Separation and modality with updates*

Joseph Boudou

Septembre 2015

1 Introduction

In this report, logics combining separation with dynamic modalities are presented. We focus on abstract logics where the separation is not dedicated to specific structures like trees, graphs or pointers memories. Moreover, we only consider propositional logics. Our main concerns for those logics are their expressivity and the decidability and complexity of their satisfiability problem.

In the next section, the logics BI and BBI are briefly recalled. This logics are the bases of most logics with separation. Moreover, we show that, to a given extend, BBI is already a modal logic with separation and update. In Section 3, we present logics with separation and temporal modalities. Section 4 introduces logics with separation and one modality for each action in some set of actions. Section 5 is dedicated to logics with separation related to updates of knowledge (or belief) bases. Finally the last section draws a conclusion.

2 Logics of Bunched Implications

The logic of Bunched Implications (BI) [26, 27] has been devised to reason about resources. Like in linear logic, the symbols of BI's language consists of a set of additive symbols (the boolean constants $\top$ and $\bot$, the conjunction $\wedge$, the disjunction $\vee$ and the implication $\rightarrow$) and a set of multiplicatives symbols (the constant I, the conjunction $*$ and the implication $\mathrel{-\!\!*}$). But in contradistinction with linear logic, the additive fragment of BI corresponds exactly to the intuitionistic logic. BI's multiplicative fragment is identical to the multiplicative fragment of intuitionistic linear logic (MILL). Intuitively, whereas the additive conjunction expresses sharing, the multiplicative conjunction expresses separation: for a state w to satisfy the formula $\varphi * \psi$ it must be possible to separate (decompose) w into two substates x and y such that x satisfies φ and y satisfies ψ. BI has an interesting sequent calculus with cut elimination [26, 27]. Moreover, its satisfiability problem is decidable: a sound, complete and terminating tableaux method has been devised in [19].

*Scientific report of the ANR project DynRes (project ANR-11-BS02-011).

In [12], a new semantics for BI based on Petri nets with inconsistency is proposed. This semantics is proved to be equivalent with the usual semantics over resources monoids. Moreover, assuming there is an injective function from the places of the Petri net with inconsistency $\mathfrak{P}$ to the propositional variables of the language, each configuration M of $\mathfrak{P}$ can be encoded by a BI's formula $\widehat{M}$ such that for all configurations M_1 and M_2 of $\mathfrak{P}$, M_2 can be reached from M_1 if and only if the empty configuration satisfies the formula $\widehat{M_1} \mathrel{-\!\!*} \widehat{M_2}$ in $\mathfrak{P}$. This result illustrates the fact that the multiplicative conjunction $\mathrel{-\!\!*}$ can express some kinds of dynamic of resources.

The Boolean logic of Bunched Implications (BBI) [27, 18] is the classical variant of BI. The language of BBI is the same as BI's one, but in BBI, the interpretation of the additive operators is classical instead of being intuitionistic. BBI is generally considered as the logical kernel of separation logics [29, 16] (a family of logics used in the industry to verify programs with pointers). Since BBI is also the base logic of most of the logics presented in this report, we give a brief formal description of its propositional variant. The language of BBI is inductively defined by the following grammar:

$$\varphi, \psi := p \mid \bot \mid (\varphi \wedge \psi) \mid (\varphi \rightarrow \psi) \mid I \mid (\varphi * \psi) \mid (\varphi \mathrel{-\!\!*} \psi)$$

where p is a propositional variable. I is the neutral element of the multiplicative conjunction $*$. The usual abbreviations are defined: $\neg\varphi \doteq \varphi \rightarrow \bot$, $\top \doteq \neg\bot$ and $\varphi \vee \psi \doteq (\neg\varphi) \rightarrow \psi$. BBI's formulas are interpreted over non-deterministic commutative monoids as defined below.

