Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 190 (2007) 43–58
www.elsevier.com/locate/entcs

Co-Algebraic Models for Quantitative Spatial Logics [1]

Vincenzo Ciancia [2] Gian Luigi Ferrari [3]

Dipartimento di Informatica
Università di Pisa
Italy

Abstract

We introduce a class of coalgebraic models and a family of modal logics that support the specification of spatial properties of distributed applications. The evaluation of a formula yields a value in a suitable multi-valued algebraic structure, giving a *measure* of the satisfaction of a requirement, induced by the decomposition of a system into subsystems, meant as available *resources*. As semantic domain we consider certain algebraic structures, called c-semirings, that allow us to generalize boolean logics to the multi-valued case, while keeping a number of the axioms of boolean algebras. Under suitable conditions on the structure of c-semirings, we show that, even if our logical formalisms are equipped with spatial operators, the interpretation of formulas fully characterizes bisimilarity.

Keywords: Modal Logics, Spatial Logics, Coalgebras, C-Semirings, Quality of Service

1 Introduction

The use of spatial logics for process calculi [6,8,9] is receiving more and more attention in the last years, in order to reason about non-behavioral aspects of computation, such as the presence of hidden resources, or the decomposition of a system into parallel components. The main difference between spatial logics and more "traditional" *observational* temporal logics (see e.g. [21]), is that the latter only allow observations about the *behavior* of a system, which is constituted by the visible effect of actions that the system performs, and synchronizations with the outside world. For this reason, temporal logics are considered *extensional*: we do not need to know how a system is made, in order to tell what properties it satisfies, but just to describe its interactions with the operational environment. Spatial logics, instead,

[1] Research partially supported by the EU, within the FETPI Global Computing, Project IST-2005-16004 Sensoria (Software Engineering for Service-Oriented Overlay Computers).

[2] Email: ciancia@di.unipi.it

[3] Email: giangi@di.unipi.it

doi:10.1016/j.entcs.2007.07.004

are usually considered *intensional*, because the meaning of formulas is not reflected in the observational semantics of programs. For example, a typical spatial operator is parallel decomposition, which is usually defined on the syntax of programs, rather than on the semantic model, because parallel components are usually forgotten in the semantics.

Being extensional is important for a logic, because relations like soundness and completeness can be established between semantic equivalence and logical equivalence. On the other hand, spatial operators are of interest not only for the very specific purpose of dealing with distributed computations, but in a more general sense to represent availability of arbitrary *resources* in computation: the fact that a system in a given state is decomposable in two parts, for example, can be interpreted as the fact that it "contains" two distinct resources. In [7], the work on spatial logics has inspired the development of a type system where it is possible to express resource access policies and ownership of resources by exploiting forms of spatial operators.

Recently, some approaches have addressed the definition on non-syntactical models of spatial logics. In [16] and [13], for example, spatial logics have been interpreted over particular kinds of graph rewriting systems, rather than on the term language of a specific process calculus. In [23], a non-syntactical interpretation of parallel decomposition is given, by adding spatial information to the semantics of the language: a coalgebraic model, called *spatial transition systems* (STS), has been developed and tested as a semantic model for a CCS-like process calculus. It has been shown that the STS semantics is fully concurrent (for example, the processes $a.b+b.a$ and $a|b$ have different meanings). In STS, the semantics of process calculi is given by reaction rules, making behavioral observations about isolated subsystems impossible.

Formal approaches to the usage and availability of resources are "hot topics" in computer science (see [18,15,2] and the references therein). To reason about resource usage, temporal logics have been modeled on more general domains than the booleans, for example in the multi-valued temporal logic of [11], or in the quantitative μ-calculus of [12]. These logics are usually equipped with some kind of metrics in order to be able to compare different systems with respects to *how much* they satisfy certain requirements. A logic for reasoning on quality of service measurement has been introduced by Lluch Lafuente and Montanari in [20]. The evaluation of a logical formula is a value of a suitable algebraic structure, namely a *constraint semiring* (c-semiring), representing the measure of the service level of the formula and not just a boolean value expressing whether or not the formula holds. C-semirings consist of a domain and two operations, called *additive* and *multiplicative,* satisfying some properties. The basic idea is that the former is used to select among values and the latter to combine values. C-semirings were originally proposed to describe and program constraint solving problems [4].

The aim of this work is to define quantitative spatial logics interpreted on multi-valued algebraic structures, whose semantics is extensional, i.e. defined on semantics rather than on syntax of process calculi. It is important to emphasize that the

semantic models that we consider do not contain quantitative information, but rather this information is inferred from the part of the model that records resource availability (e.g. spatial decomposition) at a given state. This formally defines quality of service as resource availability, in a natural way. We choose c-semirings as the domain of satisfaction values, in order to extend the approach introduced in [20], which generalizes various kinds of quantitative model-checking. Finally, in order to get a standard notion of semantic equivalence, namely *bisimilarity*, we use coalgebras to define our semantic models.

In section 3.1, we introduce *spatial labeled transition systems* (SLTS), a class of coalgebras, reminiscent of *spatial transition systems* of [23], that extends labeled transition systems with spatial decomposition. Then, in section 3.2, we introduce a logic that is interpreted over SLTS, the *quantitative observational spatial μ-calculus* (QOS-μ). The logic we obtain is sort of an "hybrid" between propositional modal logics interpreted over Kripke structures, and Hennessy-Milner logics, because we have both spatial formulas without labels, inducing valuations about states, and temporal formulas with action-labeled modalities, allowing extensional reasoning about observable actions of the system. Using the general framework of coalgebras for the definition of SLTS, we define cost-optimizing morphisms between systems, thus formally characterizing the notion of "costs less than".

In section 4, we discuss the expressiveness of QOS-μ logics, when varying the c-semiring used for the interpretation of formulas. Our main result is that for a certain class of c-semirings, also including infinite domains, not only QOS-μ is decidable for finite SLTS, but also it characterizes bisimilarity, i.e., semantic equivalence. Moreover, due to the presence of fixed point operators, and the fact that c-semirings can be infinite, a single formula can be strictly more expressive than any formula interpreted over the booleans (see remark 4.7).

