TRUST DOMAINS IN SYSTEM MODELS: ALGEBRA, LOGIC, UTILITY, AND COMBINATORS

GABRIELLE ANDERSON AND DAVID PYM

ABSTRACT. Understanding the boundaries of trust is a key aspect of accurately modelling the structure and behaviour of multi-agent systems with heterogeneous motivating factors. Reasoning about these boundaries in highly interconnected, information-rich ecosystems is complex, and dependent upon modelling at the correct level of abstraction. Building on an established mathematical systems modelling framework that captures the classical view of distributed systems, we develop a modelling framework that incorporates both logical and cost-based descriptions of systems, which allows us to establish a definition of an agent's trust domain based on the satisfaction of logical properties at acceptable utility (handled here simply as cost) to the agent, of verification. In addition to the technical properties of the modelling framework itself, we establish a theory of logical combinators, including substitution, for composing trust domains to form relatively complex models of trust. We illustrate the ideas with examples throughout.

1. INTRODUCTION

Complex systems of interacting agents, be they artificial or natural, are ubiquitous. For example, complex networks of devices and services underpin most of the systems upon which modern societies depend. Such systems are difficult to conceptualize and reason about effectively.

When agents within complex systems must interact with one another and collaborate in order to achieve their goals, the concept of trust — between agents — is important. An agent, situated within a system that contains also other agents, may establish a part of the system, or a collection of other agents within the system, that it trusts. Similarly, a system's designer or manager might establish a collection of parts of the system such that, within any given part, the agents trust one another. We shall refer to such a part of the system, or such a collection of agents, as a 'trust domain' [42].

We illustrate these motivating questions with an example: Which contractors will a company trust with its data when contracting out some internal process? What data sources are trustworthy when making a benefit-cost analysis? Where are there opportunities for colluding parties to trust each other and commit fraud? Establishing the boundaries of what an agent, or group of agents, trusts, is extremely useful when attempting to understand the interconnected, information-rich ecosystems upon which the world is more-or-less wholly dependent. The trust domain of an entity (an agent or collection of agents) consists of an ecosystem of other agents with which the entity willing to interact [1, 2]. This interaction may consist of work sharing, resource sharing, location sharing, and so on. An entity will be concerned with what the trusted agents can do (or not do).

In this paper, we propose a characterization of trust domains that has two components. First, a logical assertion that expresses the properties that the entity requires to be possessed by the trusted ecosystem. Second, a bound on the cost, to the entity, of establishing that the required properties hold. This latter requirement captures the extent to which establishing a required property for trust will affect the overall utility of the entity's actions.

This set-up is illustrated informally in Figure 1, in which we intend an implicit notion of logical or physical location.

Here the agent A may be given one of two different choices of cost function. If $K_A = K$, then B is not within A's trust domain at either the k_1 or k_2 levels. If, however, $K_A = l$, then B is within A's trust domain at the l_2, but not at the l_1 level. Agent B's cost function, m, includes agent C at the m_2 level, but not at the m_1 level ($m_1 \leq m_2$). B' is in no-one's domain at any of the given levels of cost.

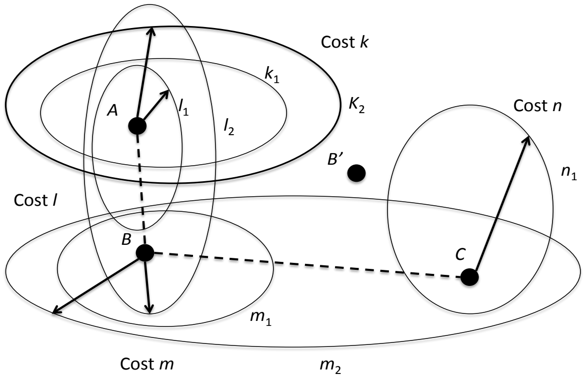


Figure 1. Trust Domains: Properties and Cost Bounds

Our use of cost and cost functions here is inspired by utility theory as used, in particular, in the theory of multi-agent systems [39, 28]. However, in this paper, we do not seek to establish, for example, the axiomatic structure of von Neumann–Morgenstern utility.

Informally, located agents manipulate their resource environments, but, in our formulation, they do so in contexts which characterize the extent to which they do so whilst maintaining a required logical property (intuitively, the 'trust' property) within a specified bound on cost. This approach stands in contrast to approaches in which constraints are expressed purely in terms of preferences, where impossible choices, that can be expressed logically in our setting, must represented by 'infinitely negative' utility. For brevity, we do not employ location explicitly, instead trusting that the intuitions suggested in Figure 1 will make a sufficiently strong indication.

The mathematical formulation of this set-up is established within a calculus of resources and processes, and its associated modal logic, introduced in [2]. This modelling framework builds directly on the ideas presented in [14, 12, 11, 13], in which a mathematical account of the classical view of distributed systems — as described, for example, in [15] — is given. The key structural components of this modelling framework are the following:

- Location: Locations are the places, with directed links between them, within and outwith the system where resources reside; locations can be logical or physical. Conceptually, locations consist of a collection of places, connected by links (which have direction). Places have arities, which express the numbers of in and out links associated with them. More generally, we can think of a location as a collection of places, with their associated links, so that more location may be substituted for another. This situation is depicted in Figure 2;

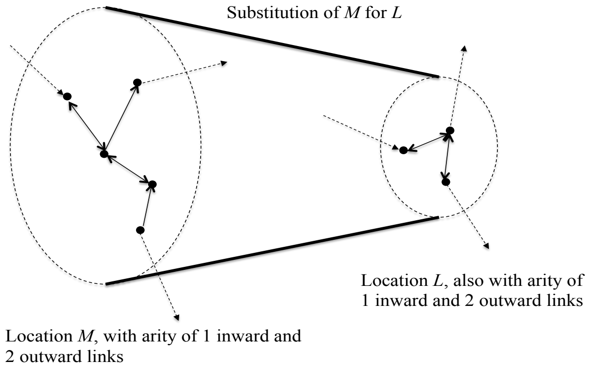

Substitution of M for L
Location L, also with arity of 1 inward and 2 outward links
Location M, with arity of 1 inward and 2 outward links

Figure 2. Substitution of Locations

- Resource: Resources are the building blocks of the system's services, such as system components, computer memory, people, or money; they can, for example, be consumed, created, and moved between locations by the system's processes. Conceptually, the axioms of resources are that resource elements can be combined and compared;
- Process: Processes deliver services within and outwith the system, manipulating the resources that are distributed around the system's logical and physical locations; they provide the dynamics of a model, and interact with the system's environment.

The remaining component of the modelling framework is the representation of the context within which models exist, described as follows:

- Environment: The system's environment is the part of the system whose structure is modelled in detail. The interaction between the model of interest and its environment is captured mathematically using stochastic processes to provide occurrences of events.

The mathematical modelling of these components is described in Section 3.1.

Along with this modelling framework we have a logic, formulated in the style of Hennessy–Milner logic [20, 19, 33], as developed for bunched systems in [11, 13, 14]. This logic provides an appropriate language for expressing the properties required, as discussed above, to characterize trust domains.

Our characterization of trust domains in this context relies on the addition of the following key concepts:

- Evolution in context: Processes that represent agents evolve with respect to an inner context and and outer context. The outer context describes the environment in which the process is considered to evolve. The inner context, which should be thought of as substituting for part of the process, describes that part of the environment that contributes to the definition of the process itself. The formal set-up is explained in Section 3.3;
- Cost-mediated choice: Our formalization of processes, explained below in Section 3.3, employs not simply non-deterministic choice between possible evolutions, but rather a choice operator in which subsequent evolutions selected according to contextually determined costs;
- Cost modalities: In the modal logic that, in the style of Hennessy–Milner logic [20, 19, 33, 11, 13, 14], is associated with with our contextual process algebra, we employ cost modalities that characterize, via possibility and necessity, cost-constrained evolutions.

The development of a process algebra that integrates, in the form that is required for this paper, these concepts into the modelling framework described in Section 3.1 is described in Section 3.3, and a corresponding cost-sensitve Hennessy–Milner- style modal logic is developed in Section 3.4.

The information systems we consider are surprisingly resistant to traditional formal specification and verification approaches, where the level of abstraction is essentially fixed at the (typically, rather low) level of the specification of the components.To model problems in these settings, it is essential to work at an appropriate level of abstraction, which often involves working at a higher level. Considering problems in these settings involves controlling the complexity, and hence the tractability, of the model. Accordingly, straightforward and intuitive techniques for constructing and combining models are essential, in order to divide and conquer the complexity.

We introduce combinators that can be used to construct new trust domains from existing ones. In particular, we lift combinators from the logic to be defined over trust domains. We make use of additive, or model preserving, combinators, as in classical propositional logic, namely $\neg, \vee, \wedge$, and $\rightarrow$. We also make use of multiplicative, or model separating, combinators, as in BI and separation logics [34, 37], namely $*$ and $\circ\!\!-\!\!*$ (a contextual version of the usual bunched $-\!\!*$). These combinators describe how models can be decomposed, and how the logical properties of separate models relate to the logical properties of composed models. All of these combinators, when raised to the level of trust domains, can be interpreted using set-theoretic techniques, simplifying the reasoning required of the modeller.

