Probabilistic Neighbourhood Logic

Dimitar P. Guelev

International Institute for Software Technology
of the United Nations University
(UNU/IIST), Macau, P.O.Box 3058.
E-mail: dg@iist.unu.edu

Abstract. This paper presents a probabilistic extension of Neighbourhood Logic (NL,[14, 1]). The study of such an extension is motivated by the need to supply the Probabilistic Duration Calculus (PDC, [10, 4]) with a proof system. The relation between the new logic and PDC is similar to that between DC [15] and ITL [12, 3]. We present a complete proof system for the new logic.

Introduction

The Probabilistic Duration Calculus (PDC) was introduced in [10] as an extension of Duration Calculus[15]. The approach to introducing PDC is as follows: Consider some finite probabilistic timed automaton $\mathbf{A}$. The behaviours of $\mathbf{A}$ can be represented as a set $\mathbf{M}$ of DC models. The probabilistic laws that govern the working of $\mathbf{A}$ are used to introduce probability on the subsets of $\mathbf{M}$. Given a DC formula D, the term $\mu(D)(t)$ denotes the probability of those models from $\mathbf{M}$ that satisfy D at the interval $[0, t]$. Terms of this sort are the component of PDC language that is new in PDC, relative to DC. In [10] the authors focused on the case of discrete time for the sake of simplicity. In a later work, [4], PDC was introduced for the case of real time too.

Both papers present examples of specification by PDC and a number of valid PDC formulas, that represent basic properties of the probabilistic operator μ. A section on specification by PDC can be found in [11] too. However, no complete proof system for PDC has been proposed so far.

DC is an extension of Interval Temporal Logic (ITL), and so is its proof system. ITL has a complete proof system with respect to an abstract class of frames[3]. In this paper, we introduce Probabilistic Neighbourhood Logic (PNL) by generalising the semantics of PDC. PNL is designed to take the role that ITL has for DC, yet for PDC. PNL is based on Neighbourhood Logic (NL, [14]), which is another interval-based temporal logic, closely related to ITL. Unlike ITL, NL has modal operators which allow reference to intervals outside the current one. This feature has proved useful for the axiomatisation of the probabilistic operator of PNL. NL has a proof system, that is complete with respect to an abstract semantics, which is similar to that of ITL[1].

In this paper we extend the proof system of NL to obtain a complete one for PNL for a similarly abstract semantics. Earlier versions of PNL have been studied by the author in [6, 7] and by Vladimir Trifonov in [13].

1 Preliminaries on Neighbourhood Logic

Neighbourhood logic is a classical first order predicate logic with equality and two unary normal modal operators.

1.1 Language

A language of NL is determined by a countable set of *individual variables* x, y, ..., and several other sets of symbols. These are *constant* symbols c, d, ..., *function* symbols f, g, ... and *relation* symbols R, S, ... Symbols of every kind can be either *rigid* or *flexible*, depending on the way they are interpreted.

Given the sets of symbols, the *terms* t and the *formulas* φ of the corresponding NL languages are defined by the BNFs:
$t ::= c|x|f(t, \ldots, t)$
$\varphi ::= \bot|R(t, \ldots, t)|(\varphi \Rightarrow \varphi)|\exists x\varphi|\Diamond_l\varphi|\Diamond_r\varphi$

Function symbols and relation symbols are assigned *arity* to denote the number of arguments they admit. Every NL language contains the rigid constant symbol 0, the rigid binary function symbol $+$, the rigid binary relation symbols $=$ and $\leq$ and the flexible constant ℓ.

Individual variables are regarded as rigid. Formulas and terms which contain no flexible symbols, are called *rigid* too. The set of individual variables that have *free occurrences* in a formula φ is denoted by $FV(\varphi)$.

1.2 Frames, Models and Satisfaction

Definition 1. *A* NL time domain *is a linearly ordered set. A* NL duration domain *is an algebraic system of the type* $\langle D, +^{(2)}, 0^{(0)}, \leq^{(2)}\rangle$ *which satisfies the axioms:*

$(D1)$ $x + (y + z) = (x + y) + z$	$(D6)$ $x \leq x$
$(D2)$ $x + 0 = x$	$(D7)$ $x \leq y \wedge y \leq x \Rightarrow x = y$
$(D3)$ $x + y = x + z \Rightarrow y = z$	$(D8)$ $x \leq y \wedge y \leq z \Rightarrow x \leq z$
$(D4)$ $\exists z(x + z = y)$	$(D9)$ $x \leq y \Leftrightarrow \exists z(x + z = y \wedge 0 \leq z)$
$(D5)$ $x + y = y + x$	

We use $\leq$ to denote both the ordering of time and duration domains.