We foresee different classes of applications, both in the area of optimization of concurrent systems and in *service-oriented computing* [25]. Spatial models can be used as fully concurrent models for process calculi. QOS-μ in this case allows to evaluate costs of *execution plans*, distinguishing between parallel and sequential execution of certain operations. In section 5 we give an SLTS semantics to pure CCS, and we show some examples of QOS-μ formulas. In service-oriented computing, on the other hand, QOS-μ, being a behavioral quantitative logic, can be exploited to define patterns for service discovery with an emphasis on quality of service and resource availability.

2 Background

2.1 Coalgebras, Coalgebra Homomorphisms and Spatial Transition Systems

Coalgebras are mathematical models used to represent the behavior of systems, abstracting from the particular structure used, and to define in a very general way properties such as bisimulation or minimization of systems.

In this section we briefly introduce the main definitions of coalgebras in the category *Set*, assuming some basic knowledge of category theory. A detailed exposition

of the theory of coalgebras and coinduction can be found in [19,26], which we use as a basis for our introduction. Hereafter, we will use the following notations: for the categorial product, π_1 and π_2 indicate the first and second projection, and $\langle f, g\rangle$ is the pairing of f and g; to represent the coproduct (disjoint union) of two sets A and B we will use the set $(\{0\} \times A) \cup (\{1\} \times B)$; the *finite power set* functor, sending each set in the set of its finite subsets, will be denoted with $\mathcal{P}_{fin}$; with $f; g$, as usual in category theory, we denote reverse composition of arrows, which in the category Set corresponds to function composition: $f; g = g \circ f$; finally, we denote with 1 a set with one element, which we write as $*$.

Definition 2.1 (coalgebra) Given a functor $F : Set \to Set$, a coalgebra for F, or F-coalgebra, is a pair $\langle S, f\rangle$ where S is a set, called the *carrier* of the coalgebra, and $f : S \to F(S)$ is the *operation* of the coalgebra.

The carrier can be often thought of as the set of states in a system, while the operation of the coalgebra gives observations about the "next" states, depending on the chosen functor. Consider for example the functor $F(S) = \mathcal{P}_{fin}(S)$. A coalgebra for F is made up of a set S and a function from S to the set of its finite subsets, i.e., a transition system with states in S. Similarly, a coalgebra for the functor $F(S) = \mathcal{P}_{fin}(L \times S)$ for a set of labels L is a labeled transition system with states in S and labels in L.

Definition 2.2 (coalgebra homomorphism) A coalgebra homomorphism from an F-coalgebra $\langle S, f\rangle$ to an F-coalgebra $\langle S', f'\rangle$ is a function $m : S \to S'$ such that $m; f' = f; F(m)$.

Intuitively, a coalgebra homomorphism is a function that preserves and reflects transitions.

Now, we introduce bisimulation and bisimilarity, usually chosen as the semantic equivalence notion when the semantics of a programming language is given using coalgebras.

Definition 2.3 (coalgebraic bisimulation and bisimilarity) Given two F-coalgebras $C = \langle S, f\rangle$ and $C' = \langle S', f'\rangle$ a bisimulation between C and C' is a relation $R \subseteq S \times S'$ over which there exists an F-coalgebra B, such that π_1 and π_2 are coalgebra homomorphisms from B to C and C', respectively. The greatest bisimulation is called *bisimilarity.* We say that s is *bisimilar* to s', written $s \sim s'$, if the pair$\langle s, s'\rangle$ is in the bisimilarity relation.

Definition 2.4 (final coalgebra) An F-coalgebra $C = \langle S, f\rangle$ is final if for any other F-coalgebra $C' = \langle S', f'\rangle$ there exists exactly one coalgebra homomorphism from C' to C.

The final coalgebra, if it exists, is unique up to isomorphism. An important class of functors that have a final coalgebra is given in the following theorem (see e.g. [26], also for the definition of polynomial functor):

Theorem 2.5 *For all functors that can be built from polynomial functors and the finite power set functor, a final coalgebra exists.*

In [23], coalgebras are used to define *spatial transition systems* (STS), that we extend in this work obtaining SLTS. A STS is made up of a set of states, and two functions that describe respectively the temporal evolution of the system, and the possible parallel decompositions, in a given state.

Definition 2.6 (spatial transition system) Given the functor

$$F(A) = \mathcal{P}_{fin}(A) \times (1 + \mathcal{P}_{fin}(A \times A))$$

a spatial transition system is a coalgebra for F over a set S, i.e., a pair

$$\langle S, f : S \to F(S) \rangle$$

The set S represents the set of states and subsystems of a given system. Notice that $f = \langle f_{tr}, f_{sp} \rangle$, where $f_{tr} : S \to \mathcal{P}_{fin}(S) = f; \pi_1$ represents the behavior of the system and $f_{sp} : S \to 1 + P_{fin}(S \times S) = f; \pi_2$ allows to observe its subsystems. These are either the set of all alternative choices for its decomposition into two parts, or 1 (i.e. a set of only one element, $\{*\}$), meaning that the system is made up of a single, undecomposable component. These two functions, named f_{tr} and f_{sp} respectively because they represent the *transitions* of the system and its *spatial* decomposition, give rise to orthogonal observations about the spatial and temporal aspects of computation.

2.2 C-Semirings and Quantitative Model-Checking

In boolean logics, the satisfaction relation is a binary predicate $- \models -$ over states and formulas, i.e., a function of signature $State \times Formula \to \{true, false\}$. There is no reason why in principle we could not use a domain different than the booleans; in [20] such a line of development is given, by defining modal logics over a particular kind of domains called *c-semirings*.

C-semirings [3] have been introduced as a formal building block to define generalized constraint solving problems [4], and have been exploited in the field of optimization, quality of service analysis [17,24], and also for defining metrics in complex domains such as those used for speech recognition [22]. We refer to [4] for the full definitions and a thorough explanation.

