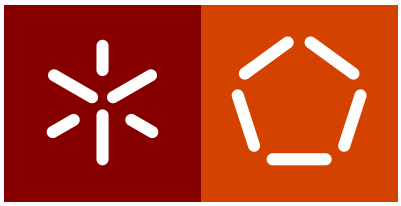


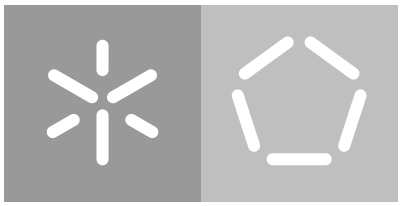
Universidade do Minho
Escola de Engenharia
Departamento de Informática

Ana Luzia Cruz

Exploring Paraconsistent Logics for Quantum Programs

Paraconsistent transition systems
Modal paraconsistent logic

December 2021



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Ana Luzia Cruz

Exploring Paraconsistent Logics for Quantum Programs

Paraconsistent transition systems
Modal paraconsistent logic

Master dissertation
Integrated Master's in Physics Engineering

Dissertation supervised by
Luís Soares Barbosa
Alexandre Madeira

December 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ACKNOWLEDGEMENTS

I once read that we have no friends, only teachers. I am lucky to have found great teachers who are great friends. I would like to thank professor Alda Pregueiro, professor Rosa Sousa, professor José Lopes and all the teachers I had the opportunity to learn with in the last 18 years. Mostly, I would like to thank professor Luís Soares Barbosa and professor Alexandre Madeira for leading me through the quest of developing the work here presented, for their suggestions and insights and for their brightness. Knowledge is either a useless or a dangerous thing in the hands of those who don't know how to share it. Thank you all for sharing your beautiful minds.

I would like to thank all my family members and all my friends for always supporting me and heartening me to succeed. No step is the last step. Thank you for witnessing and greeting all my steps, even those which lead me nowhere.

ABSTRACT

Superconducting quantum circuits are a promising model for quantum computation, although their physical implementation faces some adversities due to the hardly unavoidable decoherence of superconducting quantum bits. This problem may be approached from a formal perspective, using logical reasoning to perform software correctness of programs executed in the non-ideal available hardware. This is the motivation for the work developed in this dissertation, which is ultimately an attempt to use the formalism of transition systems to design logical tools for the engineering of quantum software.

A transition system to capture the possibly unexpected behaviors of quantum circuits needs to consider the phenomena of decoherence as a possible error factor. In this way, we propose a new family of transition systems, the Paraconsistent Labelled Transition Systems (PLTS), to describe processes that may behave differently from what is expected when facing specific contexts. System states are connected through transitions which simultaneously characterize the possibility and impossibility of that being the system's evolution. This kind of formalism may be used to represent processes whose evolution is impossible to be sharply described and, thus, should be able to cope with inconsistencies, as well as with vagueness or missing information. Besides giving the formal definition of PLTS, we establish how they are related under the notions of morphism, simulation, bisimulation and trace equivalence.

It is a common practice to combine transition systems through universal constructions, in a suitable category, which forms a basis for a process description language. In this dissertation, we define a category of PLTS and propose a number of constructions to combine them, providing a basis for such a language.

Transition systems are usually associated with modal logics which provide a formal setting to express and prove their properties. We also propose a modal logic, more specifically, a modal intuitionistic paraconsistent logic (MIPL), to talk about PLTS and express their properties, studying how the equivalence relations defined for PLTS extend to relations on MIPL models and how the satisfaction of formulas is preserved along related models.

Finally, we illustrate how superconducting quantum circuits may be represented by a PLTS and propose the use of PLTS equivalence relations, namely that of trace equivalence, to compare circuit effectiveness.

Keywords: Paraconsistency, Transition systems, Modal logic, Quantum computation.

RESUMO

Os circuitos quânticos que operam qubits supercondutores são um modelo promissor para a arquitetura de computadores quânticos. No entanto, a sua implementação física pode tornar-se ineficaz, devido a fenómenos de decoerência a que os qubits em questão estão altamente sujeitos. Uma possível abordagem a este problema consiste em empregar a lógica e as suas ferramentas para a correção de programas a executar nestes dispositivos.

A proposta desta dissertação é que se utilize o formalismo dos sistemas de transição para modelar e descrever o comportamento dos circuitos quânticos, que, por vezes, pode ser imprevisível. Para tal, considera-se a decoerência de qubits como um possível fator de erro nas computações. Assim surge uma nova família de sistemas de transição, os *Paraconsistent Labelled Transition systems* (PLTS), como um modelo para descrever processos que, em determinados contextos, se comportam de forma diferente do que é esperado. Os estados de um PLTS estão conectados por transições que caracterizam, simultaneamente, a possibilidade e a impossibilidade de o sistema evoluir transitando de um estado para o outro. Este é um modelo em que a informação acerca das transições pode ser incompleta ou mesmo contraditória. Além da definição formal dos PLTS, são também sugeridas, como relações entre PLTS, as noções de morfismo, simulação, bissimulação e equivalência por traços.

Muitas vezes, os sistemas de transição são combinados através de construções universais numa categoria adequada, de forma a definir uma álgebra de processos. Também neste trabalho é definida uma categoria de PLTS e são propostas algumas construções, típicas nas álgebras de processos, para os combinar.

O sistemas de transição são geralmente associados a lógicas modais, que permitem expressar e provar as suas propriedades. A definição dos PLTS conduziu à definição de uma lógica modal, MIPL, que permitiu determinar de que forma as relações de equivalência definidas para PLTS, e estendidas para modelos da lógica MIPL, se refletem na preservação da satisfação de fórmulas sobre os modelos relacionados.

Por fim, propõe-se utilizar PLTS para a representação de circuitos quânticos e comparar a eficácia dos circuitos através da relação de equivalência por traços.

Palavras-chave: Paraconsistência, Sistemas de transição, Lógica modal, Computação quântica.

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

Logic for what?

Logics are used to study the validity of arguments, or in other words, to verify if a particular statement, the conclusion, may be fairly inferred from a set of other statements, the premises. To do so, it is necessary to define a collection of rules that determines what conclusions follow from the assertion of a number of statements, called a *theory*. That is, if the information contained in a theory is assumed or known to be true, the logic rules of inference establish what other statements are provable, allowing to derive further truths. The concept of provability needs to be complemented with the notion of truth so that it is possible to make distinction between arguments that are *valid*, or, in other words, logically provable, and arguments that in addition to being valid also lead to a true conclusion, called *plausible* arguments. The difference between a conclusion being provable and a conclusion being true is illustrated by the argument below.

All computer scientists are logicians.
All logicians are mathematicians.
Thus, all computer scientists are mathematicians.

This argument has the form of a *syllogism*, a particular kind of argument defined by Aristotle. The last statement, the conclusion, is provable from the previous ones, the premises. Indeed, if the premises are assumed to be true, then the conclusion is necessarily true. However, it is reasonable to question the veracity of the information expressed by the premises, and if at least one of the premises is false then it is no longer possible to ensure the veracity of the conclusion. To distinguish true and false statements, logics are complemented with *semantics*, whose role is to interpret statements according to their *truth value*. When logics are enriched with a semantical layer, the veracity of premises, or their truth value, is accounted for in establishing if a conclusion is true or false.

Classical logics: when contradiction and triviality are inseparable

Classical logics are a family of logics whose semantics is usually bivalent, meaning that each statement has one of two possible truth values: true or false. Bivalent classical logics obey the *Principle of Non-Contradiction*. Formulated by Aristotles, and later by Łukasiewicz [Jaś69a], the Principle of Non-Contradiction states that two *contradictory* propositions cannot be simultaneously true. Aristotle considers two kinds of simple propositions, affirmations and denials, which he defines as statements affirming the presence and the absence of a characteristic in a subject, respectively. To Aristotle, the affirmation and denial of a given characteristic in the same subject form a pair of *contradictory* statements, since from the fact that one is true, it must follow that the other is false, and vice-versa. Then, a logic to reason over Aristotle's simple propositions must entail a set of rules that ensures it is never possible to derive both an affirmation and its denial as valid conclusions of a given theory, as long as the theory is itself free from contradictions.

In bivalent classical logics, the Principle of Non-Contradiction may be formulated in terms of one of the two following rules. Considering a negation connective $\neg$, which reads as "not", conjunction $\wedge$, which reads as "and", and disjunction $\vee$, which reads as "or", the first rule, usually referred to as the law of non-contradiction, asserts that a *logical contradiction*, *i.e.* a sentence of the form $(p \wedge \neg p)$, is false under all circumstances: that is, a sentence of the form $\neg(p \wedge \neg p)$ is always true, regardless of the meaning of p. This has a direct logical interpretation, it simply says that p and $\neg p$ should never be simultaneously provable. The other rule, known as the law of the excluded middle, asserts that either p is true or $\neg p$ is true, for any proposition p, implying that a sentence of the form $p \vee \neg p$ is always provable. It expresses the semantical consideration pointed by Aristotle: if p is true, then $\neg p$ is necessarily false, and vice-versa. Actually, in bivalent classical logic these two rules are derivable from each other, that is, each of them may be defined in terms of the other, and therefore the conclusions one reaches by considering them separately or both together are exactly the same.

Nonetheless, it is possible to design other logical systems where these rules are not dual. It is even possible to design logics whose set of rules does not include them, and furthermore it is possible to design logics where some of the connectives we used to state these rules are not even available... Aristotle knew, although being convinced that the Principle of Non-contradiction was a fundamental and ultimate truth, that classical logics would not be a universal reasoning tool. For instance, he noticed that *modal propositions*, *i.e.* those sentences asserting or denying "possibility or contingency, impossibility or necessity", would represent an exception to his principle: a sentence like "it may be that..." and its negation are not contradictory statements, but in fact they imply each other. From the fact that "it may be" it follows straightforwardly that "it may not be" as well. He concludes that this kind of propositions have a different behavior with respect to negation than that of simple propositions, but considers that an alternative definition of their contradictory is needed to

solve the issue and ensure that the Principle of Non-Contradiction is met. Such sentences are the object of analysis of *Modal Logics*. Aristotle introduced modal propositions in his Term Logic for syllogisms but "modern" modal logic was founded by C. Lewis, with his famous *S*-systems [LL34]. A possibility connective $\Diamond$ is used to express the modal status of a proposition p: $\Diamond p$ reads as "it is possible that p" and $\neg\Diamond\neg p$ ("it is not possible that not p") is interpreted as "it is necessary that p". Because of the duality between possibility and necessity, these are considered "classical" modal logics. A necessity connective $\Box$ may be introduced in the logic but is redundant, as it can be expressed in terms of $\Diamond$ and $\neg$. Latter, Kripke developed the *possible world semantics* for modal logics, based on Leibniz's conception of possible worlds. The "possible worlds" are understood as points of evaluation, each having its own assignment of truth values for non-modal propositions. A semantic model includes a set of possible worlds and a binary relation, the *accessibility relation*, establishing "connections" between the worlds, needed for the evaluation of modal propositions. A modal formula $\Diamond p$ is "true" in a world w if p is "true" in at least one world accessible from w; $\Box p$ is "true" if p is "true" in all worlds accessible from w.

Łukasiewicz objects that the Principle of Non-Contradiction applies solely to objects from which one obtains a sensible perception and should not be taken as a basic logical principle, "since it is valid only as an assumption". In fact, a resembling consideration was already pointed out, in different terms, by Aristotle, when explaining why the principle would not apply to modal formulas or sentences of a future tense. That propositions of a future tense represent an objection to the Principle of Non-Contradiction is well illustrated in Aristotle's famous argument of the sea-fight: in short, there is no way of knowing today which of the sentences *"A sea-fight will take place tomorrow"* or *"A sea-fight will not take place tomorrow"* is true, but clearly, either a sea-fight will take place tomorrow or not. Although our intuition that only one of these sentences is true (and the other false) seems correct, which is which is impossible to determine unequivocally. Moreover, this applies to any sentence about a subject whose existence is somehow limited in time, *"that which is not always existent or not always nonexistent"* - in the Sagirite's words. Then, the logic that rules what exists *actually* - at all instants of time - should not be the same as that which rules what exists *potentially*, also called the *undeterminate*, since the Principle of Non-Contradiction fails to describe the latter. If, like Aristotle, we accept the undeterminate as an existing object, or moreover, if we acknowledge that existing objects, as in Łukasiewicz formulation, might not be perceptible, then the Principle of Non-Contradiction is no longer a basic logical principle that applies to all existing things. In fairness, even before the Aristotelian principle was formulated, Heraclitus (535 - 475 BC) was already convinced that such a principle would fail to describe reality. Instead, based on the assumption that existing objects continuously change throughout time, Heraclitus believed that all things an object can possibly become, including something that is not at a present time, must *potentially exist* within it at its present

form. An object may have one characteristic today and the opposite characteristic tomorrow and these two contradictory characteristics must potentially exist within it simultaneously. The principle of Non-Contradiction applies to the non-contradictory, but these arguments suggest that some existing things do not conform to the principle because they live in a contradictory state (or do they live in a contradictory state because they do not conform to the principle?). Łukasiewicz also points out the fact that many logical principles hold independently of whether *the Principle of Non-Contradiction* holds or not and, as mentioned above, there is no reason to believe that a logic allowing contradiction cannot be constructed in the first place.

More recently, with the development of quantum theory [BvN37], the mysterious indeterminate object found a concrete physical realization in quantum systems. Indeed, a quantum system may be described by a collection of characteristics, possibly opposite, representing the potential states the system may evolve towards. A paradigmatic example of how a quantum system behaves is that two copies of the same system, meaning that they have exactly the same potentialities, may, upon the same interaction, evolve to different states. Thus, before such interaction, it is not possible to determine, but only to predict, what characteristic will accurately describe the system and, just as *the Principle of Non-Contradiction* does not apply to the indeterminate object, it does not apply to quantum systems.

The notion of *consistency* is used to characterize logics where no contradictory statements are simultaneously provable, *i.e.* classical logics. On the other hand, if for a certain logic statements p and $\neg p$ are simultaneously provable, giving rise to what we intuitively call a contradiction, then the logic under consideration is said to be inconsistent. [CCM07]

Classical logics become *trivial* in the presence of contradictions. For example, consider the Propositional Calculus:

The statement $(p \vee \neg p)$ is taken as a rule and thus it is always provable, for any proposition p. Starting with this generic statement and simply applying other rules of the Propositional Calculus, it is possible to derive the rule: $(p \wedge \neg p) \rightarrow q$ [Com98]. This means that a sentence of the form $(p \wedge \neg p) \rightarrow q$ is always provable, for any statements p and q in the Calculus. Given a contradictory theory, *i.e.* a theory which contains at least one contradiction, Propositional Calculus rules of inference set each and every statement in the Calculus as a valid consequence of that theory. In particular, any other contradiction is itself a consequence following from that theory.

Ideally, what is fairly inferred from a theory is true: a deductive system which verifies this property is said to be *sound*. The fact that any statement within the Propositional Calculus, or any other classical logic, is a consequence of a contradictory theory Γ is suspicious. Indeed, if a contradiction, which is supposed to be a false statement, is derived from Γ, then classical logics fail to distinguish truth from falsity when reasoning over contradictory theories. That is why under such circumstances, classical logics become trivial: among the

conclusions of a contradictory theory are truths and falsities, but the logic fails to make distinction between them. In a sense, truth and falsity collapse into same thing when reasoning over contradictory information.

Why reasoning in the presence of contradiction?

Logics that become trivial in the presence of a contradiction are said to satisfy *the Principle of Explosion* or to be *explosive*. Whenever the Principle of Explosion holds in a logic, a theory where a contradiction occurs entails all possible consequences. In fact, it is the Principle of Explosion that condemns contradictory theories to dud. The sense of studying contradictory theories is recovered as long as they are not trivial. In particular, for an explosive logic, all theories where at least one contradiction occurs are equivalent, since they entail exactly the same set of consequences. It may be difficult to internalize that truth or falsehood may be properly derived from a non-consistent body of knowledge, but it is probably easier to convince ourselves that not all contradictory theories within a logic are equivalent. In order to make distinction between them the Principle of Explosion must be given up. Furthermore, it is quite lazy and unambitious to say that studying non-consistent bodies of knowledge is worthless, especially with so many real examples where the information at our disposal is even expected to be contradictory. For instance, when we try to track down some event to which there are multiple witnesses it is not surprising that these heterogenous sources may provide conflicting information.

The fact that classical logics have their scope of application limited to what is consistent does not challenge their effectiveness in this domain, nor their practical importance. It should be enough evidence to say that classical logic is used to describe the behavior of *Boolean circuits* which serve as a basis for numerous digital components used nowadays as building blocks of classical computers. That being said, to question the universality of the Principle of Non-contradiction serves the purpose of trying to understand if other domains, besides the consistent one, may be subject to logical reasoning, without detracting from the merits of classical logics.

Although Aristotle's main concern was (probably) to find fundamental truths about the world that surrounds us, logics has throughout the years evolved to a mathematical tool and found lots of practical applications in more specific domains. As said before, there are useful logical systems where the Principle of Non-Contradiction fails and even useful logical systems designed without a negation connective. For instance, the bridge established between logic and computation by the Curry-Howard correspondence equates the natural deduction system (for the **positive** fragment of the Propositional Calculus) with the λ-calculus and its typing rules. Thus, λ-calculus programs may be reduced to proofs in a deduction system where negation is not available, which is an evidence that the nega-

tion connective is not required for a logic to provide useful reasoning. This may suffice to discard the classical intuition that *the Principle of Non-Contradiction* should be taken as an axiom of any logic. In removing the connective $\neg$ from Propositional Calculus the theorem $(p \vee \neg p)$ is also removed, giving a non-explosive version of the Calculus where the statement $(p \wedge \neg p) \rightarrow q$ is no longer provable.

Before we proceed, let us formally define the Principles being evoked. Let L be a logic with a set of formulas Fm and a consequence relation $\vdash$, which is monotonic, reflexive and transitive. Let $\Gamma \subseteq Fm$ be a theory of L, *i.e.* a set of formulas closed under the consequence relation. In addition we assume the connective negation $\neg$ is available in L.

Principle of Non-Contradiction: L is non-contradictory if
$(\exists \Gamma)(\forall \alpha \in Fm) : \Gamma \nvdash \alpha$ or $\Gamma \nvdash \neg\alpha$.

Principle of Non-Triviality: L is non-trivial if $(\exists \Gamma)(\exists \alpha \in Fm) : \Gamma \nvdash \alpha$.

Principle of Explosion: L is explosive if $(\forall \Gamma)(\forall \alpha)(\forall \beta)\Gamma, \alpha, \neg\alpha \vdash \beta$.

These principles are stated as in [CCM07]. The following assertions are easily seen to be true: a *trivial* logic is both *contradictory* and *explosive*; a *contradictory* logic is *trivial* **if and only if** it is *explosive*.

Contradiction and paraconsistency

Reasoning in the presence of contradiction has proved to be particularly useful and important in the discipline of *dialectics*, where it is accepted that fair conclusions may follow from asserting contradictory propositions. Such a discipline applies, for instance, in resembling debate and trying to establish the truth from conflicting points of view. The Socratic dialogues are a well known form of the dialectic method. Nonetheless, the first system of dialectical logic, where the co-existence of contradictory propositions appears as a fundamental necessity of the reasoning process, is attributed to Hegel [Jaś69b], in spite of the large debate on whether Hegel's *dialectical contradictions* constitute logical contradictions in the above sense. Assuming this is so, there are still two alternative points of view about Hegel's dialectics: the first is to consider, like Popper, that (it conforms to classical logic and thus) is trivial; the second is to consider that it does not conform to classical logic, as argued by Priest [PBW18]. Priest is convinced that the central notion of contradiction in Hegel is indeed the logical one and that Hegel's dialectics is explained by *dialetheism*, *i.e.* the view that *true* logical contradictions, or *dialethias*, exist, which opposes to the hypothesis sustained by *the Principle of Non-Contradiction* that all contradictions are necessarily false. In classical logics, the behavior of negation and conjunction (for instance, given in the form of truth tables) is such that a logical contradiction is always a "false" statement. While

classical logics have a semantic valuation for propositions which makes them either "true" or "false", dialetheic logics have a more expressive semantics admitting a third possible truth value that could be read as "true and false". The introduction of this truth value is justified by the consideration that some propositions are indeed "two-way" truths, where truth and falsity coexist and overlap. In dialetheic logic, the negation of a "true and false" proposition is itself "true and false" and so is the conjunction of two "true and false" propositions. With this artifact it is possible for some contradictions, not all, to be "true and false" which to a dialetheist implies they don't meet the Aristotelian principle. A logic where the law of non-contradiction does not hold, like dialetheic logic, is said to be *paraconsistent*. In short, paraconsistent logics exclude the Principle of Explosion, allowing some theories to be contradictory yet non-trivial. Priest suggests that classical logics applies to "the static and changeless", which are *consistent*, and that dialetheic logic has a much more general domain, giving a description to what is dynamic as well. The dynamic is by nature contradictory or *inconsistent*. Indeed, in classical logics, consistency is equated with freedom from contradiction, meaning that once a contradiction occurs, consistency may no longer be recovered. But this notion acquires a more expressive meaning in dialetheic (or more generically, paraconsistent) logics, where it is a characteristic attributable to any proposition. In dialetheic logics, consistent propositions are interpreted as either "true" or "false", and inconsistent propositions are interpreted as "true and false". Paraconsistent logics deal with inconsistencies in a more expressive way since contradictions are not necessarily equivalent to one another and not all contradictory theories necessarily entail the same set of consequences. Indirectly, inconsistency has been defined as a glut in a propositions truth value. Paraconsistent logics are then extensions of classical logic, preserving its behavior for what is consistent, represented by truth values "true" and "false", but endowed with a notion of inconsistency at a propositional level. They are considered weaker forms of classical logics since they extract less consequences from a theory, or at most the same set of consequences.

The definition of a Paraconsistent logic as above coincides with Stanislaw Jaskowski formulation [Jaś69b]. A logic is paraconsistent if it does not fulfill *the Principle of Explosion*. This is, for a theory Γ of a logic L in which negation $\neg$ is available

$$(\exists\Gamma)(\exists\alpha)(\exists\beta) : \Gamma, \alpha, \neg\alpha \nvdash \beta.$$

Another possible definition was pointed out by Newton da Costa who argued that a logic is paraconsistent *wrt* to negation $\neg$ if it serves as a basis for $\neg$-contradictory yet non-trivial theories [CCM07] . That is

$$(\exists\Gamma)(\exists\alpha)(\exists\beta) : (\Gamma \vdash \alpha \text{ and } \Gamma \vdash \neg\alpha \text{ and } \Gamma \nvdash \beta).$$

Paraconsistent logics are inconsistent, just like trivial logics. But whereas trivial logics allow any inference, the same does not apply to paraconsistent logics. From this fact follows a third definition of a paraconsistent logic:

a logic is paraconsistent if it is inconsistent yet non-trivial.

There may be special formulas within a logic, called *bottom particles* and oftenly denoted by $\bot$, which are sufficient to trivialize any theory of a given logic. Formally, a *bottom particle* ξ is such that

$$(\forall \Gamma)(\forall \beta)\Gamma, \xi \Vdash \beta.$$

The symbol $\bot$ is commonly called absurd or, sometimes, **contradiction**. In the scope of classical logics, a formula that consists of a contradiction is a *bottom particle*. In order to dissociate contradiction from triviality it is necessary to establish that a contradiction may or may not be a bottom particle. Paraconsistent logics allow (some) contradictions to be different from the absurd $\bot$ and that is how contradiction and triviality are turned apart. This could be done, for instance, by defining consistency at a propositional level: then, a consistent proposition is associated with a contradiction which is a bottom particle, whereas an inconsistent proposition is associated with anon-trivial contradiction.