Definition 2. *Given a time domain* $\langle T, \leq\rangle$*, the set of the closed* intervals $\{[\tau_1, \tau_2] : \tau_1, \tau_2 \in T, \tau_1 \leq \tau_2\}$ *in* T *is denoted by* $\mathbf{I}(T)$*. Given a time domain* $\langle T, \leq\rangle$ *and a duration domain* $\langle D, +, 0, \leq\rangle$*, a* measure function m *is a surjective function of type* $\mathbf{I}(T) \to D$*, which satisfies the axioms:*
$(M1)$ $m(\sigma) = m(\sigma') \wedge \min\sigma = \min\sigma' \Rightarrow \max\sigma = \max\sigma'$
$(M2)$ $\max\sigma_1 = \min\sigma_2 \Rightarrow m(\sigma_1) + m(\sigma_2) = m(\sigma_1 \cup \sigma_2)$
$(M3)$ $m(\sigma) = x + y \Rightarrow \exists\sigma'(\min\sigma' = \min\sigma \wedge m(\sigma') = x)$.

Definition 3. *A tuple of the kind* $\langle\langle T, \leq\rangle, \langle D, +, 0, \leq\rangle, m\rangle$*, where* $\langle T, \leq\rangle$ *is a time domain,* $\langle D, +, 0, \leq\rangle$ *is a duration domain, and* m *is a measure from* $\mathbf{I}(T)$ *to* D*, is called* NL frame.

Clearly, if a measure function from a time domain $\langle T, \leq \rangle$ to a duration domain $\langle D, +, 0, \leq \rangle$ exists, $\langle D, \leq \rangle$ is isomorphic to $\langle T, \leq \rangle$. For this reason NL is usually regarded as having just duration domains in its frames. We keep the two components of NL frames distinct for the sake of compatibility with ITL semantics, where they may differ more.

Let **L** be an NL language.

Definition 4. *Let* $F = \langle \langle T, \leq \rangle, \langle D, +, 0, \leq \rangle, m \rangle$*, where* $\langle T, \leq \rangle$ *be an* NL *frame. A function* I *which is defined on the set of symbols of* **L** *and satisfies the requirements:*
$I(x), I(c) \in D$ *for individual variables* x *and rigid constants* c
$I(f) \in (D^n \to D)$ *for* n*-place rigid function symbols* f
$I(R) \in (D^n \to \{0,1\})$ *for* n*-place rigid relation symbols* R
$I(c) \in (\mathbf{I}(T) \to D)$ *for flexible constants* c
$I(f) \in (\mathbf{I}(T) \times D^n \to D)$ *for* n*-place flexible function symbols* f
$I(R) \in (\mathbf{I}(T) \times D^n \to \{0,1\})$ *for* n*-place flexible relation symbols* R
$I(0) = 0$, $I(+) = +$, $I(\ell) = m$, $I(\leq)$ *is* $\leq$ *and* $I(=)$ *is* $=$
is called interpretation of **L** into F.

Definition 5. *A* model for **L** *is a tuple of the kind* $\langle F, I \rangle$*, where* F *is a frame, and* I *is an interpretation of* **L** *into* F*.*

Given a frame F, we denote its components by $\langle T_F, \leq_F \rangle$, $\langle D_F, +_F, 0_F, \leq_F \rangle$ and m_F, respectively. The same applies to models. We denote the frame and the interpretation of a model M by F_M and I_M, respectively.

Given a symbol s from **L**, interpretations I and J of **L** into frame **F** are said to *s-agree*, if $I(s') = J(s')$ for **L** symbols s' other than s.

Definition 6. *Let* M *be a model for* **L**. *Let* $\sigma \in \mathbf{I}(T_M)$*. The* values $I_\sigma(t)$ *of terms* t *from* **L** *are defined as follows:*
$I_\sigma(x) = I_M(x)$, $I_\sigma(c) = I_M(c)$ *for variables* x *and rigid constants* c
$I_\sigma(f(t_1, \ldots, t_n)) = I_M(f)(I_\sigma(t_1), \ldots, I_\sigma(t_n))$ *for rigid* n*-place function symbols* f
$I_\sigma(f(t_1, \ldots, t_n)) = I_M(f)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_n))$ *for flexible* n*-place function symbols* f
The relation $M, \sigma \models \varphi$ *for formulas* φ *from* **L** *is defined as follows:*
$M, \sigma \not\models \bot$
$M, \sigma \models R(t_1, \ldots, t_n)$ *iff* $I_M(R)(I_\sigma(t_1), \ldots, I_\sigma(t_n)) = 1$ *for rigid relation symbols* R
$M, \sigma \models R(t_1, \ldots, t_n)$ *iff* $I_M(R)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_n)) = 1$ *for flexible relation symbols* R
$M, \sigma \models (\varphi \Rightarrow \psi)$ *iff either* $M, \sigma \models \psi$*, or* $M, \sigma \not\models \varphi$
$M, \sigma \models \exists x \varphi$ *iff* $\langle F_M, J \rangle, \sigma \models \varphi$ *for some* J *that* x*-agrees with* I_M
$M, \sigma \models \Diamond_l \varphi$ *iff* $M, \sigma' \models \varphi$ *for some* $\sigma' \in \mathbf{I}(T_M)$ *such that* $\max \sigma' = \min \sigma$
$M, \sigma \models \Diamond_r \varphi$ *iff* $M, \sigma' \models \varphi$ *for some* $\sigma' \in \mathbf{I}(T_M)$ *such that* $\min \sigma' = \max \sigma$

1.3 Abbreviations

Along with ordinary classical first order predicate logic abbreviations and infix notation, the following NL-specific abbreviations are used:

$\Diamond_d^c\varphi \rightleftharpoons \Diamond_d\Diamond_{\overline{d}}\varphi \quad \Box_d\varphi \rightleftharpoons \neg\Diamond_d\neg\varphi \quad \Box_d^c\varphi \rightleftharpoons \neg\Box_d^c\neg\varphi \qquad d \in \{l, r\}$, $\bar{l} = r$ and $\overline{r} = l$.