Definition 2.7 A c-semiring is a tuple $\langle A, \Cup, \Cap, \mathbf{0}, \mathbf{1} \rangle$ where

- A is a set, with $\mathbf{0}$ and $\mathbf{1}$ elements of A.
- $\Cup : 2^A \to A$ satisfies $\Cup\{a\} = a$, $\Cup\emptyset = \mathbf{0}$, $\Cup A = \mathbf{1}$, $\Cup(\bigcup_i A_i) = \Cup\{\Cup A_i\}$ for $A_i \subseteq A, i \geq 0$
- $\Cap : A \times A \to A$ is commutative, associative, distributive over $\Cup$; $\mathbf{1}$ is its unit element, and $\mathbf{0}$ is its absorbing element.

Usually, $\Cup$ is called the *additive* operation of the semiring, while $\Cap$ is referred to as the *multiplicative* operation. The additive operation models the *selection* of the "best" alternative in a set of choices, while the multiplicative one *combines* several

choices (and their costs/values). A c-semiring is *distributive* if the multiplicative operation is idempotent; in this case the additive operation distributes over the multiplicative one. The $\Cup$ operation induces a relation $\leq \subseteq A \times A$, defined as $a \leq b \Leftrightarrow a \Cup b = b$. Being $\Cup$ idempotent, this relation is a partial order with minimum $\mathbf{0}$ and maximum $\mathbf{1}$; moreover, $\Cup$ and $\Cap$ are monotone over $\leq$.

The first, and simpler, c-semiring we consider is the domain of booleans: $Bool = \langle \{false, true\}, \vee, \wedge, false, true \rangle$. C-semirings also arise in the analysis of access rights for a given resource [24]. In this case, an appropriate c-semiring domain could be $\langle \mathcal{P}(S), \cup, \cap, \emptyset, S \rangle$, where S is the set of all objects in the model.

Other examples include:

- $\langle \mathbb{R}^+, min, +, +\infty, 0 \rangle$, modeling transmission costs, where the best choice is the minimum cost, and the combination of two operations yields the sum of the costs;
- $\langle \mathbb{R}^+, max, min, 0, +\infty \rangle$, modeling bandwidth, in a situation where data size is constant, and if we do several transfers in parallel we have to wait for each one to complete;
- $\langle [0,1], max, \cdot, 0, 1 \rangle$, modeling probability of events.

C-semirings are closed under Cartesian product, functional domain and power domain constructions, allowing the theory to be applied to problems having more than one quantitative dimension to model, e.g., both bandwidth usage and probability of certain events.

3 Spatial Labeled Transition Systems and QOS-μ

3.1 Spatial Labeled Transition Systems

We now introduce the notion of *spatial labeled transition systems* (SLTS).

Definition 3.1 (spatial labeled transition system) A spatial labeled transition system over a set of labels L is a coalgebra for the functor

$$F_L(S) = \mathcal{P}_{fin}(L \times S) \times (1 + \mathcal{P}_{fin}(S \times S))$$

Given a coalgebra $f : S \to F_L(S)$, we represent with $f_{tr} = f; \pi_1$ and $f_{sp} = f; \pi_2$ the temporal and spatial transition structures of the system. Notice that f_{tr} gives, for each state, a set of pairs, consisting of a label and a state. Hereafter, we abbreviate $\langle a, s_1 \rangle \in f_{tr}(s)$ with $s \xrightarrow{a} s_1$.

The intuition behind SLTS is that both the behavior (represented by labeled transitions) and the decomposition of a system can be observed. For instance, if we consider a calculus with a parallel composition operator, we might have $f_{sp}(P_1 \parallel P_2 \parallel P_3) = \langle 1, \{\langle P_1,\ P_2 \parallel P_3 \rangle, \langle P_2,\ P_1 \parallel P_3 \rangle, \langle P_3,\ P_1 \parallel P_2 \rangle\} \rangle$. Decomposition could be defined as well in a non-syntactical way, using semantic information (e.g. a process being *active* or *inactive*) similarly to what happens in the location semantics of CCS [5].

We emphasize that SLTS are *semantic* models, and that they have been intro-

duced to give a resource-conscious semantics to different formalisms, even if there is no quantitative data in the model itself: the subsystems of a system are the available resources, and their availability determines quantities such as costs, or quality of service, directly from the analysis of the semantics of programs.

3.2 The Quantitative Spatial Logic QOS-μ

In this section, we introduce the *quantitative observational spatial μ-calculus* (QOS-μ), a spatial logic with action-labeled temporal modalities which is able to express behavioral properties of systems, and we establish a connection between coalgebra homomorphisms and cost optimization. The syntax of the logic extends with action-labeled modalities and spatial operators the logic in [20], which in turn is a generalization to c-semirings of the fragment of the propositional μ-calculus without negation. We fix here a set $\mathcal{L}$ of labels, which we use throughout the paper.

Definition 3.2 (QOS-μ syntax) Given a set of constants A and a set of variables Var, the set of *QOS-μ* formulas over A is defined by the following syntax[4]:

$$\Phi^A ::= k \mid \Phi^A \vee \Phi^A \mid \Phi^A \wedge \Phi^A \mid \langle\alpha\rangle\,\Phi^A \mid [\alpha]\,\Phi^A \mid$$
$$\circlearrowleft \mid \Phi^A \parallel \Phi^A \mid \Phi^A \asymp \Phi^A \mid V \mid \mu V.\Phi^A \mid \nu V.\Phi^A$$

where $\alpha \in \mathcal{L}$, $k \in A$, $V \in Var$, variables are bound by μ and ν and are subject to the usual rules of α-conversion.

Now we give an intuition about the semantics of connectives. The easy ones are k, the constant formula, and minimum and maximum fixed points, allowing to measure respectively unbound and infinite behaviors. Disjunction chooses the best valuation out of two formulas, while conjunction combines the valuations. Diamond modality $\langle\alpha\rangle\,\phi$ selects the best valuation of ϕ in the set of states reachable by a transition labeled α, ensuring that there is *at least one* state reachable by an α transition whose valuation is at most the obtained one, and box modality $[\alpha]\,\phi$ combines all those valuations, ensuring that *all* reachable states have at least the obtained valuation. The operator $\circlearrowleft$ stands for a *local* system, i.e. a system that cannot be decomposed. Formula $\phi_1 \parallel \phi_2$ decomposes a system into its parallel components, combining the valuations of ϕ_1 and ϕ_2 on the components and selecting the best result, thus ensuring that there is at least a decomposition giving a certain valuation of ϕ_1 and ϕ_2, while $\phi_1 \asymp \phi_2$ is its dual.

