce a new kind of labels and constraints to deal with states. Let us now define labels and constraints for DBI.

Definition 5 (Resource labels). *L_r is a set of* resource labels *built from a constant* 1*, an infinite countable set of constants $\gamma_r = \{c_1, c_2, ...\}$ and a function denoted $\circ$,*

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

where $c_i \in \gamma_r$. Moreover $\circ$ is a function on L_r that is associative, commutative and 1 *is its unit. A* resource constraint *is an expression of the form $x \leq y$ where x and y are resource labels.*

For example the resource label $c_1 \circ 1 \circ c_2 \circ c_1$ is equal to the resource label $c_1 \circ c_1 \circ c_2$. We denote xy the resource label $x \circ y$. Moreover we say that x is a *resource sub-label* of y if and only if there exists z such that $x \circ z = y$. The set of resource sub-labels of x is denoted $\mathcal{E}(x)$.

Definition 6 (State labels). *L_s is an infinite countable set of* state labels *($L_s = \{l_1, l_2, ...\}$). A* state constraint *on such labels is an expression of the form $x \lhd y$, where x and y are state labels.*

Definition 7 (Domain). *Let $\mathcal{C}_r$ be a resource constraints set, the* domain *of $\mathcal{C}_r$, denoted $\mathcal{D}_r(\mathcal{C}_r)$, is the set of all resource sub-labels appearing in $\mathcal{C}_r$. In particular: $\mathcal{D}_r(\mathcal{C}_r) = \bigcup_{x \leq y \in \mathcal{C}_r} (\mathcal{E}(x) \cup \mathcal{E}(y))$.*

Definition 8 (Alphabet). *The* alphabet *of a set of resource / state constraints is the set of all label constants appearing in $\mathcal{C}_r$ / $\mathcal{C}_s$.*
In particular we have $\mathcal{A}_r(\mathcal{C}_r) = \gamma_r \cap \mathcal{D}_r(\mathcal{C}_r)$ and $\mathcal{A}_s(\mathcal{C}_s) = \bigcup_{u \lhd v \in \mathcal{C}_s} \{u, v\}$.

We can remark that $\sqsubseteq$ is reflexive, transitive and compatible. Moreover, $\preceq$ is reflexive and transitive. These properties have to be captured by the constraint sets. For that we introduce a notion of closure of constraints.

Definition 9 (Closure of resource constraints). *Let $\mathcal{C}_r$ be a set of resource constraints, the* closure *of $\mathcal{C}_r$ (denoted $\overline{\mathcal{C}_r}$) is the least relation closed under the following rules such that $\mathcal{C}_r \subseteq \overline{\mathcal{C}_r}$*

$$\frac{x \leq y \qquad y \leq z}{x \leq z}\ \langle t_r \rangle \qquad \frac{xy \leq xy}{x \leq x}\ \langle d_r \rangle \qquad \frac{ky \leq ky \qquad x \leq y}{kx \leq ky}\ \langle c_r \rangle \qquad \frac{x \leq y}{x \leq x}\ \langle l_r \rangle \qquad \frac{x \leq y}{y \leq y}\ \langle r_r \rangle$$

We can remark that as these rules do not introduce new resource label constants, then $\mathcal{A}_r(\mathcal{C}_r) = \mathcal{A}_r(\overline{\mathcal{C}_r})$.

Definition 10 (Closure of state constraints). *Let $\mathcal{C}_s$ be a set of state constraints, the* closure *of $\mathcal{C}_s$ (denoted $\overline{\mathcal{C}_s}$) is the least relation closed under the following rules such that $\mathcal{C}_s \subseteq \overline{\mathcal{C}_s}$:*

$$\frac{x \lhd y}{x \lhd x}\ \langle l_s \rangle \qquad \frac{x \lhd y}{y \lhd y}\ \langle r_s \rangle \qquad \frac{x \lhd y \qquad y \lhd z}{x \lhd z}\ \langle t_s \rangle$$

As illustration we consider $\mathcal{C}_s = \{l_1 \lhd l_2, l_2 \lhd l_3, l_3 \lhd l_4\}$. We have $l_1 \lhd l_2 \in \overline{\mathcal{C}_s}$ because $\mathcal{C}_s \subseteq \overline{\mathcal{C}_s}$ and we have $l_1 \lhd l_4 \in \overline{\mathcal{C}_s}$ because

$$\dfrac{\dfrac{l_1 \lhd l_2 \qquad l_2 \lhd l_3}{l_1 \lhd l_3}\ \langle t_s \rangle \qquad l_3 \lhd l_4}{l_1 \lhd l_4}\ \langle t_s \rangle$$

Proposition 1. *Let $\mathcal{C}_r$ be a set of resource constraints, the following properties hold:*

1. *If $kx \leq y \in \overline{\mathcal{C}_r}$ then $x \leq x \in \overline{\mathcal{C}_r}$*
2. *If $x \leq ky \in \overline{\mathcal{C}_r}$ then $y \leq y \in \overline{\mathcal{C}_r}$*

Corollary 1. *Let $\mathcal{C}_r$ be a set of resource constraints, $x \in \mathcal{D}_r(\overline{\mathcal{C}_r})$ iff $x \leq x \in \overline{\mathcal{C}_r}$.*

Lemma 3 (Compactness). *Let $\mathcal{C}_r$ (resp. $\mathcal{C}_s$) be a (possibly infinite) set of resource constraints (resp. state constraints). If $x \leq y \in \overline{\mathcal{C}_r}$ (resp. $u \lhd v \in \overline{\mathcal{C}_s}$) then there exists a finite set $\mathcal{C}_f$ such that $\mathcal{C}_f \subseteq \mathcal{C}_r$ (resp. $\mathcal{C}_f \subseteq \mathcal{C}_s$) and $x \leq y \in \overline{\mathcal{C}_f}$ (resp. $u \lhd v \in \overline{\mathcal{C}_f}$).*

4.2 A labelled tableaux method for DBI

We now define a labelled tableaux method for DBI in the spirit of previous works for BI [4] and BBI [8].