The modal operator $(.;.)$ of ITL is defined as an abbreviation in NL by putting:

$$(\varphi;\psi) \rightleftharpoons \exists x \exists y (x + y = \ell \wedge \Diamond_l^c(\varphi \wedge \ell = x) \wedge \Diamond_r^c(\psi \wedge \ell = y)),\ x, y \notin FV((\varphi;\psi)).$$

1.4 Proof System for NL

The proof system of NL consists of axioms for classical first order predicate logic with equality, the axioms $D1$-$D9$ and the following axioms and rules:

$(A1)$ $\Diamond_d\varphi \Rightarrow \varphi$ if φ is rigid.
$(A4')$ $\Diamond_d\exists x\varphi \Rightarrow \exists x\Diamond_d\varphi$
$(A2)$ $0 \leq l$
$(A5)$ $\Diamond_d(\ell = x \wedge \varphi) \Rightarrow \Box_d(\ell = x \Rightarrow \varphi)$
$(A3)$ $0 \leq x \Rightarrow \Diamond_d(\ell = x)$
$(A6)$ $\Diamond_d^c\varphi \Rightarrow \Box_d\Diamond_{\overline{d}}\varphi$
$(A4)$ $\Diamond_d(\varphi \vee \psi) \Rightarrow \Diamond_d\varphi \vee \Diamond_d\psi$
$(A7)$ $\ell = x \Rightarrow (\varphi \Leftrightarrow \Diamond_d^c(\ell = x \wedge \varphi))$
$(A8)$ $0 \leq x \Rightarrow 0 \leq y \Rightarrow \Diamond_d(\ell = x \wedge \Diamond_d(\ell = y \wedge \Diamond_d\varphi)) \Rightarrow \Diamond_d(\ell = x + y \wedge \Diamond_d\varphi)$

$(Mono)$ $\dfrac{\varphi \Rightarrow \psi}{\Diamond_d\varphi \Rightarrow \Diamond_d\psi}$ (Nec) $\dfrac{\varphi}{\Box_d\varphi}$ (MP) $\dfrac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$ (G) $\dfrac{\varphi}{\forall x\varphi}$

Substitution $[t/x]\varphi$ of variable x by term t in formula φ *is allowed in proofs* only if either t is rigid, or x does not occur in the scope of modal operators in φ. This system is complete with respect to the above semantics[1].

2 Probabilistic Timed Automata: an Introductory Example to PNL

Here we slightly generalise the notion of finite probabilistic timed automaton from [4].

Definition 1. *A* finite probabilistic timed automaton *is a system of the kind* $\mathbf{A} = \langle S, A, s_0, \langle D, +, 0, \leq\rangle, \langle q_a, a \in A\rangle, \langle P_a : a \in A\rangle\rangle$, *where*

S *is a finite set of* states*;*
$A \subset \{\langle s, s'\rangle : s, s' \in S, s \neq s'\}$ *is a set of* transitions*;*
$s_0 \in S$ *is called* initial state*;*
$\langle D, +, 0, \leq\rangle$ *is a duration domain;*
$q_a \in [0, 1]$ *is the* choice probability *for transition* $a \in A$*;*
$P_a \in (D \to [0, 1])$ *is the* duration distribution *of transition* a*.*

Given $\mathbf{A}$ *with its components named as above,* A_s*, denotes* $\{s' \in S : \langle s, s'\rangle \in A\}$. $\langle q_a : a \in A\rangle$ *are required to satisfy* $\sum\limits_{a \in A_s} q_a = 1$ *for* $A_s \neq \emptyset$. $\langle P_a : a \in A\rangle$, *are required to satisfy* $P_a(0) \geq 0$*, to be non-strictly monotonic and to converge towards* 1.

An automaton $\mathbf{A}$ of the above kind works by going through a finite or infinite sequence of states s_0, s_1, $\ldots$, s_n, $\ldots$ such that $\langle s_i, s_{i+1}\rangle \in A$ for all i. Each transition has a duration, d_i. Thus, individual *behaviours* of $\mathbf{A}$ are recorded as sequences of the kind $\langle a_0, d_0\rangle, \ldots \langle a_n, d_n\rangle, \ldots$, $a_i \in A$, $d_i \in D$, where the initial state of a_0 is s_0, and every transition arrives at the initial state of the next one.

Having arrived at state s, $\mathbf{A}$ chooses transition $a \in A_s$ with probability q_a. The probability for its duration to be no bigger than d is $P_a(d)$.

Given a language $\mathbf{L}$ for NL with a 0-place temporal relation symbol a for every $a \in A$, a behaviour $\langle a_i, d_i \rangle$, $i = 0, 1, \dots$ can be represented as a model $\langle F, I \rangle$ for this language, where $F = \langle \langle D, \leq \rangle, \langle D, +, 0, \leq \rangle, \lambda\sigma. \max\sigma - \min\sigma \rangle$ by putting $I(a)(\sigma) = 1$ iff $a = a_i$ and $\sigma = \left[\sum\limits_{j<i} d_i, \sum\limits_{j \leq i} d_i \right]$ for some i.