Additive conjunction can be used to strengthen the logical properties that an agent expects of its trusted ecosystem. The trust domain of the logical conjunct is the set intersection of the sets of trusted ecosystems for each of the sub-properties. Additive disjunction can be used to weaken the logical properties that an agent expects of its trusted ecosystem. The trust domain of the logical conjunct is the set union of the sets of trusted ecosystems for each of the sub-properties.

A trust domain defines the ecosystems where an agent has a logical property, and is able to achieve this property within a given cost-bound; this makes use of the 'possibility' modal operator. The negation of a trust domains defines ecosystems where all possible choices an agent have within a given cost-bound achieve a certain property. This makes use of the 'necessity' modal operator, and can be used to define properties such as that a given individual is never trusted, or that an agent always avoids data leaks.

Additive implication is treated normally, and can be used, for example, to ensure that either a given individual is excluded from a trust domain, or that a higher level of security is in place. The trust domain of a logical implication is the set union between the negation of the trust domain of the premiss, and the trust domain of the conclusion.

Multiplicative conjunction is used to describe how two separate agents can determine the extent of their *combined* trust, with respect to their individual extents of trust. In order to find some new bound, we often increase one agent's cost-bound, and decrease the other agent's, providing a trade-off between the possible cost-bounds. This approach enables us to explore notions of transitivity within trust domains.

In addition, we consider the notion of substitution for trust domains. We discuss how to perform substitution for each of the constituent parts of a trust domain, namely the agent, the logical properties, and the cost-bounds. These techniques can then be used to extend the work on combinators.

Our existing formal characterization of trust domains is built on a mathematical systems modelling framework [14, 1, 2]. The key components of our approach are the following: first, a resource-sensitive process algebra, with a decision-theoretic notion of utility-centric choice, and a corresponding, resource-sensitive modal logic of processes; second, a conceptual notion of trust domain, characterized using our algebraic, logical, and utility-theoretic tools; and third, a meta-theory of how trust domains can be constructed, which provides a topological interpretation of how different agents' trust interacts. The literature on models of trust is very large and cannot be surveyed comprehensively in this short article, but a good survey with a relevant perspective for us is [36].

In Section 3, we provide a gentle introduction to our modelling approach. This followed by a summary of the mathematical formulation, consisting of a process algebra that describes the costs of different decisions, and a modal, substructural logic, including cost modalities. We also formally define trust domains, in terms of the process algebra and cost logic. In Section 5, with the use of a running example about contract choices, we describe the meta-theory of trust domains in terms of the combinators introduced above. In Section 6, with the use of a running example about hospital costs, we describe the meta-theory of trust domains in terms of substitution. In Section 7, we conclude and discuss some directions for future research.

2. Related Work

We include a brief discussion of work that is related to the three key concepts mentioned above. Whilst there is quite little work that is immediately related to our approach, the broader relevant literature is very large. Accordingly, our discussion of related work is intended to be illustrative rather than comprehensive. Work related to our general programme has been discussed above.

The notion of weighted choice in a process calculus setting is explored elegantly in [43]. An abstract notion of weight is introduced, with a two-sorted transition system. The weights can be used to represent notions of priority, as in our approach to cost, infinite weighted priority, and frequency (in the sense of probability theory).

In our work, the notion of utility is based on contextual information that reaches beyond the agent making the decision. While we do not explicitly consider how the structure of that context is reflected in the utility functions, it would be valuable to do so, and certainly possible within the formulation.

There are various concepts that could be explored for such structure, such as contextual information through situation theory [4, 5], the distinction between risk and uncertainty [29], and the notion of the trade-off between the cost of establishing facts about the context verses the utility available through possible choices [27]. Contextual notions of trust for agent based software engineering, which encompasses high-level dependence, described through a modal logic, occurs in [40].

The use of process calculi for decision support is a well established technique. The Performance Evaluation Process Algebra (PEPA) [21, 22, 23] establishes a compositional theory of Markov Processes, separable product-form distributions for efficient storage, congruence results, and model checking tool support [10, 30, 23]. This technology can be used to model performance for different designs, enabling decisions to be made at a meta level between such designs. Resource–process reasoning can be used to establish costs for different design decisions, such as for different security policies [6, 9, 12].

Markov Chains are studied in [26]: they support reasoning about complex notions such as average utility with a given time discount, but do not provide compositionality results over model structures. Process calculi for Markov decision processes, which include both stochastic and cost-based decision-making, provide such compositionality results for the class of systems that do not permit negative utility, and then only for a notion of simulation [17]. It may be possible to extend work on probabilistic synchronous calculi [3] to provide a process calculus with more general notions of cost, and with full bisimulation.

Different types of decisions can interact in unexpected and non-trivial ways. Non-deterministic choice and probabilistic choices do not distribute over each other straightforwardly [41, 16]. Early attempts to design probabilistic calculi replaced non-deterministic choice with probabilistic choice, as we have replaced non-deterministic choice with cost based choice. The interaction between non-deterministic, probabilistic, and cost-based choice would be an interesting direction for future research.

In defining trust domains, we make use of cost modalities of possibility and necessity. The possibility modalities $\langle \leq n \rangle \phi$ and $\langle > n \rangle \phi$ denote that there exists an evolution whose cost m is less than or equal to, or greater than, n, respectively, where the resulting state satisfies ϕ. The necessity modalities $[\leq n]$ and $[> n]$ denote that in all evolutions whose cost m is less than or equal to, or greater than, n, respectively, where the resulting state satisfies ϕ.

These modalities are related to the preference modalities described by various authors, with various motivations and formulations too numerous to discuss here. See, for example, von Wright [44, 45], van Bentham, Otterloo, and Roy [8], Girard [18], van Bentham, Girard, and Roy [7], Osherson and Weinstein [35], and other work cited in these papers.

3. Systems Modelling and Decision-making

This section has three parts. First, in Section 3.1, we explain our previous work in system modelling based on concepts of location, resource, process, and environment [11, 13, 12, 14]. Second, in Section 3.2, we explain our approach, building on [1, 2], to integrating cost as a basis for determining choices made during process execution. Third, in Section 3.3, we describe formally a process algebra with contextual costs, supporting the approach sketched in the previous section, and establish its essential meta-theory. This is a substantial technical section. Finally, in Section 3.4, we set up, in the style of Hennessy–Milner logic [20, 19, 11, 13, 14], a modal logic that includes, in the style of the bunched logic BI, both additive and multiplicative propositional connectives, additive action modalities, and, importantly in this setting, cost modalities.

3.1. Systems Modelling Background.

In Section 1, we have explained that, following the the classical model of distributed systems, such as described in [15], the core components of the structural aspect of our modelling framework, which builds on [14, 1, 2], are location, resource, and process, together with the stochastically modelled environment. In order to introduce the specific approach to systems modelling that we employ in this paper, we explain below how these concepts are modelled mathematically. The detailed theoretical development is provided in [11, 13, 12, 14].

- *Location*. Mathematically, locations can convenient modelling using a range of graph-theoretic and topological structures [13, 14]. For the purposes of this paper, in which we do not develop a mathematical account of location, it is sufficient to think of locations as being given by directed graphs (e.g., as in Figure 2).
- *Resource*. Mathematically, resources are assumed to form a preordered partial commutative *resource monoid*,

  $$\mathbf{R} = (\mathbf{R}, \sqsubseteq, \circ, e),$$

  in which resource elements $R_1, R_2 \in \mathbf{R}$ can be combined, using the monoid operation to form $R_1 \circ R_2$ (with unit e) or compared, $R_1 \sqsubseteq R_2$, say, using the preorder. The partiality ensures that not all combinations need be considered (for example, such as those beyond a certain size in a resource monoid based on the natural numbers). The structure of the monoid is subject to some coherence conditions [34, 14]. A key example of a monoid of resources is given by the natural numbers (with 0), with addition as the monoid operation and less-than-or-equals as the order: $(\mathbb{N}, \leq, +, 0)$.

- *Process*. Mathematically, our treatment of process is based on Milner's synchronous calculus of communicating systems (SCCS) [32], as developed as a basis for systems modelling in [14]. Note that asynchronous calculi can be encoded within such synchronous calculi [32].

For simplicity of presentation, and with little loss of generality for our present purposes, we suppress locations in the remainder of this presentation, though make informal use of them in our examples, in Section 5. The reader might think of them either as implicitly present, or consider them to be rolled up into the definition of resources (see [14] for relevant technical support).

The key idea is that resources and processes co-evolve, according one of the following judgements: first, $R, E \xrightarrow{a} R', E'$, which is read as 'the process E, using resources R, performs action a and so becomes the process E' that is able to evolve using resources R''; or second, $R, E \overset{n}{\Longrightarrow} R', E'$, which is read as 'the process E, using resources R, makes some choice(s) that incur cost n and so becomes the process E' that is able to evolve using resources R''.