The semantics ${}^T[\![\phi]\!]^C_\Gamma : A^S$ of a formula $\phi \in \Phi^A$, given using an SLTS $T \in F_{\mathcal{L}}(S)$ for some set of states S, the c-semiring $C = \langle A, \Cup, \Cap, \mathbf{0}, \mathbf{1}\rangle$, and the environment $\Gamma : Var \to A^S$, is a *valuation* in A^S, giving the value of ϕ, for each state in S, as an element of A. In the following, T and C will be freely omitted when clear from the context, and Γ will be omitted when it is $\emptyset$.

Definition 3.3 (QOS-μ semantics)

[4] Hereafter, we overload the notation Φ^A to denote both the syntax, and the set of all formulas over A.

$$
\begin{array}{llll}
[\![k]\!]_\Gamma(s) & = k & [\![V]\!]_\Gamma & = \Gamma(V) \\
[\![\circlearrowleft]\!]_\Gamma(s) & = local(s) & [\![\mu V.\phi]\!]_\Gamma & = fix\ \lambda X.[\![\phi]\!]_{\Gamma[X/V]} \\
[\![\phi_1 \parallel \phi_2]\!]_\Gamma(s) & = par(s,\phi_1,\phi_2,\Gamma) & [\![\nu V.\phi]\!]_\Gamma & = Fix\ \lambda X.[\![\phi]\!]_{\Gamma[X/V]} \\
[\![\phi_1 \asymp \phi_2]\!]_\Gamma(s) & = Par(s,\phi_1,\phi_2,\Gamma) & [\![\langle\alpha\rangle\,\phi]\!]_\Gamma(s) & = \bigcup\limits_{\{\langle x,s_i\rangle \in f_{tr}(s) | x=\alpha\}} [\![\phi]\!]_\Gamma(s_i) \\
[\![\phi_1 \vee \phi_2]\!]_\Gamma(s) & = [\![\phi_1]\!]_\Gamma(s) \Cup [\![\phi_2]\!]_\Gamma(s) & [\![[\alpha]\,\phi]\!]_\Gamma(s) & = \bigcap\limits_{\{\langle x,s_i\rangle \in f_{tr}(s) | x=\alpha\}} [\![\phi]\!]_\Gamma(s_i) \\
[\![\phi_1 \wedge \phi_2]\!]_\Gamma(s) & = [\![\phi_1]\!]_\Gamma(s) \Cap [\![\phi_2]\!]_\Gamma(s) &
\end{array}
$$

where fix and Fix are the least and greatest fixed point operators, and

$$
local(s) = \begin{cases} \mathbf{1} & \text{if } f_{sp}(s) = \langle 0, * \rangle \\ \mathbf{0} & \text{otherwise} \end{cases}
$$

$$
par(s,\phi_1,\phi_2,\Gamma) = \begin{cases} \bigcup\limits_{\langle s',s''\rangle \in P} ([\![\phi_1]\!]_\Gamma(s') \Cap [\![\phi_2]\!]_\Gamma(s'')) & \text{if } f_{sp}(s) = \langle 1, P\rangle \\ \mathbf{0} & \text{otherwise} \end{cases}
$$

$$
Par(s,\phi_1,\phi_2,\Gamma) = \begin{cases} \bigcap\limits_{\langle s',s''\rangle \in P} ([\![\phi_1]\!]_\Gamma(s') \Cup [\![\phi_2]\!]_\Gamma(s'')) & \text{if } f_{sp}(s) = \langle 1, P\rangle \\ \mathbf{1} & \text{otherwise} \end{cases}
$$

In the definitions for $[\alpha]\,\phi$ and $\phi_1 \asymp \phi_2$, there is a slight abuse of notation because the multiplicative operation of the c-semiring, which is binary, is applied to a finite set. However this operation is commutative and associative, thus it can be easily extended to finite sets.

We can now state our first result: if there is a coalgebra homomorphism from a system a to another system b, then b "costs less" than a: the valuation of any formula over b is less or equal than the valuation of the same formula over a. Notice that this has particular relevance for the final coalgebra, which is the *minimal* system in terms of QOS-μ valuations.

Theorem 3.4 *Given two SLTS $\langle S, f\rangle$ and $\langle S', f'\rangle$, if there is a coalgebra homomorphism $m : S' \to S$ such that $m(s') = s$ for $s \in S$ and $s' \in S'$, then for every QOS-μ formula ϕ over A, and appropriate c-semiring structure for the valuation of ϕ, we have $[\![\phi]\!](s) \leq [\![\phi]\!](s')$.*

Proof (sketch) We proceed by induction on the structure of the formula, and by coinduction on the two coalgebras, assuming a homomorphism m such that $m(s') = s$. We only show here the case for $\langle\alpha\rangle\,\phi$. By definition of homomorphism, each s_i such that $\langle\alpha, s_i\rangle \in f_{tr}(s)$ is equal to $m(s'_j)$ for some $\left\langle \alpha, s'_j \right\rangle \in f_{tr}(s')$, so

we can choose one such s'_j for each s_i, call it $\bar{m}^{-1}(s_i)$. We have $f_{tr}(s') = K \cup K'$, where $K \subseteq f_{tr}(s')$, $K' = \left\{\left\langle \alpha, \bar{m}^{-1}(s_i)\right\rangle \mid s_i \in f_{tr}(s)\right\}$, $K \cap K' = \emptyset$. We now prove that $[\![\langle\alpha\rangle\,\phi]\!](s) \uplus [\![\langle\alpha\rangle\,\phi]\!](s') = [\![\langle\alpha\rangle\,\phi]\!](s')$:

$$
\begin{aligned}
& [\![\langle\alpha\rangle\,\phi]\!](s) \uplus [\![\langle\alpha\rangle\,\phi]\!](s') \\
= {} & \left(\uplus_{\{\langle x,s_i\rangle \in f_{tr}(s) \mid x=\alpha\}} [\![\phi]\!]\,(s_i)\right) \uplus \left(\uplus_{\left\{\left\langle x,s'_j\right\rangle \in f_{tr}(s') \mid x=\alpha\right\}} [\![\phi]\!]\left(s'_j\right)\right) \\
= {} & \left(\uplus_{\{\langle x,s_i\rangle \in f_{tr}(s) \mid x=\alpha\}} [\![\phi]\!]\,(s_i)\right) \uplus \left(\uplus_{\{\langle x,s_i\rangle \in f_{tr}(s) \mid x=\alpha\}} [\![\phi]\!]\left(\bar{m}^{-1}(s_i)\right)\right) \uplus \\
& \uplus \left(\uplus_{\left\{\left\langle x,s'_j\right\rangle \in K \mid x=\alpha\right\}} [\![\phi]\!]\left(s'_j\right)\right) \\
= {} & \{\text{properties of } \uplus\} \\
& \left(\uplus_{\{\langle x,s_i\rangle \in f_{tr}(s) \mid x=\alpha\}} [\![\phi]\!](s_i) \uplus [\![\phi]\!](\bar{m}^{-1}(s_i))\right) \uplus \left(\uplus_{\left\{\left\langle x,s'_j\right\rangle \in K \mid x=\alpha\right\}} [\![\phi]\!]\left(s'_j\right)\right) \\
= {} & \left\{\text{by coinductive hypothesis, } [\![\phi]\!](s_i) \uplus [\![\phi]\!](\bar{m}^{-1}(s_i)) = [\![\phi]\!](\bar{m}^{-1}(s_i))\right\} \\
& \left(\uplus_{\{\langle x,s_i\rangle \in f_{tr}(s) \mid x=\alpha\}} [\![\phi]\!](\bar{m}^{-1}(s_i))\right) \uplus \left(\uplus_{\left\{\left\langle x,s'_j\right\rangle \in K \mid x=\alpha\right\}} [\![\phi]\!]\left(s'_j\right)\right) \\
= {} & \uplus_{\left\{\left\langle x,s'_j\right\rangle \in f_{tr}(s') \mid x=\alpha\right\}} [\![\phi]\!](s'_j) \\
= {} & [\![\langle\alpha\rangle\,\phi]\!](s')
\end{aligned}
$$

□

Remark 3.5 QOS-μ does not include a negation operator, since c-semirings usually do not have complementation. However, certain classes of c-semirings admit a complementation operator, and in that case we can introduce negation in the logic (see also [20] on the subject), thus obtaining pairs of operators from each other by De Morgan duality (for example, obtaining $\phi_1 \asymp \phi_2$ as $\neg(\neg\phi_1 \parallel \neg\phi_2)$. It should be noticed that there are interesting cases where complementation can't be defined, like $\langle \mathbb{R}^+, max, min, 0, +\infty\rangle$, hence in this work we only deal with the more general case.

4 Expressiveness of QOS-μ

In this section we show that QOS-μ characterizes bisimilarity (the so-called *adequacy* property) when interpreted over a distributive c-semiring, and that logic equivalence is strictly finer than bisimulation for non-distributive c-semirings. Moreover, we state that non-boolean (e.g. infinite) c-semirings give to the logic a strictly higher expressive power than the booleans. The first result is composed by three parts:

(i) we show that bisimilarity implies logical equivalence

(ii) we prove adequacy in the boolean case

(iii) we show that logical equivalence over any distributive c-semiring implies logical equivalence over the booleans, thus concluding the proof.

Theorem 4.1 *Given a distributive c-semiring $C = \langle A, \Cup, \Cap, \mathbf{0}, \mathbf{1}\rangle$, s and s' states of an SLTS, we have*

$$s \sim s' \Rightarrow \forall \phi \in \Phi^A . [\![\phi]\!]^C(s) = [\![\phi]\!]^C(s')$$

Proof (sketch) Assuming that $s \sim s'$ we can proceed by induction on the structure of the formula. One interesting case is $[\alpha]\,\phi_1$, since it involves the use of distributivity. By definition we have:

$$[\![[\alpha]\,\phi_1]\!](s) = [\![[\alpha]\,\phi_1]\!](s')$$

$$\Leftrightarrow \Cap_{\{\langle x, s_i\rangle \in f_{tr}(s) | x = \alpha\}} [\![\phi_1]\!]\Gamma(s) = \Cap_{\left\{\langle \alpha, s'_j\rangle \in f_{tr}(s')\right\}} [\![\phi_1]\!]\Gamma(s')$$

Now, by bisimilarity, for each s_i there is an s_j such that $s_i \sim s_j$, and thus, by coinduction, $[\![\phi_1]\!](s_i) = [\![\phi_1]\!](s_j)$. Thus, for each value $k = [\![\phi_1]\!](s_i)$ for $\langle \alpha, s_i\rangle \in f_{tr}(s)$, there is at least an $\langle \alpha, s_j\rangle \in f_{tr}(s')$ such that $k = [\![\phi_1]\!](s_j)$, and possibly more than one. This also holds in the opposite direction, so there is such an s_i for each s_j. Since in a distributive c-semiring the $\Cap$ operation is idempotent, the above equation is satisfied. □

Remark 4.2 If the c-semiring is not distributive, this result does not hold. Consider for example an SLTS with states $\{s_1, \ldots, s_5\}$ such that $f_{tr}(s_1) = \{\langle \alpha, s_2\rangle\}$, and $f_{tr}(s_3) = \{\langle \alpha, s_4\rangle, \langle \alpha, s_5\rangle\}$. States s_1 and s_3 are bisimilar, but the formula $[a]\,k$ distinguishes them, if we choose k such that $k \Cap k \neq k$, which necessarily exists if C is not distributive.

Now we show that, when interpreted over the booleans, QOS-μ characterizes bisimilarity.