Some properties of $\mathbf{A}$ behaviours can be straightforwardly expressed under this convention. For example, if $a = \langle s, s' \rangle \in A$,

$$\Diamond_r^c a \Rightarrow \neg \left(\bigvee_{b \notin A_{s'}} \Diamond_r b \right)$$

means that a behaviour which ends at a can only continue with a transition whose initial state is the final state of a, and

$$a \Rightarrow \neg(a; \ell \neq 0)$$

means that no transition can begin at some time point and end in two distinct time points.

Now consider the set $\mathbf{M}$ of all the interpretations of $\mathbf{L}$ into F that represent behaviours of $\mathbf{A}$ in the above way. We need the following definition:

Definition 2. *Let $\tau \in T_F$. We say that interpretations I and J of $\mathbf{L}$ into F τ-*agree *iff they coincide for rigid symbols from $\mathbf{L}$, and coincide for flexible symbols from $\mathbf{L}$ on intervals σ such that* $\max\sigma \leq \tau$.

The probabilistic components $\langle q_a : a \in A \rangle$ and $\langle P_a : a \in A \rangle$ of $\mathbf{A}$ can be used to endow $\mathbf{M}$ with probabilistic structure as follows:

> For every $\tau \in D$ and every $I \in \mathbf{M}$ a probability measure $P_{I,\tau}$ is introduced on the subsets of $\mathbf{M}_{I,\tau} = \{I' \in \mathbf{M} : I'\ \tau\text{-agrees with } I\}$. Given $\mathbf{N} \subset \mathbf{M}_{I,\tau}$, $P_{I,\tau}(\mathbf{N})$ denotes the probability for $\mathbf{A}$ to continue a behaviour that is described by I up to time τ by one from $\mathbf{N}$.

Assume that, in addition to the temporal relation symbols $a \in A$, $\mathbf{L}$ contains the rigid symbols q_a and P_a, $a \in A$, and they are interpreted by the corresponding components of $\mathbf{A}$ in all the interpretations from $\mathbf{M}$. Assume that we introduce an operator p to $\mathbf{L}$ in the following way

- If φ is a formula, then $p(\varphi)$ is a term.
- $I_\sigma(p(\varphi)) = P_{I,\max\sigma}(\{I' \in \mathbf{M}_{I,\max\sigma} : \exists\tau \geq \min\sigma\ \langle F, I' \rangle, [\min\sigma, \tau] \models \varphi\})$

In words, let $p(\varphi)$ evaluate under I to the probability of the set of those interpretations of $\mathbf{L}$ which are continuations of I from time $\max\sigma$ on and satisfy φ at some interval starting at time $\min\sigma$.

Using p, the probabilistic law about $\mathbf{A}$ behaviours can be expressed as follows:

$a \Rightarrow p((a; b \wedge \ell = x)) = q_b.P_b(x)$, if $a = \langle s, s' \rangle$ and $b = \langle s', s'' \rangle$ for some $s, s', s'' \in S$;

$a \Rightarrow p((a; b)) = 0$ otherwise.

To carry out this way of introducing p and its interpretation rigorously, we can consider models that consist of a NL frame F, a set of interpretations $\mathbf{M}$ of $\mathbf{L}$

into F, and a system of probability measures $P_{I,\tau}$, $I \in \mathbf{M}, \tau \in T_F$, as specified as above.

Having in mind that the values of p-terms are not necessarily similar to durations, F should contain a separate domain for probabilities too. Accordingly, languages that interpretations from $\mathbf{M}$ are defined on should have a sort for this domain. This is essentially what PNL models are.

3 A Formal Definition of PNL

3.1 Languages

A PNL language is built starting from the same kinds of symbols as a NL language. PNL languages are two-sorted. Together with the well-known sort of *durations*, they have a sort of *probabilities*. Along with the arity, each non-logical symbol of a PNL language has a description of the sorts of each of its arguments, and of its value, in case it is an individual variable, a constant or a function symbol. For example, the function symbols P_a from automata-related languages take an argument of the duration sort to make a term of the probability sort. A PNL language should contain countably many individual variables of both sorts. Together with the symbols 0, $+$, $=$, $\leq$ and ℓ of the sort of durations, PNL languages always contain the rigid constants 0 and 1, the rigid function symbol $+$, and the rigid relation symbols $\leq$ and $=$ of the new sort of probabilities. Using the same notation for both probability and duration 0, $+$ and $=$ does not cause confusion.