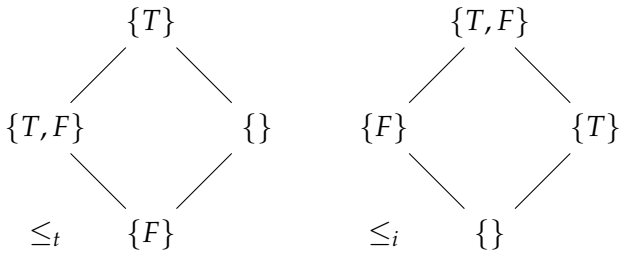
Incompleteness and Intuitionism

From the definition given above, it is straightforward that the law of non-contradiction fails in a paraconsistent logic but this has no direct implications on its dual, the law of the excluded middle. It is possible to have a paraconsistent logic where the law of the excluded middle holds and, on the other hand, there are logics, termed *intuitionistic*, where the law of the excluded middle fails but the law of non-contradiction holds. The first system of intuitionistic logic is attributed to Kolmogorov [Usp92], in 1925, but the basic observations leading to the development of this logic are due to Brouwer [vAS15]. Bouwer argues that the law of the excluded middle is abstracted from finite situations and he gives convincing mathematical evidence that it does not extend properly to statements about infinite collections. In classical logics, the law of the excluded middle is sustained by the semantical consideration that truth and falsity are mutually exclusive (they are dual *wrt* negation and the only available options for assigning a truth value to a proposition). Intuitionistic logic is different because in a sense it allows the truth value of propositions to be left undetermined, thus challenging the law of the excluded middle. As in dialetheic logics, this is achieved by extending semantics to encompass a third truth value, that could read as "neither true nor false". Classical logic requires knowledge about whether the propositions in a theory are false or true, but in intuitionistic logic the weaker hypothesis that some propositions are not known to be true and not known to be false allows reasoning in a broader domain, starting from much less specific assumptions. These propositions may be defined as *vague* or as having a gap in their truth value, opposing to those which are distinctly "true" or "false". While the existence of truth value gluts challenges the law of non-contradiction, the

existence of truth value gaps challenges the law of the excluded middle and leaves room for *vagueness*, or the lack of information.

Infinitely many truth values

Intuitionistic and Paraconsistent logic are logics with more than two semantical truth values. Motivated by Aristotles' discussion on the principle of Non-Contradiction, Łukasiewicz also developed a three-valued logic where the semantic value "possible", besides the typical "true" and "false", is introduced to handle modal propositions. It can be argued that the third truth value represents the lack of information and as such Łukasiewicz logic is within the intuitionistic domain. He also proposed a logic where the truth space is the interval $[0, 1]$, replacing the notion of "truth value" for a "truth degree", and completely rejecting the principle of bivalence, which states that every proposition is either true or false. Once again, the standard interpretations stress that the truth space $[0, 1]$ may represent the lack of information, but not its excess. This form of *many-valued logic*, with infinitely many truth degrees, was afterwords termed *fuzzy logic*. The first attempt to glue intuitionistic and paraconsistent logics together is Belnap's Four-Valued logic [Bel77], where a semantics is given by the four-element lattice $\mathcal{FOUR}$. The possible truth values in $\mathcal{FOUR}$ are the classical "true" and "false", together with "both true and false", to represent the excess of information, and "neither true nor false", to represent the lack of information. The elements of $\mathcal{FOUR}$ may be represented as subsets of $\{T, F\}$ where T and F stand for the classical truth values "true" and 'false". Moreover, it is possible to define two orderings in $\mathcal{FOUR}$: a truth ordering, $\leq_t$ and an information ordering, $\leq_i$, presented below.



With this kind of semantics, the veracity of sentences is complemented with considerations about their informative character.

The discussion above proves that in some situations classical logic is not expressive enough to be a good reasoning tool. Classically, truth and falsity are opposite characteristics attributable to any proposition, but experience seems to suggest that some propositions are characterized by being simultaneously true and false. A logic splitting reality into what is true and what is false obviously excludes from its description what is in essence

contradictory. On the other hand, classical "true" and "false" are also insufficient to correctly assign a truth value to propositions about the future, since one cannot predict it, or any other proposition characterized by incomplete information. This is well illustrated by the argument of the sea-fight. Paraconsistent and Intuitionistic logic, respectively, seem to handle these two situations.

Logics for computation

To add to the picture, logic is no more an abstract domain of mathematics, but also a key ingredient for the development of computational systems. All operations a computer performs are the result of some Boolean circuit execution at the machine level and these are designed following the rules of the two-element Boolean algebra, which is a specific mathematical structure that captures the properties of the classical propositional calculus. On the other hand, functions defined in a recursive programming language may be encoded in λ-calculus expressions and this correspondence extends to an association between programs consisting in the execution of functions for specific input arguments and proofs in the intuitionistic variant of bivalent logics.

The use of computers and the nature of programs has developed far beyond the execution of input-output functions and the possibilities of recursive programming languages. Indeed, input-output functions are only suited for programs which are supposed to terminate and are not adequate for dealing with programs representing dynamic systems as, for instance, operating systems. This class of programs requires more expressive logics to capture such properties, allowing to talk and reason about their states, changes or evolutions and results. This leads to the development of different variants of *Dynamic logics* [HTK00].

Correctness of classic imperative programs was first addressed by the *Floyd-Hoare logic* [Flo67, Hoa69] manipulating Hoare triples, usually written as $\{\varphi\}\pi\{\psi\}$, where π represents a program. The validity of a Hoare triple proves partial correctness of a program and it means that starting from a state where φ is satisfied, the execution of π results in state where ψ is satisfied, or π does not terminate. Moreover, for total correctness, it is also necessary to prove that π terminates.

More complex computational processes are generally described as a set of configurations and the available transitions from one configuration to another, which gives rise to a *transition system* [Kel76, Plo04, MN04]. Many families of transition systems have been defined, according to the nature of the transitions between process configurations, also called *worlds* or *states*. In their simplest version, transitions are given by a binary accessibility relation that expresses whether a transition between two configurations s_1 and s_2 exists or not. Since there are only two possibilities, either a transition exists or it doesn't, this is considered the classical framework. It is also possible to consider a non-classical binary relation, for instance taking values in $[0,1]$. Such is the case of *probabilistic transition systems* [LS91] and

fuzzy transition systems [KS14] where the $[0,1]$-interval values are interpreted respectively as a probability or as a certainty degree of a transition being present. Often transition systems allow transitions to have a specific label encoding some action the system needs to perform for the transition to occur. They are then qualified as *labelled transition systems*.

From a mathematical point of view, transition systems are a kind of *relational structure*, just as Kripke models or graphs are, and thus modal languages provide a precise way to describe and prove some of their properties. Dynamic logics are extensions of modal logics complemented with an algebra of regular events, where programs are described as syntactic constructions. This is essentially the result of using modal logic to perform Floyd-Hoare reasoning. Whereas modal logics are simply concerned with proving propositions, incorporating the alethic modalities of possibility and necessity related to the notion of reachability in a relational frame, dynamic logics take programs as a built in syntactic category expressing properties of transitions. Loosely speaking, modal logics allow to argue whether a a particular result may be attained from some configuration, while dynamic logics include a modal core to express properties of computational states and also a programming language to express the properties of transitions between states.

As computational systems grow in complexity to capture a great variety of systems, from the standard imperative systems to interactive systems, probabilistic systems [Koz85], hybrid systems [Pla10] or, more recently, quantum systems [BS11], many dynamic logics emerged, defined over suitable transition models, in order to reason and verify their behavior.

Quantum computation comes in

Quantum computers challenge the fundamental basis of classical computation. Indeed, classical and quantum computers correspond to very distinct physical machines. As classical computers use bits for information storage, which means information storage units may be assigned with one out of two possible values, 0 and 1, quantum computers units of information are quantum bits, which besides the classical states 0 or 1, could be in a superposition of these definite states. Considering a unit of information as state vectors, with vectors $|0\rangle$ and $|1\rangle$ the representing states 0 and 1, respectively, a superposition state is given by a linear combination of $|0\rangle$ and $|1\rangle$, that is $\alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are complex numbers subject to a normalization condition $|\alpha|^2 + |\beta|^2 = 1$. Superposition states represent a computational advantage since they could be thought as encoding a much vast scope of information [NC11].

Quantum circuits are an under-development model for quantum computation. Based on the classical formalism of Boolean circuits, they use quantum gates to encode operations to be performed over qubits, wich are then fed to these gates. To make use of these devices, algorithms should be convertible in sequential gate executions over a collection of

qubits. A problem in quantum computation is the decay of quantum superposition states to their ground state $|0\rangle$, a phenomenon known as *decoherence*. It occurs due to unwanted interactions with the system which cause interference. Since state of the art qubits are still very fragile and susceptible to interferences, quantum circuits are still very prone to error especially in the context of NISQ (Noisy Intermediate-Scale Quantum) technology [Pre18] in which levels of decoherence of quantum memory need to be articulated with the length of the circuits to assess program quality.

While the demand to construct effective quantum machines continues, either by incorporating more robust hardware or by conceiving alternative architectures, the use of logical tools for verifying quantum programs executing in the currently available hardware is a must for the development of a mature quantum software engineering discipline. With this motivation, in this dissertation we propose a model for quantum circuit computations in the form of a transition system. For this model to be robust it should describe the desired circuit execution as well as the realistic results of that execution when facing decoherence. Since decoherence is not an exact measure, but usually given in terms of a time interval, the states of the transition systems we propose are "connected" by two accessibility relations, one describing how coherent a transition could be and other describing how decoherent a transition might be. The information regarding the presence of a transition is used to model opposite scenarios simultaneously. The affirmative and negative perspectives approached simultaneously provide a basis to analyse the impact of qubit decoherence in quantum circuits and guidance in circuit optimization. As usual, a modal logic is required to express and verify properties of these transition systems. Such was the motivation for this dissertation. Along its development a new sort of transition systems was proposed, the corresponding algebra defined and a suitable logic built.

This work

In this work we define a new family of transition systems, called Paraconsistent Labelled Transition Systems (PLTS), as a modeling formalism where transitions are weighted in two opposite ways: one represents the evidence of the transition being present and other represents the evidence of it being absent. These two weights come from a residuated lattice and jointly express scenarios of consistency, inconsistency and vagueness. Furthermore, we develop a modal logic for reasoning about these systems and also define a category of such transition systems and propose a number of constructions to combine them, providing a possible framework for parallel (superconducting based) quantum computations. We finish with the proposal of a method to convert quantum circuits in PLTS and illustrate how the logic may be used to express program properties and compare their effectiveness.

Dissertation structure

Chapter 2: Paraconsistent Labelled Transition Systems

In this chapter we give the formal definition of a Paraconsistent Labelled Transition System, where each transition is characterised by three components, two of which, as mentioned above, quantifying the evidence for its existence and non-existence. To better understand the nature of these latter components and how they relate to each other, the chapter includes the definition of a residuated lattice, along with some examples, as well as the definition of a twist-structure. The introduction of the truth space that underlies our model is followed by the model definition, the definition of a morphism between PLTS, a state preserving relation, and the definitions of simulation, bisimulation and trace equivalence between PLTS, which allow us to inspect behavioral equality.

Chapter 3: Constructions over PLTS

Here we define a category of PLTS and propose a number of categorical constructions to operate over pre-existing PLTS and obtain a new ones, either through modifications in one PLTS, as in the operations of restriction, relabelling, prefixing and other less common constructions, or through the combination of two (or more) PLTS, with the operations of parallel composition and sum. The work developed in this chapter serves as a basis for the definition of a process algebra to study concurrent processes modeled by PLTS, since (some of) the universal constructions we have defined correspond to the operations of process calculi.

Chapter 4: MIPL - A modal intuitionistic paraconsistent logic

In this chapter, PLTS are regarded as semantic models for the definition of a modal logic called MIPL. MIPL models consist of a relational frame given by a PLTS and a valuation of atomic propositions in each world of the PLTS. The valuation of propositions consists of pairs which are elements of a twist-structure over a residuated lattice.

Our grammar includes a familiar non-modal core but differently from classical modal logics we consider four modal operators: $\Box$ and $\Diamond$, giving formulas whose satisfaction is witnessed by the evidence for the existence of PLTS transitions; and $\not\Box$ and $\not\Diamond$, giving formulas whose satisfaction is witnessed by the evidence against the existence of PLTS transitions. We give a recursive definition for the satisfaction of formulas in a MIPL model and prove some tautologies and semantical equivalences in any MIPL model. We define relations of simulation as bisimulation between MIPL models and inspect what preservations regarding the satisfaction of formulas one might expect to find for PLTS associated with either of

these relations.

Chapter 5: Conclusion

This chapter begins with a brief summary of the contributions presented in Chapter 2, Chapter 3 and Chapter 4.

Following our baseline motivation, we included the proposal of a method to convert quantum circuits into PLTS, giving a transition system like formalism for the description of superconducting quantum computations. Transitions' evidence of existence and non-existence are determined by best and worst case scenarios values of qubit coherence, allowing to model both scenarios simultaneously.

Finally, we give a glimpse of the future developments and improvements to the work presented in this thesis.

2

PARACONSISTENT LABELLED TRANSITION SYSTEMS PLTS

In a generic Labelled Transition System, connections between states are tagged by some action or program and transitions are regarded as the result of performing such action. In Weighted Transition Systems (WTS) each transition is associated with a numerical value. For example, a particular family of WTS, the Fuzzy Transition Systems, where the weights of transitions belong to the interval $[0,1]$, is used to capture a notion of uncertainty with respect to the existence of transitions, so that models incorporate transitions for which there is no absolute conviction on whether they are possible or not. Usually in this framework, the weights are not merely numerical values, but elements of a fuzzy truth space, *i.e.* an algebraic structure over that interval. In this dissertation, we propose a family of transition systems as models to simultaneously characterize information, possibly incomplete or contradictory, regarding the existence and non-existence of a transition between states, in terms of a positive and a negative relation. The first requirement is to define the algebraic structure where the positive and negative relations will take their values and afterwords it is possible to define the model in a generic way.

2.1 TRUTH SPACE

Graded transitions considered in the model proposed in this dissertation take values in a truth space whose definition is based on the notion of a residuated lattice [WD38]. Residuated lattices (over a set A) are algebraic structures with signature $\Sigma = \langle \sqcap, \sqcup, \odot, \hookrightarrow, e \rangle$ of arity $(2,2,2,2,0)$ where:

- $\langle A, \sqcap, \sqcup \rangle$ is a lattice
- $\langle A, \odot, e \rangle$ is a monoid and
- the operation $\odot$ is residuated, with $\hookrightarrow$ being its right residuum, that is, for all $a, b, c \in A$

$$a \odot b \leq c \Leftrightarrow b \leq a \hookrightarrow c$$

As a direct consequence of this adjunction, we have

$$\text{if } a \leq b \text{ then } (c \hookrightarrow a) \leq (c \hookrightarrow b) \tag{1}$$

$$\text{if } a \leq b \text{ then } (b \hookrightarrow c) \leq (a \hookrightarrow c) \tag{2}$$

We will only consider *complete* residuated lattices, *i.e.* lattices where any arbitrary subset of A has an infimum and a supremum. Any complete lattice is *bounded* by a maximal element and a minimal element, which we denote by 1 and 0, respectively. Another requirement is that the residuated lattice is *integral*, that is, with $1 = e$.

In an integral residuated lattice the following property, which plays an essential role in the associated logic, holds (see [MNM16] for a proof).

$$a \hookrightarrow (b \hookrightarrow c) = (b \sqcap a) \hookrightarrow c \tag{3}$$

Two other conditions need to be considered. First, a prelinearity condition

$$(a \hookrightarrow b) \sqcup (b \hookrightarrow a) = 1 \tag{4}$$

is enforced; the resulting structure is known as a MTL-algebra in the literature [EG01]. The other condition is that $\hookrightarrow$ is the residuum of $\sqcap$, which leads operations $\sqcap$ and $\odot$ to coincide.

In this thesis, the term *MTL-algebra* will be used to refer to a residuated lattice satisfying all the conditions mentioned above. Since the identity of the monoid coincides with the top lattice element and the operations $\sqcap$ and $\odot$ coincide, these will be omitted and we will write the tuple $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ to designate an MTL-algebra.

Example 1.

1.1 A first example of this structure is a Boolean algebra over $\{0, 1\}$,

$$\mathbf{2} = \langle \{0, 1\}, \wedge, \vee, 1, 0, \rightarrow \rangle$$

with the standard interpretation of the Boolean connectives.

1.2 Another example, well-known from the fuzzy logic literature, is the Gödel algebra

$$G = \langle [0, 1], min, max, 0, 1, \rightarrow \rangle$$

where $max, min, \rightarrow \colon [0, 1]^2 \longrightarrow [0, 1]$ are defined for any $a, b \in [0, 1]$ as follows.

$$max(a, b) = \begin{cases} a \text{ if } a \geq b \\ b \text{ otherwise} \end{cases} ; min(a, b) = \begin{cases} a \text{ if } a \leq b \\ b \text{ otherwise} \end{cases} ; a \rightarrow b = \begin{cases} 1, \text{ if } a \leq b \\ b, \text{ otherwise} \end{cases} .$$

The following properties will be needed in the sequel.

Lemma 1. *Let* $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, o, \hookrightarrow \rangle$ *be a complete MTL-algebra. Then, for any* $a_1, \ldots, a_n, b \in A$

$$b \hookrightarrow \left(\bigsqcap_i a_i\right) = \bigsqcap_i (b \hookrightarrow a_i) \tag{5}$$

$$\left(\bigsqcup_i a_i\right) \hookrightarrow b = \bigsqcap_i (a_i \hookrightarrow b) \tag{6}$$

$$b \hookrightarrow \left(\bigsqcup_i a_i\right) = \bigsqcup_i (b \hookrightarrow a_i) \tag{7}$$

$$\left(\bigsqcap_i a_i\right) \hookrightarrow b = \bigsqcup_i (a_i \hookrightarrow b) \tag{8}$$

where $\bigsqcap$ *and* $\bigsqcup$ *are the distributed versions of* $\sqcap$ *and* $\sqcup$*, respectively.*

The proof of properties Eq. (5) - Eq. (8) can be found in [BEGR09]. Actually, Eq. (5) and Eq. (6) are true for any residuated lattice.

One novel aspect of the transition systems we define is that transitions between states are characterized by three distinct components, one being an element of a designated set of atomic actions $\{p, q, r, s, ...\}$, interpreted as an action label in Labelled Transition systems, and the other two being elements of an MTL-algebra, which characterize each transition in opposite ways: one represents the evidence of its presence and other the evidence of its absence. We refer to these as the positive accessibility relation and the negative accessibility relation, respectively. Our main concern is to study the case where the positive and negative accessibility relations are elements of an MTL-algebra over $[0, 1]$. In this fuzzy setting, 0 and 1 are maximal pieces of information; all other values are considered to carry some degree of uncertainty.

For each transition, the values of the positive and negative accessibility relations form a pair which may be considered as an element of a bilattice [Gin86], since it is possible to define a truth ordering $\preccurlyeq_t$ and an information ordering $\preccurlyeq_i$ on such pairs. For a complete lattice $\langle A, \leq \rangle$, Fitting [Fit89] defines these two orderings as follows: for any (a, b), (c, d) $\in A \times A$

- $(a, b) \preccurlyeq_t (c, d)$ if and only if $a \leq c$ and $b \geq d$
- $(a, b) \preccurlyeq_i (c, d)$ if and only if $a \leq c$ and $b \leq d$.

As one easily checks, this construction over **2** gives Belnap's bilattice $\mathcal{FOUR}$. Here, the points (a, b),(c, d) such that $(a, b) \not\preccurlyeq_i (c, d)$ and $(c, d) \not\preccurlyeq_i (a, b)$ are simply $(1, 0)$ and $(0, 1)$, and the points (a', b'),(c', d') such that $(a', b') \not\preccurlyeq_t (c', d')$ and $(c', d') \not\preccurlyeq_t (a', b')$ are simply $(1, 1)$ and $(0, 0)$. Indeed, the definitions given above for the truth and information orderings correctly apply and are in accordance with the interpretation given to the elements in $\mathcal{FOUR}$.

However, this construction over a fuzzy lattice has a less convincing interpretation. Consider, as an example, that the pairs $(0.1, 0.9)$ and $(0.2, 0.2)$ are elements of a bilattice $\langle [0, 1] \times$

$[0,1], \preccurlyeq_t, \preccurlyeq_i\rangle$. According to the definition, $(0.1, 0.9) \not\preccurlyeq_i (0.2, 0.2)$ and $(0.2, 0.2) \not\preccurlyeq_i (0.1, 0.9)$. In the present context, the pairs give the evidence of a transition existing and not existing, respectively, so the fact that $(0.2, 0.2) \not\preccurlyeq_i (0.1, 0.9)$ misses our intuition that a transition with a positive accessibility relation of 0.1 and a negative accessibility relation of 0.9 is fully characterized, while a transition with positive and negative accessibility relations of 0.2 seem to represent the case where the information about that transition is incomplete. Moreover, one could also ask if there is a more expressive way of ordering the pairs (a, b) and (c, d) such that $(a, b) \not\preccurlyeq_t (c, d)$ and $(c, d) \not\preccurlyeq_t (a, b)$.

Whether it is possible to define a bilattice over truth and information orderings in a more expressive way leaves the scope of this work. Actually, we will consider the pairs defined by the positive and negative accessibility relations as elements of a bilattice $\mathcal{A} = \langle A \times A, \preccurlyeq_t, \preccurlyeq_i\rangle$ where $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow\rangle$ is an MTL-algebra and $\preccurlyeq_t$, $\preccurlyeq_i$ are defined as above. Thus, the information ordering won't suffice to establish if a particular pair represents a scenario of inconsistency, arising when the values for presence and absence of a transition are contradictory; vagueness, when the values for presence and absence of a transition are neither complementary nor contradictory; or consistency, when the values for presence and absence of a transition are complementary.

When $\mathbf{A}$ is a fuzzy lattice, *i.e.* a lattice over $[0, 1]$, elements of $\mathcal{A}$ may be characterized in terms of their consistency with the help of the conflation operation $\frown$, defined for $(a, b) \in \mathcal{A}$ as $\frown (a, b) = (1 - b, 1 - a)$ [Fit89], in the following way:

- the pair (a, b) represents inconsistent information, that is, the positive and negative accessibility relations sum to a value greater than or equal to 1, if and only if $\frown (a, b) \preccurlyeq_i (a, b)$
- the pair (a, b) represents vague information, or the positive and negative accessibility relations sum to a value less than or equal to 1, if and only if $(a, b) \preccurlyeq_i \frown (a, b)$
- the pair (a, b) represents consistent information, or the positive and negative accessibility relations sum to exactly 1, if and only if $(a, b) \preccurlyeq_i \frown (a, b)$ and $\frown (a, b) \preccurlyeq_i (a, b)$

Pairs containing contradictory information are represented within the upper triangle in Fig. 1, filled in grey. Those for which the information provided is vague lie in the bottom triangle in Fig. 1, filled in pink. And finally consistent pairs are represented by the red line in Fig. 1. The method above applies to bilattices over $[0, 1]$. However, in Chapter 5 we show how to characterize the consistency of elements in a generic bilattice.

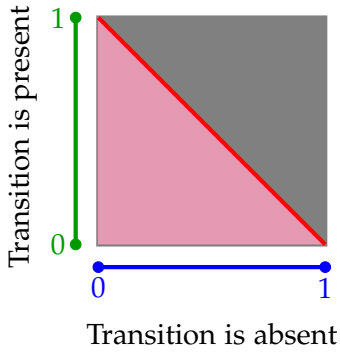


Figure 1: Intuitionistic (pink), paraconsistent (grey) and classic (red) domains characterizing truth degrees for "Transition is present" and "Transition is absent".

Besides conflation, other useful operations may be defined over the bilattice elements characterizing transitions in our model. In particular, some will be important for the definition of the logic in Chapter 4.