Definition 11 (Labelled formula / CSS). *A* labelled formula *is a 4-uplet $(\mathbb{S}, \phi, x, u) \in \{\mathbb{T}, \mathbb{F}\} \times \mathcal{L} \times L_r \times L_s$ written $\mathbb{S}\phi : (x, u)$. A* constrained set of statements *(CSS) is a triple $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$, where $\mathcal{F}$ is a set of labelled formulae, $\mathcal{C}_r$ is a set of resource constraints and $\mathcal{C}_s$ is a set of state constraints, such that the following property, called (P_{css}), holds: if $\mathbb{S}\phi : (x, u) \in \mathcal{F}$ then $x \leq x \in \overline{\mathcal{C}_r}$ and $u \lhd u \in \overline{\mathcal{C}_s}$.*

A CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ is a representation of a branch in which the formulae are the labelled formulae of $\mathcal{F}$ and the constraints on labels are the elements of $\mathcal{C}_r$ and $\mathcal{C}_s$. Our calculus extends some principles of BI calculus by adding a second kind of labels (state labels) and a set of constraints ($\mathcal{C}_s$) for state labels.
A CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ is *finite* iff $\mathcal{F}$, $\mathcal{C}_r$ and $\mathcal{C}_s$ are finite. We define the relation $\preccurlyeq$ by: $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle \preccurlyeq \langle \mathcal{F}', \mathcal{C}'_r, \mathcal{C}'_s \rangle$ iff $\mathcal{F} \subseteq \mathcal{F}'$ and $\mathcal{C}_r \subseteq \mathcal{C}'_r$ and $\mathcal{C}_s \subseteq \mathcal{C}'_s$. Moreover we denote $\langle \mathcal{F}_f, \mathcal{C}_{r_f}, \mathcal{C}_{s_f} \rangle \preccurlyeq_f \langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ when $\langle \mathcal{F}_f, \mathcal{C}_{r_f}, \mathcal{C}_{s_f} \rangle \preccurlyeq \langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ holds and $\langle \mathcal{F}_f, \mathcal{C}_{r_f}, \mathcal{C}_{s_f} \rangle$ is finite.

Definition 12 (Inconsistent label). *Let $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ be a CSS and x be a resource label. x is* inconsistent *if there exist two resource labels y and z such that $yz \leq x \in \overline{\mathcal{C}_r}$ and $\mathbb{T}\bot : (y, u) \in \mathcal{F}$. A label is* consistent *if it is not inconsistent.*

Proposition 2. *Let $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ be a CSS. The following properties hold:*

1. *If $y \leq x \in \overline{\mathcal{C}_r}$ and x is a consistent label then y is a consistent label.*
2. *If $xy \in \mathcal{D}_r(\overline{\mathcal{C}_r})$ is a consistent label then x and y are consistent labels.*

Figure 1 presents rules of labelled tableaux method for DBI. Let us remark that c_i and c_j are new label constants ($c_i \neq c_j \in \gamma_r \setminus \mathcal{A}_r(\mathcal{C}_r)$) and that l_i is a new label constant means $l_i \in L_s \setminus \mathcal{A}_s(\mathcal{C}_s)$. We note $\oplus$ the concatenation of lists. For example $[e_1; e_2; e_4] \oplus [e_4; e_3] = [e_1; e_2; e_4; e_4; e_3]$.

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

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

$$\frac{\mathbb{T}\mathrm{I} : (x,u) \in \mathcal{F}}{\langle \emptyset, \{1 \leq x\}, \emptyset \rangle} \langle \mathbb{T}\mathrm{I} \rangle$$

$$\frac{\mathbb{T}\phi \rightarrow \psi : (x,u) \in \mathcal{F} \text{ and } x \leq y \in \overline{\mathcal{C}_r}}{\langle \{\mathbb{F}\phi : (y,u)\}, \emptyset, \emptyset \rangle \mid \langle \{\mathbb{T}\psi : (y,u)\}, \emptyset, \emptyset \rangle} \langle \mathbb{T}\rightarrow \rangle \qquad \frac{\mathbb{F}\phi \rightarrow \psi : (x,u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (c_i,u), \mathbb{F}\psi : (c_i,u)\}, \{x \leq c_i\}, \emptyset \rangle} \langle \mathbb{F}\rightarrow \rangle$$

$$\frac{\mathbb{T}\phi * \psi : (x,u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (c_i,u), \mathbb{T}\psi : (c_j,u)\}, \{c_i c_j \leq x\}, \emptyset \rangle} \langle \mathbb{T}* \rangle \qquad \frac{\mathbb{F}\phi * \psi : (x,u) \in \mathcal{F} \text{ and } yz \leq x \in \overline{\mathcal{C}_r}}{\langle \{\mathbb{F}\phi : (y,u)\}, \emptyset, \emptyset \rangle \mid \langle \{\mathbb{F}\psi : (z,u)\}, \emptyset, \emptyset \rangle} \langle \mathbb{F}* \rangle$$

$$\frac{\mathbb{T}\phi \mathrel{-\!\!*} \psi : (x,u) \in \mathcal{F} \text{ and } xy \leq xy \in \overline{\mathcal{C}_r}}{\langle \{\mathbb{F}\phi : (y,u)\}, \emptyset, \emptyset \rangle \mid \langle \{\mathbb{T}\psi : (xy,u)\}, \emptyset, \emptyset \rangle} \langle \mathbb{T}\mathrel{-\!\!*} \rangle \qquad \frac{\mathbb{F}\phi \mathrel{-\!\!*} \psi : (x,u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (c_i,u), \mathbb{F}\psi : (xc_i,u)\}, \{xc_i \leq xc_i\}, \emptyset \rangle} \langle \mathbb{F}\mathrel{-\!\!*} \rangle$$

$$\frac{\mathbb{T}\Diamond\phi : (x,u) \in \mathcal{F}}{\langle \{\mathbb{T}\phi : (x,l_i)\}, \emptyset, \{u \lhd l_i\} \rangle} \langle \mathbb{T}\Diamond \rangle \qquad \frac{\mathbb{F}\Diamond\phi : (x,u) \in \mathcal{F} \text{ and } u \leq v \in \overline{\mathcal{C}_s}}{\langle \{\mathbb{F}\phi : (x,v)\}, \emptyset, \emptyset \rangle} \langle \mathbb{F}\Diamond \rangle$$

$$\frac{\mathbb{T}\Box\phi : (x,u) \in \mathcal{F} \text{ and } u \leq v \in \overline{\mathcal{C}_s}}{\langle \{\mathbb{T}\phi : (x,v)\}, \emptyset, \emptyset \rangle} \langle \mathbb{T}\Box \rangle \qquad \frac{\mathbb{F}\Box\phi : (x,u) \in \mathcal{F}}{\langle \{\mathbb{F}\phi : (x,l_i)\}, \emptyset, \{u \lhd l_i\} \rangle} \langle \mathbb{F}\Box \rangle$$

Note: c_i, c_j and l_i are new label constants.