Definition. A *non-deterministic commutative monoid* is a tuple $(M, \circ, e)$ where M is a set, $\circ$ is a function from $M \times M$ to the powerset $\mathcal{P}(M)$ of M and $e \in M$ such that:

$$\forall r \in M, \quad e \circ r = \{r\} \qquad \text{(identity)}$$
$$\forall r, s \in M, \quad r \circ s = s \circ r \qquad \text{(commutativity)}$$
$$\forall r, s, t \in M, \quad r \circ (s \circ t) = (r \circ s) \circ t \qquad \text{(associativity[1])}$$

A *partial commutative monoid* is a non-deterministic commutative monoid which further satisfies the following partiality property:

$$\forall r, s, t, u \in M, \text{ if } \{r, s\} \subseteq t \circ u \text{ then } r = s \qquad \text{(partiality)}$$

A model for BBI is a tuple $(M, \circ, e, V)$ where $(M, \circ, e)$ is a non-deterministic commutative monoid and V is a valuation assigning a subset of M to each propositional variable of the language. For any resource $r \in M$ and any BBI's formula φ, $\mathcal{M}, r \vDash \varphi$ denotes that r satisfies φ in $\mathcal{M}$. The satisfiability relation

[1] For the associativity, the outermost $\circ$ must be understood as the extension of $\circ$ to $\mathcal{P}(M)$ defined by $S \circ T = \bigcup\{s \circ t \mid s \in S \text{ and } t \in T\}$ for all $S, T \in \mathcal{P}(M)$.

$\models$ is defined as usual for the additives (classical) symbols and as follows for the multiplicative symbols:

$$\begin{array}{ll} \mathcal{M}, r \models I & \text{iff } r = e \\ \mathcal{M}, r \models \varphi * \psi & \text{iff } \exists s, t \in M \text{ such that } r \in s \circ t, \mathcal{M}, s \models \varphi \text{ and } \mathcal{M}, t \models \psi \\ \mathcal{M}, r \models \varphi -\!\!* \psi & \text{iff } \forall s, t \in M \text{ if } t \in r \circ s \text{ and } \mathcal{M}, s \models \varphi \text{ then } \mathcal{M}, t \models \psi \end{array}$$

A Hilbert-style axiomatization for BBI is given in [18]. Let us define the operators $\varphi \circledast \psi \doteq \neg((\neg\varphi) * (\neg\psi))$ and $\varphi \multimap\!\!\square\, \psi \doteq \neg((\neg\varphi) -\!\!* \psi)$. Using the axiomatization from [18], it can be easily proved that the K axioms for $\circledast$ and $\multimap\!\!\square$ hold in BBI. Similarly, it can be easily proved that the necessitation rules for $\circledast$ and $\multimap\!\!\square$ are admissible. Since I is trivially a nullary normal modality, BBI can be seen as a normal multimodal logic. A Kripke model can be easily constructed from a model in the previous semantics by defining the ternary relation $\lhd \doteq \{(r,s,t) \mid r \in s \circ t\}$ to interpret $*$ and the ternary relation $\widetilde{\lhd} \doteq \{(s,t,r) \mid r \in s \circ t\}$ to interpret $\multimap\!\!\square$.

Moreover, since a formula of the form $\varphi -\!\!* \psi$ intuitively means that ψ will be satisfied after the addition of a resource satisfying φ, the multiplicative implication $-\!\!*$ can be seen as an update modality. Therefore, BBI is already itself a modal logic with separation and update. Of course, updates in BBI are limited to the addition of some resources. The modal logics presented in the remaining sections weaken this restriction, allowing different updates to be expressed.

The satisfiability problem for BBI is undecidable [25, 8]. From a modal logic perspective, this can be seen from the fact that modal logics with an associative binary modalities are usually undecidable, as shown in [23].

3 Separation and temporal modalities

In this section we present logics which combine separation with some unary modalities meant to express the evolution of the system over time. We are only interested here in *abstract* logics and we will not mention logics devised for some particular structures like ambient logic [9].

The Dynamic logic of Bunched Implications (DBI) [14, 12] extends BI with two dual unary modalities $\Box$ and $\Diamond$. DBI's formulas are interpreted with respect to a pair (r, w) where r is a resource from a resource monoid and w is a state of an unlabeled transition system. It has to be noticed that the resource monoid and the unlabeled transition system are independent. Moreover, the evaluation of the constructors inherited from BI depends only on the resource r whereas the evaluation of the unary modality depends only on the state w. The only dependence between the resources and the states holds in the valuation function, which assigns a subset of pairs of resources and states to each propositional variable. Therefore, in DBI only the characteristics of the resources may change over time; the structure of the resources always stays the same:

if a given resource can be separated at some time, it will always remain separable. We say that DBI can capture *dynamic properties of resources*. In [12], an example on prices evolution illustrates that this kind of dynamic may be of interest. A sound and complete tableaux method for DBI is given in [14, 12].