Definition 1. Given an MTL-algebra $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$, the algebra $\mathbf{A}^{\bowtie} = \langle \mathbf{A}, \overset{\vee}{\vee}, \overset{\wedge}{\wedge}, \Longrightarrow, \neg \rangle$ is an $\mathbf{A}$-*twist-structure* where the twist-operations are defined for $(a, b), (a', b') \in \mathbf{A} \times \mathbf{A}$ as follows:

- $(a, b) \overset{\vee}{\vee} (a', b') = (a \sqcup a', b \sqcap b')$
- $(a, b) \overset{\wedge}{\wedge} (a', b') = (a \sqcap a', b \sqcup b')$
- $(a, b) \Longrightarrow (a', b') = (a \hookrightarrow a', a \sqcap b')$
- $\neg(a, b) = (b, a)$

Originally a "twist-structure" [Kra98] was defined as a construction over an Heyting algebra, but a construction of this kind had already been proposed under the name of "special $\mathcal{N}$-lattice" [Vak77] and also over a generic bilattice [Gin88]. In [OW10], for instance, the term twist-structure is used to refer to Ginsberg's construction over the bilattice $\mathcal{FOUR}$.

Although the operation $\rightarrow$ is the residuum of $\sqcap$ in a MTL-algebra, the operation $\Longrightarrow$, which corresponds to the weak implication used in [RJJ15], is not residuated in $\mathbf{A}^{\bowtie}$, neither *wrt* the truth ordering nor *wrt* the information ordering. If it was, the following conditions would hold:

- $(a, b) \overset{\wedge}{\wedge} (c, d) \preccurlyeq_t (e, f)$ if and only if $(c, d) \preccurlyeq_t (a, b) \Longrightarrow (e, f)$ and
- $(a, b) \overset{\wedge}{\wedge} (c, d) \preccurlyeq_i (e, f)$ if and only if $(c, d) \preccurlyeq_i (a, b) \Longrightarrow (e, f)$.

A counter-example using the Gödel algebra shows this is not the case. Let $(a, b) = (0.8, 0.4)$, $(c, d) = (0.5, 0.2)$ and $(e, f) = (0.6, 0.3)$. Then $(a, b) \overset{\wedge}{\wedge} (c, d) = (min\{0.8, 0.5\}, max\{0.4, 0.2\}) = (0.5, 0.4)$ and $(a, b) \Longrightarrow (e, f) = (0.8 \rightarrow 0.6, min\{0.8, 0.3\}) = (0.6, 0.3)$. Indeed, $(0.5, 0.4) \preccurlyeq_t (0.6, 0.3)$ but $(0.5, 0.2) \not\preccurlyeq_t (0.6, 0.3)$. On the other hand, $(0.5, 0.2) \preccurlyeq_i (0.6, 0.3)$ but $(0.5, 0.4) \not\preccurlyeq_i (0.6, 0.3)$.

This result may seem odd, since implication is usually defined as a residuum of conjunction but the idea behind the definition of twist-operations, explained by Vakarelov, is the following. The elements of a twist-structure should be treated as sentences: in this case, each transition is associated with an element $(a_1, a_2) \in \mathbf{A}^{\bowtie}$ where the first component is interpreted as the truth degree of "transition is present" and the second as the truth degree of "transition is absent". Thus a_2 should be interpreted as a counterexample to a_1. For (a_1, a_2), $(b_1, b_2) \in \mathbf{A}^{\bowtie}$, the twist-operations $\overset{\vee}{\vee}$, $\overset{\wedge}{\wedge}$, $\Longrightarrow$ and $\dashv$ say how to construct classical counterexamples to $a_1 \sqcup b_1$, $a_1 \sqcap b_1$, $a_1 \to b_1$ and a_2, respectively. The counterexample to a_2 follows straightforward: it is a_1. The counterexamples to $a_1 \sqcap b_1$, $a_1 \sqcup b_1$ and $\neg a_1 \sqcap b$ are given by $\neg(a_1 \sqcap a_2)$, $\neg(a_1 \sqcup a_2)$ and $\neg(\neg a_1 \sqcap b_1)$, which reduce to the expressions in Definition 1 with the application of De Morgan's law.

In [Vak77] it is required that the elements (a, b) of a special $\mathcal{N}$-lattice are such that $a \sqcap b = 0$ *wrt* to the operation $\sqcap$ of the underlying Boolean algebra, which implies that consistency holds for the pairs in a special $\mathcal{N}$-lattice and that there is only one element, *i.e.* $(0, 0)$, such that $(a, b) = \dashv(a, b)$. Although we also interpreter the second element of our twist-structure as a counter-example to the first, we do not impose any consistency requirement, since both values freely take values in the carrier of a residuated lattice, and moreover any element of the form (a, a) satisfies the mentioned property.

2.2 MODEL DEFINITION

Definition 2. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra. An **A**-*paraconsistent labelled transition system* (**A**-PLTS) defined over a set of atomic actions Π is a structure $\langle W, R \rangle$ where:

- W is a finite non-empty set whose elements are called worlds or states;

- $R \subseteq W \times \Pi \times W \times A \times A$ is the *(paraconsistent) accessibility relation* such that, for any two states $w_1, w_2 \in W$ and any $\pi \in \Pi$, there is at most one transition $(w_1, \pi, w_2, \alpha, \beta) \in R$.

The relation R characterizes the transitions between states. Each tuple $(w_1, a, w_2, \alpha, \beta) \in R$ represents a transition from w_1 to w_2 labelled by (a, α, β), where α is the degree to which the action a causes a transition between w_1 and w_2, and β the degree to which the action a prevents (the occurence of) a transition between w_1 and w_2.

Elements in the accessibility relation R of an **A**-PLTS are given by tuples, as above, but may be represented in the following manner:

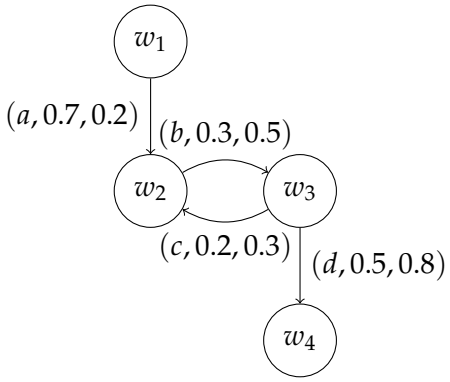
$$w_1 \xrightarrow{(a,\alpha,\beta)} w_2 \equiv (w_1, a, w_2, \alpha, \beta) \in R$$

.

Example 2. Consider the G-PLTS, where G stands for the Gödel algebra, $\langle W, R \rangle$ over $\Pi = \{a, b, c, d\}$ where

- $W = \{w_1, w_2, w_3, w_4\}$ and
- $R = \{(w_1, a, w_2, 0.7, 0.2), (w_2, b, w_3, 0.3, 0.5), (w_3, c, w_c, 0.2, 0.3), (w_3, d, w_4, 0.5, 0.8)\}$

$\langle W, R\rangle$ is depicted below.



Since the labels in a PLTS quantify the degree to which a transition is present, as well as the degree to which it is absent, it is useful to realize a simple way to access either of these values. We do so by splitting the accessibility relation into a **positive accessibility relation** and a **negative accessibility relation**.

Thus, for an **A**-PLTS $\langle W, R\rangle$ over Π its **positive accessibility relation** is a function $r^+ : \Pi \longrightarrow A^{W^W}$ such that for $\pi \in \Pi$ and $w, w \in W$,

$$r^+(\pi, w, w') = \begin{cases} \alpha \text{ if } (w, \pi, w', \alpha, \beta) \in R \\ 0 \text{ otherwise} \end{cases}$$

Similarly, its **negative accessibility relation** is a function $r^- : \Pi \longrightarrow A^{W^W}$ such that for $\pi \in \Pi$ and $w, w \in W$,

$$r^-(\pi, w, w') = \begin{cases} \beta \text{ if } (w, \pi, w', \alpha, \beta) \in R \\ 1 \text{ otherwise} \end{cases}$$

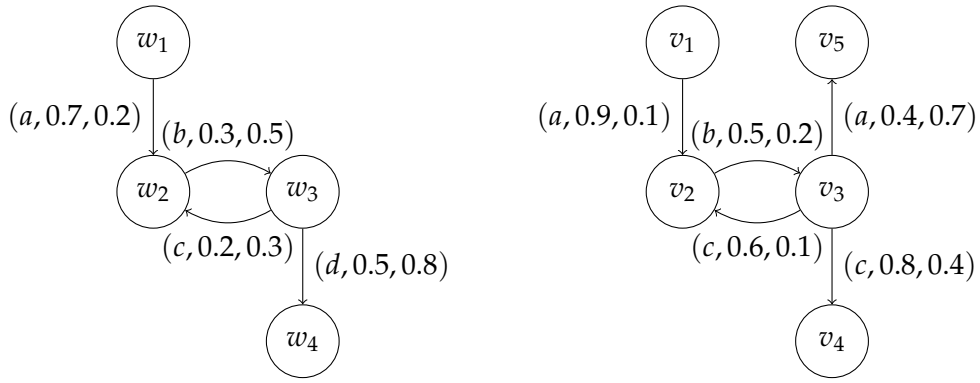
We also use the terms "positive accessibility relation" and "negative accessibility relation" to refer to the values yield by the functions r^+ and r^- for each particular transition of a PLTS. In general, these values will be elements of an MTL-algebra and form a pair which is an element of a bilattice and also of a twist-structure over that bilattice.

2.3 Morphism, Simulation and Bisimulation

Definition 3. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow\rangle$ be an MTL-algebra and let $T_1 = \langle W_1, R_1\rangle$, $T_2 = \langle W_2, R_2\rangle$ be two **A**-PLTS defined over the same set of actions Π. Denote the positive and negative accessibility relations for T_1 and T_2 by r_1^+,r_1^- and r_2^+,r_2^-, respectively. A **morphism**

relating these two PLTS is a function $h : W_1 \to W_2$ such that $\forall \pi \in \Pi$, $r_1^+(\pi, w_1, w_2) \leq r_2^+(\pi, h(w_1), h(w_2))$ and $r_1^-(\pi, w_1, w_2) \geq r_2^-(\pi, h(w_1), h(w_2))$.

Example 3. For $\Pi = \{a, b, c, d\}$ consider two G-PLTS M_1 and M_2 given by the following diagrams.



The mapping $h = \{w_1 \mapsto v_1, w_2 \mapsto v_2, w_3 \mapsto v_3\}$ is a morphism. It is easy to check that h satisfies the conditions of Definition 3.

Definition 4. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra and let $T_1 = \langle W_1, R_1 \rangle$, $T_2 = \langle W_2, R_2 \rangle$ be two $\mathbf{A}$-PLTS defined over the same set of actions Π. Denote the positive and negative accessibility relation functions for T_1 and T_2 by r_1^+,r_1^- and r_2^+,r_2^-, respectively. A relation $S \subseteq W_1 \times W_2$ is a **simulation** provided that, for all $\langle p, q \rangle \in S$, the following condition holds:

- if there is a transition in T_1 from p to p' caused by $a \in \Pi$, then there is a transition in T_2 from q to q' caused by a such that $r_2^+(a, q, q') \geq r_1^+(a, p, p')$, $r_2^-(a, q, q') \leq r_1^-(a, p, p')$ and $\langle p', q' \rangle \in S$.

In short, $S \subseteq W_1 \times W_2$ is a simulation if, for all $\langle p, q \rangle \in S$ and $a \in \Pi$,

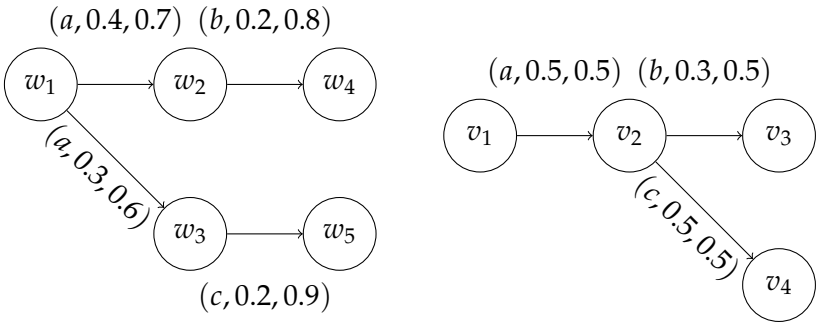
$$p \xrightarrow{(a,\alpha,\beta)}_{T_1} p' \Rightarrow \langle \exists q' \in W_2, \exists \gamma, \delta \in [0,1] : q \xrightarrow{(a,\ \gamma,\ \delta)}_{T_2} q' \wedge \langle p', q' \rangle \in S \wedge \gamma \geq \alpha \wedge \delta \leq \beta \rangle.$$

From here on, the following abbreviation is used to represent the above condition:

$$p \xrightarrow{(a,\alpha,\beta)}_{T_1} p' \Rightarrow \langle \exists q' \in W_2 : q \xrightarrow{(a,\ \gamma:\ \gamma \geq \alpha\ ,\ \delta:\ \delta \leq \beta)}_{T_2} q' \wedge \langle p', q' \rangle \in S \rangle.$$

Two states p and q are **similar**, written $p \lesssim q$, if there is a simulation S such that $\langle p, q \rangle \in S$.

Example 4. In the G-PLTS given by the diagrams below, $w_1 \lesssim v_1$, because there is a simulation, $S = \{\langle w_1, v_1 \rangle, \langle w_2, v_2 \rangle, \langle w_3, v_2 \rangle, \langle w_4, v_3 \rangle, \langle w_5, v_4 \rangle\}$, that contains $\langle w_1, v_1 \rangle$.



Lemma 2. *The similarity relation is a preorder, i.e. a reflexive and transitive relation.*

Proof.

(i) Reflexivity: $p \lesssim p$

This follows from the fact that the identity relation is a simulation. Indeed, for a PLTS $\langle W, R\rangle$, the relation $S \subseteq W \times W$ such that $\langle w, w\rangle \in S$ for all $w \in W$ satisfies the conditions of Definition 4.

(ii) Transitivity: if $p \lesssim_{S_1} q$ and $q \lesssim_{S_2} t$ then $p \lesssim_{S_3} t$

$p \lesssim_{S_1} q \Rightarrow \exists$ a simulation $S_1 : \langle p, q\rangle \in S_1$

$q \lesssim_{S_2} t \Rightarrow \exists$ a simulation $S_2 : \langle q, t\rangle \in S_2$

To prove that $p \lesssim t$ we must find a simulation S_3 such that $\langle p, t\rangle \in S_3$. Let $S_3 = S_2 \cdot S_1$. Indeed, $\langle p, t\rangle \in S_3$ since $\langle p, q\rangle \in S_1$ and $\langle q, t\rangle \in S_2$. Now we must prove S_3 satisfies the conditions in Definition 4.

$\langle p, q\rangle \in S_1 \Rightarrow$ if $p \xrightarrow{(a,\alpha,\beta)}_{T_1} p'$ then $\langle \exists q' : q \xrightarrow{(a,\ \gamma:\ \gamma\geq\alpha\ ,\ \delta:\ \delta\leq\beta)}_{T_2} q' \wedge \langle p', q'\rangle \in S_1\rangle$

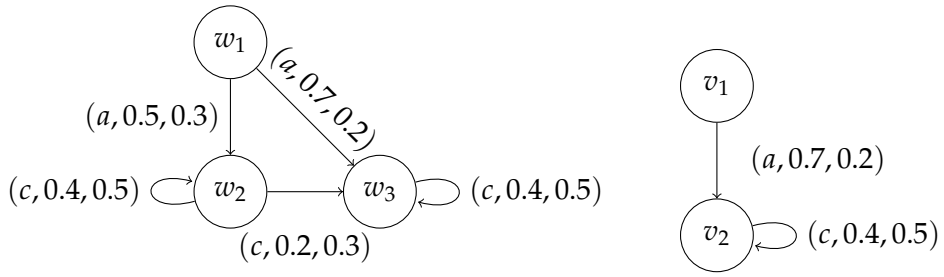
But since $\langle q, t\rangle \in S_2$ then $\exists t' : t \xrightarrow{(a,\ \mu:\ \mu\geq\gamma\ ,\ \nu:\ \nu\leq\delta)}_{T_3} t' \wedge \langle q', t'\rangle \in S_2\rangle$

It is straightforward that $\mu \geq \gamma \Rightarrow \mu \geq \alpha$ and $\nu \leq \delta \Rightarrow \nu \leq \beta$. Thus, for $\langle p, t\rangle \in S_3$, $p \xrightarrow{(a,\alpha,\beta)}_{T_1} p' \Rightarrow \langle \exists t' : t \xrightarrow{(a,\ \mu:\ \mu\geq\alpha\ ,\ \nu:\ \nu\leq\beta)}_{T_3} t'\rangle$. To check that $\langle p', t'\rangle \in S_3$ just notice that $\langle p', q'\rangle \in S_1$ and $\langle q', t'\rangle \in S_2$.

□

Definition 5. Two states p and q are **equisimilar** if $q \lesssim_{S_1} q$ and $q \lesssim_{S_2} p$.

Example 5. Consider the two G-PLTS depicted below and a relation $S = \{\langle w_1, v_1\rangle \langle w_2, v_2\rangle, \langle w_3, v_2\rangle\}$. We have $w_1 \lesssim_S v_1$ and $v_1 \lesssim_S w_1$, so w_1 and v_1 are equisimilar.



Definition 6. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra and let $T_1 = \langle W_1, R_1 \rangle$ and $T_2 = \langle W_2, R_2 \rangle$ be two $\mathbf{A}$-PLTS defined over the same set of actions Π. A relation $B \subseteq W_1 \times W_2$ is a **bisimulation** if for $\langle p, q \rangle \in B$ and $a \in \Pi$

$p \xrightarrow{(a,\alpha,\beta)}_{T_1} p' \Rightarrow \langle \exists q' \in W_2 : q \xrightarrow{(a,\alpha,\beta)}_{T_2} q' \wedge \langle p', q' \rangle \in B \rangle$ and

$q \xrightarrow{(a,\alpha,\beta)}_{T_2} q' \Rightarrow \langle \exists p' \in W_1 : p \xrightarrow{(a,\alpha,\beta)}_{T_1} p' \wedge \langle p', q' \rangle \in B \rangle$.

It follows straightforward that any transition in the first PLTS is mapped to an exactly equal transition in the second PLTS, and vice-versa. Thus, the bisimulation is an equivalence relation, i.e. a reflexive, transitive and symmetric relation.

Two states p and q are **bisimilar**, written $p \sim q$, if there is a bisimulation B such that $\langle p, q \rangle \in B$.

2.4 TRACES AND TRACE EQUIVALENCE

Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra and $\langle W, R \rangle$ be an $\mathbf{A}$-PLTS defined over a set of actions Π. Furthermore let r^+,r^- denote its positive and negative accessibility relations, respectively.

Definition 7. A **path** from $w \in W$ in a PLTS $\langle W, R \rangle$ is a sequence $[(w_1, a_1, w_2), (w_2, a_2, w_3), ...]$ such that $w_i \in W, a_i \in \Pi$ of worlds connected by transitions available in $\langle W, R \rangle$, with $w_1 = w$. The set of all paths from w is denoted by Paths(w).

Example 6. Consider the PLTS given in Example 2. The following are some paths from w_1: $[(w_1, a, w_2)]$, $[(w_1, a, w_2), (w_2, b, w_3)]$, $[(w_1, a, w_2), (w_2, b, w_3), (w_3, c, w_2)]$.

Note that a path, as defined above, only specifies the action causing each transition, and leaves out the values for the accessibility relations. These values are easily obtained using $r+$ and r^-: for each tuple (w, a, w') in a path, $r^+(a, w, w')$ and $r^-(a, w, w')$ are the values for the accessibility relation characterizing the transition to which the tuple refers. We define the function $t : W \times \Pi \times W \to \Pi \times A \times A$ such that $t(w, a, w') = (a, r^+(a, w, w'), r^-(a, w, w'))$.

Thus, from a path $\rho = [(w_1, a_1, w_2), (w_2, a_2, w_3), ...]$ one obtains a sequence of transition labels in a PLTS by mapping each (w_i, a_j, w_k) in ρ to $t(w_i, a_j, w_k)$. The list trait$(\rho) = t^*(\rho)$ is the list derived by successively applying t to the tuples in ρ and we call it **trait of** ρ.

Example 7. We now give the trait of each path pointed out in Example 6.
$\text{trait}[(w_1, a, w_2)] = [(a, 0.7, 0.2)]$
$\text{trait}[(w_1, a, w_2), (w_2, b, w_3)] = [(a, 0.7, 0.2), (b, 0.3, 0.5)]$
$\text{trait}[(w_1, a, w_2), (w_2, b, w_3), (w_3, c, w_2)] = [(a, 0.7, 0.2), (b, 0.3, 0.5), (c, 0.2, 0.3)]$

Definition 8. Given a path $\rho = [(w_1, a_1, w_2), (w_2, a_2, w_3), ...]$ from $w_1 \in W$,

(i) An **unweighted trace** is the sequence of actions $[a_1, a_2, ...]$ in such path, given by $t_{unweighted}(\rho) = P_2^*\rho$, where P_2^* is the sequence extension of P_2, and

(ii) A **weighted trace** is the sequence of actions in such path, together with the maximum value for the positive accessibility relation and the minimum value for the negative accessibility relation occurring in $trait(\rho)$. Formally, a weighted trace is defined by: $t_{weighted}(\rho) = \langle P_2^* p, \bigsqcup(P_2^* trait(\rho)), \bigsqcap(P_3^* trait(\rho))\rangle$, where $\bigsqcup$ and $\bigsqcap$ are the distributed versions of $\sqcup$ and $\sqcap$ in the underlying MTL-algebra, respectively.

Where P_2 and P_3 are the second and third projections of the tuples to which they are applied.

Example 8. For the path $[(w_1, a, w_2), (w_2, b, w_3), (w_3, c, w_2)]$ given in Example 6, with corresponding trait $[(a, 0.7, 0.2), (b, 0.3, 0.5), (c, 0.2, 0.3)]$, its unweighted trace is $t_{unweighted} : t = [a, b, c]$; and its weighted trace is $t_{weighted} = \langle [a, b, c], 0.7, 0.2\rangle$.

Definition 9. Let $t = \langle [a_1, a_2, ..., a_n], \alpha, \beta\rangle, t' = \langle [b_1, b_2, ..., b_m], \gamma, \delta\rangle$ be two weighted traces of some given PLTS(s). We say t is a **subtrace** of t' if the three following conditions hold: (i) the list $[a_1, a_2, ..., a_n]$ is a prefix of $[b_1, b_2, ..., b_m]$, that is, $n \leq m$ and $a_i = b_i$ for $i = 1, 2, ..., n$; (ii) $\gamma \geq \alpha$ and (iii) $\delta \leq \beta$.

Example 9. Let t, t', t'' be the weighted traces correspondent to the first, second and third paths given in Example 6, respectively. Both t and t' are subtraces of t''. For a more complex example, consider the PLTS given in Example 4. The weighted traces from w_1 are $\{t_0 = \langle [a], 0.4, 0.7\rangle, t_1 = \langle [a], 0.3, 0.6\rangle, t_2 = \langle [a, b], 0.4, 0.7\rangle, t_3 = \langle [a, c], 0.3, 0.6\rangle\}$ and the traces from v_1 are $:= \{t'_1 = \langle [a], 0.5, 0.5\rangle, t'_2 = \langle [a, b], 0.5, 0.5\rangle, t'_3 = \langle [a, c], 0.5, 0.5\rangle\}$. Thus, t_0 and t_1 are both subtraces of t'_1, t_2 is a subtrace of t'_2 and t_3 is a subtrace of t'_3.

Definition 10. Let P be the set of all paths from $w \in W$.

(i) The **set of unweighted traces** from w, $\text{Tr}_{unweighted}(w)$, is the set that contains the unweighted traces associated to each element in $\text{Paths}(w)$, that is $\text{Tr}_{unweighted}(w) = \{t_{unweighted}(\rho) | \rho \in \text{Paths}(w)\}$, and

(ii) the **set of weighted traces** from w, $\text{Tr}_{weighted}(w)$, is the set that contains the weighted traces associated to each element in $\text{Paths}(w)$, that is $\text{Tr}_{weighted}(w) = \{t_{weighted}(\rho) | \rho \in \text{Paths}(w)\}$.