The BNF for formulas is as in NL. The BNF for terms in NL languages is extended to capture the terms that express probability in PNL languages as follows:

$t ::= x|c|f(t,\ldots,t)|p(\varphi,t,\ldots,t)$

Terms of the kind $f(t,\ldots,t)$ are well-formed only if the sorts of the subterms t match the requirements for f. A similar condition applies to atomic formulas. Terms of the kind $p(\varphi,t,\ldots,t)$ (p-terms) have the probability sort. They contain one formula-argument φ and as many term arguments, as are the free variables of φ. Let $x_1,\ldots,x_n$ be the free variables of φ, listed in the order of their first free occurrences in φ. Then $p(\varphi,t_1,\ldots,t_n)$ is well-formed, iff t_i has the sort of x_i, $i=1,\ldots,n$. Besides $p(\varphi,x_1,\ldots,x_n)$ is abbreviated to $p(\varphi)$. This looks the same as for closed φ, but is no source of confusion. We put

$$FV(p(\varphi,t_1,\ldots,t_n)) = \bigcup_{i=1}^{n} FV(t_i)$$

and

$$[t/x]p(\varphi,t_1,\ldots,t_n) \rightleftharpoons p(\varphi,[t/x]t_1,\ldots,[t/x]t_n).$$

The symbol p is not a non-logical symbol in PNL languages. Its role is rather like that of modal operators, yet it is used to construct terms, not formulas.

3.2 Frames, Models and Satisfaction

In order to enable a finite complete first order proof system for PNL, we introduce probability domains in PNL abstractly, like the other NL domains:

Definition 1. *A system of the kind* $\langle U, +^{(2)}, 0^{(0)}, 1^{(0)}\rangle$ *is* a probability domain*, if it satisfies the axioms:*

$(U1)$ $x + (y + z) = (x + y) + z$

$(U2)$ $x + 0 = x$

$(U3)$ $x + y = y + x$

$(U4)$ $x + y = x + z \Rightarrow y = z$

$(U5)$ $x + y = 0 \Rightarrow x = 0$

$(U6)$ $\exists z(x + z = y \vee y + z = x)$

$(U7)$ $0 \neq 1$

The classical probability domain is $\langle \mathbf{R}_+, +, 0, 1\rangle$. Another example is $\langle\{\frac{i}{n} : i < \omega\}, +, 0, 1\rangle$, where n is a fixed positive integer.

We assume that the linear ordering $\leq$ which is defined by the equivalence $x \leq y \leftrightarrow \exists z(x + z = y)$ is available for probability domains.

Definition 2. *A tuple of the kind* $\langle\langle T, \leq\rangle, \langle D, +, 0\rangle, \langle U, +, 0, 1\rangle, m\rangle$ *is a* PNL frame*, if* $\langle\langle T, \leq\rangle, \langle D, +, 0\rangle, m\rangle$ *is an (ordinary, one-sorted)* NL *frame, and* $\langle U, +, 0, 1\rangle$ *is a probability domain.*

Interpretations of symbols from PNL languages into PNL frames are defined like in (one-sorted) NL languages. Of course, the types of the functions and relations that symbols evaluate to should match the types of the symbols. Besides, the obligatory symbols 0, 1, +, and $\leq$ of the probability sort should be interpreted by the corresponding components of the frame's probability domain.

The setting given in the previous section makes it clear that the values of the probability measures $P_{I,\tau}$ are relevant to the interpretation of p-terms only for some, *formula-definable* subsets of the set of interpretations that is part of every PNL model. These subsets are difficult to describe prior to defining the relation $\models$ in corresponding model. On the other hand, requesting $P_{I,\tau}$ to be defined on the entire powersets of interpretations would render the forthcoming completeness theorem unreasonably difficult to prove.

That is why, before defining PNL models, we introduce an auxiliary notion of *partial* PNL models:

Definition 3. *Let* $\mathbf{L}$ *be a language for* PNL*. A triple* $\langle F, \mathbf{M}, P\rangle$ *is a* partial PNL model for $\mathbf{L}$ *if* F *is a* PNL *frame,* $\mathbf{M}$ *is a set of interpretations of the non-logical symbols of* $\mathbf{L}$ *into* F*, and* $P = \langle P_{I,\tau} : I \in \mathbf{M}, \tau \in T_F\rangle$ *is a system of* partial *functions* $P_{I,\tau} \in (2^{\mathbf{M}_{I,\tau}} \to U_F)$*, where* $\mathbf{M}_{I,\tau} = \{I' \in \mathbf{M} : I'$ *τ-agrees with* $I\}$*, which satisfy the equalities*

$$P_{I,\tau}(\emptyset) = 0,\ P_{I,\tau}(\mathbf{M}_{I,\tau}) = 1,\ P_{I,\tau}(\mathbf{N}_1) + P_{I,\tau}(\mathbf{N}_2) = P_{I,\tau}(\mathbf{N}_1 \cup \mathbf{N}_2) + P_{I,\tau}(\mathbf{N}_1 \cap \mathbf{N}_2).$$

for whichever $\mathbf{N}_1, \mathbf{N}_2 \subseteq \mathbf{M}_{I,\tau}$ $P_{I,\tau}$ *is defined.*

In the above definition $P_{I,\tau}$ are partial probability functions on the sets of interpretations $\mathbf{M}_{I,\tau}$. They take the abstract kind of probabilities we introduced as their values.