The Logic with Separating Modalities (LSM) [12] extends BBI with unary modalities $\Box$ and $\Box_\bullet$ along with one unary modality $\Box_\rho$ for each ρ in a set Σ of resource variables. A model for LSM is a tuple $\mathcal{M} = (W, M, \circ, e, R, \sigma, V)$ such that:

- W is a non-empty set of states;
- $(M, \circ, e)$ is a partial commutative monoid;
- R is a reflexive and transitive binary relation over $W \times M$;
- σ is a function from Σ to M;
- V is a valuation function assigning a subset of $W \times M$ to each propositional variable of the language.

The satisfiability relation is defined with respect to a pair $(w, r) \in W \times M$. For the constructs inherited from BBI, the definition of the satisfiability relation is similar to the one given in Sect. 2 and depends only on r. For the added unary modalities the definition of the satisfiability relation is as follows:

$$
\begin{aligned}
\mathcal{M}, w, r \vDash \Box\varphi \quad & \text{iff } \forall (x,s) \in W \times M, \\
& \qquad \text{if } (w,r)\ R\ (x,s) \text{ then } \mathcal{M}, x, s \vDash \varphi \\
\mathcal{M}, w, r \vDash \Box_\bullet\varphi \quad & \text{iff } \forall (x,s) \in W \times M, \forall t, u \in M, \\
& \qquad \text{if } u \in r \circ t \text{ and } (w,u)\ R\ (x,s) \text{ then } \mathcal{M}, x, s \vDash \varphi \\
\mathcal{M}, w, r \vDash \Box_\rho\varphi \quad & \text{iff } \forall (x,s) \in W \times M, \forall u \in M, \\
& \qquad \text{if } u \in r \circ \sigma(\rho) \text{ and } (w,u)\ R\ (x,s) \text{ then } \mathcal{M}, x, s \vDash \varphi
\end{aligned}
$$

Since the accessibility relation R relates pairs from $W \times M$, resources in LSM can be produced and consumed over time, which were not the case with DBI. We say that LSM can capture *dynamic of resources*. A sound and complete tableaux method for LSM is provided in [12].

4 Separation and modalities for actions

In the previous section, we presented logics with modalities expressing the evolution of a system after some anonymous actions or events. We present now some modal logics with separation in which actions are named. In those logics, it is possible to reason about the modifications of resources by some specific actions.

The Modal logic of Bunched Implications (MBI) [28, 11] is a Hennessy-Milner-style modal logic with separation. We give here a brief description of the quantifier-free fragment MBIc of MBI [10], for comparison purpose. The language of MBIc is the extension of BBI with two unary modalities $[a]$ and $[a]_\nu$ for each action $a \in \mathcal{A}$ from a given monoid $(\mathcal{A}, \|, \mathrm{skip})$ of actions. A model for MBIc is constructed by means of a process calculus called the synchronous calculus of resource processes (SCRP) from a partial commutative monoid $(M, \circ, e)$ of resources and a partial function $\mu : \mathcal{A} \times M \rightharpoonup M$ describing the modifications of the resources by the actions. For each action, the process calculus SCRP provides a binary relation $\xrightarrow{a} \subseteq (M \times P) \times (M \times P)$ where P is the set of processes of the calculus. Formulas of MBIc are evaluated on pairs composed of a process and a resource. We give the interesting definitions of the satisfiability relation for MBIc:

$$
\begin{aligned}
\mathcal{M}, r, p \vDash \varphi * \psi \ \text{ iff } & \exists s_1, s_2 \in M, \exists q_1, q_2 \in P \text{ such that} \\
& r \in s_1 \circ s_2, p \sim q_1 \times q_2, \mathcal{M}, r_1, s_1 \vDash \varphi \text{ and } \mathcal{M}, r_2, s_2 \vDash \psi \\
\mathcal{M}, r, p \vDash \varphi \mathrel{-\!\!*} \psi \ \text{ iff } & \forall s, t \in M, \forall q \in P, \\
& \text{if } t \in r \circ s \text{ and } \mathcal{M}, s, q \vDash \varphi \text{ then } \mathcal{M}, t, p \times q \vDash \psi \\
\mathcal{M}, r, p \vDash [a]\varphi \ \text{ iff } & \forall s \in M, \forall q \in P, \\
& \text{if } (r, p) \xrightarrow{a} (s, q) \text{ then } \mathcal{M}, s, q \vDash \varphi \\
\mathcal{M}, r, p \vDash [a]_\nu \varphi \ \text{ iff } & \forall s, t, u \in M, \forall q \in P, \\
& \text{if } u \in r \circ t \text{ and } (u, p) \xrightarrow{a} (s, q) \text{ then } \mathcal{M}, s, q \vDash \varphi
\end{aligned}
$$

where $\mathcal{M} = (M, \circ, e, \mu, V)$, $\sim$ is the bisimulation relation over processes and $\times$ is a binary operator over processes. It has to be noted that $\|$ and $\times$ are understood as parallel composition of actions and processes respectively.