Fig. 1. Tableaux rules for DBI

Definition 13 (DBI-tableau). *A* DBI-tableau *for a finite CSS* $\langle \mathcal{F}_0, \mathcal{C}_{r_0}, \mathcal{C}_{s_0} \rangle$ *is a list of CSS (*branches*), built inductively according the following rules:*

1. *The one branch list* $[\langle \mathcal{F}_0, \mathcal{C}_{r_0}, \mathcal{C}_{s_0} \rangle]$ *is a DBI-tableau for* $\langle \mathcal{F}_0, \mathcal{C}_{r_0}, \mathcal{C}_{s_0} \rangle$
2. *If the list* $\mathcal{T}_m \oplus [\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle] \oplus \mathcal{T}_n$ *is a DBI-tableau for* $\langle \mathcal{F}_0, \mathcal{C}_{r_0}, \mathcal{C}_{s_0} \rangle$ *and*

$$\frac{cond(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle)}{\langle \mathcal{F}_1, \mathcal{C}_{r_1}, \mathcal{C}_{s_1} \rangle \mid ... \mid \langle \mathcal{F}_k, \mathcal{C}_{r_k}, \mathcal{C}_{s_k} \rangle}$$

is an instance of a rule of Figure 1 for which $cond(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle)$ *is fulfilled, then the list* $\mathcal{T}_m \oplus [\langle \mathcal{F} \cup \mathcal{F}_1, \mathcal{C}_r \cup \mathcal{C}_{r_1}, \mathcal{C}_s \cup \mathcal{C}_{s_1} \rangle; ...; \langle \mathcal{F} \cup \mathcal{F}_k, \mathcal{C}_r \cup \mathcal{C}_{r_k}, \mathcal{C}_s \cup \mathcal{C}_{s_k} \rangle] \oplus \mathcal{T}_n$ *is a DBI-tableau for* $\langle \mathcal{F}_0, \mathcal{C}_{r_0}, \mathcal{C}_{s_0} \rangle$.

A DBI-tableau for a formula ϕ *is a DBI-tableau for* $\langle \{\mathbb{F}\phi : (1,l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\} \rangle$.

It is possible to prove, by observing rules of the tableaux method for DBI, that new CSS, obtained by applying a rule, respect the condition (P_{css}) of Definition 11. Then, for all branches $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ of a DBI-tableau for a formula ϕ, as $\mathbb{F}\phi : (1,l_1) \in \mathcal{F}$, then $1 \leq 1 \in \overline{\mathcal{C}_r}$. Thereby, for all branches $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$, it holds that $1 \in \mathcal{D}_r(\overline{\mathcal{C}_r})$.

A first kind of rules concerns $\langle \mathbb{T}\mathrm{I} \rangle$, $\langle \mathbb{F} \rightarrow \rangle$, $\langle \mathbb{T} * \rangle$, $\langle \mathbb{F} \multimap \rangle$, $\langle \mathbb{T} \Diamond \rangle$ and $\langle \mathbb{F} \Box \rangle$. These rules introduce new constraints and also new label constants (c_i, c_j and l_i), except for $\langle \mathbb{T}\mathrm{I} \rangle$ that only introduces a new constraint. Let us illustrate the $\langle \mathbb{T} \Diamond \rangle$ rule. To apply this rule on a CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ on a labelled formula $\mathbb{T}\Diamond\phi : (c_1, l_3) \in \mathcal{F}$, we choose a new label which does not appear in $\mathcal{C}_s$. For example, we say that $l_{10} \notin \mathcal{C}_s$. Thus, by choosing l_{10}, we can apply the rule, getting the new CSS $\langle \mathcal{F} \cup \{\mathbb{T}\phi : (c_1, l_{10})\}, \mathcal{C}_r, \mathcal{C}_s \cup \{l_3 \lhd l_{10}\} \rangle$. We notice the new state constraint $l_3 \lhd l_{10}$ added to the set of constraints. Let us observe that the $\langle \mathbb{T} * \rangle$ rule introduces two new resource labels. Concerning the rule $\langle \mathbb{F} \multimap \rangle$, as $\mathbb{F}\psi : (xc_i, u)$ is added to the set of labelled formulae, xc_i has to belong to $\overline{\mathcal{C}_r}$ in order to satisfy the condition (P_{css}) of Definition 11. By adding $xc_i \leq xc_i$ to $\mathcal{C}_r$, xc_i belongs to $\overline{\mathcal{C}_r}$ and so (P_{css}) is satisfied.
A second kind of rules concerns $\langle \mathbb{T} \rightarrow \rangle$, $\langle \mathbb{F} * \rangle$, $\langle \mathbb{T} \multimap \rangle$, $\langle \mathbb{F} \Diamond \rangle$ and $\langle \mathbb{T} \Box \rangle$. These rules have a condition on a closure of label constraints. In order to apply one of these rules we have to choose an existing label which satisfies the condition and then apply the rule using it. Otherwise, we cannot apply such rules. We illustrate the $\langle \mathbb{T} \Box \rangle$ rule: let a CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ such that $\mathbb{T}\Box\phi : (c_1, l_1) \in \mathcal{F}$. To apply this rule, we have to choose a state label l such that $l_1 \lhd l \in \overline{\mathcal{C}_s}$. If we consider that $l_1 \leq l_2 \in \overline{\mathcal{C}_s}$ then we can decide to apply the rule using l_2, getting the CSS $\langle \mathcal{F} \cup \{\mathbb{T}\phi : (c_1, l_2)\}, \mathcal{C}_r, \mathcal{C}_s \rangle$. Let us observe that $\langle \mathbb{F} * \rangle$ rule needs to choose two labels y and z such that $yz \leq x \in \overline{\mathcal{C}_r}$.