For two sets X and Y of unweighted traces, we say $X \subseteq_u Y$ if $\forall_{t \in X} \exists_{t' \in Y} : t = t'$.

For two sets X and Y of weighted traces, we say $X \subseteq_w Y$ if $\forall_{t \in X} \exists_{t' \in Y} : t$ is a subtrace of t'.

Two states p and q are **trace equivalent**, written $\text{Tr}_{weighted}(p) =_w \text{Tr}_{weighted}(q)$, if $\text{Tr}_{weighted}(p) \subseteq_w \text{Tr}_{weighted}(q)$ and $\text{Tr}_{weighted}(q) \subseteq_w \text{Tr}_{weighted}(p)$.

Lemma 3. *Consider two PLTS $T_1 = \langle W_1, R_1 \rangle$ and $T_2 = \langle W_2, R_2 \rangle$. If two states $p \in W_1$ and $q \in W_2$ are similar, i.e., $p \lesssim q$, then the set of weighted traces from p, call it X, and the set of weighted traces from q, call it Y, are such that $X \subseteq_w Y$.*

Proof.

$(p, q) \in S$. Therefore, if $p \xrightarrow{(a,\alpha,\beta)} p'$ then $\exists\ q' : q \xrightarrow{(a,\ \gamma:\ \gamma \geq \alpha\ ,\ \delta:\ \delta \leq \beta)} q' \wedge \langle p', q' \rangle \in S \rangle$

Since $\langle p, q \rangle \in S$, we can perform the same collection of actions in p and in q. Moreover, if we perform the same action in p, causing a transition to p', and in q, causing a transition to q', the pair $\langle p', q' \rangle$ is also in S and again the same collection of actions is available in p' and q'. From this follows that any sequence of actions available in T_1, starting from p, is also available in T_2, starting from q. Or, in other words, the set of unweighted traces from p is contained in the set of unweighted traces from q: $\text{Tr}_{unweighted}(p) \subseteq_u \text{Tr}_{unweighted}(q)$.

Now, for $t \in \text{Tr}_{unweighted}(p)$, let $\{\alpha_1, \alpha_2, ..., \alpha_i\}$ and $\{\beta_1, \beta_2, ..., \beta_i\}$ be the values of the positive and negative accessibility relation, respectively, characterizing each transition in t, so that $T = \langle t, A = \bigsqcup(\{\alpha_1, \alpha_2, ..., \alpha_i\}), B = \bigsqcap(\{\beta_1, \beta_2, ..., \beta_i\}) \rangle$, call it T is the weighted trace corresponding to t.

Furthermore, each element $t \in \text{Tr}_{unweighted}(p)$ is also an element of $\text{Tr}_{unweighted}(q)$. So for $t' \in \text{Tr}_{unweighted}(q) : t' = t$, let $\{\alpha'_1, \alpha'_2, ..., \alpha'_i\}$ and $\{\beta'_1, \beta'_2, ..., \beta'_i\}$ be the values of the positive and negative accessibility relation, respectively, characterizing each transition in t', and similarly $T' \langle t', A' = \bigsqcup(\{\alpha'_1, \alpha'_2, ..., \alpha'_i\}), B' = \bigsqcap(\{\beta'_1, \beta'_2, ..., \beta'_i\}) \rangle$, call it T', is the weighted trace corresponding to t'. Since $p \lesssim q$, we know that $\{\alpha_1 \leq \alpha'_1, \alpha_2 \leq \alpha'_2, ..., \alpha_i \leq \alpha'_i\}$ and $\{\beta_1 \geq \beta'_1, \beta_2 \geq \beta'_2, ..., \beta_i \geq \beta'_i\}$. From this it follows that $A' \geq A$ and $B' \leq B$. Thus, T is a subtrace of T', according to Definition 9. Moreover, this will be true for any trace in $\text{Tr}_{unweighted}(p)$, so the set of weighted traces from p, denoted by X, and the set of weighted traces from q, denoted by Y, verify $X \subseteq_w Y$.

□

Lemma 4. *Consider two PLTS $T_1 = \langle W_1, R_1 \rangle$ and $T_2 = \langle W_2, R_2 \rangle$. If two states $p \in W_1$ and $q \in W_2$ are bisimilar, i.e., $p \sim q$, then they are trace equivalent.*

Proof.

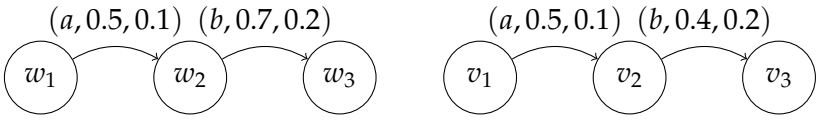
$p \sim q \Rightarrow \exists$ a bisimulation $B : \langle p, q \rangle \in B$

For p and q to be trace equivalent, we require that set of weighted traces from p, X, and the set of weighted traces from q, Y, are such that $X = Y$.

Recall that a bisimulation is a simulation whose converse is also a simulation. So we may write $p \lesssim q$ and conclude, from Lemma 3, that $X \subseteq_w Y$. Since the bisimilarity relation is symetric, we have $p \sim q \Rightarrow q \sim p$, and, again, we may write $q \lesssim p$ and conclude that $Y \subseteq_w X$. □

Remark 1. *Notice that the converse of Lemma 4 is not true: two states may be trace equivalent and not bisimilar.*

Example 10. Consider the PLTS depicted below.



Indeed, $\mathrm{Tr}_{weighted}(w_1) = \mathrm{Tr}_{weighted}(v_1) = \{\langle [a], 0.5, 0.1\rangle, \langle [a, b], 0.7, 0.1\rangle\}$. Nonetheless it is clear that $w_1 \nsim v_1$.

3

CONSTRUCTIONS OVER PLTS

For the next chapter, we follow the formalism in [WN95] to propose some useful constructions over PLTS. Winskel and Nielsen show that a category of Transitions Systems, where the objects are generic Labelled Transition Systems and the arrows correspond to morphisms between them, provides a mathematical formalism for modelling distributed computations, since it is possible to describe the operations of process calculi as universal constructions in this category. Moreover, the same applies to other categories of more generic transition systems with an appropriate definition of morphism. Our goal is to define such a category where the objects are PLTS.

However, we need a definition of morphism different from Definition 3. A morphism was defined as a relation between two PLTS over the same set of actions but this definition will be extended so that a morphism relates PLTS over (possibly) distinct sets of actions to grant a useful model for parallel computation, capable of combining processes over different atomic actions into a valid model. Definition 3 forms a category where PLTS over different sets of actions can not be combined through universal constructions. We adjust the definition of morphism to define category where it is possible to do so.

Now a morphism between two PLTS T and T' will be a pair of functions: a total function from the worlds of T onto the worlds of T' and a partial function from the set of actions of T onto the set of actions of T'. We avoid giving the (new) formal definition of morphism in terms of a partial function by considering any partial function $\lambda : \Pi \to \Pi'$ as a total function $\lambda' : \Pi \to \Pi' \cup \{\bot\}$, also written $\lambda' : \Pi \to_\bot \Pi'$, where $\bot$ is a distinguished element standing for undefined: whenever $\lambda(a)$ is undefined, $\lambda'(a) = \bot$. We consider an extension of the accessibility relation R of any PLTS, given by $R^\bot = R \cup \{(w, \bot, w, 1, 0) | w \in W\}$. Any $t \in R^\bot \setminus R$ is called an **idle transition**. We also extend the definition of a PLTS to incorporate a distinguished initial state, which will be useful for the definition of some particular constructions. We write $T = \langle W, i, R, \Pi \rangle$ for the PLTS $\langle W, R \rangle$ over Π with initial state $i \in W$.

Definition 11. Let $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$ and $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ be two PLTS. An **extended morphism** relating T_1 and T_2 is a pair of functions $(\sigma : W_1 \to W_2 \ , \ \lambda : \Pi_1 \to_\bot \Pi_2)$ such that

- $\sigma(i_1) = i_2$,
- if $(w, a, w', \alpha, \beta) \in R_1$ then $(\sigma(w), \lambda(a), \sigma(w'), \alpha', \beta') \in R_2^{\perp}$, with $\alpha \leq \alpha'$ and $\beta' \leq \beta$.

If two PLTS have the same underlying set of actions then λ is just the identity function and it is easy to check that this definition becomes equivalent to Definition 3.

We define the category $\mathbf{T}_{PL}$ whose objects and arrows are PLTS and extended morphisms between PLTS, respectively. The composition of two morphisms $f = (\sigma, \lambda) : T_0 \to T_1$ and $f' = (\sigma', \lambda') : T_1 \to T_2$ in $\mathbf{T}_{PL}$ is $f' \circ f = (\sigma' \circ \sigma, \lambda' \circ \lambda) : T_0 \to T_2$ and the identity morphism of a PLTS T is $(1_W, 1_\Pi)$ where 1_W identity function on the worlds of T and 1_Π the identity function on the set of actions of T. It is easy to prove that the composition of two extended morphisms is associative and $\mathbf{T}_{PL}$ is indeed a category.

Lemma 5. *Consider four PLTS: $T_0 = \langle W_0, i_0, R_0, \Pi_0 \rangle$, ..., $T_3 = \langle W_3, i_3, R_3, \Pi_3 \rangle$ and extended morphisms $f = (\sigma, \lambda) : T_0 \to T_1$, $g = (\sigma', \lambda') : T_1 \to T_2$ and $h = (\sigma'', \lambda'') : T_2 \to T_3$. Then $h \circ (g \circ f) = (h \circ g) \circ f$.*

Proof.

$$
\begin{aligned}
& h \circ (g \circ f) \\
= & \quad \{ \text{ definition of } h, g, f \ \} \\
& (\sigma'', \lambda'') \circ ((\sigma', \lambda') \circ (\sigma, \lambda)) \\
= & \quad \{ \text{ functuriality } \} \\
& (\sigma'', \lambda'') \circ (\sigma' \circ \sigma, \lambda' \circ \lambda) \\
= & \quad \{ \text{ definition of } \circ \ \} \\
& (\sigma'' \circ (\sigma' \circ \sigma), \lambda'' \circ (\lambda' \circ \lambda)) \\
= & \quad \{ \circ \text{ is associative } \} \\
& ((\sigma'' \circ \sigma') \circ \sigma, (\lambda'' \circ \lambda') \circ \lambda) \\
= & \quad \{ \text{ functuriality } \} \\
& ((\sigma'', \lambda'') \circ (\sigma', \lambda')) \circ (\sigma, \lambda) \\
= & \quad \{ \text{ definition of } h, g, f \ \} \\
& (h \circ g) \circ f
\end{aligned}
$$

□

The PLTS which consists of a single initial state is $T_{nil} = \langle \{*\}, *, \emptyset, \emptyset \rangle$.

Lemma 6. *T_{nil} is the initial and terminal object of the category $\mathbf{T}_{PL}$.*

Proof.

For any other PLTS $T = \langle W, i, R, \Pi \rangle$ in $\mathbf{T}_{PL}$

1. there is a unique morphism $! : T \to T_{\text{nil}}$, given by $((!_W, ()))$, where $!_W(i) = *$ and $(w, a, w', \alpha, \beta) \in R \Rightarrow (*, \bot, *, 1, 0) \in R_{T_{nil}}{}^{\bot}$.

2. Similarly, there is a unique morphism $? : T_{nil} \to T$, given by $(\underline{i}, ())$. Clearly, $\underline{i}(*) = i$ and the other condition is necessarily true.

□

Given a PLTS $T = \langle W, i, R, \Pi \rangle$, a world $w \in W$ is **reachable** if there is a path in T from i to w. We follow the terminology in [WN95] and say T is **reachable** if it is possible to reach every world of T starting from the initial state. Furthermore, T is **acyclic** if there is only one path from each world to itself and it is an idle transition. Morphisms preserve the initial state and also reachable states: that is, if w is reachable in T and there is a morphism $(\sigma, \lambda) : T \to T'$, then $\sigma(w)$ is reachable in T'.

3.1 RESTRICTION

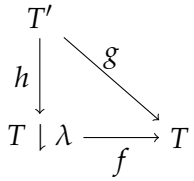
In a generic labelled transition system, the operation of restriction takes a subset of the system's set of labels and removes all transitions whose labels are not in that set. Since the labels represent actions causing transitions, the operation of restriction narrows the actions that the system is able to perform. PLTS have labels with three distinct components but only one refers to the actions causing transitions, thus the operation of restriction has its version on the PLTS setting.

Definition 12. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. For $\Pi' \subset \Pi$ let $\lambda : \Pi' \to \Pi$ be a mapping taking $a \in \Pi'$ to $a \in \Pi$. The **restriction** $T \restriction \lambda$ is a PLTS $\langle W, i, R', \Pi' \rangle$ over Π' where

$$R' = \{(w, \pi, w', \alpha, \beta) \in R | \pi \in \Pi'\}.$$

There is an extended morphism from the restricted PLTS $T \restriction \lambda$ to the original one, given by $f = (1_W, \lambda)$ and a functor $p : \mathbf{T}_{PL} \to \mathbf{Set}_*$ between the category $\mathbf{T}_{PL}$ and the category of sets with partial functions, which sends a morphism $(\sigma, \lambda) : T_1 \to T_2$ of PLTS $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$ and $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$, to the partial function $\lambda : \Pi_1 \to \Pi_2$. The morphism f associated with the restriction is a cartesian morphism in $\mathbf{T}_{PL}$, since it satisfies the following universal property:

For any morphism $g : T' \to T$ in T_{PL} such that $p(g) = \lambda$ there is a unique morphism $h : T' \to T \restriction L$ such that $p(h) = 1_{\Pi'}$ and $f \circ h = g$. In a diagram:

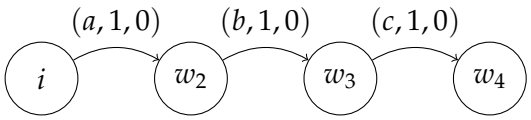


$$\Pi' \xrightarrow{\lambda} \Pi$$

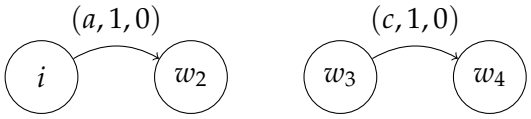
A cartesian morphism $f = (\sigma, \lambda)$ in $\mathbf{T}_{PL}$ is a cartesian lifting of the morphism $p(f) = \lambda$ in $\mathbf{Set}_*$.

In general, the operation of restriction does not preserve the reachable states.

Example 11. Consider the PLTS T depicted below, with initial state i.

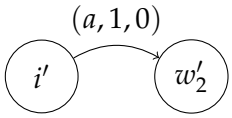


The restriction $T \lfloor \{a, c\}$ is the PLTS:



Which clearly has a distinct set of reachable states from i.

Now consider the following PLTS T'.



The state i of $T \lfloor \{a, c\}$ from Example 11 is bisimilar to the state i' of T'. In terms of reachable states, T' and $T \lfloor \{a, c\}$ have the same behaviors. Thus it could be useful to have an additional construction giving the reachable component of a (restricted) PLTS.

Definition 13. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. The **reachable component** of T is the PLTS $T_{reach} = \langle W', i, R', \Pi' \rangle$ where

$$R' = \{(w, a, w', \alpha, \beta) \in R | w = i\} \cup \{(w, a, w', \alpha, \beta) \in R | \text{ there is a path in T from } i \text{ to } w\}.$$

W' is the set of reachable worlds in T and Π' is the set of actions labelling the transitions in T_{reach}. Clearly, $W' \subseteq W$ and $\Pi' \subseteq \Pi$.

3.2 RELABELLING

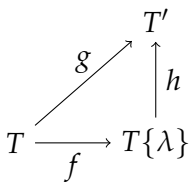
Given a labelled transition system T whose labelling set is Π and a total function $\lambda : \Pi \to \Pi'$, the relabelling construction preserves the underlying structure of T but consistently changes its labels according to λ. In essence, this construction renames the actions in T. It is also possible to define a construction which renames the actions in a PLTS.

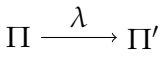
Definition 14. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. Let $\lambda : \Pi \to \Pi'$ be a total function. The **relabelling** $T\{\lambda\}$ is the PLTS $\langle W, i, R', \Pi' \rangle$ where

$$R' = \{(w, \lambda(a), w', \alpha, \beta) | (w, a, w', \alpha, \beta) \in R\}$$

There is a morphism from a PLTS T to the relabeled PLTS $T\{\lambda\}$, given by $f = (1_W, \lambda)$ and it is a cocarteasian morphism, since it satisfies the following universal property:

For any morphism $g : T \to T'$ in T_{PL} such that $p(g) = \lambda$, there is a unique morphism $h : T\{\lambda\} \to T'$ such that $p(h) = 1_{\Pi'}$ and $h \circ f = g$. In a diagram:





A cocartesian morphism in $\mathbf{T}_{PL}$ is associated with a construction dual to the cartesian lifting, which is the cocartesian lifting. Then, a cocartesian morphism $f = (\sigma, \lambda)$ in $\mathbf{T}_{PL}$ is a cocartesian lifting of the morphism $p(f) = \lambda$ in $\mathbf{Set}_*$.

3.3 PARALLEL COMPOSITION

The product of two transition systems is an operation for producing a parallel composition of the two systems. The parallel composition of two systems models a process combining the execution of all processes inherent to each component system. Essentially, states of the product system are obtained through the combination of a state from each component system, and, similarly, transitions are obtained through the combinations of a transition from each component system (idle transitions included). In [WN95] parallelism is modeled through the construction of a product, which combines the two systems allowing all conceivable synchronizations. That is, any combination of states of the component systems is

a valid state in the product system and any transition in the first component system may be performed synchronously with any other transition from the second component system. For this reason, the product operation is combined with the operations of restriction and relabelling in order to remove unwanted synchronizations and produce more specific parallel compositions. The product of two generic LTS is a LTS where transitions are either caused by performing one action in each component system at the same time, or by performing an action at a time in one of the component systems.

Definition 15. Let $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$ and $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ be two PLTS. Their **product** $T_1 \times T_2$ is the PLTS $\langle W_1 \times W_2, (i_1, i_2), R, \Pi \rangle$, such that

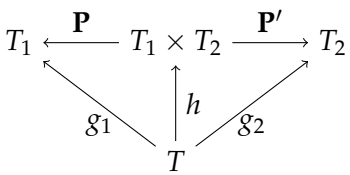
- $\Pi = \Pi_1 \times_\perp \Pi_2 = \{(a, \perp) | a \in \Pi_1\} \cup \{(\perp, b) | b \in \Pi_2\} \cup \{(a, b) | a \in \Pi_1, b \in \Pi_2\}$, and
- $(w, a, w', \alpha, \beta) \in R$ if and only if $(P_1(w), P_1(a), P_1(w'), \alpha_1, \beta_1) \in R_1^\perp$ and $(P_2(w), P_2(a), P_2(w'), \alpha_2, \beta_2) \in R_{2*}$ and $\alpha = \alpha_1 \sqcap \alpha_2$ and $\beta = \beta_1 \sqcup \beta_2$.

The set of actions of a product has elements of the form (a, b), which represent synchronizations between the component processes, that is, the execution of an action by each component simultaneously; and elements of the form $(a, \perp)$ or $(\perp, b)$, which represent the execution of an action by one of the component systems and inaction in the other.

The product $T_1 \times T_2 = \langle W, i, R, \Pi \rangle$ of two PLTS has projection morphisms $\mathbf{P} : T_1 \times T_2 \to_\perp T_1$ and $\mathbf{P}' : T_1 \times T_2 \to_\perp T_2$ given by $\mathbf{P} = (P_1, P_1)$ and $\mathbf{P}' = (P_2, P_2)$. For a synchronous transition $(w, e, w', \alpha, \beta)$ it follows straightforward from Definition 15 that there is a transition $(P_1(w), P_1(e), P_1(w'), \alpha_1, \beta_1) \in R_1$ such that $\alpha \leq \alpha_1$ and $\beta \geq \beta_1$; and a transition $(P_2(w), P_2(e), P_2(w'), \alpha_2, \beta_2) \in R_2$ such that $\alpha \leq \alpha_2$ and $\beta \geq \beta_2$. For asynchronous transitions $(w, (\perp, b), w', \alpha, \beta)$ or $(v, (a, \perp), v', \alpha', \beta')$, note that $P_1(w) = P_1(w')$ and $P_2(v) = P_2(v')$ and indeed $(P_1(w), \perp, P_1(w'), 1, 0) \in R_1^\perp$ and $(P_2(v), \perp, P_2(v'), 1, 0) \in R_2^\perp$. For any values of α, α', β, β' it is true that $\alpha \leq 1$, $\alpha' \leq 1$, $\beta \geq 0$ and $\beta' \geq 0$.

These two morphisms form a product in T_{PL} since they satisfy the following universal property:

For any morphism $g_1 : T \to T_1$ and $g_2 : T \to T_2$, there is a unique morphism $h : T \to T_1 \times T_2$, given by $\langle g_1, g_2 \rangle$, such that $\mathbf{P} \circ h = g_1$ and $\mathbf{P}' \circ h = g_2$. In a diagram:



Proof.

To check that the diagram commutes just note that:

– $(\mathbf{P} \circ h)(x) = \mathbf{P}(\langle g_1(x), g_2(x)\rangle) = g_1(x)$ and
– $(\mathbf{P}' \circ h)(x) = \mathbf{P}'(\langle g_1(x), g_2(x)\rangle) = g_2(x)$.

We must also prove that h is a morphism and that it is unique. Let $T = \langle W, i, R, \Pi\rangle, T_1 = \langle W_1, i_1, R_1, \Pi_1\rangle, T_2 = \langle W_2, i_2, R_2, \Pi_2\rangle, T_1 \times T_2 = \langle W_1 \times W_2, (i_1, i_2), R', \Pi'\rangle$, $g_1 = (\sigma_1, \lambda_1)$ and $g_2 = (\sigma_2, \lambda_2)$. If $(w, a, w', \alpha, \beta) \in R$ then there is a transition $(\sigma_1(w), \lambda_1(a), \sigma_1(w'), \alpha_1, \beta_1) \in R_1^{\perp}$ such that $\alpha \leq \alpha_1$ and $\beta \geq \beta_1$; and also a transition $(\sigma_2(w), \lambda_2(a), \sigma_2(w'), \alpha_2, \beta_2) \in R_2^{\perp}$ such that $\alpha \leq \alpha_2$ and $\beta \geq \beta_2$. Moreover, according to 15 there is a transition

$$(\langle \sigma_1, \sigma_2\rangle(w), \langle \lambda_1, \lambda_2\rangle(a), \langle \sigma_1, \sigma_2\rangle(w'), \alpha_1 \sqcap \alpha_2, \beta_1 \sqcup \beta_2) \in R'$$

. Thus for any $(w, a, w', \alpha, \beta) \in R$ there is a transition

$$(\langle \sigma_1, \sigma_2\rangle(w), \langle \lambda_1, \lambda_2\rangle(a), \langle \sigma_1, \sigma_2\rangle(w'), \alpha', \beta')) \in R'$$

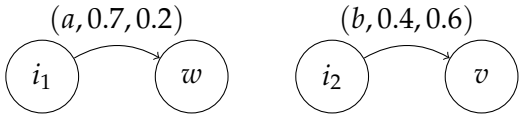
such that $\alpha \leq \alpha'$ and $\beta \geq \beta'$. Furthermore, initial states are preserved since $\langle \sigma_1, \sigma_2\rangle(i) = (\sigma_1(i), \sigma_2(i)) = (i_1, i_2)$ so $h = \langle g_1, g_2\rangle$ is a morphism.

Let $f : T \to T_1 \times T_2$ be some other morphism such that $\mathbf{P} \circ f = g_1$ and $\mathbf{P}' \circ f = g_2$. For some x let $f(x) = \langle a, b\rangle$. Then we have $g_1(x) = (\mathbf{P} \circ f)(x) = \mathbf{P}(f(x)) = \mathbf{P}\langle a, b\rangle = a$ and $g_2(x) = (\mathbf{P}' \circ f)(x) = \mathbf{P}'(f(x)) = \mathbf{P}'\langle a, b\rangle = b$. Therefore $f(x) = \langle g_1(x), g_2(x)\rangle = h(x)$, which proves $f = h$ and thus there is a unique morphism satisfying the universal property.