The Dynamic Modal logic of Bunched Implication (DMBI) [15, 12] extends BBI with a modality $\Box$, like in DBI, along with a modality $[a]$ for each action $a \in \mathcal{A}$ from a monoid $(\mathcal{A}, ;, \mathrm{skip})$. It has to be noted though that the composition ; of the monoid of actions does not denote parallel execution like in MBIc but sequential composition. A model for DMBI is a tuple $(W, M, \circ, e, R, \mu, V)$ where W is a non-empty set of states, $(M, \circ, e)$ is partial commutative monoid of resources, R is an accessibility function assigning a binary relation over W to each action in $\mathcal{A}$, μ is a resource modification partial function from $\mathcal{A} \times M$ to M and V is a valuation function assigning a subset of $W \times M$ to each propositional variable. Moreover, a model must satisfy the following conditions, for all $w, x, y \in W$, all $r \in M$ and all $a, b \in \mathcal{A}$: [2]

- $w\ R(\mathrm{skip})\ w$
- if $w\ R(a)\ x$ and $x\ R(b)\ y$ then $w\ R(a\,;b)\ y$
- $\mu(\mathrm{skip}, r)\downarrow$ and $\mu(\mathrm{skip}, r) = r$

[2] For any partial function f, we write $f(x)\downarrow$ to denote that f is defined at x.

- if $\mu(a,r)\downarrow$ and $\mu(b,\mu(a,r))\downarrow$ then $\mu(a\,;b,r)\downarrow$ and $\mu(a\,;b,r) = \mu(b,\mu(a,r))$

For each $a \in \mathcal{A}$, the accessibility relation $\xrightarrow{a}$ over $M \times W$ is defined such that $r,w \xrightarrow{a} s,x$ iff $\mu(a,r)\downarrow$, $\mu(a,r) = s$ and $w\ R(a)\ x$. The binary relation $\rightsquigarrow$ over $M \times W$ is the transitive closure of $\bigcup_{a\in\mathcal{A}} \xrightarrow{a}$. The satisfiability relation is defined as follows for the multiplicative operators and the unary modalities (the other constructs being classical):

$$\mathcal{M},r,w \vDash \varphi * \psi \text{ iff } \exists s,t \in M \text{ such that } r \in s \circ t, \mathcal{M},s,w \vDash \varphi \text{ and } \mathcal{M},t,w \vDash \psi$$
$$\mathcal{M},r,w \vDash \varphi \mathrel{-\!\!*} \psi \text{ iff } \forall s,t \in M, \text{ if } t \in r \circ s \text{ and } \mathcal{M},s,w \vDash \varphi \text{ then } \mathcal{M},t,w \vDash \psi$$
$$\mathcal{M},r,w \vDash [a]\varphi \quad \text{iff } \forall s \in M, \forall x \in W, \text{ if } (r,w) \xrightarrow{a} (s,x) \text{ then } \mathcal{M},s,x \vDash \varphi$$
$$\mathcal{M},r,w \vDash \Box\varphi \quad \text{iff } \forall s \in M, \forall x \in W, \text{ if } (r,w) \rightsquigarrow (s,x) \text{ then } \mathcal{M},s,x \vDash \varphi$$

Compared with MBIc, DMBI is more abstract since the transition system is not constrained by a process calculus. A more involved difference can be observed by comparing the satisfiability of the multiplicative conjunction $*$ of those logics. Whereas in MBIc both resources and processes are decomposed by this operator, in DMBI only resources are decomposed, states being left unchanged. This difference explain why there is no need for multiplicative modalities $[\cdot]_\nu$ in DMBI: they can be defined by $[a]_\nu\,\varphi \doteq \top \mathrel{-\!\!*} [a]\varphi$. In fact, MBIc has been devised to reason about synchronous actions which is not the case of DMBI. Nevertheless, it is possible to reason about synchronous actions in DMBI, by constructing a model representing a set of parallel processes, as illustrated in [15, 12]. A sound and complete tableaux calculus for DMBI with countermodel extraction is proposed in [15, 12].