Theorem 4.3 *For s and s' states of an SLTS, it holds that*

$$(\forall \phi \in \Phi^{\{true, false\}} . [\![\phi]\!]^{Bool}(s) = [\![\phi]\!]^{Bool}(s')) \Leftrightarrow s \sim s'$$

Proof (sketch) The $\Leftarrow$ part is a direct consequence of theorem 4.1, but also similar to standard results in modal logics for labeled transition systems. For the $\Rightarrow$ part, one can show that the relation $R(s, s') = \forall \phi \in \Phi^{\{true, false\}} . [\![\phi]\!]^{Bool}(s) = [\![\phi]\!]^{Bool}(s')$ is a bisimulation. For the temporal part of any SLTS, which is indeed a LTS, and the temporal fragment of QOS-μ, the result is well-known. Proofs for many modal logics can be done following the ones in [21]. For example, suppose that $R(s, s')$ and $s \xrightarrow{a} s_1$, but $\neg \exists s' \xrightarrow{a} s'_i$ such that $R(s_1, s'_i)$, then for every $s' \xrightarrow{a} s'_i$ there exists ϕ_i such that $[\![\phi_i]\!](s_1) \neq [\![\phi_i]\!](s'_i)$. Then $[\![\phi]\!](s) \neq [\![\phi]\!](s')$, where $\phi = \langle a\rangle \wedge_i \phi_i$, and this is a contradiction. For the spatial component, the proof can be done following the same pattern. □

The last theorem needed to establish our result is that logical equivalence for formulas interpreted over a (distributive or not) c-semiring C implies logical equivalence for formulas interpreted over *Bool*.

Theorem 4.4 *For $C = \langle A, \Cup, \Cap, \mathbf{0}, \mathbf{1}\rangle$ c-semiring, and s, s' states of an SLTS, we have:*

$$(\forall \phi \in \Phi^A.[\![\phi]\!]^C(s) = [\![\phi]\!]^C(s')) \Rightarrow (\forall \phi \in \Phi^{\{true,false\}}.[\![\phi]\!]^{Bool}(s) = [\![\phi]\!]^{Bool}(s'))$$

Proof (sketch) The structure $C' = \langle \{\mathbf{0}, \mathbf{1}\}, \Cup, \Cap, \mathbf{0}, \mathbf{1}\rangle$, obtained from C by removing all constants from A except from the top and bottom element, is indeed a well defined c-semiring, isomorphic to the boolean one and respecting the same axioms. So, given any formula in $\Phi^{\{\mathbf{0},\mathbf{1}\}}$ interpreted over C', we can cast the same formula into $\Phi^{\{true,false\}}$, with an isomorphic interpretation; due to logical equivalence over C, we also have logical equivalence over C' and thus over $Bool$. □

Now we are able to state the main theorem of this paper, encompassing all the results of this section:

Theorem 4.5 *QOS-μ interpreted over any non distributive c-semiring is strictly finer than QOS-μ interpreted over a distributive c-semiring. Moreover, QOS-μ interpreted over a distributive c-semiring characterizes bisimilarity.*

Proof We first prove adequacy, i.e. for $C = \langle A, \Cup, \Cap, \mathbf{0}, \mathbf{1}\rangle$ distributive c-semiring, $s \sim s' \iff \forall \phi \in \Phi^A.[\![\phi]\!]^C(s) = [\![\phi]\!]^C(s')$.

The $\Rightarrow$ part is given by theorem 4.1. Now suppose that $\forall \phi \in \Phi^A.[\![\phi]\!]^C(s) = [\![\phi]\!]^C(s')$, then by theorem 4.4 we have $\forall \phi \in \Phi^{\{true,false\}}.[\![\phi]\!]^{Bool}(s) = [\![\phi]\!]^{Bool}(s')$, and by theorem 4.3 we get $s \sim s'$, concluding the adequacy proof.

By remark 4.2, it follows that QOS-μ interpreted over non-distributive c-semirings is finer than QOS-μ interpreted over distributive c-semirings. □

Remark 4.6 Regarding decidability of the model-checking problem, in [20] it is stated that the fragment without function symbols of the c-semiring μ-calculus defined there is decidable (for finite models) when using distributive c-semirings. This result can be casted in our framework, even in the case of QOS-μ, when interpreted over distributive c-semirings.

Remark 4.7 It is well-known that formulas in multi-valued logics over finite lattices can be checked using boolean formulas which express *cuts* on the state-transition graph of the system [14]. C-semirings, however, can be infinite, thus this technique cannot be used; for certain fixed-points formulas of QOS-μ, indeed, we get a strictly higher expressiveness than using finite lattices, since the equivalence classes induced by these formulas are infinite (with just a single formula, we can separate the set of all systems into infinite subsets).

5 Applications

5.1 SLTS Semantics for Process Calculi

In this section, as a case study, we introduce a SLTS semantics for pure CCS. This semantics is inspired both by the STS semantics of [23] and by the location semantics of [5]. A common criticism of spatial logics is that the induced equivalence, in many cases, is too intensional, and coincides with structural equivalence. The spatial semantics we introduce here, being not purely syntactical, but rather behavioral, does not coincide with structural equivalence, yet it allows to check properties related to the parallel composition of processes. In the following we assume an enumerable set Act of actions, and use the notation $\overline{Act} = \{\bar{x} \mid x \in Act\}$.

Definition 5.1 The syntax of CCS processes is given as follows

$$P ::= P \parallel P \mid \sum_{i \in I} \alpha_i.P_i \mid (\nu l)P \mid R(\overrightarrow{x})$$

where $\alpha_i \in \mathcal{L} = \{\tau\} \cup Act \cup \overline{Act}$, $\tau \notin Act \cup \overline{Act}$, I is a finite set, $l \in Act$, $\overrightarrow{x} \in Act^*$, and for each process identifier R there is a defining equation $R(\overrightarrow{x}) = P_R$, where each process identifier in P_R is in the scope of an action α (guarded recursion).

The sum over an empty set is written 0, name l in $(\nu l)P$ is bound in P and subject to the rules of α-conversion, and $fn(P)$ is the set of free names in the labels of a process.

Definition 5.2 (Structural congruence) Structural congruence is the least congruence $\equiv$ satisfying commutativity, associativity and identity law of $\sum$ and $\parallel$ (with 0 as the unit element), the laws of α-conversion for the restriction operator $(\nu l)P$, and the axioms:

- $(\nu\alpha)(P \parallel Q) \equiv P \parallel (\nu\alpha)Q$ if $\alpha \notin fn(P)$;
- $(\nu\alpha)0 \equiv 0$, $(\nu\alpha)(\nu\beta)P \equiv (\nu\beta)(\nu\alpha)P$.