Definition 14 (Closure condition). *A CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ is* closed *if one of the following conditions hold:*

1. $\mathbb{T}\phi : (x, u) \in \mathcal{F}$, $\mathbb{F}\phi : (y, u) \in \mathcal{F}$ *and* $x \leq y \in \overline{\mathcal{C}_r}$
2. $\mathbb{F}\mathrm{I} : (x, u) \in \mathcal{F}$ *and* $1 \leq x \in \overline{\mathcal{C}_r}$
3. $\mathbb{F}\top : (x, u) \in \mathcal{F}$
4. $\mathbb{F}\phi : (x, u) \in \mathcal{F}$ *and x is inconsistent*

A CSS is open *if it is not closed. A DBI-tableau is closed if all its branches are closed.*

Definition 15 (DBI-proof). *A* DBI-proof *for a formula ϕ is a DBI-tableau for ϕ which is closed.*

Let us recall that we deal with labelled formulae with two kinds of labels: resource labels and state labels. Each CSS (branch) contains two sets of constraints, one for resources and another for states. Moreover the closure of such constraints can be represented by graphs. There are rules which modify constraint sets (graphs) and introduce new labels. Other rules have a set of conditions that must be satisfied, by finding labels satisfying it and then to solve constraints on the constraint graphs.

Let us now consider the formula $\phi \equiv (\Box(P \rightarrow \Diamond Q) \wedge \Diamond P) \multimap \Diamond Q$ and give a DBI-proof for it. By Definition 13, the following DBI-tableau $[\langle \{\mathbb{F}\phi : (1, l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\} \rangle]$ is a DBI-tableau for ϕ. We introduce a new representation for a DBI-tableau, which is

$[\mathcal{F}]$	$[\mathcal{C}_r]$	$[\mathcal{C}_s]$
$\mathbb{F}(\Box(P \rightarrow \Diamond Q) \wedge \Diamond P) \multimap \Diamond Q : (1, l_1)$	$1 \leq 1$	$l_1 \lhd l_1$

We can observe that there are three columns, one for the labelled formula sets of the CSS of the DBI-tableau ($[\mathcal{F}]$), one for the resource constraint sets of the CSS of the DBI-tableau ($[\mathcal{C}_r]$) and one for the state constraint sets of the CSS of the DBI-tableau ($[\mathcal{C}_s]$). By applying some rules, we obtain the following DBI-tableau:

We decorate a labelled formula with $\surd_i$ to show that we apply a rule on this formula at step i. We remark that columns ($[\mathcal{F}]$, $[\mathcal{C}_r]$ and $[\mathcal{C}_s]$) are trees that contain two branches. There are two branches because there are two CSS in the DBI-tableau. The branches on the left (resp. right) contain the elements of the first (resp. second) CSS. We also remark that all CSS are closed (denoted $\times$). The CSS of the left is closed because $\mathbb{T}P : (c_1, l_2) \in \mathcal{F}$, $\mathbb{F}P : (c_1, l_2) \in \mathcal{F}$ and $c_1 \leq c_1 \in \overline{\mathcal{C}_r}$. Thus, by definition, this DBI-tableau is a DBI-proof of $(\Box(P \to \Diamond Q) \wedge \Diamond P) \mathrel{-\!\!*} \Diamond Q$.

5 Soundness and completeness results

The soundness proof uses similar techniques than the ones used in BI for a labelled tableaux method [4]. The key point is the notion of *realizability* of a CSS $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$, that means there exists a dynamic model $\mathcal{K}$ and embeddings from resource labels to the resource set ($\lfloor \cdot \rfloor$) and state labels to the state set ($\lceil \cdot \rceil$) of $\mathcal{K}$ such that if $\mathbb{T}\phi : (x, u) \in \mathcal{F}$ then $\lfloor x \rfloor, \lceil u \rceil \vDash_{\mathcal{K}} \phi$ and if $\mathbb{F}\phi : (x, u) \in \mathcal{F}$ then $\lfloor x \rfloor, \lceil u \rceil \nvDash_{\mathcal{K}} \phi$.

Definition 16 (Realization). *Let $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ be a CSS. A* realization *of it is a triple $(\mathcal{K}, \lfloor . \rfloor, \lceil . \rceil)$ such that $\mathcal{K} = (\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ is a dynamic resource model, $\mathcal{M} = (R, \bullet, e, \pi, \sqsubseteq, S, \preceq)$, $\lfloor . \rfloor : \mathcal{D}_r(\overline{\mathcal{C}_r}) \to R$ and $\lceil . \rceil : \mathcal{A}_s(\mathcal{C}_s) \to S$, such that:*

- $\lfloor 1 \rfloor = e$
- $\lfloor x \circ y \rfloor = \lfloor x \rfloor \bullet \lfloor y \rfloor$
- *If* $\mathbb{T}\phi : (x,u) \in \mathcal{F}$ *then* $\lfloor x \rfloor, \lceil u \rceil \vDash_{\mathcal{K}} \phi$
- *If* $\mathbb{F}\phi : (x,u) \in \mathcal{F}$ *then* $\lfloor x \rfloor, \lceil u \rceil \nvDash_{\mathcal{K}} \phi$
- *If* $x \leq y \in \mathcal{C}_r$ *then* $\lfloor x \rfloor \sqsubseteq \lfloor y \rfloor$
- *If* $u \lhd v \in \mathcal{C}_s$ *then* $\lceil u \rceil \preceq \lceil v \rceil$

We say that a CSS/branch is *realizable* if there exists a realization of it. We say that a tableau is *realizable* if it contains a realizable CSS/branch.

Lemma 4. *Let* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *be a CSS and* $(\mathcal{K}, \lfloor . \rfloor, \lceil . \rceil)$ *a realization of it. For all* $x \leq y \in \overline{\mathcal{C}_r}$, $\lfloor x \rfloor \sqsubseteq \lfloor y \rfloor$ *and for all* $u \lhd v \in \overline{\mathcal{C}_s}$, $\lceil u \rceil \preceq \lceil v \rceil$.