The Propositional Dynamic Logic with Parallel composition, Recovering and Storing (PRSPDL) [6, 5, 2, 3, 7] is an extension of the Propositional Dynamic Logic (PDL) [17, 20] with parallel composition of programs and some new special programs. There is no multiplicative conjunction in the language of PRSPDL, but as we will see, the combination of parallel compositions and tests allows to express separation. This logic does not add dynamics to a logic with separation but adds separation to a dynamic logic. Like for PDL, the language of PRSPDL is twofold. There are both programs and formulas, defined by simultaneous induction as follows:

$$\alpha,\beta := a \mid (\alpha \cup \beta) \mid (\alpha\,;\beta) \mid \alpha^* \mid \varphi? \mid r_1 \mid r_2 \mid s_1 \mid s_2 \mid (\alpha \parallel \beta) \quad \text{(programs)}$$
$$\varphi,\psi := p \mid \bot \mid (\varphi \wedge \psi) \mid (\varphi \to \psi) \mid [\alpha]\varphi \quad \text{(formulas)}$$

where a is an atomic program and p a propositional variable. Informally, $\cdot \cup \cdot$ is the non-deterministic choice between two programs, $\cdot\,;\cdot$ is the sequential composition, $\cdot^*$ is the iteration, $\cdot?$ is the test, r_i and s_i are the recovering and storing programs and $\cdot \parallel \cdot$ is the parallel composition. We define the usual abbreviations: $\neg\varphi \doteq \varphi \to \bot$, $\top \doteq \neg\bot$, $\varphi \vee \psi \doteq (\neg\varphi) \to \psi$ and $\langle\alpha\rangle\varphi \doteq \neg[\alpha]\neg\varphi$.

A model for PRSPDL is a tuple $\mathcal{M} = (W, R, \circ, V)$ where W is a non-empty set of states, R is an accessibility function assigning a binary relation over W to each atomic action, $\circ$ is a function from $W \times W$ to $\mathcal{P}(W)$ and V is a valuation function assigning a subset of W to each propositional variable. The satisfiability relation $\vDash$ is defined by induction simultaneously with the extension of R to all programs of the language. We only give here the definitions of R and $\vDash$ which are important for the following discussion and refer the reader to [20] for the missing ones.

$$
\begin{array}{ll}
w\,R(\varphi?)\,x & \text{iff } w = x \text{ and } \mathcal{M}, w \vDash \varphi \\
w\,R(r_i)\,x & \text{iff } \exists y_1, y_2 \in W \text{ such that } w \in y_1 \circ y_2 \text{ and } x = y_i \\
w\,R(s_i)\,x & \text{iff } \exists y_1, y_2 \in W \text{ such that } x \in y_1 \circ y_2 \text{ and } w = y_i \\
w\,R(\alpha \parallel \beta)\,x & \text{iff } \exists y_1, y_2, z_1, z_2 \in W \text{ such that} \\
 & \qquad w \in y_1 \circ y_2, y_1\,R(\alpha)\,z_1, y_2\,R(\beta)\,z_2 \text{ and } x \in z_1 \circ z_2 \\
\mathcal{M}, w \vDash [\alpha]\varphi & \text{iff } \forall x \in W, \text{ if } w\,R(\alpha)\,x \text{ then } \mathcal{M}, x \vDash \varphi
\end{array}
$$

It has to be noted that here the $\circ$ function does not necessarily has a neutral element and is not constrained to be commutative or associative. Therefore we do not have a non-deterministic commutative monoid. Nevertheless, we can define an operator $*$ by $\varphi * \psi \doteq \langle \varphi? \parallel \psi? \rangle \top$. This operator has a semantics similar to the semantics of BBI's multiplicative conjunction:

$$\mathcal{M}, w \vDash \varphi * \psi \text{ iff } \exists x, y \in W \text{ such that } w \in x \circ y, \mathcal{M}, x \vDash \varphi \text{ and } \mathcal{M}, y \vDash \psi$$

Hence, even though PRSPDL is not a resource logic since resources usually have a non-deterministic commutative monoid structure, PRSPDL still is a logic with separation, its parallel composition being separating.