Actions are required to form a commutative monoid (a and b combine to form ab). the relationship between actions and resources must be specified using a *modification function* that specifies the effect of performing an action a on a resource element R: that is, $\mu : (a, R) \mapsto R'$. Modification functions must satisfy some (mild) coherence conditions relating the monoid structure of actions and the monoidal structure of resources (details may be found in [14]). This treatment of resource just as in bunched logic [34] and in various versions of separation logic [37] and, for brevity, we refrain from further rehearsing its justification here.

These judgements are defined using a structural operational semantics, such as in the definition of SCCS [32, 14]. In its basic form, the operational semantics admits rules such as

$$\frac{}{R, a : E \to^{a} \mu(a, R), E} \quad \text{(Prefix)}$$

$$\text{(1)} \qquad \frac{R, E_i \to^{a} R', E'}{R, E_1 + E_2 \to^{a} R', E'}\, i = 1, 2 \quad \text{(Sum)}$$

$$\frac{R, E \to^{a} R', E' \quad S, F \to^{b} S', F'}{R \circ S, E \times F \to^{ab} R' \circ S', E' \times F'} \quad \text{(Prod)}$$

giving a process evolution via its head action where its resources are modified accordingly, non-deterministic choice, and concurrent product, respectively. Other familiar process combinators can be handled similarly [11, 14].

In the presence of explicitly modelled locations, we work with a basic judgement of the form

$$L, R, E \to^{a} L', R', E'$$

describing the co-evolution of locations, resources, and processes.

A given system that is modelled exists in the context of its interaction with other systems that may not be modelled. Such unmodelled systems constitute the environment within which the modelled system interacts. Events originating in the environment may be incident upon the the system of interest, and the system of interest may cause events to be incident upon the environment.

- *Environment*. The natural way to model the interaction of the system of interest with its environment is to use stochastic methods [38, 25, 21, 22, 12, 14]. Events that are incident upon the system are determined by sampling specified probability distributions. For example, the arrival of entities at a system portal — such as people joining a queue or ships entering a harbour — might be captured using a negative exponential distribution. Essentially, sampling a distribution 'creates' an action [12, 14].

3.2. **Integrating Evolution and Cost.** Processes model the dynamics of a system — for example, the steps of employing a contractor to perform a specialized task — but they say nothing about the costs involved in carrying out those tasks. Cost, as previously discussed, is an essential aspect of determining trust boundaries. For example, sharing confidential data to an off-site analyst comes at a high risk of leakage, while employing an analyst to work with the data on-site does not. This can be represented by assigning the cost 0.7 to the process $R, \mathrm{off\_site} : E$ that models working off-site, and 0.2 to the process

$R, \text{on\_site} : F$ that models working on-site. Cost is used, as in utility theory, to encompass uplift for revenue (or another measure of value). Cost functions map resource–process pairs (called process contexts, the set of which is denoted *Cont*) to real numbers. It is important to note that costs are often contextually dependent. For example, when the contractor has a robust IT security strategy, the cost associated with sharing the data off-site may be lower: $0.4 = u(R \circ S, (\text{off\_site} : E) \times IT)$, where S are the resources allocated to the IT process. We assume, for each formal cost u (in a given set U of such symbols, used below in the formal definition of processes), an associated, real-valued *cost function* $u : Cont \longrightarrow \mathbb{R}$ (see [28]) that fixes an interpretation for each formal symbol $u \in U$. The identically zero function is associated with 0_U. Henceforth, we do not distinguish between formal costs and their costs functions.

When a process makes a choice, we annotate the cost n of the chosen summand on the evolution (e.g., $R, \text{off\_site} : E +_u \text{on\_site} : F \Longrightarrow^{0.7} R, \text{off\_site} : E$). As the cost depends on the context, a choice point needs to take account of the context in which it appears, when it determines the costs of the possible summands. We henceforth annotate the context in which a process is evolved on the underside of the evolution arrow (e.g., $R, \text{off\_site} : E +_u \text{on\_site} : F \xRightarrow[S,IT\times[\,]]{}^{0.7} R, \text{off\_site} : E$), where $[\,]$ denotes the hole into which $\text{off\_site} : E +_u \text{on\_site} : F$ may be substituted to regain the complete system $(\text{off\_site} : E +_u \text{on\_site} : F) \times IT$. In addition, any choices in $[\,] \times IT$ will make use of the process that is substituted into the hole $[\,]$. We therefore annotate the process that is substituted, into the process being evolved, on top of the evolution arrow; for example, $S, [\,] \times IT \xRightarrow{R,\text{off\_site}:E+_u\text{on\_site}:F}{}^{n} S', [\,] \times IT'$.

Essentially, the judgement for evolution for processes with cost of the form

$$C \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} C' \tag{2}$$

denotes how a context C, that exists in a system that can be decomposed as $C_1(C(C_2))$, evolves in terms of its choices. We refer to C as the *(primary) context*, C_1 as the *outer context*, and C_2 as the *substituted*, or *inner context*. Intuitively, this set-up describes the evolution of one part, C, of an entire system, $C_1(C(C_2))$. In order to reason compositionally, we wish to be able to describe the evolution of C independently and structurally. As choices take account of context, this is not possible. The semantics of choice, however, makes use of just the definition of the inner and outer context, disregarding their structure. So we do not need to make use of the structure of C_1 and C_2, as we do with C, but need only record their definitions, for reference at choice points. They are therefore annotated on the evolution arrow, but are not evolved in that relation.

3.3. **A Process Algebra with Contextual Costs.** We now have introduced all of the concepts required to describe the theoretical set-up, introduced, without proofs, in [2]. We now describe the theoretical set-up in detail.

Assume the set U of symbols, called *formal costs*, with a distinguished element 0_U, called the *neutral cost*. *Processes* are generated by the grammar

$$E ::= \mathbf{1} \mid [\,] \mid a : E \mid \sum_{i \in I}{}_u E_i \mid E \times E, \tag{3}$$

as discussed below. These are really process contexts: the term $[\,]$ is a *hole* into which other processes may be substituted. For this work, it turns out to be convenient to develop contexts as first-class citizens rather than merely meta-theoretic tools. The *choice* $\sum_{i \in I}{}_u E_i$ is the key construct: it describes situations in which an agent has a choice between alternatives E_i, where $i \in I$ for a finite indexing set I, and its cost (in a larger context) is codified by the cost $u \in U$. The infix operator $E +_u F$ may be used for binary sums, and the subscript u may be dropped when $u = 0_U$. The *zero* process $\mathbf{0}$ is defined to be the sum indexed by the empty set and the neutral cost. The zero process, *unit* process $\mathbf{1}$, and *synchronous products* $E \times F$ are well-known in process calculus, as are *prefixes* $a : E$, where $a \in \text{Act}$. We conjecture that other process combinators, such as those considered in [11, 14], can be included in the framework presented here.

A process E is *well-formed* if it contains at most one hole and that hole is not guarded by action prefixes. From now on we assume all processes are well-formed. The process E is *closed* if it has no holes and *open* otherwise. Let *PCont* be the set of all well-formed processes, *PCCont* be the set of all closed well formed processes, and *POCont* be the set of all open well-formed processes. Let $\mathbf{R}$ be a resource monoid and μ be a fixed modification function, as previously defined. Define the products of sets $Cont = \mathbf{R} \times PCont$,

$$\frac{}{R,\mathbf{1} \xrightarrow[C_1]{C_2}{}^{1} R,\mathbf{1}}\ (\textsc{Tick}) \qquad \frac{}{R, a : E \xrightarrow[C_1]{C_2}{}^{a} \mu(a,R), E}\ (\textsc{Prefix})$$

$$\frac{C_2 \xrightarrow[C_1]{(e,\mathbf{1})}{}^{a} C_2'}{e,[\,] \xrightarrow[C_1]{C_2}{}^{1} e,[\,]}\ (\textsc{Hole})$$

$$(S\times)\frac{R,E \xrightarrow[C_3]{C_2}{}^{a} R',E' \qquad S,F \xrightarrow[C_4]{C_2}{}^{b} S',F'}{R \circ S, E \times F \xrightarrow[C_1]{C_2}{}^{ab} R' \circ S', E' \times F'}\ (\textsc{Prod})$$

Figure 3. Action Operational Semantics

$$\frac{}{R,\mathbf{1} \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{0} R,\mathbf{1}}\ (\textsc{Tick}\mathrm{W}) \qquad \frac{}{R, a : E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{0} R, a : E}\ (\textsc{Prefix}\mathrm{W})$$

$$\frac{C_2 \overset{(e,\mathbf{1})}{\underset{C_1}{\Longrightarrow}}{}^{n} C_2'}{e,[\,] \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{0} e,[\,]}\ (\textsc{Hole}\mathrm{W}) \qquad \frac{n = u(C_1(R, E_i(C_2)))}{R, \sum_{u\,I} E_i \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R, E_i}\ (\textsc{Sum}\mathrm{W})$$

$$(S\times)\frac{R,E \overset{C_2}{\underset{C_3}{\Longrightarrow}}{}^{n} R,E' \qquad S,F \overset{C_2}{\underset{C_4}{\Longrightarrow}}{}^{m} S,F'}{R \circ S, E \times F \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n+m} R \circ S, E' \times F'}\ (\textsc{Prod}\mathrm{W})$$

Figure 4. Operational Semantics of Cost

$CCont = \mathbf{R} \times PCCont$ and $OCont = \mathbf{R} \times POCont$. The letter C is reserved for contexts. Define $C_\emptyset = e, [\,]$. Brackets will be freely used to disambiguate both processes and contexts. For $C = R, E$, the notational abuses $C \times F = R, (E \times F)$ and $C +_u F = R, (E +_u F)$ will sometimes be used. Substitution in processes, $E(F)$, replaces all occurrences of $[\,]$ in E with F; for example, $(([\,] +_u E) \times G)(F) = (F +_u E) \times G$. Substitution of contexts $C_1(C_2)$, where $C_1 = R, E$ and $C_2 = S, F$, is defined as follows: if E is open, then $C_1(C_2) = R \circ S, E(F)$, where $E(F)$ is process substitution; if E is closed, then $C_1(C_2) = C_1$.