We now give the SLTS semantics of the calculus.

Definition 5.3 (SLTS semantics of CCS) The temporal transition function f_{tr} is

defined by the following rules:

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \xrightarrow{\alpha} P'}{(\nu l)P \xrightarrow{\alpha} (\nu l)P'} \text{ if } \alpha \neq l, \bar{l}$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\bar{\alpha}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \qquad \frac{P \equiv P' \qquad P' \xrightarrow{\alpha} Q' \qquad Q' \equiv Q}{P \xrightarrow{\alpha} Q}$$

$$\frac{P\left[\overrightarrow{k}/_{\overrightarrow{x}}\right] \xrightarrow{\alpha} P'}{A(\overrightarrow{k}) \xrightarrow{\alpha} P'} \text{ if } A(\overrightarrow{x}) = P$$

The spatial decomposition function f_{sp} is defined as:

$$f_{sp}(P) = \begin{cases} \langle 0, * \rangle & \text{if } reactive(P) \wedge dec(P) = \emptyset \\ \langle 1, dec(P) \rangle & \text{otherwise} \end{cases}$$

where

$$reactive(P) = \exists n \geq 0. P \xrightarrow{\tau} \ldots \xrightarrow{\tau} P_n \xrightarrow{\alpha} P' \wedge \alpha \neq \tau$$

$$dec(P) = \left\{ \langle P_1, P_2 \rangle \,\middle|\, P \equiv P_1 \parallel P_2,\ reactive(P_1),\ reactive(P_2) \right\}$$

This definition, differently from the one in [23], allows us to observe the parallel decomposition of a process only if the involved subprocesses are *reactive*, i.e. they can synchronize with other processes at some point (similarly to the location semantics of CCS [5]). The obtained SLTS bisimulation obviously contains ordinary CCS bisimulation, and it is coarser than structural congruence: consider the processes $(\nu x)\,(x.0 \parallel \bar{x}.0) \parallel (\nu y)\,(y.0 \parallel \bar{y}.0) \parallel a$ and $(\nu x)\,(x.x.0 \parallel \bar{x}.\bar{x}.0) \parallel a$. These two processes are SLTS and CCS bisimilar, but not structurally congruent.

5.2 Examples of QOS-μ usage

We provide two examples of QOS-μ usage. We adopt the following shorthand: $\langle L \rangle\, \phi \triangleq \vee_{\alpha \in L} \langle \alpha \rangle\, \phi$, where L is a finite subset of $\mathcal{L}$.

Example 5.4 (lower bound of resource availability) Consider $Act = \{a, b, c\}$, $\mathcal{L} = \{\tau\} \cup Act \cup \overline{Act}$, and the c-semiring $C_{opt} = \langle \mathbb{R}, min, +, +\infty, 0 \rangle$. The formula

$$\phi_1 = \mu X. \left((\langle a \rangle\, 1 \vee [a] + \infty) \wedge \circlearrowleft \right) \vee (X \parallel X)$$

counts the number of parallel processes that can do the action a in a given state. Now consider the formula

$$\phi_2 = \langle b \rangle \, \mu X. \, (\phi_1 \vee (\langle \mathcal{L} \rangle \, X))$$

this formula measures the minimum available quantity of a certain kind of resource (modeled by a process that can do a), after the occurrence of event b. This kind of formula can be compared, in the setting of multi-valued logics, to the issues dealt with when considering *history dependent access control policies* [1]. In our case, the *measure* of resource usage policies depends on the past history of execution, as it is the case for the *validity* of policies in history dependent access control.

Example 5.5 (different cost of parallel and sequential execution) Consider the following two process definitions:

$$\begin{aligned} P_1 &= R \parallel R \parallel R \\ P_2 &= S \parallel S \parallel S \\ P_R &= a.R + b.R + c.R \\ P_S &= a.0 + b.0 + c.0 \end{aligned}$$

The process identifiers R and S represent a basic resource that is employed to implement services offered by system P_1 and P_2, respectively, and whose minimum availability during a service invocation is used to determine the specific cost that the client has to pay.

Let us consider the two formulas ϕ_3 and ϕ_4 interpreted over the c-semiring C_{opt}:

$$\phi_3 = \phi_1 \parallel (\circlearrowleft \wedge \langle b \rangle \, 0) \parallel (\circlearrowleft \wedge \langle c \rangle \, 0)$$

and

$$\phi_4 = (\phi_1 \parallel (\circlearrowleft \wedge \langle b \rangle \, 0)) \vee \langle b \rangle \, (\phi_1 \parallel (\circlearrowleft \wedge \langle c \rangle \, 0))$$

In both cases, the formulas count how many parallel processes in the current state are enabled to do a, using ϕ_1. The number of processes that can do a is the number of available resources, and if put in parallel with other formulas, it gives a measure of the *remaining* resources, when some of them are used by a client. Formulas ϕ_3 and ϕ_4 are execution plans to run the actions b and c on the server. In ϕ_3 we consider the parallel execution of both actions, while in ϕ_4 we consider the sequential execution of b and c, thus we have to measure the available resources in both states reached. Applying both formulas, we can check whether in a given state it costs less to execute b and c sequentially or in parallel. We have $[\![\phi_3]\!](P_1) = 1$, while $[\![\phi_4]\!](P_1) = 2$, so if the system is in state P_1, it costs less, for the client, to use the plan ϕ_3. On the other hand, we have $[\![\phi_3]\!](P_2) = [\![\phi_4]\!](P_2) = 1$, thus the cost of the two plans is the same. The intuition behind this phenomenon is that in P_2 resources get consumed, so it is important to acquire them as soon as possible. More complex systems and plans could as well make the parallel execution of two tasks less resource-consuming than their execution in sequence.

6 Concluding Remarks

We have introduced a family of extensional spatial logics for a class of models which have labeled transitions, thus allowing behavioral observations. These logics are generalized over c-semirings, and if the c-semirings used for the interpretation are distributive, they are decidable over finite models and adequate with respects to bisimilarity. Using these logics, quantitative analysis of systems can be performed in a generalized way.