Lemma 5. *The closed DBI-tableaux are not realizable.*

Lemma 6. *The expansion rules preserve realizability, i.e., if a rule of the DBI-tableau method is applied on a labelled formula of a realizable CSS then one of the obtained CSS is realizable.*

Theorem 1 (Soundness). *Let* ϕ *be a formula, if there exists a DBI-proof of* ϕ *then* ϕ *is valid.*

Proof. Let $\mathcal{T}$ be a DBI-proof of ϕ. Let us assume that ϕ is not valid. Then there exits a dynamic resource model $\mathcal{K}$ such that $e, s \nvDash_{\mathcal{K}} \phi$. If we consider $\lfloor 1 \rfloor = e$ and $\lceil l_1 \rceil = s$ we obtain a realisation $(\mathcal{K}, \lfloor . \rfloor, \lceil . \rceil)$ of the initial CSS $\langle \{\mathbb{F}\phi : (1, l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\} \rangle$. Thus, by Lemma 6, one branch of $\mathcal{T}$ is realizable. But by Lemma 5 it is contradictory, because as $\mathcal{T}$ is a DBI-proof, then $\mathcal{T}$ is closed. Thus ϕ is valid.

Before to study completeness we consider the countermodel extraction for DBI tableaux method. The main idea consists in transforming resource and state constraints into a dynamic resource monoid, from a branch $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ which is not closed.
In order to obtain a countermodel, this transformation has to verify two properties: if $\mathbb{T}\phi : (x,u) \in \mathcal{F}$ then $x, u \vDash_{\mathcal{K}} \phi$ and if $\mathbb{F}\phi : (x,u) \in \mathcal{F}$ then $x, u \nvDash_{\mathcal{K}} \phi$. In order to satisfy them, our method needs to *saturate* labelled formulae (to obtain a Hintikka CSS), that means, for instance, if $\mathbb{T}\Box\phi : (x,u) \in \mathcal{F}$ then we want that $x, u \vDash_{\mathcal{K}} \Box\phi$, so for all state labels v such that $u \lhd v \in \overline{\mathcal{C}_s}$, $\mathbb{T}\phi : (x,v) \in \mathcal{F}$ has to be verified.

Definition 17 (Hintikka CSS). *A CSS* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *is a* Hintikka CSS *if for any formula* $\phi, \psi \in \mathcal{L}$ *and any label* $x, y \in L_r$ *and* $u, v \in L_s$*:*

1. $\mathbb{T}\phi : (x,u) \notin \mathcal{F}$ *or* $\mathbb{F}\phi : (y,u) \notin \mathcal{F}$ *or* $x \leq y \notin \overline{\mathcal{C}_r}$
2. $\mathbb{F}\mathrm{I} : (x,u) \notin \mathcal{F}$ *or* $1 \leq x \notin \overline{\mathcal{C}_r}$
3. $\mathbb{F}\top : (x,u) \notin \mathcal{F}$
4. $\mathbb{F}\phi : (x,u) \notin \mathcal{F}$ *or* x *is consistent*
5. *If* $\mathbb{T}\mathrm{I} : (x,u) \in \mathcal{F}$ *then* $1 \leq x \in \overline{\mathcal{C}_r}$
6. *If* $\mathbb{T}\phi \wedge \psi : (x,u) \in \mathcal{F}$ *then* $\mathbb{T}\phi : (x,u) \in \mathcal{F}$ *and* $\mathbb{T}\psi : (x,u) \in \mathcal{F}$
7. *If* $\mathbb{F}\phi \wedge \psi : (x,u) \in \mathcal{F}$ *then* $\mathbb{F}\phi : (x,u) \in \mathcal{F}$ *or* $\mathbb{F}\psi : (x,u) \in \mathcal{F}$
8. *If* $\mathbb{T}\phi \vee \psi : (x,u) \in \mathcal{F}$ *then* $\mathbb{T}\phi : (x,u) \in \mathcal{F}$ *or* $\mathbb{T}\psi : (x,u) \in \mathcal{F}$
9. *If* $\mathbb{F}\phi \vee \psi : (x,u) \in \mathcal{F}$ *then* $\mathbb{F}\phi : (x,u) \in \mathcal{F}$ *and* $\mathbb{F}\psi : (x,u) \in \mathcal{F}$

10. *If* $\mathbb{T}\phi \rightarrow \psi : (x,u) \in \mathcal{F}$ *then* $\forall y \in L_r, x \leq y \in \overline{\mathcal{C}_r} \Rightarrow \mathbb{F}\phi : (y,u) \in \mathcal{F}$ *or* $\mathbb{T}\psi : (y,u) \in \mathcal{F}$
11. *If* $\mathbb{F}\phi \rightarrow \psi : (x,u) \in \mathcal{F}$ *then* $\exists y \in L_r, x \leq y \in \overline{\mathcal{C}_r}$ *and* $\mathbb{T}\phi : (y,u) \in \mathcal{F}$ *and* $\mathbb{F}\psi : (y,u) \in \mathcal{F}$
12. *If* $\mathbb{T}\phi * \psi : (x,u) \in \mathcal{F}$ *then* $\exists y,z \in L_r$, $yz \leq x \in \overline{\mathcal{C}_r}$ *and* $\mathbb{T}\phi : (y,u) \in \mathcal{F}$ *and* $\mathbb{T}\psi : (z,u) \in \mathcal{F}$
13. *If* $\mathbb{F}\phi * \psi : (x,u) \in \mathcal{F}$ *then* $\forall y,z \in L_r, yz \leq x \in \overline{\mathcal{C}_r} \Rightarrow \mathbb{F}\phi : (y,u) \in \mathcal{F}$ *or* $\mathbb{F}\psi : (z,u) \in \mathcal{F}$
14. *If* $\mathbb{T}\phi \twoheadrightarrow \psi : (x,u) \in \mathcal{F}$ *then* $\forall y \in L_r, xy \in \mathcal{D}_r(\overline{\mathcal{C}_r}) \Rightarrow \mathbb{F}\phi : (y,u) \in \mathcal{F}$ *or* $\mathbb{T}\psi : (xy,u) \in \mathcal{F}$
15. *If* $\mathbb{F}\phi \twoheadrightarrow \psi : (x,u) \in \mathcal{F}$ *then* $\exists y \in L_r$, $xy \in \mathcal{D}_r(\overline{\mathcal{C}_r})$ *and* $\mathbb{T}\phi : (y,u) \in \mathcal{F}$ *and* $\mathbb{F}\psi : (xy,u) \in \mathcal{F}$
16. *If* $\mathbb{T}\Diamond\phi : (x,u) \in \mathcal{F}$ *then* $\exists v \in L_s$, $u \lhd v \in \overline{\mathcal{C}_s}$ *and* $\mathbb{T}\phi : (x,v) \in \mathcal{F}$
17. *If* $\mathbb{F}\Diamond\phi : (x,u) \in \mathcal{F}$ *then* $\forall v \in L_s$, $u \lhd v \in \overline{\mathcal{C}_s} \Rightarrow \mathbb{F}\phi : (x,v) \in \mathcal{F}$
18. *If* $\mathbb{T}\Box\phi : (x,u) \in \mathcal{F}$ *then* $\forall v \in L_s$, $u \lhd v \in \overline{\mathcal{C}_s} \Rightarrow \mathbb{T}\phi : (x,v) \in \mathcal{F}$
19. *If* $\mathbb{F}\Box\phi : (x,u) \in \mathcal{F}$ *then* $\exists v \in L_s$, $u \lhd v \in \overline{\mathcal{C}_s}$ *and* $\mathbb{F}\phi : (x,v) \in \mathcal{F}$