In developing the formulation sketched above, we separate the operational semantics into two dimensions: the evolution system for performing actions (Figure 3) and the evolution system for determining the cost of possible choices (Figure 4), as in [43], and building on [1]. Overall, the evolution sequences for the calculus are interleavings of the two dimensions.

The operational semantics for performing actions is defined in Figure 3.

Although the evolution relation for actions does not evolve choices, we also annotate the inner and outer contexts on the evolution relation for consistency of presentation. The unit process always ticks, effecting no change. The prefix process evolves via its head action. The hole rule is a technical one used to terminate evolution derivations of open contexts. An important feature of this system is that contextual information about conclusions is propagated up to premisses. In the product case, information about each premiss is propagated up from the conclusion to the other premiss, so that derivations of transitions occur in context. This is effected by the side-condition $(S\times)$ is which states that $C_3 = C_1((S, F(C_2)) \times [\,])$ and $C_4 = C_1((R, E(C_2)) \times [\,])$, which passes the details of one sub-process to the reduction of the other.

The operational semantics for determining the cost of possible choices is defined in Figure 4.

A neutral cost is given to tick, prefix, and hole processes, as they contain no choices. The sum process $\sum_{u\,I} E_i$ represents a preference-based choice by the agent: it evolves to one of its summands, annotating the value of that summand in the wider context on the evolution arrow, according to its cost function u. A special case of the sum is for the zero process $\mathbf{0}$, which never evolves. The product evolves two processes synchronously in parallel, according to the decomposition of the associated resources, and annotates the sum of the sub-processes' costs on the evolution arrow. This approach to combining costs, and the value given to tick or prefix processes, is one possible design decision, and will be considered more fully in future work. We make use of the abbreviation $C \overset{n}{\Longrightarrow} C'$ and $C \overset{a}{\rightarrow} C'$ to denote $C \xRightarrow[e,[\,]]{e,\mathbf{1}}{}^{n} C'$ and $C \xrightarrow[e,[\,]]{e,\mathbf{1}}{}^{a} C'$, respectively.

Lemma 1. *If* $R, E \xrightarrow[C_1]{C_2}{}^{a} S, F$*, then* $S = \mu(a, R)$*.*

Proof. Straightforward, by induction over the derivation of $R, E \xrightarrow[C_1]{C_2}{}^{a} S, F$. □

We show how the substitute is irrelevant for the purposes of reduction when the primary context is closed. This is used to show reduction under any substitute, given an existing reduction under a specific substitute.

Lemma 2. *If C is closed, then for all* $C_1, C_2, C_3, C' \in \mathit{Cont}$*:*

(1) $C \xrightarrow[C_1]{C_2}{}^{a} C' \;\Leftrightarrow\; C \xrightarrow[C_1]{C_3}{}^{a} C'$

(2) $C \xRightarrow[C_1]{C_2}{}^{n} C' \;\Leftrightarrow\; C \xRightarrow[C_1]{C_3}{}^{n} C'$.

Proof.

(1) Property 1:

$\Rightarrow$ Straightforward, by induction over the derivation of $C \xrightarrow[C_1]{C_2}{}^{a} C'$.

$\Leftarrow$ Straightforward, by induction over the derivation of $C \xrightarrow[C_3]{C_4}{}^{a} C'$.

(2) Property 2:

$\Rightarrow$ Straightforward, by induction over the derivation of $C \xRightarrow[C_1]{C_2}{}^{n} C'$. The key case is for SumW. As C is closed then by the definition of substitution For all C_3, $C_1(R, E_i(C_2)) = C_1(R, E_i(C_3))$ and hence, for all C_3, $u(C_1(R, E_i(C_2))) = u(C_1(R, E_i(C_3))) = n$. By the induction hypothesis, we know that if $C_i \xRightarrow[C_4]{C_3}{}^{m} C_i'$, where C_4 is as in the side-condition, then C_i' is closed. Hence, by the SumW rule we have that $R, \underset{I}{\textstyle\sum_u} E_i \xRightarrow[C_1]{C_3}{}^{n} R, E_i'$, as required.

$\Leftarrow$ Straightforward, by induction over the derivation of $C \xRightarrow[C_3]{C_4}{}^{n} C'$.

□

We now give the technical results that are required for our arguments. A fundamental aspect of process calculus is the ability to reason equationally about behavioural equivalence of processes [32]. We define bisimilarity to suit the calculus in this paper, which incorporates ideas from [14].

The *bisimulation relation* $\sim\; \subseteq \mathit{PCont} \times \mathit{PCont}$ is the largest binary relation such that, if $E \sim F$, then for all $a \in \mathrm{Act}$, for all $R, R', S, T \in \mathbf{R}$, and for all $G, H, I, J \in \mathit{PCont}$ with $G \sim I$ and $H \sim J$, then

(1) for all $E' \in \mathit{PCont}$, if $R, E \xrightarrow[S,G]{T,H}{}^{a} R', E'$, then there is F' such that $R, F \xrightarrow[S,I]{T,J}{}^{a} R', F'$ and $E' \sim F'$, and if $R, E \xRightarrow[S,G]{T,H}{}^{n} R, E$, then there is F' such that $R, F \xRightarrow[S,I]{T,J}{}^{n} R, F'$ and $E' \sim F'$, and

(2) for all $F' \in \mathit{PCont}$, if $R, F \xrightarrow[S,I]{T,J}{}^{a} R', F'$, then there is E' such that $R, E \xrightarrow[S,G]{T,H}{}^{a} R', E'$ and $E' \sim F'$, and if $R, F \xRightarrow[S,I]{T,J}{}^{n} R, F'$, then there is E' such that $R, E \xRightarrow[S,G]{T,H}{}^{n} R, E'$ and $E' \sim F'$.

The union of any set of relations that satisfy these two conditions also satisfies these conditions, so the largest such relation is well-defined. Define $\sim\; \subseteq \mathit{Cont} \times \mathit{Cont}$ by: if $E \sim F$ then $R, E \sim R, F$ for all $R \in \mathbf{R}$ and $E, F \in \mathit{Cont}$.

Definition 3. *A cost function, u,* respects bisimilarity *if, for all* $C_1, C_2 \in \mathit{Cont}$*,* $C_1 \sim C_2$ *implies* $u(C_1) = u(C_2)$*.*

That is, behaviourally equivalent (bisimilar) states are required to be indistinguishable by u. The set U of utilities respects bisimilarity if every $u \in U$ respects bisimilarity. Any real-valued function defined on the quotient $Cont/\sim$ defines a cost that respects bisimilarity. Henceforth cost functions are assumed to respect bisimilarity.

We can show that if bisimilar contexts are substituted into each other, then the result is bisimilar.

Proposition 4 (Bisimulation Closure Under Substitution)**.** *If $E \sim G$ and $F \sim H$, then $E(F) \sim G(H)$.*

Proof. The bisimulation relation $\sim$ is the largest bisimulation relation, and contains all other bisimulation relations. In order to show that $E(F) \sim G(H)$ it is sufficient, therefore, to define a relation $\mathcal{R}$, where $E(F)\,\mathcal{R}\,G(H)$, for which the required substitution property holds, and show that the relation R is a bisimulation.

Let $\mathcal{R} = \{(E(F), G(H)) \mid E \sim G \text{ and } F \sim H\} \cup \sim$. The relation is a bisimulation if and only if the following holds: for all T, T', I', $C_1 \sim C_3$, and $C_2 \sim C_4$, if $T, E(F) \xrightarrow[C_1]{C_2}{}^{c}\ T', I'$ (respectively $T, E(F) \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n}\ T, I'$), then there exists some J' such that $T, G(H) \xrightarrow[C_3]{C_4}{}^{c}\ T', J'$ (respectively $T, G(H) \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n}\ T, J'$), where $(I', J') \in \mathcal{R}$; and, for all T, T', J', $C_1 \sim C_3$, and $C_2 \sim C_4$, if $T, G(H) \xrightarrow[C_3]{C_4}{}^{c}\ T', J'$ (respectively $T, G(H) \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n}\ T, J'$), then there exists some I' such that $T, E(F) \xrightarrow[C_1]{C_2}{}^{c}\ T', I'$ (respectively $T, E(F) \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n}\ T, I'$), where $(I', J') \in \mathcal{R}$.