PRSPDL differs from both MBIc and DMBI in the kind of currency which is expressible. A first expression of concurrency can be captured by a formula of the form $\langle\alpha\rangle\varphi * \langle\beta\rangle\psi$ which informally means that the current state can be divided in two substates, one from which the action α can be executed to reach a substate satisfying φ and another substate from which the action β can be executed to reach a substate satisfying ψ. This kind of concurrency is the only one expressible in DMBI and is expressible in MBIc and PRSPDL too. The drawbacks of such formula are that it does not ensure that the resulting substates after the parallel execution of α and β can be substates of a global final state and even if that global final state exists, the formula can not express its properties. In PRSPDL, the formula $\langle(\alpha\,;\varphi?) \parallel (\beta\,;\psi?)\rangle\chi$ ensures that the final global state exists and satisfies χ. In MBIc, another kind of concurrency can be expressed by formulas of the form $\langle a \parallel b\rangle\varphi$. This formula informally means that the actions a and b can be executed synchronously to reach a state (or process) satisfying φ. A similar property can be expressed in PRSPDL by the same formula $\langle a \parallel b\rangle\varphi$. But in PRSPDL atomic actions are not executed synchronously and a program of the form $(a\,;b) \parallel a$ is executable in PRSPDL whereas it is not expressible in MBIc.

Since the operator $*$ is not associative in PRSPDL, PRSPDL's satisfiability problem is decidable and has been proved to be inside 2EXPTIME in [3]. By removing the special programs r_1, r_2, s_1, s_2 and adding the partiality condition, the satisfiability problem becomes easier and has been proved to be in NEXPTIME in [7]. A sound and complete tableaux method has been devised for this variant in [3]. Finally by adding the following condition of separation, the satisfiability problem has been proved to be undecidable in [5].

$$\forall w, x_1, y_1, x_2, y_2 \in W,\ \text{if } w \in x_1 \circ y_1 \text{ and } w \in x_2 \circ y_2 \text{ then } x_1 = x_2 \text{ and } y_1 = y_2 \qquad \text{(separation)}$$

The iteration-free fragment of PRSPDL with the previous separation condition has been axiomatized in [2].

5 Separation and epistemic updates

In this section, we present logics with separation designed for epistemic reasoning and update of knowledge or belief bases.

The Epistemic Separation Logic (ESL) [13, 12] extends BBI with one epistemic unary modality K_a for each agent a from a set A of agents. Informally, the formula $K_a\varphi$ means that the agent a knows that φ. A model for ESL is a tuple $\mathcal{M} = (M, \circ, e, (\sim_a)_{a\in A}, V)$ where $(M, \circ, e)$ is a partial commutative monoid of resources, $(\sim_a)_{a\in A}$ is a family of equivalence relations over M and V is a valuation function assigning a subset of M to each propositional variable of the language. The satisfiability relation for the epistemic modalities is:

$$\mathcal{M}, r \vDash K_a\varphi \text{ iff } \forall s \in M,\ \text{if } r \sim_a s \text{ then } \mathcal{M}, s \vDash \varphi$$

Combining multiplicative operators with epistemic modalities leads to some interesting formulas, as illustrated in [12]. For instance, the formula $\varphi \mathrel{-\!\!*} K_a\psi$ intuitively means that if a resource satisfying φ were added to the current resource then agent a would know that ψ. Conversely the formula $\varphi * K_a\psi$ intuitively means that if a resource satisfying φ were removed from the current resource then agent a would know that ψ. A sound and complete tableaux method for ESL is given in [12].

An extension of ESL with public announcement is proposed in [12]. Let us call this extension ESL-PA. ESL-PA extends ESL with the public announcement construct $[\varphi]\psi$. A model for ESL-PA has the same structure as a model for ESL. The satisfiability relation for the public announcement construct is the following:

$$\mathcal{M}, r \vDash [\varphi]\psi \text{ iff } \mathcal{M}, r \vDash \neg\varphi \text{ or } \mathcal{M} \mid \varphi, r \vDash \psi$$

where:

- $\mathcal{M} \mid \varphi = (M, \circ, e, (\sim'_a)_{a\in A}, V)$ and
- $\sim'_a = \sim_a \cap \{(s,t) \in M \times M \mid \mathcal{M}, s \vDash \varphi \text{ iff } \mathcal{M}, t \vDash \varphi\}$.