The conditions (1), (2), (3) and (4) of Definition 17 certify that a Hintikka CSS is not closed. Others conditions certify that all labelled formulae of a Hintikka CSS are saturated. Let us now define a function Ω that allows us to extract a countermodel from a Hintikka CSS.

Definition 18 (Function Ω). *Let* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *be a Hintikka CSS and* $\overline{\mathcal{C}_{r\omega}}$ *be the restriction of* $\overline{\mathcal{C}_r}$ *to constraints including only consistent labels. The function* Ω *associates to* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *a triple* $\Omega(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle) = (\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ *where* $\mathcal{M} = (R, \bullet, e, \pi, \sqsubseteq, S, \preceq)$*, such that:*

- $R = \mathcal{D}_r(\overline{\mathcal{C}_{r\omega}}) \cup \{\pi\}$*, with* $\pi \notin \mathcal{D}_r(\overline{\mathcal{C}_r})$
- $S = \mathcal{A}_s(\mathcal{C}_s)$
- $e = 1$
- $\bullet$ *is defined by:* $\forall r_1, r_2 \in R \begin{cases} r_1 \bullet r_2 = r_1 \circ r_2 & \text{if } r_1 \circ r_2 \in \mathcal{D}_r(\overline{\mathcal{C}_{r\omega}}) \\ r_1 \bullet r_2 = \pi & \text{otherwise} \end{cases}$
- $r_1 \sqsubseteq r_2$ *iff* $r_1 \leq r_2 \in \overline{\mathcal{C}_{r\omega}}$ *or* $r_2 = \pi$
- $s_1 \preceq s_2$ *iff* $s_1 \lhd s_2 \in \overline{\mathcal{C}_s}$
- $(r,s) \in [\![P]\!]$ *iff* $(r = \pi)$ *or* $(\exists r' \in R, r' \sqsubseteq r$ *and* $\mathbb{T}P : (r',s) \in \mathcal{F})$

Let $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ be a CSS and $x \in \mathcal{D}_r(\overline{\mathcal{C}_r})$. We remark that x is a consistent label resource if and only if $x \in \mathcal{D}_r(\overline{\mathcal{C}_{r\omega}})$. Indeed, if $x \in \mathcal{D}_r(\overline{\mathcal{C}_r})$ then by Corollary 1, $x \leq x \in \overline{\mathcal{C}_r}$. Thus, as x is consistent, all resource labels and sub-labels of x are consistent by Proposition 2. Thus $x \leq x \in \overline{\mathcal{C}_{r\omega}}$ and $x \in \mathcal{D}_r(\overline{\mathcal{C}_{r\omega}})$. Now, if $x \in \mathcal{D}_r(\overline{\mathcal{C}_{r\omega}})$ then there exist $xy \leq z \in \overline{\mathcal{C}_{r\omega}}$ or $z \leq xy \in \overline{\mathcal{C}_{r\omega}}$. Therefore x is consistent otherwise $xy \leq z \notin \overline{\mathcal{C}_{r\omega}}$ or $z \leq xy \notin \overline{\mathcal{C}_{r\omega}}$.

Lemma 7. *Let* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *be a Hintikka CSS and* $\Omega(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle) = (\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ *where* $\mathcal{M} = (R, \bullet, e, \pi, \sqsubseteq, S, \preceq)$*.* $(\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ *is a dynamic resource model.*

Lemma 8. *Let* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *be a Hintikka CSS. Let* $\Omega(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle) = (\mathcal{M}, [\![\cdot]\!], \vDash_{\mathcal{K}})$ *where* $\mathcal{M} = (R, \bullet, e, \pi, \sqsubseteq, S, \preceq)$*. For any formula* ϕ *the following properties hold:*

1. $\pi, s \vDash_{\mathcal{K}} \phi$
2. *If* $\mathbb{F}\phi : (r,s) \in \mathcal{F}$ *and* r *consistent then* $r,s \nvDash_{\mathcal{K}} \phi$
3. *If* $\mathbb{T}\phi : (r,s) \in \mathcal{F}$ *and* r *consistent then* $r,s \vDash_{\mathcal{K}} \phi$

Lemma 9. *Let* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$ *be a Hintikka CSS such that* $\mathbb{F}\phi : (1,s) \in \mathcal{F}$. ϕ *is not valid.*

Proof. If the resource label 1 is inconsistent, then it is contradictory because $\mathbb{F}\phi : (1,s) \in \mathcal{F}$ and by condition (4) of Definition 17. Thus 1 is consistent. By Lemma 7, $\Omega(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle)$ is a dynamic resource model. By Lemma 8, $e,s \nvDash_{\mathcal{K}} \phi$ in this model. Thus $\Omega(\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle)$ is a countermodel of ϕ and then ϕ is not valid.