All processes are defined by a finite number of applications of the operators of the language. We proceed by induction on the derivation of this structure according to the rules of the operational semantics.

Consider the case in which $T, E(F) \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n}\ T, I'$. We prove that there exists some J' such that $T, G(H) \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n}\ T, J'$ by induction on the structures of E, G, F, and H, and over the (process) structures of C_1, C_3, C_2, and C_4, in that order. Here the induction is on the number of operators in a process term.

Consider the case of this nested induction in which $E = \mathbf{1}$, $G = G_1 \times G_2$, $F = \mathbf{1}$, and $H = \mathbf{1}$, where $C_1 = e, [\,]$, $C_3 = e, [\,]$, $C_2 = e, \mathbf{1}$, and $C_4 = e, \mathbf{1}$.

In SCCS, if $\mathbf{1} \sim G_1 \times G_2$, then G_1 and G_2 would necessarily be bisimilar to $\mathbf{1}$, but here that is not the case. Consider the process $G_1 = \mathbf{1} +_u \mathbf{1}$, where

$$u(C) = \begin{cases} 0 & \text{if there exists } C' \in OCont, R \text{ where } C = C'(R, G_2) \\ n & \text{otherwise,} \end{cases}$$

for some $n > 0$.

We then have that $(\mathbf{1} +_u \mathbf{1}) \times G_2 \sim \mathbf{1}$; a sketch of the proof follows below.

Consider some contexts $C_5 \sim C_7$ and $C_6 \sim C_8$, and resources $R = R_1 \circ R_2$. We then have that $u(C_5(R, \mathbf{1} \times G_2)) = 0$, for all C_5, and hence by the (SumW) rule that $R_1, \mathbf{1} +_u \mathbf{1} \overset{C_2}{\underset{C_5(R_1,[\,]\times G_2)}{\Longrightarrow}}{}^{0}\ R_1, \mathbf{1}$. Note, however, that $\mathbf{1} +_u \mathbf{1}$ is not bisimilar to $\mathbf{1}$; for the empty outer context $e, [\,]$ the former can perform a n-cost transition while the latter cannot.

As a result, bisimulation does not work component-by-component (solely because of cost-moderated choice). Note, however, that following a cost transition by $\mathbf{1} +_u \mathbf{1}$, the resulting process $\mathbf{1}$ is bisimilar to $\mathbf{1}$. As the reduction of cost-moderated choices applies to strictly simpler terms, we can apply the induction hypothesis and show that substituted processes are bisimilar, and hence the choices that are formed from them are also bisimilar.

This argument extends to more complex cases, where the sub-components of a product are not merely tick processes. There, cost moderated choices as sub-components will also eventually reduce to a product subcomponent to which the induction hypothesis can be applied.

The proof sketched above makes use of an eight-fold nested induction, which has a very large number of cases. Unfortunately, the complexity of this nested induction precludes an exhaustive presentation. However, it is easy to see that the difficult cases in the induction arise solely from use of cost-moderated choices and, since these always reduce to simpler processes to which the induction hypothesis can be applied, the key cases are all similar to the one described above.

The remaining cases are routine. □

Note that the argument presented above would not be applicable in the presence of general (guarded) fixed points; our contextual calculus must be a finite modelling framework.

With this result, we can obtain a key property for reasoning compositionally, that bisimulation is a congruence.

Theorem 5 (Bisimulation Congruence)**.** *The relation $\sim$ is a congruence. It is reflexive, symmetric, and transitive, and, for all a, E, F, G, with $E \sim F$ and all families $(E_i)_{i\in I}$, $(F_{i\in I})_I$ with $E_i \sim F_i$ for all $i \in I$, $a : E \sim a : F$, $E \times G \sim F \times G$, and $\underset{i\in I}{\sum\nolimits_u} E_i \sim \underset{i\in I}{\sum\nolimits_u} F_i$.*

Proof. By induction on the structure of bisimulations. We give illustrative cases.

(1) Reflexive. Let $\mathcal{R} = \{(E, E)\}$. As the evolution of each of the process E is decorated with different outer and inner contexts (C_1 and C_2, and C_3 and C_4, respectively) we do not immediately have reflexivity. We prove this property by induction on the derivation of $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R, E'$.

The relation $\mathcal{R}$ is a bisimulation if and only if, for all E and F such that $E \mathcal{R} F$, the following holds: for all R, R', E', a, n, $C_1 \sim C_3$, and $C_2 \sim C_4$, if $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$ (respectively $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R, E'$), then there exists some F' such that $R, F \xrightarrow[C_3]{C_4}{}^{a} R', F'$ (respectively $R, F \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n} R, F'$), where $E' \mathcal{R} F'$; and, for all R, R', F', a, n, $C_1 \sim C_3$, and $C_2 \sim C_4$, if $R, F \xrightarrow[C_3]{C_4}{}^{a} R', F'$ (respectively $R, F \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n} R, F'$), then there exists some E' such that $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$ (respectively $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R, E'$), where $E' \mathcal{R} F'$.

- Consider some $E \mathcal{R} F$, R, R', E', a, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$. By the definition of $\mathcal{R}$, we know that $F = E$. We then prove that there exists some E'', such that $R, E \xrightarrow[C_3]{C_4}{}^{a} R', E''$ and $E' \sim E''$, by induction over the derivation of $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$.
- Consider some $E \mathcal{R} F$, R, R', E', n, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{a} R, E'$. By the definition of $\mathcal{R}$, we know that $F = E$. We then prove that there exists some E'', such that $R, E \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{a} R, E''$ and $E' \sim E''$ by induction over the derivation of $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{a} R, E'$.
- Consider some $E \mathcal{R} F$, R, R', F', a, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, F \xrightarrow[C_3]{C_4}{}^{a} R', F'$. By the definition of $\mathcal{R}$, we know that $F = E$ and $F' = E''$. We then prove that there exists some E', such that $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$ and $E' \sim E''$, by induction over the derivation of $R, E \xrightarrow[C_3]{C_4}{}^{a} R', E''$.
- Consider some $E \mathcal{R} F$, R, R', F', n, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, F \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{a} R, F'$. By the definition of $\mathcal{R}$, we know that $F = E$ and $F' = E''$. We then prove that there exists some E', such that $R, E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{a} R, E'$ and $E' \sim E''$, by induction over the derivation of $R, E \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{a} R, E''$.

Hence $\mathcal{R}$ is closed and a bisimulation.

(2) Symmetric. Let $\mathcal{R} = \{(F, E) \mid E \sim F\} \cup \sim$. If $R, F \xrightarrow[C_1]{C_2}{}^{a} S, F'$, $C_1 \sim C_3$, and $C_2 \sim C_4$, then we need to show that $R, E \xrightarrow[C_3]{C_4}{}^{a} S, E'$, where $F' \mathcal{R} E'$. As $E \sim F$, by the definition of bisimulation, we have that if $C_5 \sim C_7, C_6 \sim C_8$, and $R, F \xrightarrow[C_5]{C_6}{}^{a} S, F'$, then $R, E \xrightarrow[C_7]{C_8}{}^{a} S, E'$, where $E' \sim F'$. Let $C_1 = C_5, C_2 = C_6, C_3 = C_7$, and $C_4 = C_8$. We then have that $R, E \xrightarrow[C_3]{C_4}{}^{a} S, E'$. As $E' \sim F'$, we have that $F' \mathcal{R} E'$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(3) Transitive. Let $\mathcal{R} = \{(E, G) \mid E \sim F \text{ and } F \sim G\}$. If $R, E \xrightarrow[C_1]{C_2}{}^{a} S, E'$, $C_1 \sim C_3$, and $C_2 \sim C_4$, then we need to show that $R, G \xrightarrow[C_3]{C_4}{}^{a} S, G'$, where $E' \mathcal{R} G'$. By the definition of bisimulation, as