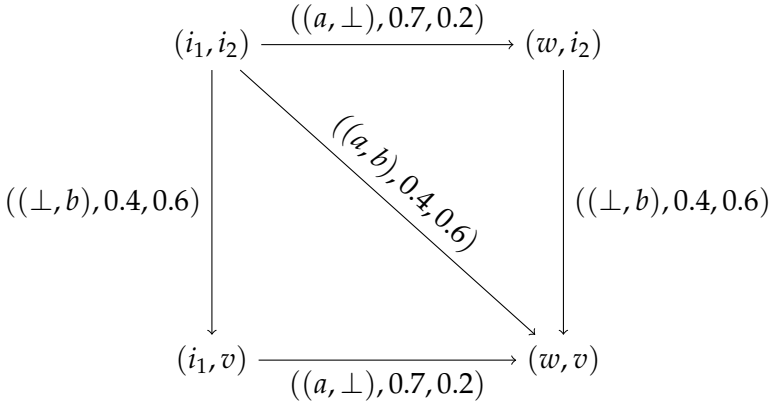
□

A state s is reachable in $T_1 \times T_2$ if and only if $P_1(s)$ is reachable in T_1 and $P_2(s)$ is reachable in T_2. We have only considered binary products but all products exist in $\mathbf{T}_{PL}$, in particular the empty product which is T_{nil}.

Example 12. Consider the PLTS T_1 and T_2 depicted below.



Their product T is the PLTS

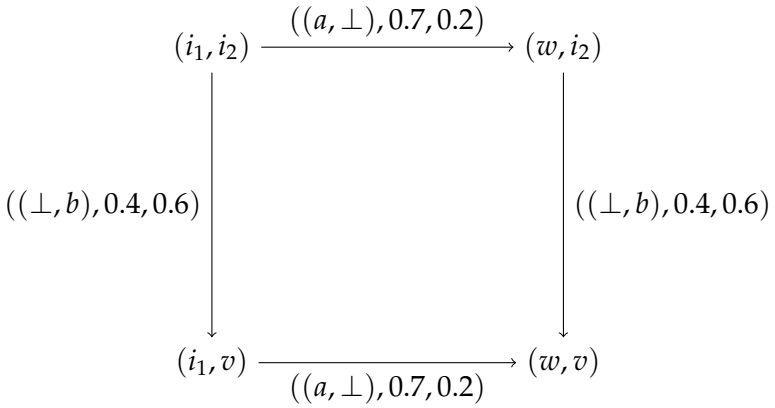


As mentioned above, it is possible to remove unwanted transitions in a PLTS that consists in the product of two transition systems by simply applying operation(s) of restriction. For instance, the parallel composition of two systems where no transitions may be performed simultaneously could be obtained through the construction of the product and then restriction to all possible synchronizations.

Definition 16. Let $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$, $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ be two PLTS and $T_1 \times T_2 = \langle W', i', R', \Pi_1 \times_\perp \Pi_2 \rangle$ be their product. Also let $\Pi = \{(a,\perp) | a \in \Pi_1\} \cup \{(\perp, b) | b \in \Pi_2\}$ and $\lambda : \Pi \to \Pi_1 \times_\perp \Pi_2$ be the mapping taking $x \in \Pi$ to $x \in \Pi_1 \times_\perp \Pi_2$. The **interleaving** or **asynchronous product** $T_1 ||| T_2 \equiv (T_1 \times T_2) \restriction \lambda$ is the PLTS $\langle W_1 \times W_2, (i_1, i_2), R, \Pi \rangle$ such that

$$R = \{(w, a, w', \alpha, \beta) \in R' | a \in \Pi\}.$$

Example 13. For the PLTS in Example 12 this gives the PLTS depicted below.



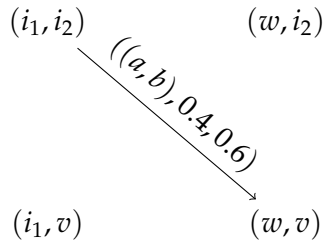
On the other hand, the parallel composition of two systems where only synchronous transitions are allowed could be obtained through the construction of the product and then restriction to all asynchronous transitions.

Definition 17. Let $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$, $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ be two PLTS and $T_1 \times T_2 = \langle W', i', R', \Pi_1 \times_\perp \Pi_2 \rangle$ be their product. Also let $\Pi = \{(a,b) | a \in \Pi_1 \text{ and } b \in \Pi_2\}$ and

$\lambda : \Pi \to \Pi_1 \times_\perp \Pi_2$ be the mapping taking $x \in \Pi$ to $x \in \Pi_1 \times_\perp \Pi_2$. The **synchronous product** $T_1 \otimes T_2 \equiv (T_1 \times T_2) \restriction \lambda$ is the PLTS $\langle W_1 \times W_2, (i_1, i_2), R, \Pi \rangle$ such that

$$R = \{(w, a, w', \alpha, \beta) \in R' | a \in \Pi\}.$$

Example 14. For the PLTS in Example 12 this gives the PLTS depicted below.



3.4 SUM

In process calculi the (nondeterministic) sum of two or more processes defines a process that can behave as each of its component processes. The sum of transition systems must be a construction capturing this property so it is capable of simulating the behavior of the alternative processes modeled by its constituent systems.

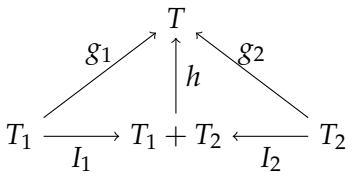
Definition 18. Let $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$ and $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ be two PLTS. Their sum $T_1 + T_2$ is the PLTS $\langle W, (i_1, i_2), R, \Pi_1 \cup \Pi_2 \rangle$, where

- $W = (W_1 \times \{i_2\}) \cup (\{i_1\} \times W_2)$,
- $t \in R$ if and only if $\exists (w, a, w', \alpha, \beta) \in R_1$ such that $t = (In_1(w), a, In_1(w'), \alpha, \beta)$ or $\exists (w, a, w', \alpha, \beta) \in R_2$ such that $t = (In_2(w), a, In_2(w'), \alpha, \beta)$

where In_1 and In_2 are the left and right injections, respectively.

Associated with the sum $T_1 + T_2$ there are injection morphisms $I_1 : T_1 \to T_1 + T_2$ and $I_2 : T_2 \to T_1 + T_2$ given by $I_1 = (In_1, 1_\Pi)$ and $I_2 = (In_2, 1_\Pi)$. These two morphisms form a coproduct in $\mathbf{T}_{PL}$, since they satisfy the following universal property:

For any morphisms $g_1 : T_1 \to T$ and $g_2 : T_2 \to T$, there is a unique morphism $h : T_1 + T_2 \to T$, given by $[g_1, g_2]$, such that $h \circ I_1 = g_1$ and $h \circ I_2 = g_2$. In a diagram:



Proof.

To check that the diagram commutes just note that:

- $(h \circ I_1)(x) = [g_1, g_2](I_1(x)) = g_1(x)$
- $(h \circ I_2)(x) = [g_1, g_2](I_2(x)) = g_2(x)$

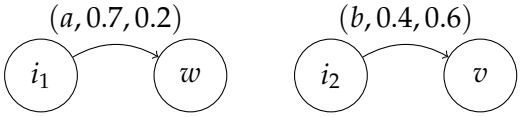
We must also prove that h is a morphism and that it is unique. Let $T = \langle W, i, R, \Pi \rangle$, $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$, $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$, $T_1 + T_2 = \langle W', (i_1, i_2), R', \Pi' \rangle$, $g_1 = (\sigma_1, \lambda_1)$ and $g_2 = (\sigma_2, \lambda_2)$. If $(w, a, w', \alpha, \beta) \in R_1$ then there is a transition $(\sigma_1(w), \lambda_1(a), \sigma_1(w'), \alpha_1, \beta_1) \in R$ such that $\alpha \leq \alpha_1$ and $\beta \geq \beta_1$; and also a transition $(In_1(w), a, In_1(w'), \alpha, \beta) \in R'$. If $(w, a, w', \alpha, \beta) \in R_2$ then there is a transition $(\sigma_2(w), a, \sigma_2(w'), \alpha_2, \beta_2) \in R$ such that $\alpha \leq \alpha_2$ and $\beta \geq \beta_2$; and also a transition $(In_2(w), a, In_2(w'), \alpha, \beta) \in R'$. Thus for any $(w, a, w', \alpha, \beta) \in R'$ there is a transition $([\sigma_1, \sigma_2](w), [\lambda_1, \lambda_2](a), [\sigma_1, \sigma_2](w'), \alpha', \beta') \in R$ such that $\alpha \leq \alpha'$ and $\beta \geq \beta'$. Furthermore, initial states are preserved since $\sigma_1(i_1) = \sigma_2(i_2) = i$, so $h = [g_1, g_2]$ is a morphism.

Now let $f : T_1 + T_2 \to T$ be some other morphism such that $f \circ I_1 = g_1$ and $f \circ I_2 = g_2$. Then $g_1(x) = (f \circ I_1)(x) = f(I_1(x))$ and $g_2(x) = (f \circ I_2)(x) = f(I_2(x))$. On the one hand we have $[g_1, g_2](x) = [f(I_2(x)), f(I_2(x))]$, and on the other hand $[g_1, g_2](x) = [g_1(x), g_2(x)] = [h(I_1(x)), h(I_2(x))]$. Thus $f = h$ and there is a unique morphism satisfying the universal property.

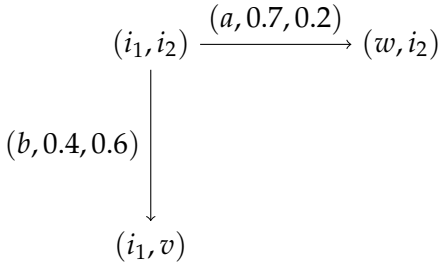
□

A state s is reachable in $T_1 + T_2$ if there is s_1 reachable in T_1 such that $s = In_1(s_1)$ or there is s_2 reachable in T_2 such that $s = In_2(s_2)$. We have only considered the coproduct of two PLTS, but all coproducts exist in $\mathbf{T}_{PL}$.

Example 15. Consider the PLTS T_1 and T_2 depicted below.



Their sum T is the PLTS



3.5 PREFIXING

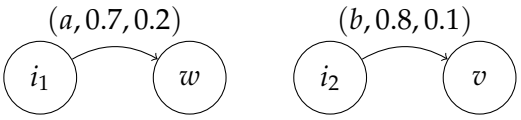
The operation of prefixing in a generic LTS adds a new initial state and introduces a transition connecting it to the former initial state. The process resulting from this operations behaves as the original process after the new initial action has taken place. Then, from a given LTS one constructs a prefix of it by specifying the label for the new transition. We add a new transition to construct the prefix of a PLTS by specifying not only the action causing the transition but also the values for the positive and negative accessible relation.

Definition 19. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS over an MTL-algebra $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$. Given an action $\{a\}$, eventually not in Π, and $\alpha, \beta \in A$ the prefix $(a, \alpha, \beta)T$ is a construction that gives the PLTS $T' = \langle W', i', R', \Pi \cup \{a\} \rangle$ where

- $W' = \{w | w \in W\} \cup \{\varnothing\}$,
- $i' = \varnothing$
- $R' = \{(w, \pi, w', \alpha', \beta') | (w, \pi, w', \alpha', \beta') \in R\} \cup \{(\varnothing, a, i, \alpha, \beta)\}$

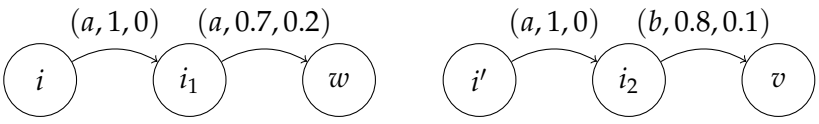
Since it is not required that the prefixing action is distinct from the former actions, the operation of prefixing does not extend to a functor in $\mathbf{T}_{PL}$. This is illustrated in the example below.

Example 16. Consider two PLTS $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$ and $T_2 = \langle W_2, i_2, R_2, \Pi_2 \rangle$ depicted below.



There is an extended morphism $(\sigma, \lambda) : T_1 \to T_2$ given by $\sigma(i_1) = i_2$, $\sigma(w) = v$ and $\lambda(a) = b$.

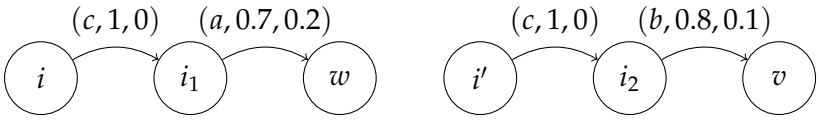
16.1 Now consider the prefixes $(a, 1, 0)T_1$ and $(a, 1, 0)T_2$ depicted below.



Clearly, a mapping from the actions in $(a, 1, 0)T_1$ to the actions in $(a, 1, 0)T_1$ does not exist so neither exists a morphism between the two prefixes.

16.2 If we consider prefixes where the transition from the initial state is caused from a fresh action, *i.e.* some action c such that $c \notin \Pi_1 \sqcup \Pi_2$, such as the prefixes $(c, 1, 0)T_1$ and $(c, 1, 0)T_2$, depicted below, there is an extended morphism $(\sigma', \lambda') : (c, 1, 0)T_1 \to (c, 1, 0)T_2$ given by

- $\sigma'(x) = \begin{cases} \sigma(x) \text{ if } x \in W_1 \sqcup W_2 \\ i' \text{ if } x = i \end{cases}$
- $\lambda'(x) = \begin{cases} \lambda(x) \text{ if } x \in \Pi_1 \sqcup \Pi_2 \\ x' \text{ otherwise} \end{cases}$



However, the operation of prefixing extends to a functor on the subcategory of action-preserving morphisms, *i.e.* the subcategory where morphisms (σ, λ) between PLTS are such that λ is an inclusion function. Given two PLTS $T_1 = \langle W_1, i_1, R_1, \Pi_1 \rangle$, $T_2 = \langle W_2, i_2, R_2, \Pi_2$ and an action-preserving morphism $(\sigma, \lambda) : T_1 \to T_2$ there is a functor which sends (σ, λ) to a morphism $(\sigma', \lambda') : (a, \alpha, \beta)T_1 \to (a, \alpha, \beta)T_2$, for some action a and some $\alpha, \beta \in A$, given by:

- $\sigma'(x) = \begin{cases} \text{ø} \text{ if } x = \text{ø} \\ \sigma(x) \text{ otherwise} \end{cases}$
- $\lambda'(x) = x$

3.6 OTHER OPERATIONS

There are of course several constructions one may perform to obtain a modified transition system from a given one (or more). Those constructions are defined depending on what is useful for different kinds of transition systems and their applications. One particularity of PLTS is that of having transitions with two accessibility relations, besides the typical labelling action, and for that reason we propose some operations which have no analogous in [WN95], but are designed specifically for PLTS over the Gödel algebra G.

First we define an operation which takes a PLTS and uniformly increases or decreases the value of the positive accessibility relation in all transitions and another one which uniformly increases or decreases the value of the negative accessibility relation. For this purpose we define the following operation for $\alpha, \beta \in [0, 1]$

$$\alpha \oplus \beta = \begin{cases} 1 \text{ if } \alpha + \beta \geq 1 \\ 0 \text{ if } \alpha + \beta \leq 0 \\ \alpha + \beta \text{ otherwise} \end{cases}$$

Definition 20. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. Taking $v \in [-1, 1]$, the **positive v-approximation** $T_{\oplus v}^{+}$ is a PLTS $\langle W, i, R', \Pi \rangle$ where

$$R' = \{(w, \pi, w', \alpha \oplus v, \beta) | (w, \pi, w', \alpha, \beta) \in R\}.$$

Given two PLTS T_1 and T_2, some $v \in [-1, 1]$ and a morphism $(\sigma, \lambda) : T_1 \to T_2$ there is a functor that sends T_i to $T_{i\oplus_v^+}$, for $i = \{1, 2\}$ and (σ, λ) to itself.

Definition 21. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. Taking $v \in [-1, 1]$, the **negative v-approximation** $T_{\oplus_v^-}$ is a PLTS $\langle W, i, R', \Pi \rangle$ where

$$R' = \{(w, \pi, w', \alpha, \beta \oplus v) | (w, \pi, w', \alpha, \beta) \in R\}.$$

Given two PLTS T_1 and T_2, some $v \in [-1, 1]$ and a morphism $(\sigma, \lambda) : T_1 \to T_2$ there is a functor that sends T_i to $T_{i\oplus_v^-}$, for $i = \{1, 2\}$ and (σ, λ) to itself.

Lastly, we propose an operation that removes all transitions in a PLTS for which the positive accessibility relation is below a certain value and the negative accessibility relation is above a certain value.

Definition 22. Let $T = \langle W, i, R, \Pi \rangle$ be a PLTS. Taking $v_1, v_2 \in [0, 1]$, the **constraint** $T_{v_1 \uparrow \downarrow v_2}$ is a PLTS $\langle W, i, R', \Pi \rangle$ where

$$R' = \{(w, \pi, w', \alpha, \beta) | (w, \pi, w', \alpha, \beta) \in R \text{ and } \alpha \geq v_1 \text{ and } \beta \leq v_2\}$$

Given two PLTS T_1 and T_2 and a morphism $(\sigma, \lambda) : T_1 \to T_2$ there is a functor that sends T_i to $T_{iv_1 \uparrow \downarrow v_2}$, for $i = 1, 2$, and (σ, λ) to $(\sigma', \lambda') : T_{1v_1 \uparrow \downarrow v_2} \to T_{2v_1 \uparrow \downarrow v_2}$ given by:

- $\sigma'(x) = \sigma(x)$ and
- $\lambda'(x) = \lambda(x)$.

4

MIPL - A MODAL INTUITIONISTIC PARACONSISTENT LOGIC

This chapter presents the design of a logic which takes PLTS as relational structures for the definition of its semantic models.

4.1 SYNTAX - SIGNATURES AND FORMULAS

Given a set of propositions Prop, the formulas of MIPL are generated by the following grammar:

$$\varphi ::= p|\bot|\top|\neg\varphi|\sim\varphi|\varphi_1 \wedge \varphi_2|\varphi_1 \vee \varphi_2|\varphi_1 \rightarrow \varphi_2|\varphi_1 \leftrightarrow \varphi_2|\Box\varphi|\Diamond\varphi|\not\Box\varphi|\not\Diamond\varphi$$

where $p \in$ Prop. We denote this set of formulas by $\mathrm{Fm}(\mathrm{Prop})$.

The nullary connective $\bot$ is a bottom particle interpreted as *False*; $\top$ a redundant (nullary) connective interpreted as *True* and it is defined in terms of $\bot$ as $\top := \neg\bot$. Binary connectives $\wedge$, $\vee$ and $\rightarrow$ have their usual interpretations as conjunction, disjunction and implication, respectively. Unary connectives $\neg$ and $\sim$ are distinct negation connectives. We refer to $\sim$ as natural negation, or simply $\sim$-negation, because $\sim\varphi$ is an abbreviation for $\varphi \rightarrow \bot$. We refer to $\neg$ as twist-negation, or simply $\neg$-negation. The binary connective $\leftrightarrow$ is also redundant: $\varphi_1 \leftrightarrow \varphi_2$ stands for $(\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$. Here, the modal unary connectives $\Box, \Diamond, \not\Box, \not\Diamond$ represent distinct operations not inter-defined, unlike classical modal logics operators $\Diamond$ and $\Box$ which are dual: in classical modal logics, $\Diamond$ is redundant and given in terms of $\Box$ as $\Diamond \equiv \neg\Box\neg$, or conversely. Moreover, in classical modal logics the sentence "is not necessary that φ" is expressed by $\neg\Box\varphi$ (which is logically equivalent to $\Diamond\neg\varphi$) and the sentence "it is not possible that φ" is expressed by $\neg\Diamond\varphi$ (logically equivalent to $\Box\neg\varphi$). In that sense, a single modal operator ($\Box$ or $\Diamond$) is required to represent two distinct alethic modalities. In MIPL a modal operator is defined to represent each of these modalities. Formulas $\Box\varphi$ and $\Diamond\varphi$ naturally stand for the sentences "it is necessary that φ" and "it is possible that φ", respectively, while the formula $\not\Box\varphi$ expresses the sentence "it is not necessary that φ" and the formula $\not\Diamond\varphi$ expresses the sentence "it is not possible that φ".

4.2 SEMANTICS AND SATISFACTION

Definition 23. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra and Prop a set of atomic propositions. An MIPL model is a triple $M = \langle W, R, V \rangle$, where

- $\langle W, R \rangle$ is an **A**-PLTS over the set of actions Π and
- $V : \text{Prop} \times W \to A \times A$ is a valuation function. For $p \in \text{Prop}$ and $w \in W$, $V(p, w) = (\gamma, \delta)$ gives a pair of truth values $\gamma, \delta \in A$, where γ the degree to which there is evidence of p being true and δ is the degree to which there is evidence of p being false, in w.

Accordingly, for a specific $p \in \text{Prop}$ and $w \in W$, the valuation function V of an MIPL model appoints two truth values and in order to consider either of these values separately, we introduce two functions that simply give the projections onto the first and second values of V. These are the **positive valuation function** $v^+ : \text{Prop} \longrightarrow A^W$ and the **negative valuation function** $v^- : \text{Prop} \longrightarrow A^W$, such that for $p \in \text{Prop}$ and $w \in W$,

$$v^+(p)(w) = P_1 V(p, w) \text{ and } v^-(p)(w) = P_2 V(p, w)$$

where P_1, P_2 are the projections onto the first and second values of V, respectively.

The satisfaction relation for an MIPL model, formally defined below, is a function $\models$: $W \times \text{Fm}(\Pi, \text{Prop}) \longrightarrow A \times A$, which gives a valuation to formulas of Fm(Prop), in the worlds of the underlying **A**-PLTS, and it is a mapping to pairs (α, β) such that $\alpha, \beta \in A$. If, for instance, $\langle W, R, V \rangle$ is an MIPL model such that $\langle W, R \rangle$ is G-PLTS then we have $\alpha, \beta \in G$ and $(\alpha, \beta) \in G^{\bowtie}$.

The positive and negative valuation functions of atomic propositions extend to a positive and negative satisfaction of a formula φ. Let the satisfaction of a formula φ in a world $w \in W$ be given by the pair (α, β); we define $(w \models \varphi)^+ \equiv (\alpha, \beta)^+ \equiv \alpha$ and $(w \models \varphi)^- \equiv (\alpha, \beta)^- \equiv \beta$.

To define the satisfaction of modal formulas $\Box\varphi$, $\Diamond\varphi$, $\boxslash\varphi$ and $\diamondslash\varphi$ we need to introduce four new operations. These are $\boxplus, \boxminus, \diamondplus, \diamondminus : W \times \text{Fm}(\text{Prop}) \times \{+, -\} \to A$ given by:

- $\boxplus(w, \varphi, *) = \bigsqcap_{w' \in R[w]} (R^+(w, w') \hookrightarrow (w' \models \varphi)^*)$
- $\boxminus(w, \varphi, *) = \bigsqcap_{w' \in R[w]} (R^-(w, w') \hookrightarrow (w' \models \varphi)^*)$
- $\diamondplus(w, \varphi, *) = \bigsqcup_{w' \in R[w]} (R^+(w, w') \sqcap (w' \models \varphi)^*)$
- $\diamondminus(w, \varphi, *) = \bigsqcup_{w' \in R[w]} (R^-(w, w') \sqcap (w' \models \varphi)^*)$

where $R[w] = \{w' | (w, w', \alpha, \beta) \in R, \text{ for some } \alpha, \beta \in A\}$.