As illustrated in [12], ESL-PA is very expressive. No proof theory have yet been devised for ESL-PA though.

The Simple Separation Logic (SSL) [21] is an extension of the propositional classical logic with two separating operators $\dot{\wedge}$ and $\dot{\|}$. SSL is interpreted over valuations. A *valuation* is a function from the set of propositional variables of the language to the set $\{0,1\}$. A *partial valuation* is a partial function from the set of propositional variables of the language to the set $\{0,1\}$. A partial valuation V' is *compatible* with a valuation V iff for all $p \in \mathrm{dom}(V')$, $V'(p) = V(p)$. In that case, V is called an *extension* of V'. A *partition* of the valuation V is a set $\{V_1, V_2\}$ of partial valuations such that $\{\mathrm{dom}(V_1), \mathrm{dom}(V_2)\}$ is a partition of the set of propositional variables and both V_1 and V_2 are compatible with V. The satisfiability relation for the separating operators of SSL is as follows:

- $V \vDash \varphi \mathbin{\dot{\wedge}} \psi$ iff there is a partition $\{V_1, V_2\}$ of V such that for *all* extensions V_1' and V_2' of V_1 and V_2, $V_1' \vDash \varphi$ and $V_2' \vDash \psi$.
- $V \vDash \varphi \mathbin{\dot{\|}} \psi$ iff there is a partition $\{V_1, V_2\}$ of V such that for *some* extensions V_1' and V_2' of V_1 and V_2, $V_1' \vDash \varphi$ and $V_2' \vDash \psi$.

Let us consider a boolean formula β representing a belief base and a boolean formula ψ, called the input formula, representing a new fact. The expression $\beta \circ \psi$ denotes a set of valuations corresponding to the result of updating (or revising) the belief base β with the new information ψ. In this context of belief revision or update, it is postulated in [21] that the separating operators of SSL express two kind of independence:

- $\dot{\wedge}$ expresses the independence of resources: the base $\beta_1 \mathbin{\dot{\wedge}} \beta_2$ can be updated by updating β_1 and β_2 independently.

$$(\beta_1 \mathbin{\dot{\wedge}} \beta_2) \circ \psi = (\beta_1 \circ \psi) \cap (\beta_2 \circ \psi) \qquad (\mathrm{REL}_s)$$

- $\dot{\|}$ expresses the independence of processes: updating by the input formula $\psi_1 \mathbin{\dot{\|}} \psi_2$ can be performed by updating by ψ_1 and ψ_2 independently.

$$\beta \circ (\psi_1 \mathbin{\dot{\|}} \psi_2) = (\beta \circ \psi_1) \circ \psi_2 = (\beta \circ \psi_2) \circ \psi_1 \qquad (\mathrm{REL}_d[3])$$

A PSPACE upper bound for both the model-checking and the satisfiability problem of SSL is given in [21] by a translation into the Dynamic Logic of Propositional Assignments (DL-PA) [4, 22].

6 Conclusion

In this report we have listed a great amount of logics with separation and update. Each of this logics express a different kind of dynamic and/or separation. All this logics have been studied as part of the ANR project DynRes. As a conclusion, we briefly enumerate all the results produced as part of the project in the field of modal logics with separation and update.

[3] In this equation, $\beta \circ \psi_1$ and $\beta \circ \psi_2$ must be understood as the formula corresponding to the set of valuations.

- The logics DBI [14, 12], LSM [12], DMBI [15, 12], ESL [13, 12], ESL-PA [13, 12] and SSL [21] have been invented.
- The undecidability of BBI [25] and of some variants of PRSPDL [5] has been proved.
- The decidability of BI [19], SSL [21] and some variants of PRSPDL [1, 3] has been shown.
- Some upper bounds have been stated for the complexity of the satisfiability problems: PSPACE for SSL [21], NEXPTIME for some variants of PRSPDL [1, 7] and 2EXPTIME for PRSPDL [3].
- Sound and complete tableaux methods have been devised for BI [19], BBI [24], DBI [14, 12], LSM [12], DMBI [15, 12], ESL [13, 12] and for some variants of PRSPDL [3].
- Sound and complete axiomatizations have been devised for BBI [18] and for some variants of PRSPDL [2].

References

[1] Philippe Balbiani and Joseph Boudou. Decidability of iteration-free PDL with parallel composition. In *ADDCT workshop*, 2014.