$R, E \xrightarrow[C_1]{C_2}{}^{a} S, E'$, we have that $R, F \xrightarrow[C_1]{C_2}{}^{a} S, F'$, where $E' \sim F'$, and, similarly, as $R, F \xrightarrow[C_1]{C_2}{}^{a} S, F'$, we have that $R, G \xrightarrow[C_3]{C_4}{}^{a} S, G'$, where $F' \sim G'$. We then have that $E' \mathcal{R} G'$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(4) Let $\mathcal{R} = \{(a : E, a : F) \mid E \sim F\} \cup \sim$. If $R, a : E \xrightarrow[C_1]{C_2}{}^{a} S, E$, $C_1 \sim C_3$ and $C_2 \sim C_4$, then we need to prove that $R, a : F \xrightarrow[C_1]{C_2}{}^{a} S, F$, where $E \mathcal{R} F$. The only applicable reduction rule for $a : E$ is the (Prefix). By this rule, which disregards C_3 and C_4, we can show that $R, a : F \xrightarrow[C_1]{C_2}{}^{a} S, F$. As $E \sim F$, we have that $E \mathcal{R} F$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(5) Let $\mathcal{R} = \{(E +_u G, F +_u G) \mid E \sim F\} \cup \sim$. If $R, E +_u G \xRightarrow[C_1]{C_2}{}^{n} R, E'$, $C_1 \sim C_3$ and $C_2 \sim C_4$, then we need to prove that $R, F +_u G \xRightarrow[C_3]{C_4}{}^{n} R, F'$ and $E' \mathcal{R} F'$. By (SumW), we know the that $n = u(C_1(R, E(C_2)))$ and $E' = E$. As $E \sim F$ and $C_5 \sim C_6$ (by Proposition 4), by Definition 1, we have that $u(C_3(R, F(C_4))) = u(C_1(R, E(C_2)))$. Then, by (SumW), we can show that $R, F +_u G \xrightarrow[C_3]{C_4}{}^{n} S, F$. As $E \sim F$, we have that $E \mathcal{R} F$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(6) Let $\mathcal{R} = \{(E \times G, F \times G) \mid E \sim F\}$. If $R \circ S, E \times G \xrightarrow[C_1]{C_2}{}^{ab} R' \circ S', E' \times G'$, $C_1 \sim C_3$ and $C_2 \sim C_4$, then we need to prove that $R \circ S, F \times G \xrightarrow[C_3]{C_4}{}^{ab} R' \circ S', F' \times G''$, where $E' \times G' \mathcal{R} F' \times G''$. By the (Prod) rule we have that $R, E \xrightarrow[C_5]{C_2}{}^{a} R', E'$ and $S, G \xrightarrow[C_6]{C_2}{}^{b} S', G'$, where $C_5 = C_1((S, F(C_2)) \times [\,])$ and $C_6 = C_1((R, E(C_2)) \times [\,])$. By Proposition 4 and $\mathcal{R}$, we have that $C_5 \sim C_7 = (C_3(S, G(C_4)) \times [\,]$ and $C_6 \sim C_8 = (C_3(R, F(C_4)) \times [\,]$. By the definition of bisimulation, we have that $R, F \xrightarrow[C_7]{C_4}{}^{a} R', F'$ and $S, G \xrightarrow[C_8]{C_4}{}^{b} S', G''$, where $E' \sim F'$ and $G' \sim G''$. We can then use the (Prod) rule to show that $R \circ S, F \times G \xrightarrow[C_1]{C_2}{}^{ab} R' \circ S', F' \times G''$, where $E' \times G' \mathcal{R} F' \times G''$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

□

In order to reason equationally about processes, it is also useful to establish various algebraic properties concerning parallel composition and choice. We derive these properties for our calculus below. We use the binary version of sum here in order to aid comprehension, but finite choices between sets of processes work straightforwardly.

Proposition 6 (Algebraic Properties)**.** *For all u, E, F, G, we have the following: (1) $E +_u F \sim F +_u E$; (2) $E \times \mathbf{0} \sim \mathbf{0}$; (3) $E \times \mathbf{1} \sim E$; (4) $E \times F \sim F \times E$; and (5) $E \times (F \times G) \sim (E \times F) \times G$.*

Proof. (1) Let $\mathcal{R} = \{(E +_u F, F +_u E) \mid E, F : PCont\} \cup \sim$. The relation $\mathcal{R}$ is a bisimulation if and only if, for all E and F such that $E \mathcal{R} F$, the following holds: for all R, R', E', a, n, $C_1 \sim C_3$, and $C_2 \sim C_4$, if $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$ (respectively $R, E \xRightarrow[C_1]{C_2}{}^{n} R, E'$), then there exists some F' such that $R, F \xrightarrow[C_3]{C_4}{}^{a} R', F'$ (respectively $R, F \xRightarrow[C_3]{C_4}{}^{n} R, F'$), where $E' \mathcal{R} F'$; and, for all R, R', F', a, n, $C_1 \sim C_3$, and $C_2 \sim C_4$, if $R, F \xrightarrow[C_3]{C_4}{}^{a} R', F'$ (respectively $R, F \xRightarrow[C_3]{C_4}{}^{n} R, F'$), then there exists some E' such that $R, E \xrightarrow[C_1]{C_2}{}^{a} R', E'$ (respectively $R, E \xRightarrow[C_1]{C_2}{}^{n} R, E'$), where $E' \mathcal{R} F'$.

- Consider some $E_1 \mathcal{R} E_2$, R, R', E_1', a, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E_1 \xrightarrow[C_1]{C_2}{}^{a} R', E_1'$. Consider the case in which $E_1 \sim E_2$. Then, by the definition of bisimulation, we have that there exists some E_2' such that $R, E_2 \xrightarrow[C_3]{C_4}{}^{a} R', E_2'$, where $E_1' \sim E_2'$, and hence $E_1' \mathcal{R} E_2'$. Consider the case in which $E_1 = E +_u F$ and $E_2 = F +_u E$. There is no action reduction rule for the sum operator, and therefore there are no such transitions $R, E_1 \xrightarrow[C_1]{C_2}{}^{a} R', E_1'$. As such, this case is vacuously true.

- Consider some $E_1 \mathcal{R} E_2$, R, R', E_1', n, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E_1 \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R', E_1'$. Consider the case in which $E_1 \sim E_2$. Then, by the definition of bisimulation, we have that there exists some E_2' such that $R, E_2 \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n} R', E_2'$, where $E_1' \sim E_2'$, and hence $E_1' \mathcal{R} E_2'$. Consider the case in which $E_1 = E +_u F$ and $E_2 = F +_u E$. By the (SumW) rule we have that either $n = u(C_1(R, E(C_2)))$ and $E_1' = E$, or $n = u(C_1(R, F(C_2)))$ and $E_1' = F$. Consider the former case (the other is symmetric). By Proposition 4, we have that $C_1(R, E(C_2)) \sim C_3(R, E(C_4))$ and, by Definition 1, we have that $n = u(C_3(R, E(C_4)))$. Then, by the (SumW) rule, we can derive $R, F +_u E \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n} R, E$.
- Consider some $E_1 \mathcal{R} E_2$, R, R', E_2', a, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E_2 \xrightarrow[C_3]{C_4}{}^{a} R', E_2'$. Consider the case where $E_1 \sim E_2$. Then, by the definition of bisimulation, we have that there exists some E_1' such that $R, E_1 \xrightarrow[C_1]{C_2}{}^{a} R', E_1'$, where $E_1' \sim E_2'$, and hence $E_1' \mathcal{R} E_2'$. Consider the case in which $E_1 = E +_u F$ and $E_2 = F +_u E$. There is no action reduction rule for the sum operator, and therefore there are no such transitions $R, E_2 \xrightarrow[C_3]{C_4}{}^{a} R', E_2'$. As such, this case is vacuously true.
- Consider some $E_1 \mathcal{R} E_2$, R, R', E_2', n, $C_1 \sim C_3$, and $C_2 \sim C_4$ such that $R, E_w \overset{C_4}{\underset{C_3}{\Longrightarrow}}{}^{n} R', E_2'$. Consider the case in which $E_1 \sim E_2$. Then, by the definition of bisimulation, we have that there exists some E_1' such that $R, E_1 \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R', E_1'$, where $E_1' \sim E_2'$, and hence $E_1' \mathcal{R} E_2'$. Consider the case in which $E_1 = E +_u F$ and $E_2 = F +_u E$. By the (SumW) rule, we have that either $n = u(C_3(R, F(C_4)))$ and $E_2' = F$, or $n = u(C_3(R, E(C_4)))$ and $E_2' = E$. Consider the former case (the other is symmetric). By Proposition 4, we have that $C_1(R, F(C_2)) \sim C_3(R, F(C_4))$, and by Definition 1 we have that $n = u(C_1(R, F(C_2)))$. Then, by the (SumW) rule, we can derive $R, F +_u E \overset{C_2}{\underset{C_1}{\Longrightarrow}}{}^{n} R, E$.

Hence $\mathcal{R}$ is closed and a bisimulation.

(2) Let $\mathcal{R} = \{(E \times \mathbf{0}, \mathbf{0}) \mid E \in \mathit{PCont}\}$. By the operational semantics we have that $\mathbf{0}$ can make neither action nor weighted transitions. The only applicable rules to $E \times \mathbf{0}$ are (Prod) and (ProdW). These both require, as sub-derivations, that $\mathbf{0}$ make a transition, which is impossible. Hence $E \times \mathbf{0}$ can also make no transitions, and is bisimilar to $\mathbf{0}$.