We proceed to define the satisfaction relation $\models$ on partial PNL models. In order to define $\models$ for formulas of the kind $\exists x\varphi$, we need a technical definition:

Definition 4. *Given an interpretation I of language $\mathbf{L}$ into frame F and a non-logical symbol s from $\mathbf{L}$, I_s^a stands for the interpretation of $\mathbf{L}$ into F that s-agrees with I and interprets s as a. Given a set of interpretations $\mathbf{N}$ of $\mathbf{L}$ into F, $\mathbf{N}_s^a$ is $\{I_s^a : I \in \mathbf{N}\}$. Given a partial function $f : 2^{\mathbf{A}} \to U_F$, the partial function $f_s^a : 2^{\mathbf{A}_s^a} \to U_F$ is defined by putting $f_s^a(\mathbf{N}_s^a) = f(\mathbf{N})$, if $f(\mathbf{N})$ is defined. If $f(\mathbf{N})$ is undefined, then $f_s^a(\mathbf{N})$ is undefined too. Given a partial PNL model $M = \langle F, \mathbf{M}, P\rangle$, M_s^a is $\langle F, \mathbf{M}_s^a, \langle (P_{I,\tau})_s^a : I \in \mathbf{M}, \tau \in T_F\rangle\rangle$.*

Obviously M_s^a is a partial PNL model, if M is one. We abbreviate $(\dots I_{s_1}^{a_1} \dots)_{s_n}^{a_n}$ to $I_{s_1,\dots,s_n}^{a_1,\dots,a_n}$. The same applies to models M.

Values $I_\sigma(t)$ of terms t and the modelling relation $\models$ are partially defined in PNL models by simultaneous induction on the length of terms and formulas. The clauses about the kinds of terms and formulas that are known from NL are as in NL: Given a PNL model $M = \langle F, \mathbf{M}, P\rangle$ and $I \in \mathbf{M}$, the clause for $M, I, \sigma \models \varphi$ is the same as that for $\langle F, I\rangle, \sigma \models \varphi$. Each clause applies only if the entities on its right side are defined. The only clause which is subjected to a somewhat greater change is the one about existential formulas:

$M, I, \sigma \models \exists x\varphi$ iff there exists an a such that $a \in D_F$, in case x is a duration variable, and $a \in U_f$, in case x is a probability variable, and $M_x^a, I_x^a, \sigma \models \varphi$

The new, PNL-specific clause is about p-terms. Given a well-formed p-term $p(\varphi, t_1, \dots, t_n)$, $I_\sigma(p(\varphi, t_1, \dots, t_n))$ is

$$P_{I,\max\sigma}(\{I' \in \mathbf{M}_{I,\max\sigma} : \exists\tau \geq \min\sigma\ M_{x_1,\ \dots\ ,x_n}^{I_\sigma(t_1),\dots,I_\sigma(t_n)}, (I')_{x_1,\ \dots\ ,x_n}^{I_\sigma(t_1),\dots,I_\sigma(t_n)}, [\min\sigma, \tau] \models \varphi\}),$$

only if $P_{I,\max\sigma}$ is defined for the given set. In case φ is closed, this definition simplifies to

$$I_\sigma(p(\varphi)) = P_{I,\max\sigma}(\{I' \in \mathbf{M}_{I,\max\sigma} : \exists\tau \geq \min\sigma\ M, I', [\min\sigma, \tau] \models \varphi\}).$$

In words, given an interpretation $I \in \mathbf{M}$ and an interval σ, $I_\sigma(p(\varphi))$ represents the probability of the set of those interpretations $I' \in \mathbf{M}$ which are like I up to the end of the interval σ and satisfy φ at some interval which has the same beginning as σ. For a modelled system's behaviour which is represented by I for the time until $\max\sigma$, this term can represent the probability for this behaviour to continue so that φ eventually gets satisfied in the specified kind of interval.

In the general case the operator p evaluates the above probability under the assumption that the free variables of φ evaluate to the values which $t_1, \dots, t_n$ have in the current interval σ.

Note that interpretations I' which $\max\sigma$-agree with the selected one I may happen to satisfy φ at intervals $[\min\sigma, \tau]$ where $\tau \leq \max\sigma$. In this case satisfaction of φ may happen to be a simple consequence of $\max\sigma$-agreeing with I, and no substantial probability evaluation is involved. For example

$(\varphi; \top) \Rightarrow p(\varphi) = 1$

is a valid PNL formula, if φ is retrospective (see definition 1), that is, if φ does not specify properties of interpretations beyond the end of the current interval.

Having defined (partial) $\models$ on partial PNL models, we are ready to define PNL total models:

Definition 5. *A partial PNL model M is a (total) PNL* model*, if values of terms and satisfaction of formulas from the corresponding language are everywhere defined in M.*

For the rest of the paper only total PNL models are considered.

4 A Complete Proof System for *PNL*

We need to specify a special class of PNL formulas, in order to introduce our proof system.

4.1 Retrospective Formulas and Interpretations Which τ-agree

Definition 1. *We call NL formulas that can be defined by the BNF*
$\varphi ::= \bot | R(t, \ldots, t) | \neg\varphi | (\varphi \wedge \varphi) | (\varphi; \varphi) | \Diamond_l \varphi | \exists x \varphi$
retrospective.