[2] Philippe Balbiani and Joseph Boudou. Iteration-free PDL with storing, recovering and parallel composition: a complete axiomatization. *J. Logic and Computation*, 2015. to appear.

[3] Philippe Balbiani and Joseph Boudou. Tableaux methods for propositional dynamic logics with separating parallel composition. In Amy P. Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 539–554. Springer, 2015.

[4] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: A well-behaved variant of PDL. In *LICS*, pages 143–152. IEEE Computer Society, 2013.

[5] Philippe Balbiani and Tinko Tinchev. Definability and computability for PRSPDL. In *Advances in Modal Logic*, pages 16–33. College Publications, 2014.

[6] Mario R. F. Benevides, Renata P. de Freitas, and Jorge Petrúcio Viana. Propositional dynamic logic with storing, recovering and parallel composition. *ENTCS*, 269:95–107, 2011.

[7] Joseph Boudou. Exponential-size model property for PDL with separating parallel composition. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *Mathematical Foundations of Computer Science*, volume 9234 of *LNCS*, pages 129–140. Springer, 2015.

[8] James Brotherston and Max I. Kanovich. Undecidability of propositional separation logic and its neighbours. In *LICS*, pages 130–139. IEEE Computer Society, 2010.

[9] Luís Caires and Luca Cardelli. A spatial logic for concurrency (part I). *Inf. Comput.*, 186(2):194–235, 2003.

[10] Matthew Collinson and David J. Pym. Algebra and logic for resource-based systems modelling. *Mathematical Structures in Computer Science*, 19(5):959–1027, 2009.

[11] Matthew Collinson, David J. Pym, and Chris M. N. Tofts. Errata for *Formal Aspects of Computing* (2006) 18: 495-517 and their consequences. *Formal Asp. Comput.*, 19(4):551–554, 2007.

[12] Jean-René Courtault. *Logiques de ressources dynamiques: modèles, propriétés et preuves*. PhD thesis, Université de Lorraine, 2015.

[13] Jean-René Courtault, Hans van Ditmarsch, and Didier Galmiche. An epistemic separation logic. In *WoLLIC*, volume 9160 of *LNCS*, pages 156–173. Springer, 2015.

[14] Jean-René Courtault and Didier Galmiche. A modal BI logic for dynamic resource properties. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 7734 of *LNCS*, pages 134–148. Springer, 2013.

[15] Jean-René Courtault and Didier Galmiche. A modal separation logic for resource dynamics. *Journal of Logic and Computation*, 2015. to appear.

[16] Stéphane Demri and Morgan Deters. Separation logics and modalities: A survey. *J. Applied Non-Classical Logics*, 25(1):50–99, 2015.

[17] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.

[18] Didier Galmiche and Dominique Larchey-Wendling. Expressivity properties of boolean BI through relational models. In *FSTTCS*, volume 4337 of *LNCS*, pages 357–368. Springer, 2006.

[19] Didier Galmiche, Daniel Méry, and David J. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15(6):1033–1088, 2005.

[20] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic logic*. MIT press, 2000.

[21] Andreas Herzig. A simple separation logic. In Leonid Libkin, Ulrich Kohlenbach, and Ruy J. G. B. de Queiroz, editors, *WoLLIC*, volume 8071 of *LNCS*, pages 168–178. Springer, 2013.

[22] Andreas Herzig. Belief change operations: A short history of nearly everything, told in dynamic logic of propositional assignments. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *KR*. AAAI Press, 2014.

[23] Ágnes Kurucz, István Németi, Ildikó Sain, and András Simon. Decidable and undecidable logics with a binary modality. *Journal of Logic, Language and Information*, 4(3):191–206, 1995.

[24] Dominique Larchey-Wendling and Didier Galmiche. Exploring the relation between Intuitionistic BI and Boolean BI: an unexpected embedding. *Mathematical Structures in Computer Science*, 19(3):435–500, 2009.

[25] Dominique Larchey-Wendling and Didier Galmiche. The undecidability of boolean BI through phase semantics. In *LICS*, pages 140–149. IEEE Computer Society, 2010.

[26] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[27] David J Pym. *The semantics and proof theory of the logic of bunched implications*, volume 26 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.

[28] David J. Pym and Chris M. N. Tofts. A calculus and logic of resources and processes. *Formal Asp. Comput.*, 18(4):495–517, 2006.

[29] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS*, pages 55–74. IEEE Computer Society, 2002.