Future developments are manifold. The SLTS semantics for CCS given in section 5 does not decompose processes under the scope of a restriction. This is a limit of our semantics, when comparing it to location semantics. Further study is required to see if it is possible to obtain a SLTS semantics equivalent to the location one. Spatial semantics of calculi like the π-calculus or the ambient calculus should be given using SLTS, and we should investigate what relation is induced on processes from bisimulation in the model.

Other applications of interest are in the area of *service-oriented computing* [25]: a formal, semantic notion of quality measurement can be used for *quality of service* requirements, and also as the basis for the implementation of *semantic service discovery*, where a client and a server establish a *contract* by defining respectively a minimum level of quality of service, and a maximum level of resource usage, in an heterogeneous situation where many different programming languages are used for the implementation of services, and thus semantic models must be used. The quantitative, resource conscious nature of QOS-μ ensures to be able to model real-world situations in which resources are limited, and there is a maximum number of clients that can simultaneously access each service.

An open question is if there are classes of non-distributive c-semirings for which decidability over finite models holds, while work is being done on coalgebraic models where the interpretation of QOS-μ using non-distributive c-semirings characterizes bisimilarity. Finally, since spatial logics have been used as the basis of query languages for semistructured data [10], applications of quantitative analysis, using QOS-μ, to this domain, could be worth further research.

Acknowledgement

The authors wish to thank Alberto Lluch-Lafuente for helpful comments.

References

[1] Abadi, M. and C. Fournet, *Access control based on execution history.*, in: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA* (2003).

[2] Bartoletti, M., P. Degano and G. L. Ferrari, *Types and effects for secure service orchestration*, in: *CSFW '06: Proceedings of the 19th IEEE Workshop on Computer Security Foundations* (2006), pp. 57–69.

[3] Bistarelli, S., "Semirings for Soft Constraint Solving and Programming," Lecture Notes in Computer Science **2962**, Springer, 2004.

[4] Bistarelli, S., U. Montanari and F. Rossi, *Semiring-based constraint satisfaction and optimization*, Journal of the ACM **44** (1997), pp. 201–236.

[5] Boudol, G., I. Castellani, M. Hennessy and A. Kiehn, *Observing localities*, Theoretical Computer Science **114** (1993), pp. 31–61.

[6] Caires, L., *Behavioral and spatial properties in a logic for the pi-calculus*, in: I. Walukiwicz, editor, *Proc. of Foundations of Software Science and Computation Structures'2004 (FOSSACS)*, number 2987 in Lecture Notes in Computer Science (2004), pp. 72–89.

[7] Caires, L., *Spatial-behavioral types for distributed services and resources*, in: *Proceedings of the 2nd International Symposium on Trustworthy Global Computing (TGC06), to appear*, 2006.

[8] Caires, L. and L. Cardelli, *A Spatial Logic for Concurrency (Part I)*, Information and Computation **186** (2003), pp. 194–235.

[9] Caires, L. and L. Cardelli, *A Spatial Logic for Concurrency (Part II)*, Theoretical Computer Science **3** (2004), pp. 517–565.

[10] Cardelli, L. and G. Ghelli, *TQL: a query language for semistructured data based on the ambient logic.* (2004).

[11] Chechik, M., B. Devereux, S. Easterbrook and A. Gurfinkel, *Multi-valued symbolic model-checking*, ACM Trans. Softw. Eng. Methodol. **12** (2003), pp. 371–408.

[12] de Alfaro, L., *Quantitative verification and control via the mu-calculus.*, in: *CONCUR*, 2003, pp. 102–126.

[13] Ferrari, G. and A. Lluch-Lafuente, *A logic for graphs with qos.*, Electr. Notes Theor. Comput. Sci. **142** (2006), pp. 143–160.

[14] Gurfinkel, A. and M. Chechik, *Multi-valued model checking via classical model checking.*, in: R. M. Amadio and D. Lugiez, editors, *CONCUR*, Lecture Notes in Computer Science **2761** (2003), pp. 263–277.

[15] Hennessy, M. and J. Riely, *Resource access control in systems of mobile agents*, Inf. Comput. **173** (2002), pp. 82–120.

[16] Hirsch, D., A. Lluch-Lafuente and E. Tuosto, *A logic for application level qos.*, Electr. Notes Theor. Comput. Sci. **153** (2006), pp. 135–159.

[17] Hirsch, D. and E. Tuosto, *SHReQ: Coordinating Application Level QoS*, in: *SEFM '05: Proceedings of the Third IEEE International Conference on Software Engineering and Formal Methods* (2005), pp. 425–434.

[18] Igarashi, A. and N. Kobayashi, *Resource usage analysis*, in: *Symposium on Principles of Programming Languages*, 2002, pp. 331–342.

[19] Jacobs, B. and J. Rutten, *A tutorial on (co)algebras and (co)induction*, Bulletin of the European Association for Theoretical Computer Science **62** (1997), pp. 222–259.

[20] Lafuente, A. and U. Montanari, *Quantitative mu-calculus and CTL defined over constraint semirings*, Electronic Notes on Theoretical Computing Systems QAPL **112** (2005), pp. 1–30.

[21] Milner, R., J. Parrow and D. Walker, *Modal logics for mobile processes*, Theoretical Computer Science **114** (1993), pp. 149–171.

[22] Mohri, M. and M. Riley, *A weight pushing algorithm for large vocabulary speech recognition*, in European Conf. on Speech Communication and Technology, Aalborg, Denmark (2001).

[23] Monteiro, L., *A noninterleaving model of concurrency based on transition systems with spatial structure.*, Electr. Notes Theor. Comput. Sci. **106** (2004), pp. 261–277.

[24] Nicola, R. D., G. L. Ferrari, U. Montanari, R. Pugliese and E. Tuosto, *A Process Calculus for QoS-Aware Applications.*, in: *COORDINATION*, 2005, pp. 33–48.

[25] Papazoglou, M. P. and D. Georgakopoulos, *Special issue on service oriented computing*, Commun. ACM **46** (2003).

[26] Rutten, J. J. M. M., *Universal coalgebra: a theory of systems*, Theoretical Computer Science **249** (2000), pp. 3–80.