The definitions are given in terms of the generic symbols $\sqcup$ and $\to$, to be replaced by the operations of a chosen residuated lattice.

Definition 24. Let $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow \rangle$ be an MTL-algebra, Prop be a set of atomic propositions and $\langle W, R \rangle$ be an $\mathbf{A}$-PLTS. The satisfaction relation of an MIPL model $M = \langle W, R, V \rangle$ is function

$$\models : W \times \mathrm{Fm}(\Pi, \mathrm{Prop}) \longrightarrow A \times A$$

defined for each $\varphi \in \mathrm{Fm}(\mathrm{Prop})$ as follows:

- $(w \models \top) = (1, 0)$
- $(w \models \bot) = (0, 1)$
- $(w \models p) = V(p, w)$
- $(w \models \neg\varphi) = \neg(w \models \varphi)$
- $(w \models \sim\varphi) \equiv (w \models \varphi \to \bot)$
- $(w \models \varphi_1 \wedge \varphi_2) = (w \models \varphi_1) \overset{\wedge}{\wedge} (w \models \varphi_2)$
- $(w \models \varphi_1 \vee \varphi_2) = (w \models \varphi_1) \overset{\vee}{\vee} (w \models \varphi_2)$
- $(w \models \varphi_1 \to \varphi_2) = (w \models \varphi_1) \Longrightarrow (w \models \varphi_2)$
- $(w \models \varphi_1 \leftrightarrow \varphi_2) \equiv (w \models (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1))$
- $(w \models \Box\varphi) = (\boxplus(w, \varphi, +), \diamondsuit(w, \varphi, -))$
- $(w \models \Diamond\varphi) = (\diamondsuit(w, \varphi, +), \boxplus(w, \varphi, -))$
- $(w \models \boxslash\varphi) = (\diamondsuit(w, \varphi, -), \boxminus(w, \varphi, +))$
- $(w \models \diamondslash\varphi) = (\diamondsuit(w, \varphi, +), \boxminus(w, \varphi, -))$

Two formulas $\varphi, \psi \in \mathrm{Fm}(\mathrm{Prop})$ are semantically equivalent, in symbols $\varphi \equiv \psi$, if for any $w \in W$, $(w \models \varphi) = (w \models \psi)$. We say that φ is valid if, for any $w \in W$, $(w \models \varphi) = (1, 0)$.

The pairs given by the satisfaction relation are interpreted in the same way as the pairs given by the valuation function, that is, the satisfaction of any formula φ in the grammar, at each $w \in W$, is a pair $(\alpha, \beta) \in \mathbf{A}^{\bowtie}$ where α quantifies the evidence of φ being true and β quantifies the evidence of it being false, in w. Indeed, for atomic propositions, the satisfaction relation (in w) is simply given by the valuation function of the proposition (in w). The truth ordering $\preccurlyeq_t$ in $\mathbf{A}^{\bowtie}$ establishes a relation between any two pairs given by the satisfaction relation of an MIPL model.

It is no surprise that the worlds of the PLTS may provide conflicting information about the formulas in the grammar. Moreover, there might be no information whatsoever on wether some φ is true or false, if its satisfaction has value $(0, 0)$. Since 0 is the least value in the underlying truth space, this means there is no evidence that φ is true and neither there is evidence that it is false.

We now explain the difference between $\neg$-negation and $\sim$-negation. It might be helpful to see how $\sim\varphi$ behaves *wrt* φ:

φ	$\sim\varphi$
$(0,0)$	$(1,0)$
$(0,\beta)$	$(1,0)$
$(\alpha,0)$	$(0,\alpha)$
(α,β)	$(0,\alpha)$

Figure 2: Satisfaction of the formula $\sim\varphi$ according to the satisfaction of φ

Note that $\sim\varphi = \top$, whenever the evidence of truthfulness of φ is zero. Thus if φ is not true (and by this we mean either it is only false or neither true nor false), $\sim\varphi$ is only true; if φ is true (only true or both true and false), $\sim\varphi$ is only false. Then $\sim\varphi$ is read as "φ is not true", meaning that $\sim$ distinguishes if the evidence of a formula being true is zero or greater than zero.

The negation $\neg$ introduces formulas $\neg\varphi$ in the grammar which may be considered the converse of φ, since the evidence of $\neg\varphi$ being true is given by the evidence of φ being false, and vice-versa.

Theorem 1. *The following formulas are valid in any MIPL model:*

$$\top \leftrightarrow \sim\bot \tag{9}$$

$$\sim(\varphi_1 \wedge \varphi_2) \leftrightarrow (\sim\varphi_1 \vee \sim\varphi_2) \tag{10}$$

$$\sim(\varphi_1 \vee \varphi_2) \leftrightarrow (\sim\varphi_1 \wedge \sim\varphi_2) \tag{11}$$

Proof.

Firstly, we note that for any $\psi, \psi' \in \mathrm{Fm}(\mathrm{Prop})$,

$$(w \models \psi \leftrightarrow \psi') = \big((w \models \psi) \Longrightarrow (w \models \psi')\big) \mathbin{\overset{\wedge}{\wedge}} \big((w \models \psi') \Longrightarrow (w \models \psi)\big) \tag{12}$$

since,

$$\begin{aligned}
& (w \models \psi \leftrightarrow \psi') \\
= & \quad \{ \text{ definition of } \leftrightarrow \} \\
& (w \models (\psi \to \psi') \wedge (\psi' \to \psi)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models \psi \to \psi') \mathbin{\overset{\wedge}{\wedge}} (w \models \psi' \to \psi) \\
= & \quad \{ \text{ definition of } \models \} \\
& \big((w \models \psi) \Longrightarrow (w \models \psi')\big) \mathbin{\overset{\wedge}{\wedge}} \big((w \models \psi') \Longrightarrow (w \models \psi)\big)
\end{aligned}$$

1. Then, in order to prove (1), we observe that

$$
\begin{aligned}
& (w \models \top \leftrightarrow \sim\bot) \\
= & \quad \{\ (12)\ \} \\
& ((w \models \top) \Longrightarrow (w \models \sim\bot)) \mathbin{\hat{\wedge}} ((w \models \sim\bot) \Longrightarrow (w \models \top)) \\
= & \quad \{\ \text{definition of } \models\ \} \\
& ((1,0) \Longrightarrow (w \models \bot \to \bot)) \mathbin{\hat{\wedge}} ((w \models \bot \to \bot) \Longrightarrow (1,0)) \\
= & \quad \{\ \text{definition of } \models\ \} \\
& ((1,0) \Longrightarrow ((0,1) \Longrightarrow (0,1))) \mathbin{\hat{\wedge}} (((1,0) \Longrightarrow (0,1))) \Longrightarrow (1,0)) \\
= & \quad \{\ \text{definition of } \Longrightarrow\ \} \\
& ((0,1) \Longrightarrow (1,0)) \mathbin{\hat{\wedge}} ((1,0) \Longrightarrow (1,0)) \\
= & \quad \{\ \text{definition of } \Longrightarrow\ \} \\
& (1,0) \mathbin{\hat{\wedge}} (1,0) \\
= & \quad \{\ \text{definition of } \mathbin{\hat{\wedge}}\ \} \\
& (1,0)
\end{aligned}
$$

2. In order prove (2) let us consider fix $(w \models \varphi_1) = (\alpha, \beta)$ and $(w \models \varphi_2) = (\alpha', \beta')$.

$$
\begin{array}{ll}
 & (w \models \sim(\varphi_1 \wedge \varphi_2)) \\
= & \quad \{ \text{ definition of } \sim \} \\
 & w \models (\varphi_1 \wedge \varphi_2) \to \bot \\
= & \quad \{ \text{ definition of } \models \} \\
 & (w \models (\varphi_1 \wedge \varphi_2)) \Longrightarrow (w \models \bot) \\
= & \quad \{ \text{ definition of } \models \} \\
 & ((w \models \varphi_1) \overset{\wedge}{\wedge} (w \models \varphi_2)) \Longrightarrow (0,1) \\
= & \quad \{ \text{ (12) } \} \\
 & \big((\alpha,\beta) \overset{\wedge}{\wedge} (\alpha',\beta')\big) \Longrightarrow (0,1) \\
= & \quad \{ \text{ definition of } \overset{\wedge}{\wedge} \} \\
 & (\alpha \sqcap \alpha', \beta \sqcup \beta') \Longrightarrow (0,1) \\
= & \quad \{ \text{ definition of } \Longrightarrow \} \\
 & ((\alpha \sqcap \alpha') \hookrightarrow 0, \alpha \sqcap \alpha')
\end{array}
\qquad
\begin{array}{ll}
 & (w \models (\sim\varphi_1 \vee \sim\varphi_2)) \\
= & \quad \{ \text{ definition of } \models \} \\
 & (w \models \sim\varphi_1) \overset{\vee}{\vee} (w \models \sim\varphi_2) \\
= & \quad \{ \text{ definition of } \sim \} \\
 & ((w \models \varphi_1) \Longrightarrow (0,1)) \overset{\vee}{\vee} \\
 & ((w \models \varphi_2) \Longrightarrow (0,1)) \\
= & \quad \{ - \} \\
 & ((\alpha,\beta) \Longrightarrow (0,1)) \overset{\vee}{\vee} \\
 & \big((\alpha',\beta') \Longrightarrow (0,1)\big) \\
= & \quad \{ \text{ definition of } \Longrightarrow \} \\
 & (\alpha \hookrightarrow 0, \alpha) \overset{\vee}{\vee} (\alpha' \hookrightarrow 0, \alpha') \\
= & \quad \{ \text{ definition of } \overset{\vee}{\vee} \} \\
 & (\alpha \to 0 \sqcup \alpha' \to 0, \alpha \sqcap \alpha')
\end{array}
$$

By Eq. (8) in Lemma 1, $((\alpha \sqcap \alpha') \hookrightarrow 0, \alpha \sqcap \alpha') = (\alpha \hookrightarrow 0 \sqcup \alpha' \hookrightarrow 0, \alpha \sqcap \alpha')$. Hence $w \models \sim(\varphi_1 \wedge \varphi_2) \leftrightarrow (\sim\varphi_1 \vee \sim\varphi_2)$.

3. In order to prove (3) let us consider fix $(w \models \varphi_1) = (\alpha, \beta)$ and $(w \models \varphi_2) = (\alpha', \beta')$.

$$
\begin{aligned}
& (w \models \sim(\varphi_1 \vee \varphi_2)) \\
= & \quad \{ \text{ definition of } \sim \} \\
& w \models (\varphi_1 \vee \varphi_2) \to \bot \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models (\varphi_1 \vee \varphi_2)) \Longrightarrow (w \models \bot) \\
= & \quad \{ \text{ definition of } \models \} \\
& ((w \models \varphi_1) \overset{\vee}{\vee} (w \models \varphi_2)) \Longrightarrow (0,1) \\
= & \quad \{ (12) \} \\
& \big((\alpha,\beta) \overset{\vee}{\vee} (\alpha',\beta')\big) \Longrightarrow (0,1) \\
= & \quad \{ \text{ definition of } \overset{\vee}{\vee} \} \\
& (\alpha \sqcup \alpha', \beta \sqcap \beta') \Longrightarrow (0,1) \\
= & \quad \{ \text{ definition of } \Longrightarrow \} \\
& ((\alpha \sqcup \alpha') \hookrightarrow 0, \alpha \sqcup \alpha')
\end{aligned}
\qquad
\begin{aligned}
& (w \models (\sim\varphi_1 \wedge \sim\varphi_2)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models \sim\varphi_1) \overset{\wedge}{\wedge} (w \models \sim\varphi_2) \\
= & \quad \{ \text{ definition of } \sim \} \\
& ((w \models \varphi_1) \Longrightarrow (0,1)) \overset{\wedge}{\wedge} \\
& ((w \models \varphi_2) \Longrightarrow (0,1)) \\
= & \quad \{ - \} \\
& ((\alpha,\beta) \Longrightarrow (0,1)) \overset{\wedge}{\wedge} \\
& \big((\alpha',\beta') \Longrightarrow (0,1)\big) \\
= & \quad \{ \text{ definition of } \Longrightarrow \} \\
& (\alpha \hookrightarrow 0, \alpha) \overset{\wedge}{\wedge} (\alpha' \hookrightarrow 0, \alpha') \\
= & \quad \{ \text{ definition of } \overset{\wedge}{\wedge} \} \\
& (\alpha \hookrightarrow 0 \sqcap \alpha' \hookrightarrow 0, \alpha \sqcup \alpha')
\end{aligned}
$$

By Eq. (6) in Lemma 1, $((\alpha \sqcup \alpha') \hookrightarrow 0, \alpha \sqcup \alpha') = ((\alpha \hookrightarrow 0) \sqcap (\alpha' \hookrightarrow 0), \alpha \sqcup \alpha')$. Hence $(w \models \sim(\varphi_1 \vee \varphi_2)) = (w \models (\sim\varphi_1 \wedge \sim\varphi_2))$.

□

Remark 2. *Note that the definition $\sim$ coincides with the definition of classic and intuitionistic logic. Intristingly, De Morgan duality applies to $\wedge$ and $\vee$ wrt $\sim$, which is not the case in intuitionistic logic. On the other hand, the rules of material implication and double negation are not valid for $\sim$ and the rule of material implication is not valid for $\neg$. This means that $\sim$ and $\neg$ do not satisfy the axioms of an intuitionistic negation and moreover they do not satisfy the axioms of a strong negation. (For the definition of strong negation see for instance [Sed16] or [Vak77].) Indeed, $\neg\varphi \to \sim\varphi$ and $\sim\varphi \to \neg\varphi$ are not valid formulas in MIPL so neither of these negations is "stronger" than the other.*

It is clear from Definition 24 that modal operators are not dual in the classical sense. Nonetheless we shall prove the following equalities between modal formulas.

Theorem 2. *The following semantical equivalences are true in any MIPL model:*

$$\Box\neg\varphi \equiv \neg\Diamond\varphi \quad (13) \qquad \text{⟡}\neg\varphi \equiv \text{⟡}\varphi \quad (16)$$

$$\Diamond\neg\varphi \equiv \neg\Box\varphi \quad (14) \qquad \Box\sim\varphi \equiv \sim\Diamond\varphi \quad (17)$$

$$\text{⟡}\neg\varphi \equiv \text{⟡}\varphi \quad (15) \qquad \text{⟡}\sim\varphi \equiv \neg\sim\text{⟡}\varphi \quad (18)$$

Proof.

(i) $\Box(\neg\varphi) = \neg(\Diamond\varphi)$

$$
\begin{aligned}
& (w \models \Box\neg\varphi) \\
= & \quad \{ \text{ definiton of } \models \} \\
& \left(\bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (w' \models \neg\varphi)^+\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \neg\varphi)^-\} \right) \\
= & \quad \{ \text{ definition of } \models \} \\
& \left(\bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (\dashv(w' \models \varphi))^+\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (\dashv(w' \models \neg\varphi))^-\} \right) \\
= & \quad \{ \text{ definition of } \dashv \} \\
& \left(\bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (w' \models \varphi)^-\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \varphi)^+\} \right) \\
= & \quad \{ \text{ definition of } \boxplus, \dot{\Diamond} \} \\
& (\boxplus(w,\varphi,-), \dot{\Diamond}(w,\varphi,+)) \\
= & \quad \{ \text{ definition of } \dashv \} \\
& \dashv(\dot{\Diamond}(w,\varphi,+), \boxplus(w,\varphi,-)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models \neg(\Diamond\varphi))
\end{aligned}
$$

(ii) $\Diamond(\neg\varphi) = \neg(\Box\varphi)$

$$
\begin{aligned}
& (w \models \Diamond(\neg\varphi)) \\
= & \quad \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \neg\varphi)^+\}, \bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (w' \models \neg\varphi)^-\} \right) \\
= & \quad \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (\neg(w' \models \varphi))^+\}, \bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (\neg(w' \models \varphi))^-\} \right) \\
= & \quad \{ \text{ definition of } \neg \} \\
& \left(\bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \varphi)^-\}, \bigsqcap_{w' \in W} \{R^+(w,w') \hookrightarrow (w' \models \varphi)^+\} \right) \\
= & \quad \{ \text{ definition of } \diamondplus, \boxplus \} \\
& (\diamondplus(w,\varphi,-), \boxplus(w,\varphi,+)) \\
= & \quad \{ \text{ definition of } \neg \} \\
& \neg(\boxplus(w,\varphi,+), \diamondplus(w,\varphi,-)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models \neg(\Box\varphi))
\end{aligned}
$$

(iii) $\boxslash\neg\varphi = \diamondslash\varphi$

$$
\begin{aligned}
& (w \models \not\Box(\neg\varphi)) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \neg\varphi)^-\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (w' \models \neg\varphi)^+\} \right) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (\dashv(w' \models \varphi))^-\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (\dashv(w' \models \varphi))^+\} \right) \\
= \quad & \{ \text{ definition of } \dashv \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^+\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (w' \models \varphi)^-\} \right) \\
= \quad & \{ \text{ definition of } \diamondarrow, \boxminus \} \\
& (\diamondarrow(w,\varphi,+), \boxminus(w,\varphi,-)) \\
= \quad & \{ \text{ definition of } \models \} \\
& (w \models \not\Diamond\varphi)
\end{aligned}
$$

(iv) $\not\Diamond\neg\varphi = \not\Box\varphi$

$$
\begin{aligned}
& (w \models \not\Diamond(\neg\varphi)) \\
= & \quad \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \neg\varphi)^+\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (w' \models \neg\varphi)^-\} \right) \\
= & \quad \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (\neg(w' \models \varphi))^+\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (\neg(w' \models \varphi))^-\} \right) \\
= & \quad \{ \text{ definition of } \neg \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^-\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (w' \models \varphi)^+\} \right) \\
= & \quad \{ \text{ definition of } \diamondsuit, \boxminus \} \\
& (\diamondsuit(w,\varphi,-), \boxminus(w,\varphi,+)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models \not\Box\varphi)
\end{aligned}
$$

(v) $\Box(\sim\varphi) = \sim(\Diamond\varphi)$

$$
\begin{aligned}
& (w \models \Box(\sim\varphi)) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\prod_{w' \in W} \{R^+(w,w') \hookrightarrow (w' \models \sim\varphi)^+\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \sim\varphi)^-\} \right) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\prod_{w' \in W} \{R^+(w,w') \hookrightarrow ((w' \models \varphi) \Longrightarrow (0,1))^+\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap ((w' \models \neg\varphi) \Longrightarrow (0,1))^-\} \right) \\
= \quad & \{ \text{ definition of } \Longrightarrow \} \\
& \left(\prod_{w' \in W} \{R^+(w,w') \hookrightarrow ((w' \models \varphi)^+ \hookrightarrow 0)\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \varphi)^+\} \right) \\
= \quad & \{ \text{ by Eq. (3) } \} \\
& \left(\prod_{w' \in W} \{(R^+(w,w') \sqcap (w' \models \varphi)^+) \hookrightarrow 0\}, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \varphi)^+\} \right) \\
= \quad & \{ \text{ by Eq. (6) in Lemma 1 } \} \\
& \left(\bigsqcup_{w' \in W} \{(R^+(w,w') \sqcap (w' \models \varphi)^+)\} \hookrightarrow 0, \bigsqcup_{w' \in W} \{R^+(w,w') \sqcap (w' \models \varphi)^+\} \right) \\
= \quad & \{ \text{ definition of } \Diamond \} \\
& (\Diamond(w,\varphi,+) \hookrightarrow 0, \Diamond(w,\varphi,+)) \\
= \quad & \{ \text{ definition of } \Longrightarrow \} \\
& (\Diamond(w,\varphi,+), \boxplus(w,\varphi,-)) \Longrightarrow (0,1) \\
= \quad & \{ \text{ definition of } \models \} \\
& (w \models \Diamond\varphi) \Longrightarrow (w \models \bot) \\
= \quad & \{ \text{ definition of } \models \} \\
& (w \models \sim(\Diamond\varphi))
\end{aligned}
$$

(vi) $\boxslash\sim\varphi = \neg\sim\Diamond\varphi$

$$
\begin{aligned}
& (w \models \not\boxdot(\sim\varphi)) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \sim\varphi)^-\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow (w' \models \sim\varphi)^+\} \right) \\
= \quad & \{ \text{ definition of } \models \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap ((w' \models \varphi) \Longrightarrow (0,1))^-\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow ((w' \models \neg\varphi) \Longrightarrow (0,1))^+\} \right) \\
= \quad & \{ \text{ definition of } \Longrightarrow \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^+\}, \bigsqcap_{w' \in W} \{R^-(w,w') \hookrightarrow ((w' \models \varphi)^+ \hookrightarrow 0)\} \right) \\
= \quad & \{ \text{ by Eq. (3) } \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^+\}, \bigsqcap_{w' \in W} \{(R^-(w,w') \sqcap (w' \models \varphi)^+) \hookrightarrow 0\} \right) \\
= \quad & \{ \text{ by Eq. (6) in Lemma 1 } \} \\
& \left(\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^+\}, (\bigsqcup_{w' \in W} \{R^-(w,w') \sqcap (w' \models \varphi)^+\}) \hookrightarrow 0 \right) \\
= \quad & \{ \text{ definition of } \diamondsuit \} \\
& (\diamondsuit(w,\varphi,+), \diamondsuit(w,\varphi,+) \hookrightarrow 0) \\
= \quad & \{ \text{ definition of } \neg \} \\
& \neg(\diamondsuit(w,\varphi,+) \hookrightarrow 0, \diamondsuit(w,\varphi,+)) \\
= \quad & \{ \text{ definition of } \Longrightarrow \} \\
& \neg((\diamondsuit(w,\varphi,+), \boxminus(w,\varphi,-)) \Longrightarrow (0,1)) \\
= \quad & \{ \text{ definition of } \models \} \\
& \neg((w \models \not\Diamond\varphi) \Longrightarrow (w \models \bot)) \\
= \quad & \{ \text{ definition of } \models \} \\
& \neg(w \models \sim(\not\Diamond\varphi)) \\
= \quad & \{ \text{ definition of } \models \} \\
& (w \models \neg(\sim(\not\Diamond\varphi)))
\end{aligned}
$$

□

4.3 MODAL PRESERVATIONS

In this section we will define the notions of simulation and bisimulation between MIPL models and study the preservation of formulas between similar and bisimilar models.

Consider two PLTS $T_1 = \langle W_1, R_1 \rangle$, $T_2 = \langle W_2, R_2 \rangle$ and MIPL models $M_1 = (T_1, V_1)$ and $M_2 = (T_2, V_2)$ over $\Sigma = (\Pi, \text{Prop})$.

Definition 25. A relation $S \subseteq W_1 \times W_2$ is a simulation between MIPL models M_1 and M_2 if

- S is a simulation between PLTS T_1 and T_2
- for any $p \in \text{Prop}$ and $\langle w, v \rangle \in S$, $V_1(w, p) \preccurlyeq_t V_2(v, p)$

Lemma 7. *If $S \subseteq W_1 \times W_2$ is a simulation between models M_1 and M_2 and $\langle w, v \rangle \in S$ then*

$$(w \models_{M_1} \varphi) \preccurlyeq_t (v \models_{M_2} \varphi) \text{ for } \varphi \in Fm^{+\Diamond}$$

where $Fm^{+\Diamond}$ is the positive fragment of MIPL with a single modal connective $\Diamond$.

Proof.

In the proof the subscript of $\preccurlyeq_t$ is omitted and $\preccurlyeq$ represents the truth ordering of $\mathbf{A}^{\bowtie}$. We consider as an induction hypothesis that for any φ, $(w \models_{M_1} \varphi) \preccurlyeq_t (v \models_{M_2} \varphi)$.