The proof of completeness consists in building a Hintikka CSS from a CSS which cannot be closed, in the spirit of the proof developed for BBI [8]. Then we need a fair strategy and a oracle which contains all finite *consistent* (not closed but saturated) CSS.

Definition 19 (Fair strategy). *A* fair strategy *is a labelled formulae sequence* $(\mathbb{S}_i F_i : (x_i, u_i))_{i \in \mathbb{N}}$ *in* $\{\mathbb{T}, \mathbb{F}\} \times \mathcal{L} \times L_r \times L_s$ *such that any labelled formula occurs infinitely many times in this sequence, that is* $\{i \in \mathbb{N} \mid \mathbb{S}_i F_i : (x_i, u_i) \equiv \mathbb{S}F : (x,u)\}$ *is infinite for any* $\mathbb{S}F : (x,u) \in \{\mathbb{T}, \mathbb{F}\} \times \mathcal{L} \times L_r \times L_s$.

Proposition 3. *There exists a fair strategy.*

The main argument is that the set of labelled formulae is countable.

Definition 20. *Let* $\mathcal{P}$ *be a set of CSS.*

1. $\mathcal{P}$ *is* $\preccurlyeq$-closed *if* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle \in \mathcal{P}$ *holds whenever* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle \preccurlyeq \langle \mathcal{F}', \mathcal{C}'_r, \mathcal{C}'_s \rangle$ *and* $\langle \mathcal{F}', \mathcal{C}'_r, \mathcal{C}'_s \rangle \in \mathcal{P}$ *hold.*
2. $\mathcal{P}$ *is of* finite character *if* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle \in \mathcal{P}$ *holds whenever* $\langle \mathcal{F}_f, \mathcal{C}_{rf}, \mathcal{C}_{sf} \rangle \in \mathcal{P}$ *holds for every* $\langle \mathcal{F}_f, \mathcal{C}_{rf}, \mathcal{C}_{sf} \rangle \preccurlyeq_f \langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle$.
3. $\mathcal{P}$ *is* saturated *if for any* $\langle \mathcal{F}, \mathcal{C}_r, \mathcal{C}_s \rangle \in \mathcal{P}$ *and any instance*

$$\frac{cond(\mathcal{F}, \mathcal{C}_r, \mathcal{C}_s)}{\langle \mathcal{F}_1, \mathcal{C}_{r1}, \mathcal{C}_{s1} \rangle \mid \ldots \mid \langle \mathcal{F}_k, \mathcal{C}_{rk}, \mathcal{C}_{sk} \rangle}$$

of a rule of Figure 1, if $cond(\mathcal{F}, \mathcal{C}_r, \mathcal{C}_s)$ *is fulfilled then* $\langle \mathcal{F} \cup \mathcal{F}_i, \mathcal{C}_r \cup \mathcal{C}_{ri}, \mathcal{C}_s \cup \mathcal{C}_{si} \rangle \in \mathcal{P}$ *for at least one* $i \in \{1, \ldots, k\}$.

Definition 21 (Oracle). *An* oracle *is a set of non closed CSS which is* $\preccurlyeq$*-closed, of finite character and saturated.*

Lemma 10. *There exists an oracle which contains every finite CSS for which there exists no closed DBI-tableau.*

This oracle is the set of all CSS such that there exists no closed DBI-tableau for their finite sub-CSS ($\preccurlyeq$). Let us assume that there exists no DBI-proof of formula φ and show that φ is not valid by constructing a Hintikka CSS. Let us note that φ denotes the formula for which we are constructing a Hintikka CSS and ϕ denotes any formula. Let $\mathcal{T}_0$ a initial DBI-tableau for φ, we have

1. $\mathcal{T}_0 = [\langle\{\mathbb{F}\varphi : (1,l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\}\rangle]$
2. $\mathcal{T}_0$ cannot be closed

By Lemma 10, there exists an oracle which contains every finite CSS for which there exists no closed DBI-tableau. Let $\mathcal{P}$ be such an oracle. By hypothesis we have $\langle\{\mathbb{F}\varphi : (1,l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\}\rangle \in \mathcal{P}$. By Proposition 3, there exists a fair strategy. Let $\mathcal{S}$ be such a strategy. We denoted $\mathbb{S}_i F_i : (x_i, u_i)$ the ith formula of $\mathcal{S}$. We built a sequence $\langle \mathcal{F}_i, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle_{0 \leqslant i}$ as follows:

- $\langle \mathcal{F}_0, \mathcal{C}_{r0}, \mathcal{C}_{s0}\rangle = \langle\{\mathbb{F}\varphi : (1,l_1)\}, \{1 \leq 1\}, \{l_1 \lhd l_1\}\rangle$
- If $\langle \mathcal{F}_i \cup \{\mathbb{S}_i F_i : (x_i, u_i)\}, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle \notin \mathcal{P}$ then $\langle \mathcal{F}_{i+1}, \mathcal{C}_{ri+1}, \mathcal{C}_{si+1}\rangle = \langle \mathcal{F}_i, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle$
- If $\langle \mathcal{F}_i \cup \{\mathbb{S}_i F_i : (x_i, u_i)\}, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle \in \mathcal{P}$ then $\langle \mathcal{F}_{i+1}, \mathcal{C}_{ri+1}, \mathcal{C}_{si+1}\rangle = \langle \mathcal{F}_i \cup \{\mathbb{S}_i F_i : (x_i, u_i)\} \cup F_e, \mathcal{C}_{ri} \cup \mathcal{C}_{re}, \mathcal{C}_{si} \cup \mathcal{C}_{se}\rangle$ such that F_e, $\mathcal{C}_{re}$ and $\mathcal{C}_{se}$ are determined by:

S_i	F_i	F_e	$\mathcal{C}_{re}$	$\mathcal{C}_{se}$
$\mathbb{F}$	$\phi \to \psi$	$\{\mathbb{T}\phi : (a,u_i), \mathbb{F}\psi : (a,u_i)\}$	$\{x_i \leq a\}$	$\emptyset$
$\mathbb{T}$	$\phi * \psi$	$\{\mathbb{T}\phi : (a,u_i), \mathbb{T}\psi : (b,u_i)\}$	$\{ab \leq x_i\}$	$\emptyset$
$\mathbb{F}$	$\phi \mathrel{-\!\!*} \psi$	$\{\mathbb{T}\phi : (a,u_i), \mathbb{F}\psi : (x_i a,u_i)\}$	$\{x_i a \leq x_i a\}$	$\emptyset$
$\mathbb{T}$	I	$\emptyset$	$\{1 \leq x_i\}$	$\emptyset$
$\mathbb{T}$	$\Diamond\phi$	$\{\mathbb{T}\phi : (x_i, c)\}$	$\emptyset$	$\{u_i \lhd c\}$
$\mathbb{F}$	$\Box\phi$	$\{\mathbb{F}\phi : (x_i, c)\}$	$\emptyset$	$\{u_i \lhd c\}$
Otherwise		$\emptyset$	$\emptyset$	$\emptyset$

with $a = c_{2i+1}$, $b = c_{2i+2}$ and $c = l_{i+2}$.

Proposition 4. *For any $i \in \mathbb{N}$, the following properties hold:*

1. $\mathbb{F}\varphi : (1,l_1) \in \mathcal{F}_i$, $1 \leq 1 \in \mathcal{C}_{ri}$ *and* $l_1 \lhd l_1 \in \mathcal{C}_{si}$
2. $\mathcal{F}_i \subseteq \mathcal{F}_{i+1}$, $\mathcal{C}_{ri} \subseteq \mathcal{C}_{ri+1}$ *and* $\mathcal{C}_{si} \subseteq \mathcal{C}_{si+1}$
3. $\langle \mathcal{F}_i, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle_{0 \leqslant i} \in \mathcal{P}$
4. $\mathcal{A}_r(\mathcal{C}_{ri}) \subseteq \{1, c_1, c_2, ..., c_{2i}\}$
5. $\mathcal{A}_s(\mathcal{C}_{si}) \subseteq \{l_1, l_2, ..., l_{i+1}\}$

We now consider the limit CSS $\langle \mathcal{F}_\infty, \mathcal{C}_{r\infty}, \mathcal{C}_{s\infty}\rangle$ of the sequence $\langle \mathcal{F}_i, \mathcal{C}_{ri}, \mathcal{C}_{si}\rangle_{0 \leqslant i}$ defined by:

$$\mathcal{F}_\infty = \bigcup_i \mathcal{F}_i \quad \text{and} \quad \mathcal{C}_{r\infty} = \bigcup_i \mathcal{C}_{ri} \quad \text{and} \quad \mathcal{C}_{s\infty} = \bigcup_i \mathcal{C}_{si}$$

Proposition 5. *We have $\langle \mathcal{F}_\infty, \mathcal{C}_{r\infty}, \mathcal{C}_{s\infty}\rangle \in \mathcal{P}$ and for all labelled formulae $\mathbb{S}\phi : (x,u)$, if $\langle \mathcal{F}_\infty \cup \{\mathbb{S}\phi : (x,u)\}, \mathcal{C}_{r\infty}, \mathcal{C}_{s\infty}\rangle \in \mathcal{P}$ then $\mathbb{S}\phi : (x,u) \in \mathcal{F}_\infty$*

Lemma 11. *The limit CSS is a Hintikka CSS.*

Theorem 2 (Completeness). *Let φ be a formula, if φ is valid then there exists a DBI-proof for φ.*

Proof. We suppose that there is no DBI-proof of φ and show that φ is not valid. Our method allows us to build a limit CSS that is a Hintikka CSS, by Lemma 11. By property 1 of Proposition 4, $\mathbb{F}\varphi : (1,l_1) \in \mathcal{F}_i$. By Lemma 9, φ is not valid.

6 Conclusion

We have defined and studied a modal extension of BI, called DBI, that allows us to express dynamic properties about resources. We propose a Kripke semantics for DBI and a labelled tableaux method that is proved sound and complete w.r.t. this semantics. Compared to previous works on proof-theory in BI, the labelled tableaux method for DBI deals not only with a so-called resource graph but also with a state graph. Moreover we show how we can generate countermodels in case of non-validity.
Future works will be devoted to the study of other extensions of BI with other modalities such that fragments of SCRP/MBI [12], in order to mix dynamic resources and processes, and to the study of semantics based on Petri nets for such extensions.

References

1. N. Biri and D. Galmiche. A Separation Logic for Resource Distribution. In *23rd Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'03, LNCS 2914*, pages 23–37, December 2003. Mumbai, India.
2. U. Engberg and G. Winskel. Completeness results for Linear Logic on Petri nets. *Annals of Pure and Applied Logic*, 86:101–135, 1997.
3. D. Galmiche and D. Méry. Tableaux and Resource Graphs for Separation Logic. *Journal of Logic and Computation*, 20(1):189–231, 2010.
4. D. Galmiche, D. Méry, and D. Pym. The semantics of BI and Resource Tableaux. *Math. Struct. in Comp. Science*, 15(6):1033–1088, 2005.
5. J.Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
6. C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, October 1969.
7. S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *28th ACM Symposium on Principles of Programming Languages, POPL 2001*, pages 14–26, London, UK, 2001.
8. D. Larchey-Wendling. Semantic Tableaux for Boolean BI: Strong Completeness and Semi-Decision. Technical report, Loria, 2012.
9. D. Larchey-Wendling and D. Galmiche. The Undecidability of Boolean BI through Phase Semantics. In *25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010*, pages 147–156, Edinburgh, UK, July 2010.
10. R. Milner. *Communication and concurrency*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
11. P.W. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
12. D. Pym and C. Tofts. Systems modelling via resources and processes: Philosophy, calculus, semantics, and logic. *Electronic Notes in Theoretical Computer Science*, 172:545–587, 2007.
13. D.J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.
14. J. Reynolds. Separation logic: A logic for shared mutable data structures. In *IEEE Symposium on Logic in Computer Science*, pages 55–74, Copenhagen, Danemark, July 2002.