There is a close connection between retrospective formulas and interpretations that τ-agree:

Proposition 1. *Let F be a frame and $\tau \in T_F$. Let I and J be interpretations of $\mathbf{L}$ into F that τ-agree. Let $\sigma \in \mathbf{I}(T_F)$ and $\max\sigma \leq \tau$. Then $\langle F, I\rangle, \sigma \models \varphi$ iff $\langle F, J\rangle, \sigma \models \varphi$ for all retrospective φ from $\mathbf{L}$.*

Since occurrences of modal operators can be removed from rigid formulas due to $A1$, rigid formulas share the properties of retrospective formulas.

4.2 The System

The proof system for PNL that we propose is an extension of that for NL with the axioms $U1$-$U7$ about probabilities, and the following axioms and rules:
$(P_\bot)$ $p(\bot) = 0$ (P_+) $p(\varphi) + p(\psi) = p(\varphi \vee \psi) + p((\varphi; \top) \wedge (\psi; \top))$
$(P_\top)$ $p(\top) = 1$ $(P_=)$ $x_1 = y_1 \wedge \ldots \wedge x_n = y_n \Rightarrow p(\varphi, x_1, \ldots, x_n) = p(\varphi, y_1, \ldots, y_n)$
$(P_\Diamond)$ $\dfrac{\varphi \Rightarrow (\Diamond_l^c \psi \Rightarrow \Diamond_l^c \chi)}{\varphi \Rightarrow p(\psi) \leq p(\chi)}$ if φ is retrospective, $(P_;)$ $\dfrac{\varphi \Rightarrow \neg(\varphi; \ell \neq 0)}{(\varphi; p(\psi) = x) \Rightarrow p((\varphi; \psi)) = x}$

Note that in the above axioms and rules terms like $p(\varphi)$ should be understood as abbreviations of the kind $p(\varphi, x_1, \ldots, x_n)$, as stated in Subsection 3.1. This means that these axioms and rules have instanced with formulas that have free variables. $P_\Diamond$ and $P_;$ can be applied only to theorems of PNL. Substitution in p-terms is allowed in proofs only if the substitute term is rigid.

The soundness of the above system is established in the ordinary way. Given a PNL language $\mathbf{L}$, we denote the set of all PNL theorems in $\mathbf{L}$ by $PNL_\mathbf{L}$.

Consistency and maximal consistency are defined for sets of formulas in a PNL language with respect to the above proof system in the ordinary way. We have the following completeness result about our proof system:

Theorem 1. *Let Γ be a set of formulas from a PNL language $\mathbf{L}$. Then Γ is consistent iff there exists a model $M = \langle F, \mathbf{M}, P\rangle$ for $\mathbf{L}$, an interpretation $I \in \mathbf{M}$ and an interval $\sigma \in \mathbf{I}(T_F)$ such that $M, I, \sigma \models \Gamma$.*

A proof of this theorem can be found in [7, 9].

5 Chapman-Kolmogorov's Equality for Composition in PNL

The means to express sequential composition of (probabilistic) processes in PNL is the defined operator $(.;.)$. In this section we extend the semantics of PNL and its proof system so that probabilities of formulas with $(.;.)$ satisfy Chapman-Kolmogorov's equality about sequential composition under reasonable assumptions.

Since this equality involves integration, probability domains are extended with multiplication, which is needed to define integration. Multiplication of probabilities is required to satisfy the axioms:

$(U8)$ $x.1 = x$	$(U11)$ $x.(y+z) = x.y + x.z$
$(U9)$ $x.(y.z) = (x.y).z$	$(U12)$ $x.y = x.z \wedge x \neq 0 \Rightarrow y = z$
$(U10)$ $x.y = y.x$	$(U13)$ $x \neq 0 \Rightarrow \exists y(x.y = z)$

We extend the proof system of PNL by the rules:

$$(\overline{P})\ \frac{\varphi \Rightarrow \neg(\varphi; \ell \neq 0)}{\ell = 0 \wedge p(\varphi \wedge \theta \Rightarrow p((\varphi;\psi)) \leq x) = 1 \Rightarrow p((\varphi \wedge \theta;\psi)) \leq x.p(\varphi \wedge \theta)}$$

$$(\underline{P})\ \frac{\varphi \Rightarrow \neg(\varphi; \ell \neq 0)}{\ell = 0 \wedge p(\varphi \wedge \theta \Rightarrow p((\varphi;\psi)) \geq x) = 1 \Rightarrow p((\varphi \wedge \theta;\psi)) \geq x.p(\varphi \wedge \theta)}$$

For a formula φ to specify a step in some process, it is natural to expect that $\vdash_{PNL} \varphi \Rightarrow \neg(\varphi; \ell \neq 0)$. That is why the latter formula is used as a premiss for $\overline{P}$ and $\underline{P}$. Let φ be a formula from some PNL language $\mathbf{L}$. Let $M = \langle F, \mathbf{M}, P\rangle$ be a model for $\mathbf{L}$. Let $\sigma_0 \in \mathbf{I}(T_F)$ be a 0-length interval and $I_0 \in \mathbf{M}$. Let $f = \lambda I.P_{I,\tau'}(\mathbf{M}_{I',[\tau,\tau'],(\varphi;\psi)})$. Then the equality of Chapman-Kolmogorov can be expressed as:

$$P_{I_0,\tau}(\mathbf{M}_{I_0,\sigma,(\varphi;\psi)}) = \int\limits_{I \in \mathbf{M}_{I_0,\sigma,\varphi}} f(I) dP_{I_0,\tau}.$$