- if $\varphi = \top$

  $(w \models_{M_1} \top) = (1, 0) = (v \models_{M_2} \top)$

  $\therefore (w \models_{M_1} \top) \preccurlyeq (v \models_{M_2} \top)$

- if $\varphi = \bot$

  $(w \models_{M_1} \bot) = (0, 1) = (v \models_{M_2} \bot)$

  $\therefore (w \models_{M_1} \bot) \preccurlyeq (v \models_{M_2} \bot)$

- if $\varphi = p : p \in \text{Prop}$

$$\begin{array}{ll} & (w \models_{M_1} p)^+ \\ = & \quad \{ \text{ definition of } \models \} \\ & V_1^+(w, p) \\ \leq & \quad \{ \text{ definition of } S \} \\ & V_2^+(v, p) \end{array}$$

$$
\begin{array}{ll}
 & (w \models_{M_1} p)^- \\
= & \quad \{ \text{ definition of } \models \} \\
 & V_1^-(w,p) \\
\geq & \quad \{ \text{ definition of } S \} \\
 & V_2^-(v,p)
\end{array}
$$

$\therefore (w \models_{M_1} p) \preccurlyeq (v \models_{M_2} p)$

- if $\varphi = \varphi_1 \wedge \varphi_2$

$$
\begin{array}{ll}
 & (w \models_{M_1} \varphi_1 \wedge \varphi_2)^+ \\
= & \quad \{ \text{ definition of } \models \} \\
 & (w \models_{M_1} \varphi_1)^+ \sqcap (w \models_{M_1} \varphi_2)^+ \\
\leq & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi)^+ \leq (v \models_{M_2} \varphi)^+ \text{ and monotonicity of } \sqcap \} \\
 & ((v \models_{M_2} \varphi_1)^+ \sqcap (v \models_{M_2} \varphi_2)^+) \\
= & \quad \{ \text{ definition of } \models \} \\
 & (v \models_{M_2} \varphi_1 \wedge \varphi_2)^+
\end{array}
$$

$$
\begin{array}{ll}
 & (w \models_{M_1} \varphi_1 \wedge \varphi_2)^- \\
= & \quad \{ \text{ definition of } \models \} \\
 & (w \models_{M_1} \varphi_1)^- \sqcup (w \models_{M_1} \varphi_2)^- \\
\geq & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi)^- \geq (v \models_{M_2} \varphi)^- \text{ and monotonicity of } \sqcup \} \\
 & ((v \models_{M_2} \varphi_1)^- \sqcup (v \models_{M_2} \varphi_2)^-) \\
= & \quad \{ \text{ definition of } \models \} \\
 & (v \models_{M_2} \varphi_1 \wedge \varphi_2)^-
\end{array}
$$

$\therefore (w \models_{M_1} \varphi_1 \wedge \varphi_2) \preccurlyeq (v \models_{M_2} \varphi_1 \wedge \varphi_2)$

- if $\varphi = \varphi_1 \vee \varphi_2$

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \vee \varphi_2)^+ \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1)^+ \sqcup (w \models_{M_1} \varphi_2)^+ \\
\leq & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi)^+ \leq (v \models_{M_2} \varphi)^+ \text{ and monotonicity of } \sqcup \} \\
& (v \models_{M_2} \varphi_1)^+ \sqcup (v \models_{M_2} \varphi_2)^+ \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \varphi_1 \vee \varphi_2)^+
\end{aligned}
$$

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \vee \varphi_2)^- \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1)^- \sqcap (w \models_{M_1} \varphi_2)^- \\
\geq & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi)^- \geq (v \models_{M_2} \varphi)^- \text{ and monotonicity of } \sqcap \} \\
& (v \models_{M_2} \varphi_1)^- \sqcap (v_{M_2} \models \varphi_2)^- \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \varphi_1 \vee \varphi_2)^-
\end{aligned}
$$

$\therefore (w \models_{M_1} \varphi_1 \vee \varphi_2) \preccurlyeq (v \models_{M_2} \varphi_1 \vee \varphi_2)$

- if $\varphi = \Diamond \varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\varphi_1)^+ \\
= \quad & \{ \text{ definition of } \models \} \\
& \bigsqcup_{w' \in W_1} \{R_1^+(w,w') \sqcap (w' \models_{M_1} \varphi_1)^+\} \\
= \quad & \{ \text{ for some } x \in W_1 \text{ such that } \langle x, x' \rangle \in S \} \\
& R_1^+(w,x) \sqcap (x \models_{M_1} \varphi_1)^+ \\
\leq \quad & \{ \text{ definition of } S \; R_1^+(w,x) \leq R_2^+(v,x') \text{ and Induction Hypothesis } (x \models_{M_1} \varphi_1)^+ \leq (x' \models_{M_2} \varphi_1)^+ \} \\
& R_2^+(v,x')^+ \sqcap (x' \models_{M_2} \varphi_1)^+ \\
\leq \quad & \{ \text{ monotonicity of } \sqcup \} \\
& \bigsqcup_{v' \in W_2} \{R_2^+(v,v') \sqcap (v' \models_{M_2} \varphi_1)^+\} \\
= \quad & \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Diamond\varphi_1)^+
\end{aligned}
$$

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\varphi_1)^- \\
= \quad & \{ \text{ definition of } \models \} \\
& \prod_{w' \in W_1} \{R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^-\}
\end{aligned}
$$

For any $w' \in W_1$ exists $v' \in W_2$ such that $\langle w', v' \rangle \in S$ and:

$$
\begin{aligned}
& R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^- \\
\geq \quad & \{ \text{ definition of } S \; R_1^+(w,w') \leq R_2^+(v,v+) \text{ and Eq. (2)} \} \\
& R_2^+(v,v') \hookrightarrow (w' \models_{M_1} \varphi_1)^- \\
\geq \quad & \{ \text{ and Induction Hypothesis } (w \models_{M_1} \varphi)^- \geq (v \models_{M_2} \varphi)^- \text{ and Eq. (1)} \} \\
& R_2^+(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^-
\end{aligned}
$$

$$
\begin{aligned}
& \prod_{w' \in W_1} \{R_1^+(w, w') \hookrightarrow (w' \models_{M_1} \varphi_1)^-\} \\
\geq \quad & \{ \text{ monotonicity of } \sqcap \ \} \\
& \prod_{v' \in W_2} \{R_2^+(v, v') \hookrightarrow (v' \models_{M_2} \varphi_1)^-\} \\
= \quad & \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Diamond \varphi_1)^-
\end{aligned}
$$

$$\therefore (w \models_{M_1} \Diamond \varphi_1) \preccurlyeq (v \models_{M_2} \Diamond \varphi_1)$$

Remark 3. *As expected, there are some formulas which are not preserved by these mappings. Such are the cases of $\neg\varphi$, $\varphi_1 \to \varphi_2$, $\Box\varphi$, $\not\Box\varphi$ and $\not\Diamond\varphi$, as presented below.*

- *if* $\varphi = \neg\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \neg\varphi_1)^+ \\
= \quad & \{ \textit{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1)^- \\
\geq \quad & \{ \textit{ Induction Hypothesis } (w \models_{M_1} \varphi)^- \geq (v \models_{M_2} \varphi)^- \} \\
& (v \models_{M_2} \varphi_1)^- \\
= \quad & \{ \textit{ definition of } \models \} \\
& (v \models_{M_2} \neg\varphi_1)^+
\end{aligned}
$$

$$
\begin{aligned}
& (w \models_{M_1} \neg\varphi_1)^- \\
= \quad & \{ \textit{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1)^+ \\
\leq \quad & \{ \textit{ Induction Hypothesis } (w \models_{M_1} \varphi)^+ \leq (v \models_{M_2} \varphi)^+ \} \\
& (v \models_{M_2} \varphi_1)^+ \\
= \quad & \{ \textit{ definition of } \models \} \\
& (v \models_{M_2} \neg\varphi_1)^-
\end{aligned}
$$

$\therefore (w \models_{M_1} \neg\varphi_1) \not\preccurlyeq (v \models_{M_2} \neg\varphi_1)$

In fact $(v \models_{M_2} \neg\varphi_1) \preccurlyeq (w \models_{M_1} \neg\varphi_1)$.

- *if* $\varphi = \varphi_1 \to \varphi_2$

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \to \varphi_2)^+ \\
= & \quad \{ \textit{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1)^+ \hookrightarrow (w \models_{M_1} \varphi_2)^+
\end{aligned}
$$

We give a counterexample, using the Gödel algebra, which proves that $(w \models_{M_1} \varphi)^+ \leq (v \models_{M_2} \varphi)^+$ *is not always true for* $\langle w, v\rangle \in S$. *For instance, it is possible to have a simulation where*

$$(w \models_{M_1} \varphi_1)^+ = 0.3 \textit{ and } (v \models_{M_2} \varphi_1)^+ = 0.8$$
$$(w \models_{M_1} \varphi_2)^+ = 0.6 \textit{ and } (v \models_{M_2} \varphi_2)^+ = 0.7$$

Here,

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \to \varphi_2)^+ \\
= & \quad \{ \textit{ definition of } \models \} \\
& 0.3 \hookrightarrow 0.6 \\
\geq & \quad \{ \textit{ by Eq. (2) } \} \\
& 0.8 \hookrightarrow 0.6 \\
\geq & \quad \{ \textit{ by Eq. (1) } \} \\
& 0.8 \hookrightarrow 0.7 \\
= & \quad \{ \textit{ definition of } \models \} \\
& (v \models_{M_2} \varphi_1 \to \varphi_1)^+
\end{aligned}
$$

$\therefore (w \models_{M_1} \varphi_1 \to \varphi_2) \not\preccurlyeq (v \models_{M_2} \varphi_1 \to \varphi_2)$

Moreover, note that $\varphi = {\sim}\varphi_1$ *is an instance of* $\varphi = \varphi_1 \to \varphi_2$.

$\therefore (w \models_{M_1} {\sim}\varphi) \not\preccurlyeq (v \models_{M_2} {\sim}\varphi)$

- *if* $\varphi = \Box\varphi$

$$
\begin{aligned}
& (w \models_{M_1} \Box\varphi_1)^+ \\
= & \quad \{ \textit{ definition of } \models \} \\
& \bigsqcap_{w' \in W_1} \{R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+\}
\end{aligned}
$$

A counterexample using the Gödel algebra shows that $(w \models_{M_1} \Box\varphi_1)^+ \leq (v \models_{M_2} \Box\varphi_1)^+$ *is not necessarily true is the case where both worlds* w *and* v *have a single transition and where*

$$R_1^+(w,w') = 0.3 \textit{ and } R_2^+(v,v') = 0.8$$

$$(w' \models_{M_1} \varphi_1)^+ = 0.6 \textit{ and } (v' \models_{M_2} \varphi_1)^+ = 0.7$$

Here $R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+ = 0.3 \hookrightarrow 0.6$ *whereas* $R_2^+(v,v') \to (v \models_{M_2} \varphi_1)^+ = 0.8 \hookrightarrow 0.7$ *and we have proved* $0.3 \hookrightarrow 0.6 \geq 0.8 \hookrightarrow 0.7$.

$\therefore (w \models_{M_1} \Box\varphi) \not\preccurlyeq (v \models_{M_2} \Box\varphi)$

- *if* $\varphi = \Diamond\!\!\!\!-\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\!\!\!\!-\varphi_1)^+ \\
= & \quad \{ \textit{ definition of } \models \} \\
& \bigsqcup_{w' \in W} \{R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^-\}
\end{aligned}
$$

For any $w' \in W_1$ *exists* $v' \in W_2$ *such that* $\langle w', v' \rangle \in S$ *and:*

$$
\begin{aligned}
& R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^- \\
\geq & \quad \{ \textit{ definition of } S \; R_1^-(w,w') \geq R_2^-(v,v') \textit{ and Induction Hypothesis } (w' \models_{M_1} \varphi)^- \geq (v' \models_{M_2} \varphi)^- \} \\
& R_2^-(v,v') \sqcap (v' \models_{M_2} \varphi_1)^-
\end{aligned}
$$

$$
\begin{aligned}
& \bigsqcup_{w' \in W_1} \{R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^-\} \\
\geq & \quad \{ \textit{ monotonicity of } \sqcup \} \\
& \bigsqcup_{v' \in W_2} \{R_2^-(v,v') \sqcap (v' \models_{M_2} \varphi_1)^-\} \\
= & \quad \{ \textit{ definition of } \models \} \\
& (v \models_{M_2} \boxslash \varphi_1)^+
\end{aligned}
$$

$$
\begin{aligned}
& (w \models_{M_1} \boxslash \varphi_1)^- \\
= & \quad \{ \textit{ definition of } \models \} \\
& \bigsqcap_{w' \in W_1} \{R_1^-(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+\}
\end{aligned}
$$

For any $w' \in W_1$ exists $v' \in W_2$ such that $\langle w', v' \rangle \in S$ and

$$
\begin{aligned}
& R_1^-(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+ \\
\leq & \quad \{ \textit{ definition of } S \ R_1^-(w,w') \geq R_2^-(v,v') \textit{ and by Eq. (2)} \} \\
& R_2^-(v,v') \hookrightarrow (w' \models_{M_1} \varphi_1)^+ \\
\leq & \quad \{ \textit{ Induction Hypothesis } (w \models_{M_1} \varphi)^+ \leq (v \models_{M_2} \varphi)^+ \textit{ and by Eq. (1)} \} \\
& R_2^-(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^+
\end{aligned}
$$

$$
\begin{aligned}
& \bigsqcap_{w' \in W_1} \{R_1^-(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+\} \\
\leq & \quad \{ \textit{ monotonicity of } \sqcap \} \\
& \bigsqcap_{v' \in W_2} \{R_2^-(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^+\} \\
= & \quad \{ \textit{ definition of } \models \} \\
& (v \models_{M_2} \boxslash \varphi_1)^-
\end{aligned}
$$

$\therefore (w \models_{M_1} \boxtimes\varphi_1) \not\preccurlyeq (v \models_{M_2} \boxtimes\varphi_1)$

In fact, $(v \models_{M_2} \boxtimes\varphi) \preccurlyeq (w \models_{M_1} \boxtimes\varphi)$

- *if* $\varphi = \diamondsuit\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \diamondsuit\varphi_1)^+ \\
= & \quad \{ \textit{ definition of } \models \} \\
& \bigsqcup_{w' \in W_1} \{R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^+\}
\end{aligned}
$$

A counterexample using the Gödel algebra shows that $(w \models_{M_1} \diamondsuit\varphi_1)^+ \leq (v \models_{M_2} \diamondsuit\varphi_1)^+$ *is not necessarily true is the case where both worlds* w *and* v *have a single transition and where*

$$R_1^-(w,w') = 0.8 \textit{ and } R_2^-(v,v') = 0.6$$
$$(w' \models_{M_1} \varphi_1)^+ = 0.7 \textit{ and } (v' \models_{M_2} \varphi_1)^+ = 0.5$$

Here,

$$
\begin{aligned}
& R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^+ \\
= & \quad \{ \text{-} \} \\
& 0.8 \sqcap 0.7 \\
\geq & \quad \{ \textit{ since } 0.8 \geq 0.6 \textit{ and } 0.7 \geq 0.5 \} \\
& 0.6 \sqcap 0.5 \\
= & \quad \{ \textit{ definition of } \models \} \\
& R_2^-(v,v') \sqcap (v \models_{M_2} \varphi_1)^+
\end{aligned}
$$

$\therefore (w \models_{M_1} \diamondsuit\varphi_1) \not\preccurlyeq (v \models_{M_2} \diamondsuit\varphi_1)$

□

Definition 26. A relation $B \subseteq W_1 \times W_2$ is a bisimulation between MIPL models M_1 and M_2 if

- B is a bisimulation between PLTS T_1 and T_2
- for any $p \in$ Prop and $\langle w, v\rangle \in B$, $V_1(w,p) = V_2(v,p)$.

Theorem 3. *Let B be a bisimulation between two MIPL models and $\langle w, v\rangle \in B$. Then $(w \models_{M_1} \varphi) = (v \models_{M_2} \varphi)$ for $\varphi \in Fm(\Pi, Prop)$.*

Proof.

We consider as an induction hypothesis that for any φ, $(w \models_{M_1} \varphi) = (v \models_{M_2} \varphi)$.

- if $\varphi = \top$ $(w \models_{M_1} \top)^+ = (v \models_{M_2} \top)^+ = 1$ $(w \models_{M_1} \top)^- = (v \models_{M_2} \top)^- = 0$
- if $\varphi = \bot$ $(w \models_{M_1} \bot)^+ = (v \models_{M_2} \bot)^+ = 0$ $(w \models_{M_1} \bot)^- = (v \models_{M_2} \bot)^- = 1$
- if $\varphi = p$ such that $p \in$ Prop

$$\begin{array}{ll} & (w \models_{M_1} p) = V_1(w, p) \\ = & \quad \{\ \langle w, v\rangle \in B\ \} \\ & (w \models_{M_1} p) = V_2(v, p) \end{array}$$

$\therefore (w \models_{M_1} p) = (v \models_{M_2} p)$

- if $\varphi = \neg\varphi_1$

$$\begin{array}{ll} & (w \models_{M_1} \neg\varphi_1) \\ = & \quad \{\ \text{definition of } \models\ \} \\ & ((w \models_{M_1} \varphi_1)^-, (w \models_{M_1} \varphi_1)^+) \\ = & \quad \{\ \text{Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi)\ \} \\ & ((v \models_{M_2} \varphi_1)^-, (v \models_{M_2} \varphi_1)^+) \\ = & \quad \{\ \text{definition of } \models\ \} \\ & (v \models_{M_2} \neg\varphi_1) \end{array}$$

$\therefore (w \models_{M_1} \neg\varphi_1) = (v \models_{M_2} \neg\varphi_1)$

- $\varphi = \varphi_1 \wedge \varphi_2$

$$\begin{array}{ll} & (w \models_{M_1} \varphi_1 \wedge \varphi_2) \\ = & \quad \{\ \text{definition of } \models\ \} \\ & (w \models_{M_1} \varphi_1) \overset{\wedge}{\wedge} (w \models_{M_1} \varphi_2) \\ = & \quad \{\ \text{Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi)\ \} \\ & (v \models_{M_2} \varphi_1) \overset{\wedge}{\wedge} (v \models_{M_2} \varphi_2) \\ = & \quad \{\ \text{definition of } \models\ \} \\ & (v \models_{M_2} \varphi_1 \wedge \varphi_2) \end{array}$$

$\therefore (w \models_{M_1} \varphi_1 \wedge \varphi_2) = (v \models_{M_2} \varphi_1 \wedge \varphi_2)$.

- if $\varphi = \varphi_1 \vee \varphi_2$

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \vee \varphi_2) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1) \overset{\vee}{\vee} (w \models_{M_1} \varphi_2) \\
= & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& (v \models_{M_2} \varphi_1) \overset{\vee}{\vee} (v \models_{M_2} \varphi_2) \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \varphi_1 \vee \varphi_2)
\end{aligned}
$$

$\therefore (w \models_{M_1} \varphi_1 \vee \varphi_2) = (v \models_{M_2} \varphi_1 \vee \varphi_2)$.

- if $\varphi = \varphi_1 \to \varphi_2$

$$
\begin{aligned}
& (w \models_{M_1} \varphi_1 \to \varphi_2) \\
= & \quad \{ \text{ definition of } \models \} \\
& (w \models_{M_1} \varphi_1) \Longrightarrow (w \models_{M_1} \varphi_2) \\
= & \quad \{ \text{ Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& (v \models_{M_2} \varphi_1) \Longrightarrow (v \models_{M_2} \varphi_2) \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \varphi_1 \to \varphi_2)
\end{aligned}
$$

$\therefore (w \models_{M_1} \varphi_1 \to \varphi_2) = (v \models_{M_2} \varphi_1 \to \varphi_2)$

- if $\varphi = \Diamond \varphi_1$

  First note that any pair $\langle w, v \rangle$ of a bisimulation between MIPL models satisfies the following condition:

  - $\forall_{w' \in W_1} \exists v' \in W_2$ such that $R_1^+(w, w') = R_2^+(v, v'), R_1^-(w, w') = R_2^-(v, v')$ and $\langle w', v' \rangle \in B$. $(*)$

  This follows directly from the definition of bisimulation between two PLTS.

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\varphi_1)^+ \\
= & \quad \{ \text{ definition of } \models \} \\
& \bigsqcup_{w' \in W_1} (R_1^+(w,w') \sqcap (w' \models_{M_1} \varphi_1)^+) \\
= & \quad \{ \text{ by } (*) \text{ and Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& \bigsqcup_{w' \in W_1} (R_2^+(v,v') \sqcap (v' \models_{M_2} \varphi_1)^+) \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Diamond\varphi_1)^+
\end{aligned}
$$

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\varphi_1)^- \\
= & \quad \{ \text{ definition of } \models \} \\
& \bigsqcap_{w' \in W_1} (R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi)^-) \\
= & \quad \{ \text{ by } (*) \text{ and Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& \bigsqcap_{v' \in W_1} (R_2^+(v,v') \hookrightarrow (v' \models_{M_2} \varphi)^-) \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Diamond\varphi_1)^-
\end{aligned}
$$

$\therefore (w \models_{M_1} \Diamond\varphi_1) = (v \models_{M_2} \Diamond\varphi_1)$.

- if $\varphi = \Box\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \Box\varphi_1) \\
= \quad & \{ \text{ definition of } \models \} \\
& (\bigsqcap_{w' \in W_1} (R_1^+(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^+), \bigsqcup_{w' \in W_1} (R_1^+(w,w') \sqcap (w' \models \varphi_1)^-)) \\
= \quad & \{ \text{ by } (*) \text{ and Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& (\bigsqcap_{v' \in W_2} (R_2^+(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^+), \bigsqcup_{v' \in V_1} (R_2^+(v,v') \sqcap (v' \models_{M_2} \varphi_1)^-)) \\
= \quad & \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Box\varphi_1)
\end{aligned}
$$

$\therefore (w \models_{M_1} \Box\varphi_1) = (v \models_{M_2} \Box\varphi_1)$

- if $\varphi = \boxslash\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \boxslash\varphi_1) \\
= \quad & \{ \text{ definition of } \models \} \\
& (\bigsqcup_{w' \in W_1} (R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^-), \bigsqcap_{w' \in W_1} (R_1^-(w,w') \rightarrow (w' \models_{M_1} \varphi_1)^+)) \\
= \quad & \{ \text{ by } (*)_{16} \text{ and Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& (\bigsqcup_{v \in W_2} (R_2^-(v,v') \sqcap (v' \models_{M_2} \varphi_1)^-), \bigsqcap_{v \in W_2} (R_2^-(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^+)) \\
= \quad & \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \boxslash\varphi_1)
\end{aligned}
$$

$\therefore (w \models_{M_1} \boxslash\varphi_1) = (v \models_{M_2} \boxslash\varphi_1)$

- if $\varphi = \Diamond\varphi_1$

$$
\begin{aligned}
& (w \models_{M_1} \Diamond\varphi_1) \\
= & \quad \{ \text{ definition of } \models \} \\
& (\bigsqcup_{w' \in W_1} (R_1^-(w,w') \sqcap (w' \models_{M_1} \varphi_1)^+), \prod_{w' \in W_1} (R_1^-(w,w') \hookrightarrow (w' \models_{M_1} \varphi_1)^-)) \\
= & \quad \{ \text{ by } (*)_{16} \text{ and Induction Hypothesis } (w \models_{M_1} \varphi) = (v \models_{M_2} \varphi) \} \\
& (\bigsqcup_{v \in W_2} (R_2^-(v,v') \sqcap (v' \models_{M_2} \varphi_1)^+), \prod_{v \in W_2} (R_2^-(v,v') \hookrightarrow (v' \models_{M_2} \varphi_1)^-)) \\
= & \quad \{ \text{ definition of } \models \} \\
& (v \models_{M_2} \Diamond\varphi_1)
\end{aligned}
$$

$\therefore (w \models_{M_1} \Diamond\varphi_1) = (v \models_{M_2} \Diamond\varphi_1)$

□

5

CONCLUSION

5.1 SUMMARY OF CONTRIBUTIONS

In this work we have proposed a new family of transition systems, the Paraconsistent Labelled Transition Systems, also written PLTS, whose transitions are described by two fuzzy relations, besides the usual labelling action. These transition systems are suited to model any dynamic process where there is some kind of contradiction or lack of information regarding the existence of transitions between states, also supporting consistent scenarios. We have established different forms of equivalence between PLTS, given by the definition of morphism, simulation and bisimulation. We also developed the notion of equivalence between states with the definition of traces and trace equivalence.