(3) Let $\mathcal{R} = \{(E \times \mathbf{1},\ F) \mid E \sim F\}$. If $R, E \times \mathbf{1} \xrightarrow[C_1]{C_2}{}^{a} S, E'$, $C_1 \sim C_3$ and $C_2 \sim C_4$, then we need to show that $R, F \xrightarrow[C_3]{C_4}{}^{a} S, F'$, where $E' \mathcal{R} F'$. Let $C_5 = C_1((e, \mathbf{1}(C_2)) \times [\,])$. By the (Prod) rule, we have that $R, E \xrightarrow[C_5]{C_2}{}^{a} S, E'$. By $\mathcal{R}$, we have that $[\,] \times \mathbf{1} \sim [\,]$ and, by Proposition 4, we then have that $C_3 \sim C_5 \sim C_1$. We can then show that $R, E \xrightarrow[C_3]{C_4}{}^{a} S, F'$, where $E' \sim F'$. We then have that $E' \times \mathbf{1} \mathcal{R} F'$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(4) Let $\mathcal{R} = \{(E \times F, F \times E) \mid E, F : \mathit{PCont}\}$. If $R \circ S, E \times F \xrightarrow[C_1]{C_2}{}^{ab} R' \circ S', E' \times F'$, $C_1 \sim C_3$ and $C_2 \sim C_4$, then we need to show that $R \circ S, F \times E \xrightarrow[C_3]{C_4}{}^{ba} R' \circ S', F'' \times E''$, where $E' \times F' \mathcal{R} F'' \times E''$. Let $C_5 = C_1((S, F(C_2)) \times [\,])$ and $C_6 = C_1((R, E(C_2)) \times [\,])$. By the (Prod) rule, we have that $R, E \xrightarrow[C_5]{C_2}{}^{a} R', E'$ and $S, F \xrightarrow[C_6]{C_2}{}^{b} S', F'$. Let $C_7 = C_3((S, F(C_4)) \times [\,])$ and $C_8 = C_3((R, E(C_4)) \times [\,])$. By $\mathcal{R}$ and Proposition 4, we have that $C_5 \sim C_7$ and $C_6 \sim C_8$. By the definition of bisimulation, we

have that $R, E \xrightarrow[C_7]{C_4}{}^{a} R', E''$ and $S, F \xrightarrow[C_8]{C_4}{}^{b} S', F''$, where $E' \sim E''$ and $F' \sim F''$. Then, by the (Prod) rule, we have that $R \circ S, F \times E \xrightarrow[C_3]{C_4}{}^{ba} R' \circ S', F' \times E'$, where $(E' \times F', F' \times E') \in \mathcal{R}$. As $E' \sim E''$ and $F' \sim F''$, we have that $E' \times F' \mathcal{R} F'' \times E''$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(5) Let $\mathcal{R} = \{(E \times (F \times G), (E \times F) \times G) \mid E, F, G : PCont\}$. If $R \circ S \circ T, E \times (F \times G) \xrightarrow[C_1]{C_2}{}^{abc} R' \circ S' \circ T', E' \times (F' \times G')$, $C_1 \sim C_3$, and $C_2 \sim C_4$, then we need to show that $R \circ S \circ T, (E \times F) \times G \xrightarrow[C_3]{C_4}{}^{abc} R' \circ S' \circ T', (E'' \times F'') \times G''$, where $E' \times (F' \times G') \mathcal{R} (E'' \times F'') \times G''$. Let $C_5 = C_1((S \circ T, F \times G(C_2)) \times [\,])$ and $C_6 = C_1((R, E(C_2)) \times [\,])$. By the (Prod) rule, we have that $R, E \xrightarrow[C_5]{C_2}{}^{a} R', E'$ and that $S \circ T, F \times G \xrightarrow[C_6]{C_2}{}^{bc} S' \circ T', F' \times G'$. Let $C_7 = C_6((T, G(C_2)) \times [\,])$ and $C_8 = C_6((S, F(C_2)) \times [\,])$. By the (Prod) rule, we have that $S, F \xrightarrow[C_7]{C_2}{}^{b} S', F'$ and that $T, G \xrightarrow[C_8]{C_2}{}^{c} T', G'$. Let $C_9 = C_3(T, G(C_4) \times [\,]), C_{10} = C_9((S, F(C_4)) \times [\,])$, $C_{11} = C_9((R, E(C_4)) \times [\,])$, and $C_{12} = C_3((R \circ S, E \times F(C_4)) \times [\,])$. By $\mathcal{R}$ and Proposition 4, we have that $C_8 \sim C_{12}$, $C_7 \sim C_{11}$, and $C_5 \sim C_{10}$. By the definition of bisimulation, we have that $R, E \xrightarrow[C_{10}]{C_4}{}^{a} R', E', S, F \xrightarrow[C_{11}]{C_4}{}^{b} S', F'$, and $T, G \xrightarrow[C_{10}]{C_4}{}^{c} T', G'$, where $E' \sim E''$, $F' \sim F''$, and $G' \sim G''$. Then, by the (Prod) rule, we have that $R \circ S, E \times F \xrightarrow[C_9]{C_4}{}^{ab} R' \circ S', E'' \times F''$, and that $R \circ S \circ T, (E \times F) \times G \xrightarrow[C_3]{C_4}{}^{abc} R' \circ S' \circ T', (E'' \times F'') \times G''$. As $E' \sim E''$, $F' \sim F''$, and $G' \sim G''$, we have that $E' \times (F' \times G') \mathcal{R} (E'' \times F'') \times G''$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

□

3.4. A Cost-sensitive Modal Logic.

As mentioned in Section 1, we make use of a modal logic, formulated in the style of Hennessy–Milner logic [20, 19, 33], as developed for bunched systems in [11, 13, 14], to express properties of the models that are built using the process algebra defined in the previous section. Of particular interest are action modalities, that describe what actions a process can (or must) perform, and cost modal connectives, that describe the cost of choices a process can (or must) make.

The semantics is given by a satisfaction relation

$$C \models_{C'} \phi, \tag{4}$$

where C is a closed context, C' is an open context, and ϕ is a formula of a (Hennessy–Milner-style) modal logic of processes: this may be read 'the *primary context* C satisfies ϕ in the *surrounding context* C''.

The context C may satisfy different logical propositions, perhaps even negations of each other, when placed in different surrounding contexts; an example of this is below. Context-sensitive logics have been studied previously [31, 5]. The structural nature of processes and resources provides a semantic framework in which such logics seem particularly natural.

The propositions of the logic are defined by the following grammar:

$$\begin{array}{rl} \phi ::= & p \mid \bot \mid \top \mid \neg\phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \phi \rightarrow \phi \\ & \mid \langle a \rangle \phi \mid [a]\phi \\ & \mid I \mid \phi * \phi \mid \phi \mathrel{\circ\!\!-\!\!*} \phi \\ & \mid \langle \leq n \rangle \phi \mid [\leq n]\phi \mid \langle > n \rangle \phi \mid [> n]\phi, \end{array} \tag{5}$$

where p ranges over atomic propositions, a over actions, and n over rational numbers. The symbols for propositions for *false*, *true*, *negation*, *disjunction*, *(additive) conjunction*, and *(additive) implication* are standard. The (*additive*) *modalities* are $\langle a \rangle$ and $[a]$. The connectives I, $*$, and $\mathrel{\circ\!\!-\!\!*}$ are the *multiplicative unit*, *conjunction*, and (a contextual version of) *multiplicative implication*, respectively. The implication $\mathrel{\circ\!\!-\!\!*}$ differs from the usual bunched $\mathrel{-\!\!*}$ in that the meanings of its constituent formulæ are given relative to each other's contexts, so capturing a degree of inter-dependency. The (*cost*) *modalities* are $\langle \leq n \rangle$, $[\leq n]$, $\langle > n \rangle$, $[> n]$, and denote possible and necessary modal bounds on costed evolutions.

$$
\begin{array}{lll}
C \models_{C'} p & \text{iff} & C \in \mathcal{V}(p) \\
C \models_{C'} \bot & & \text{never} \\
C \models_{C'} \top & & \text{always} \\
C \models_{C'} \neg\phi & \text{iff} & C \not\models_{C'} \phi \\
C \models_{C'} \phi \vee \psi & \text{iff} & C \models_{C'} \phi \text{ or } C \models_{C'} \psi \\
C \models_{C'} \phi \wedge \psi & \text{iff} & C \models_{C'} \phi \text{ and } C \models_{C'} \psi \\
C \models_{C'} \phi \rightarrow \psi & \text{iff} & C \models_{C'} \phi \text{ implies } C \models_{C'} \psi \\
C_1 \models_{C_2} \langle a \rangle \phi & \text{iff} & \text{for some } C_1', C_2', b \text{ such that} \\
& & C_1 \xrightarrow[C_2]{e,\mathbf{1}}{}^{a}\ C_1' \text{and } C_2 \xrightarrow[C_\emptyset]{C_1}{}^{b}\ C_2', \\
& & C_1' \models_{C_2'} \phi \\
C_1 \models_{C_2} [a]\phi & \text{iff} & \text{for all } C_1', C_2', b \text{ such that} \\
& & C_1 \xrightarrow[C_2]{e,\mathbf{1}}{}^{a}\ C_1' \text{ and } C_2 \xrightarrow[C_\emptyset]{C_1}{}^{b}\ C_2', \\
& & C_1' \models_{C_2'} \phi
\end{array}
$$

FIGURE 5. Interpretation of Additive Propositional formulæ

A *valuation*, $\mathcal{V}$, is a function that maps each atomic proposition to a $\sim$-closed set of closed contexts. $\bot$, $\top$, $\neg$, $\vee$, $\wedge$, and $\rightarrow$ are all interpreted (essentially) classically. The action modalities are interpreted normally (see, for example, [11, 14]) as possibility and necessity. The interpretation of the multiplicative connectives here is similar to that for the logic MBI [14]. In the interpretation of atoms, the surrounding context is wrapped around the primary context, and the valuation of the atom consulted to see if it contains this compound context. This is what makes the logic context-sensitive, rather than just having worlds with two parts.