The integral which occurs above is defined as the least upper bound of the sums of the kind $\sum\limits_{i=1}^{n} \inf\limits_{I \in A_i} f(I)P_{I_0,\tau}(A_i)$, in case it is equal to the greatest lower bound of the sums of the kind $\sum\limits_{i=1}^{n} \sup\limits_{I \in A_i} f(I)P_{I_0,\tau}(A_i)$, where $\{A_1, \ldots, A_n\}$ ranges over the finite partititions of $\mathbf{M}_{I_0,\sigma,\varphi}$ for which $P_{I_0,\tau}(A_i)$, $i = 1, \ldots, n$ is defined.

In order to enable this definition, we need to require that the linear ordering of U_F is complete. Unfortunately, this cannot be enforced by first-order means. In the general case we can show that the above rules entail the following approximation of Chapman-Kolmogorov's equality for models $M = \langle F, \mathbf{M}, P\rangle$ which validate them:

Let $I_0 \in \mathbf{M}$, and $\sigma \in \mathbf{I}(T_F)$ be a 0-length interval. Let φ and ψ be formulas, and φ satisfy the premiss of our rules. Let $n < \omega$. Then there exists a partitition $\{A_i : i \leq n\}$ of $\mathbf{M}_{I_0,\sigma,\varphi}$ such that $(i-1) < n.f(I) \leq i$ for $I \in A_i$, $i = 0, \ldots, n$, and moreover $\sum\limits_{i=1}^{n}(i-1).P_{I_0,\tau}(A_i) \leq n.P_{I_0,\tau}(\mathbf{M}_{I_0,\sigma,(\varphi;\psi)}) \leq \sum\limits_{i=0}^{n} i.P_{I_0,\tau}(A_i)$.

Clearly, in the case $U = \mathbf{R}_+$, this is equivalent to the precise equality.

Conclusions

We believe that, by introducing PNL and finding a complete proof system for it, we have made the task of obtaining a similar system for PDC a lot simpler. In fact, PNL has the expressive power of PDC, except for state expressions and their durations. However they can be introduced to PNL using the constructions presented in, e.g. [5]. This makes it reasonable to believe that PNL is an appropriate tool for the specification and verification of probabilistic behaviour of real-time systems.

Acknowledgements

Thanks are due to Dang Van Hung for his remarks and suggestions on draft versions of this paper.

References

[1] Barua, R. S. Roy and Zhou Chaochen. Completeness of Neighbourhood Logic. *Proceedings of STACS'99*, Trier, Germany, LNCD 1563, Springer Verlag, 1999.

[2] Chang, C. C. and H. J. Keisler. *Model Theory.* North Holland, Amsterdam, 1973.

[3] Dutertre, B. *On First Order Interval Temporal Logic* Report no. CSD-TR-94-3 Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, 1995.

[4] Dang Van Hung and Zhou Chaochen. Probabilistic Duration Calculus for Continuous Time. *Formal Aspects of Computing*, 11, pp. 21-44, 1999.

[5] Guelev, D. P. *A Calculus of Durations on Abstract Domains: Completeness and Extensions.* Technical Report 139, UNU/IIST, P.O.Box 3058, Macau, May 1998.

[6] Guelev, D. P. *Probabilistic Interval Temporal Logic*, Technical Report 144, UNU/IIST, P.O.Box 3058, Macau, August 1998, Draft.

[7] Guelev, D. P. *Probabilistic and Temporal Modal Logics*, Ph.D. thesis, submitted.

[8] Guelev, D. P. *Interval-related Interpolation in Interval Temporal Logics*, submitted, 1999.

[9] Guelev, D. P. *Probabilistic Neighbourhood Logic.* Technical Report 196, UNU/IIST, P.O.Box 3058, Macau, April 2000.

[10] Liu Zhiming, A. P. Ravn, E. V. Sørensen and Zhou Chaochen. A Probabilistic Duration Calculus. In: *Proceedings of the Second International Workshop on Responsive Computer Systems.* KDD Research and Development Laboratories, Saitama, Japan, 1992.

[11] Mathai, J. *Real-Time Systems.* Prentice Hall, 1995.

[12] Moszkowski, B. Temporal Logic For Multilevel Reasoning About Hardware. *IEEE Computer*, 18(2):10-19, February 1985.

[13] Trifonov, V. T. *A completeness theorem for the probabilistic interval temporal logic with respect to its standard semantics*, M.Sc. thesis, Sofia University, July 1999. (In Bulgarian)

[14] Zhou Chaochen, and M. R. Hansen. An Adequate First Order Interval Logic. *International Symposium, Compositionality - The Significant Difference*, H. Langmaack, A. Pnueli and W.-P. de Roever (eds.), Springer, 1998.

[15] Zhou Chaochen, C. A. R. Hoare and A. P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40(5), pp. 269-276, 1991.

Remark: UNU/IIST technical reports are available from URL
`http://www.iist.unu.edu/newrh/III/1/page.html`