Motivated by the possible use of PLTS as a model for quantum computation, illustrated below, we defined a category of PLTS and their morphisms, which we endowed with constructions that could serve as a basis for the definition of a process algebra and thus as a possible formalism for parallel quantum computations. We have proposed a modal intuitionistic paraconsistent logic, MIPL, whose semantic models are defined over PLTS. The valuation of propositions and the satisfaction of formulas in MIPL model the lack and excess of information, besides consistent valuations. MIPL supports inconsistency and vagueness both at the level of the accessibility relations in its underlying relational structure and at the level of proposition variables. Given some process modeled by a PLTS, MIPL is a tool for talking about its properties as well as to compare the satisfaction of formulas in models related by the equivalence notions mentioned above.

5.2 MODELING QUANTUM CIRCUITS - AN APPLICATION OF PLTS

Quantum circuits (QC) are a promising model for quantum computing, but superconducting qubits may hold in a superposition state for a limited period of time, *i.e.* the coherence time. Decoherence consists in decay of a qubit in superposition to its ground state and may be caused by distinct physical phenomena, each with a certain probability of occurring. A

quantum circuit is effective only if gate operations and measurements are performed to superposition states within a limited period of time after their preparation. When that time is exceeded there is an increasing probability that the circuit does not behave according to its design, since there may be decayed states in place of superposition states. One approach to solve this problem is to enhance superconducting qubits performance, increasing their coherence time. If the coherence time is large enough to ensure that no decay will occur over the time the circuit is being executed, decoherence is no longer an error factor to quantum computations. Here we do not explore this possibility. Instead, we provide a model for quantum circuits which incorporates the possible decoherence of state of the art superconducting qubits as an error factor. This way, quantum circuits are translated into a structure which models not only the desired computation but the behavior of the circuit when executed in a real setting.

5.2.1 *From quantum circuits to PLTS*

We will use PLTS to model quantum circuits, taking advantage of their accessibility relations in the following way. Coherence of qubits is not an exact measure, but usually given by a time interval. This fixes two values of coherence, corresponding to a worst case scenario and a best case scenario. We employ the two accessibility relations in a PLTS to model both scenarios simultaneously.

Other important observation for the conversion of quantum circuits to PLTS is that quantum circuits always have a sequential execution. Simultaneous operations performed to distinct qubits are combined using the tensor product construction $\otimes$ into a single operation to the whole collection of qubits treated by the circuit.

Thus a quantum circuit may be described by a sequence of executions $e_1, e_2, e_3, ...$ where each e_i is the tensor product of the operations performed to the qubits at each execution step. Starting from the initial state where, for instance, all qubits are in the state $|0\rangle$, each e_i takes the collection of qubits to a new state, specified by the state of each qubit after e_i is performed.

Example 17. Consider, for instance, the following circuit designed with IBM Quantum Composer, an online tool for designing and testing quantum circuits. It is a simple circuit which creates a superposition state in a qubit resister, with the application of the Hadamard gate, and after performs a measurement to that qubit.

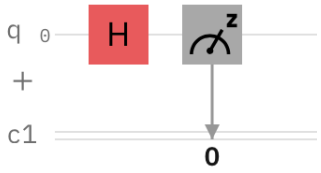


Figure 3: Circuit 1 (designed with IMB Quantum Composer online software)

This quantum circuit deals with a single qubit register q. Its initial state s_1 is given by the initial state of q and its first execution e_1 is the application of the Hadamard gate to q, which leaves q in the familiar superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle$. The last execution e_2 measures the state of q leaving it in a definite state, either $|0\rangle$ or $|1\rangle$. In short, this circuit is described by the following sequence of states s_1, s_2 and s_3.

$$s_1 : q[0] = |0\rangle; s_2 : H(q[0]); s_3 : M(H(q[0]))$$

H and M stand for Hadamard gate and measurement, respectively.

Now consider another circuit, given below, which differs from the previous one only by the addition of a new qubit register.

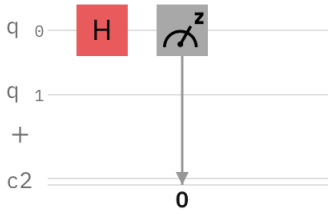


Figure 4: Circuit 2 (designed with IMB Quantum Composer online software)

Although the addition of a second qubit does not change the outcome of this computation, the states of the latter circuit differ from those in the former, since each circuit state is specified by the states of all its qubits at each particular instant. Moreover, when the Hadamard gate is applied to the first qubit, it is being omitted that the Identity is simultaneously applied to the second qubit. Then, the sequence of states describing this circuit is distinct from the sequence above, and given by:

$$s_1 : q[0] = |0\rangle\,; q[1] = |0\rangle; s_2 : H \otimes I(q[0] \otimes q[1]); s_3 : M \otimes I(H \otimes I(q[0] \otimes q[1]))$$

This leads to the first rules of conversion from QC to PLTS.

1. Each circuit state, given by the states of all qubit registers between executions, corresponds to a state of a PLTS, and the starting state of the circuit is the initial state of that PLTS;

2. Each execution, taking the circuit from a state to another, defines a transition between the corresponding states of the PLTS, and the action labeling this transition is given by the tensor product of the gates in that execution (no gate means the Identity gate).

Now we need a procedure to assign values to the positive and negative accessibility relation of each transition that takes into account the possibile decoherence of superposition states. The main idea is that when performing a gate operation with execution time τ_G or a measurement with execution time τ_M to a superposition state whose coherence time is τ, there is a probability τ_G/τ that the gate is applied to a decayed state, and a probability τ_M/τ that the measurement is applied to a decayed state. These values are defined under the following considerations:

- After τ the state has decayed with probability 1 and
- the probability of decoherence has a linear time evolution.

Of course, the last assumption may be suitably replaced given another (maybe more realistic) time evolution for the probability of decoherence.

Considering the minimum coherence time $\tau_{\min}$ and the maximum coherence time $\tau_{\max}$, two values are yield for the possibility of decoherence during some circuit execution. The value of $\tau_{\max}$ will determine the positive accessibility relation, interpreted as the probability to which the system remains coherent, and the value of $\tau_{\min}$ will determine the negative accessibility relation, interpreted as the probability to which the system evolves to a decoherent state. How coherence and execution times shape the accessibility relations is illustrated in the examples below.

For a somewhat realistic model we take the values of coherence time and gate or measurement execution from [ZDL+19] and define:

- The largest and smallest coherence time of a superconducting qubit are $\tau_{\max} = 100\mu s$ and $\tau_{\min} = 10\mu s$, respectively;
- The execution time of a single qubit gate is $\tau_G = 20\mu s$ and the execution time of a two qubit gate is $2\tau_G = 40\mu s$;
- The time execution of a measurement is $\tau_M = 300ns \sim 1\mu s$.

For a simplified model we consider the time execution of a measurement to be $\tau_M = 1\mu s$ hereafter.

Example 18. We start with the simple example given by the first circuit of 17. We give an intermediate representation of this circuit obtained through the application of rules 1 and 2. This gives the states and the transitions with respective labelling actions.

$$
\begin{array}{c}
q[0] : |0\rangle \\
\Big\downarrow H \\
q[0] : H(|0\rangle) \\
\Big\downarrow M \\
q[0] : M(H(|0\rangle))
\end{array}
$$

The positive and negative accessibility relations characterizing each transition are given as follows.

1. For the first transition, labelled with H:

   Note that the Hadamard gate is applied to a qubit in a definite state $|0\rangle$. The problem of decoherence does not concern qubits in a definite state, but only those in superposition states. Then coherence of the qubit is guaranteed during this execution and the probability of decoherence is just 0. We conclude the values for the positive and negative accessibility relations are 1 and 0, respectively.

2. For the second transition, labelled with M:

   M is applied to a superposition state, so indeed there is some probability that the state decays during this execution.

   On the one hand, we consider the best case scenario, *i.e.* that the qubit has a coherence time of $100\mu s$. After $100\mu s$ the state has decayed with probability 1. So after $\tau_M = 1\mu s$ the state has decayed with probability $1\mu s/100\mu s = 0.01$. In this case the state is coherent with probability 0.99. This is the value for the positive accessibility relation.

   On the other hand, we consider the worst case scenario, *i.e.* that the qubit has a coherence time of $10\mu s$. In this case the state decays with probability $1\mu s/10\mu s = 0.1$. This is the value for the negative accessibility relation.

Therefore the circuit translates into the following PLTS.

$$
\begin{array}{c}
q[0] : |0\rangle \\
\Big\downarrow (H, 1, 0) \\
q[0] : H(|0\rangle) \\
\Big\downarrow (M, 0.99, 0.1) \\
q[0] : M(H(|0\rangle))
\end{array}
$$

Moreover, the PLTS for the second circuit of 17 is constructed following the same steps and depicted below.

$$q[0];q[1] \xrightarrow{(H \otimes Id, 1, 0)} q[0];q[1] \xrightarrow{(M \otimes Id, 0.99, 0.1)} q[0];q[1]$$

Example 19. Now consider the following quantum circuit, which creates two superposition states by applying the Hadamard gate to two distinct qubit registers and then creates an entangled state with the application of a CNOT gate to both qubits.

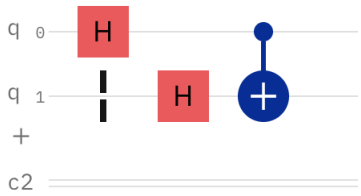


Figure 5: Circuit 3 (designed with IMB Quantum Composer online software)

The intermediate representation of this circuit obtained through the application of rules 1 and 2 is the following.

$$q[0];q[1] \xrightarrow{H \otimes I} q[0];q[1] \xrightarrow{I \otimes H} q[0];q[1] \xrightarrow{CNOT} q[0];q[1]$$

Now we must give the values of the positive and negative accessibility relations characterizing each transition.

1. For the first transition, labelled with $H \otimes I$:

   Again, it is straightforward that the positive and negative accessibility relations are 1 and 0, respectively.

2. For the second transition, labelled with $I \otimes H$:

   Note that when the Hadamard gate is applied to a register in a definite state which will not decay during this transition. Nonetheless, the first qubit is in a superposition state and it may decay, leaving the system in an unwanted state.

   Through the time of this execution, given by $\tau_G = 20\mu s$, the first qubit has, at most, a probability $1 - (20\mu s/100\mu s) = 0.8$ of staying coherent. The probability that the state decays to its ground state cannot be given by the formula $\tau_G/10\mu s$, since this yields a value greater than 1. When $\tau_G/10\mu s > 1$ we simply say that the state decays with probability 1.

3. For the last transition, labelled with $CNOT$:

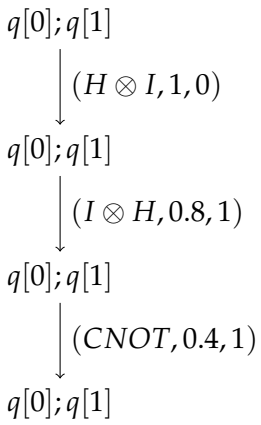
   $CNOT$ is a two qubit gate, with an execution time of $40\mu s$. By the time this transition is performed, both qubits are in a superposition state so the system may collapse due to decoherence of either of them. The qubit which has been in superposition for the longest time obviously has a greater chance of decaying to its ground state. The circuit performs correctly if:

   - The first qubit holds in superposition for the time execution of the second and third transitions, given by $20\mu s + 40\mu s = 60\mu s$.
   - The second qubit holds in superposition for the time execution of the third transition, which is $40\mu s$.

   The first qubit has at most a probability $1 - (60\mu s/100\mu s) = 0.4$ of staying coherent and the second qubit has at most a probability of $1 - (40\mu s/100\mu s) = 0.6$ of staying coherent. The positive accessibility relation is given by the minimum of these values, *i.e.* 0.4.

   Both qubits decay with probability 1, since the $60\mu s/10\mu s$ yields a value greater than 1, as well as $40\mu s/10\mu s$.

Therefore the circuit translates into the following PLTS.



Note that the CNOT gate acts as if fresh superposition states were prepared, resetting their coherence. Any gate leaving a qubit in a "new" superposition state restores its coherence and only the circuit executions after this preparation should be accounted to inspect coherence or decoherence of the system subsequently.

Given a quantum circuit the following steps give a systematic procedure to convert that circuit into a PLTS:

1. Each circuit state, given by the states of all qubit registers between executions, corresponds to a state of a PLTS, and the starting state of the circuit is the initial state of that PLTS;
2. Each execution, taking the circuit from a state to another, defines a transition between the corresponding states of the PLTS, and the action labeling this transition is given by the tensor product of the gates in that execution (no gate means the Identity gate);
3. Compute the positive and negative accessibility relations characterizing each transition as follows. Let t be a transition between s_1 and s_2.

3.1 If state s_1 is such that all registers are in a definite state, the positive and negative accessibility relations are 1 and 0, respectively.

3.2 If state s_1 is such that some registers $q_1, q_2, ..., q_n$ are in superposition, for each q_i compute: (i) the execution time τ of all transitions immediately after q_i's superposition was prepared and until s_1; (ii) the probability that q_i stays coherent in the best case scenario, given by $P_i^+ = \tau/\tau_{\max}(q_i)$, and (iii) the probability that q_i decays in the worst case scenario, given by $P_i^- = \tau/\tau_{\min}(q_i)$, where $\tau_{\max}(q_i)$ and $\tau_{\min}(q_i)$ are the maximal and minimal coherence times of q_i. Finally (iv) compute the value for the positive accessibility relation as

$$r^+ = \begin{cases} 1 \text{ if } \bigwedge_{i=1}^n \{1 - P_i^+\} \geq 1 \\ \bigwedge_{i=1}^n \{1 - P_i^+\} \text{ if } \bigwedge_{i=1}^n \{1 - P_i^+\} \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

and the value for the negative accessibility relation as

$$r^- = \begin{cases} 1 \text{ if } \bigvee_{i=1}^{n}\{P_i^-\} \geq 1 \\ \bigvee_{i=1}^{n}\{P_i^-\} \text{ otherwise} \end{cases}$$

5.3 PROSPECT FOR FUTURE WORK

Enrich MIPL with a consistency connective

In the same way the conflation operation was used to determine if a particular transition of an A-PLTS, where A is an MTL-algebra over $[0,1]$, was within the intuitionistic, strictly consistent or paraconsistent domain, it could be used to determine the consistency of formulas in an MIPL model over a such a PLTS. A formula φ is considered consistent if the evidence of it being true and the evidence of it being false are non-contradictory, which in the fuzzy framework means they add to 1 or to a value less than 1. In other words, if the satisfaction of φ is given by the pair (a,b), φ is consistent only if $(a,b) \leq_{\frown} (a,b)$ where $\frown$ is, as before, the conflation operation.

In light of this consideration, we could extend MIPL grammar with a consistency connective $\circ$ and define the satisfaction of a formula $\circ\varphi$ in a world w of an MIPL model over an A-PLTS as follows.

$$(w \models \circ\varphi) = \begin{cases} (1,0) \text{ iff } (a,b) \leq_{\frown} (a,b) \\ (0,1) \text{ otherwise} \end{cases}$$

Another possibility is to endow the underlying MTL-algebra of a PLTS with the structure of a metric space and a notion of distance.

Definition 27. $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow, d\rangle$ is a metric MTL-algebra if

1. $\mathbf{A} = \langle A, \sqcap, \sqcup, 1, 0, \hookrightarrow\rangle$ in an MTL-algebra, and
2. (A,d) is a metric space, that is, for any $x,y,z \in A$, $d : A \times A \to \mathbb{R}_0^+$ is such that:
   - $d(x,y) = 0$ iff $x = y$
   - $d(x,y) \leq d(x,z) + d(z,y)$

Moreover, we can define a bilattice over $\mathbf{A}$ and a distance metric $D : (A \times A) \times (A \times A) \to \mathbb{R}_0^+$ over bilattice pairs. For instance, for any $(a,b),(c,d) \in A \times A$ we could have

$$D((a,b),(c,d)) = \sqrt{d(a,c)^2 + d(c,d)^2}$$

or

$$D((a,b),(c,d)) = d(a,c) + d(c,d)$$

In this way we could define the sets of intuitionistic and paraconsistent pairs, Δ_P, Δ_I, respectively, as

$$\Delta_I = \{(a,b) | D((a,b),(0,0)) \leq D((a,b),(1,1))\}$$

$$\Delta_P = \{(a,b) | D((a,b),(1,1)) \leq D((a,b),(0,0))\}$$

Then, it is possible to establish if a bilattice element (a,b) represents incomplete or contradictory information by comparing its distance to the elements $(0,0)$ and $(1,1)$. Moreover, the bilattice elements equally distant from $(0,0)$ and $(1,1)$ are those representing complete information, lying in the set of strictly consistent pairs Δ, formally defined as

$$\Delta = \Delta_P \cap \Delta_I$$

A formula in MIPL is consistent when its satisfaction is given by (a,b) such that $(a,b) \in \Delta_I$ (note that $\Delta \subset \Delta_I$); it is inconsistent when $(a,b) \in \Delta_P \setminus \Delta$. Thus satisfaction of a formula $\circ\varphi$ in a world w of a MIPL model over a metric MTL-PLTS would be defined as

$$(w \models \circ\varphi) = \begin{cases} (1,0) \text{ iff } (w \models \varphi) \in \Delta_I \\ (0,1) \text{ otherwise} \end{cases}$$

Develop a process language and design a dynamic extension of MIPL

The constructions defined in Chapter 3 could be explored to develop a process language. Having programs described by PLTS, as, for instance, quantum circuit executions, such a language would allow to represent parallel executions of those programs. Following this line of work, the next step would be to extend MIPL to a dynamic logic.

Improve the description of quantum circuits as PLTS

Finally, it would be interesting to further explore the description of quantum circuits as PLTS and improve the method described above for this conversion. The notions of simulation and bisimulation fail to describe the equivalence of circuits in the examples above, so one needs to study how to better characterize suitable mappings between PLTS. If the conversion from QC to PLTS is worthwhile in the engineering of quantum software, it is possible to design a tool for representing classes of equivalent quantum algorithms and to define metrics of quality, establishing which circuit implementation better performs a given algorithm.

BIBLIOGRAPHY

[BEGR09] Félix Bou, Francesc Esteva, Lluís Godo, and Ricardo Oscar Rodríguez. On the Minimum Many-Valued Modal Logic over a Finite Residuated Lattice. *Journal of Logic and Computation*, 21(5):739–790, 10 2009.

[Bel77] Nuel Belnap. A useful four-valued logic. 1977.

[BS11] Alexandru Baltag and Sonja Smets. Quantum logic as a dynamic logic. *Synthese*, 179(2):285–306, 2011.

[BvN37] Garrett Birkhoff and John von Neumann. The logic of quantum mechanics. *Journal of Symbolic Logic*, 2(1):44–45, 1937.

[CCM07] Walter Carnielli, Marcelo E. Coniglio, and João Marcos. Logics of Formal Inconsistency. *Handbook of Philosophical Logic*, pages 1–93, 2007.

[Com98] Stephen D. Comer. Paul halmos and steven givant. logic as algebra. the dolciani mathematical expositions, no. 21. the mathematical association of america, washington 1998, ix 141 pp. *Journal of Symbolic Logic*, 63(4):1604–1604, 1998.

[EG01] Francesc Esteva and Lluis Godo. Godo, l.: Monoidal t-norm based logic: Towards a logic for left-continuous t-norms. fuzzy sets and systems 124(3), 271-288. *Fuzzy Sets and Systems*, 124:271–288, 12 2001.

[Fit89] Melvin Fitting. Bilattices and the theory of truth. *Journal of Philosophical Logic*, 18(3):225–256, 1989.

[Flo67] Robert W. Floyd. Assigning meanings to programs. *Mathematical aspects of computer science*, 19(19-32):1, 1967.

[Gin86] Matthew L. Ginsberg. Multi-valued logics. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86, pages 243–247. AAAI Press, 1986.

[Gin88] Matthew Ginsberg. Multivalued logics: A uniform approach to reasoning in ai. *Computer Intelligence*, 4(1):256–316, 1988.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, October 1969.

[HTK00] David Harel, Jerzy Tiuryn, and Dexter Kozen. *Dynamic Logic*. MIT Press, Cambridge, MA, USA, 2000.

[Jaś69a] Stanisław Jaśkowski. Propositional calculus for contradictory deductive systems. *Studia Logica*, 24(1):143–157, 1969.

[Jaś69b] Stanisław Jaśkowski. Propositional calculus for contradictory deductive systems. *Studia Logica*, 24(1):143–157, 1969.

[Kel76] Robert Keller. Formal verification of parallel programs. *Commun. ACM*, 19:371–384, 07 1976.

[Koz85] Dexter Kozen. A probabilistic pdl. *Journal of Computer and System Sciences*, 30(2):162–178, 1985.

[Kra98] Marcus Kracht. On extensions of intermediate logics by strong negation. *Journal of Philosophical Logic*, 27(1):49–73, 1998.

[KS14] Sofia Kouah and Djamel Eddine Saidouni. Fuzzy labeled transition refinement tree: Application to stepwise designing multi agent systems. *International Journal of Agent Technologies and Systems*, 6:1–31, 07 2014.

[LL34] C. I. Lewis and C. H. Langford. Symbolic logic. *Erkenntnis*, 4(1):65–66, 1934.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[MN04] Erik Meineche Schmidts Mikkel Nygaard. *DAIMI FN : Transition systems : algorithms and data structures*. Matematisk Institut, Aarhus Universitet, Datalogisk Afdeling, Aarhus, 2004.

[MNM16] Alexandre Madeira, Renato Neves, and Manuel A. Martins. An exercise on the generation of many-valued dynamic logics. *Journal of Logical and Algebraic Methods in Programming*, 85(5, Part 2):1011–1037, 2016. Articles dedicated to Prof. J. N. Oliveira on the occasion of his 60th birthday.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[OW10] Sergei P. Odintsov and Heinrich Wansing. Modal logics with belnapian truth values. *Journal of Applied Non-Classical Logics*, 20(3):279–301, 2010.

[PBW18] Graham Priest, Francesco Berto, and Zach Weber. Dialetheism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2018 edition, 2018.

[Pla10] Andr Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[Plo04] Gordon Plotkin. A structural approach to operational semantics. *J. Log. Algebr. Program.*, 60-61:17–139, 07 2004.

[Pre18] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, Aug 2018.

[RJJ15] Umberto Rivieccio, Achim Jung, and Ramon Jansana. Four-valued modal logic: Kripke semantics and duality. *Journal of Logic and Computation*, 27(1):155–199, 06 2015.

[Sed16] Igor Sedlár. Propositional dynamic logic with belnapian truth values. 08 2016.

[Usp92] Vladimir A. Uspensky. Kolmogorov and mathematical logic. *Journal of Symbolic Logic*, 57(2):385–412, 1992.

[Vak77] D. Vakarelov. Notes on n-lattices and constructive logic with strong negation. *Studia Logica*, 36(1):109–125, 1977.

[vAS15] Mark van Atten and Göran Sundholm. L.e.j. brouwer's 'unreliability of the logical principles'. a new translation, with an introduction, 11 2015.

[WD38] Morgan Ward and R. P. Dilworth. Residuated lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 24(3):162–164, 1938.

[WN95] Glynn Winskel and Mogens Nielsen. *Models for Concurrency*, pages 1–148. Oxford University Press, Inc., USA, 1995.

[ZDL+19] Yu Zhang, Haowei Deng, Quanxi Li, Haoze Song, and Leihai Nie. Optimizing quantum programs against decoherence: Delaying qubits into quantum superposition. *2019 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, Jul 2019.

A

SUPPORT MATERIAL

Auxiliary results which are not main-stream; or

Details of results whose length would compromise readability of main text; or

Specifications and Code Listings: should this be the case; or

Tooling: Should this be the case.

NB: place here information about funding, FCT project, etc in which the work is framed. Leave empty otherwise.