The satisfaction relation for additive formulæ is specified in Figure 5 and that for multiplicative formulæ is specified in Figure 6.

$$
\begin{array}{ll}
R, E \models_{C'} I & \text{iff } R = e \text{ and } E \sim \mathbf{1} \\
\\
R, E \models_{C'} \phi * \psi & \text{iff there are } S, T, F, G \text{ such that if} \\
& \quad R = S \circ T,\ E \sim F \times G, \text{ then} \\
& \quad S, F \models_{C'(T,[\,]\times G)} \phi \text{ and } T, G \models_{C'(S,F\times[\,])} \psi \\
\\
R, E \models_{C'} \phi \circ\!\!\!-\!\!* \psi & \text{iff for all } S, F \text{ such that } R \circ S \text{ is defined,} \\
& \quad S, F \models_{C'(R,E\times[\,])} \phi \text{ and } R, E \models_{C'(S,F\times[\,])} \psi
\end{array}
$$

FIGURE 6. Interpretation of Multiplicative Propositional formulæ

The interpretation of cost modalities is straightforward. The possibility modalities $\langle \leq n \rangle \phi$ and $\langle > n \rangle \phi$ denote that there exists an evolution whose cost m is less than or equal to, or greater than, n, respectively, where the resulting state satisfies ϕ. The necessity modalities $[\leq n]$ and $[> n]$ denote that in all evolutions whose cost m is less than or equal to, or greater than, n, respectively, where the resulting state satisfies ϕ. The satisfaction relation for cost modalities is specified in Figure 7.

The standard interpretation of logics formulated in Hennessy–Milner-style uses the relation specified by the operational semantics as a Kripke structure to support the modalities. In our work, the operational semantics is more complex: a context occurs, and evolves alongside an outer context. Therefore, when we consider whether $C_1 \models_{C_2} \langle \leq n \rangle \phi$ holds, we have to consider whether there are evolutions of the form $C_1 \xRightarrow[C_2]{e,\mathbf{1}}{}^{m}\ C_1'$ *and* $C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o}\ C_2'$ such that $C_1' \models_{C_2'} \phi$ and $m \leq n$. The occurrence of the tick process and the empty context ensure that no extraneous contextual information is introduced into the evolutions of interest. This is what makes our logic context-sensitive. Other modal operators are interpreted similarly.

$$C_1 \models_{C_2} \langle \leq n\rangle\phi \quad \text{iff} \quad \text{for some } C'_1, C'_2, m, o \text{ such that } C_1 \xRightarrow[C_2]{e,\mathbf{1}}{}^{m} C'_1,\ C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C'_2, \text{ and } m \leq n, \; C'_1 \models_{C'_2} \phi$$

$$C_1 \models_{C_2} [\leq n]\phi \quad \text{iff} \quad \text{for all } C'_1, C'_2, m, o \text{ such that } C_1 \xRightarrow[C_2]{e,\mathbf{1}}{}^{m} C'_1,\ C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C'_2, \text{ and } m \leq n, \; C'_1 \models_{C'_2} \phi$$

$$C_1 \models_{C_2} \langle > n\rangle\phi \quad \text{iff} \quad \text{for some } C'_1, C'_2, m, o \text{ such that } C_1 \xRightarrow[C_2]{e,\mathbf{1}}{}^{m} C'_1,\ C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C'_2, \text{ and } m > n, \; C'_1 \models_{C'_2} \phi$$

$$C_1 \models_{C_2} [> n]\phi \quad \text{iff} \quad \text{for all } C'_1, C'_2, m, o \text{ such that } C_1 \xRightarrow[C_2]{e,\mathbf{1}}{}^{m} C'_1,\ C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C'_2, \text{ and } m > n, \; C'_1 \models_{C'_2} \phi$$

Figure 7. Interpretation of Propositional Cost Modalities

The logic admits the usual classical (or, if preferred, intuitionistic) propositional connectives, as well as thus usual 'separating' or 'resource-sensitive' multiplicatives from bunched logic [34], as in [14], and action modalities, as in Hennessy–Milner logic [20, 19, 33].

Behaviourally equivalent processes are also logically equivalent (they satisfy the same logical properties). This is half of the Hennessy–Milner property [20, 19, 33].

Theorem 7. *If $C_1 \models_{C_2} \phi$, and $C_1 \sim C_3$, and $C_2 \sim C_4$, then $C_3 \models_{C_4} \phi$.*

Proof. By induction over the derivation of of $C_1 \models_{C_2} \phi$.

- Case $C_1 \models_{C_2} p$. By the definition of $\mathcal{V}$ we have that if $C \sim C'$ and $C \in \mathcal{V}(p)$ then $C' \in \mathcal{V}(p)$. By Proposition 4, we have that $C_2(C_1) \sim C_4(C_3)$, and hence $C_4(C_3) \in \mathcal{V}(p)$.
- Case $C_1 \models_{C_2} \bot$. As the premisses assume $C_1 \models_{C_2} \phi$, we have a contradiction and can disregard this case.
- Case $C_1 \models_{C_2} \top$. We have that $C_3 \models_{C_4} \top$, straightforwardly.
- Case $C_1 \models_{C_2} \phi \wedge \psi$. By the induction hypothesis, we know that $C_3 \models_{C_4} \phi$ and $C_3 \models_{C_4} \psi$. Hence we have that $C_3 \models_{C_4} \phi \wedge \psi$.
- Case $C \models_{C'} \phi \vee \psi$. By the induction hypothesis, we know that $C_3 \models_{C_4} \phi$ or $C_3 \models_{C_4} \psi$. Hence we have that $C_3 \models_{C_4} \phi \vee \psi$.
- Case $C \models_{C'} \phi \rightarrow \psi$. By the induction hypothesis, we know that $C_3 \models_{C_4} \phi$ whenever $C_1 \models_{C_2} \phi$ and $C_3 \models_{C_4} \psi$ whenever $C_1 \models_{C_2} \psi$. Hence we have that $C_3 \models_{C_4} \phi \rightarrow \psi$.
- Case $C_1 \models_{C_2} \langle a\rangle\phi$. As there exist C'_1, C'_2, and b such that $C_1 \xrightarrow[C_2]{C_\emptyset}{}^{a} C'_1$ and $C_2 \xrightarrow[C_\emptyset]{C_1}{}^{b} C'_2$ and $C'_1 \models_{C'_2} \phi$, then, by the definition of bisimulation, we know that there exist C'_3 and C'_4 such that $C_3 \xrightarrow[C_4]{C_\emptyset}{}^{a} C'_3$ and $C_4 \xrightarrow[C_\emptyset]{C_3}{}^{b} C'_4$. By the induction hypothesis, we know that, as $C'_1 \models_{C'_2} \phi$, then $C'_3 \models_{C'_4} \phi$. Hence we have that $C_3 \models_{C_4} \langle a\rangle\phi$.
- Case $C_1 \models_{C_2} [a]\phi$. We have that, for all C'_1, C'_2, b such that $C_1 \xrightarrow[C_2]{C_\emptyset}{}^{a} C'_1$ and $C_2 \xrightarrow[C_\emptyset]{C_1}{}^{b} C'_2$, $C'_1 \models_{C'_2} \phi$. By the definition of bisimulation, we know that, for all C'_3 and C'_4 such that $C_3 \xrightarrow[C_4]{C_\emptyset}{}^{a} C'_3$ and $C_4 \xrightarrow[C_\emptyset]{C_3}{}^{b} C'_4$, $C'_1 \sim C'_3$ and $C'_2 \sim C'_4$. By the induction hypothesis, we know that, as $C'_1 \models_{C'_2} \phi$ then $C'_3 \models_{C'_4} \phi$. Hence we have that $C_3 \models_{C_4} [a]\phi$.
- Case $R, E \models_{C_2} I$. Let $C_3 = R, F$. By Theorem 5, we have that, as $E \sim \mathbf{1}$ and $E \sim F$, $E \sim \mathbf{1}$. Hence we have that $R, F \models_{C_4} I$.
- Case $R \circ S, E \models_{C_2} \phi * \psi$. Let $C_3 = R, H$. By Theorem 5, we have that as $E \sim H$ and $E \sim F \times G$ that hence $H \sim F \times G$. By the induction hypothesis, we have that $R, F \models_{C_2(S,[\,]\times G)} \phi$ and $S, G \models_{C_2(S,F\times[\,])}$

ψ. By Proposition 4, we have that $C_2(S, [\,] \times G) \sim C_4(S, [\,] \times G)$ and $C_2(S, F \times [\,]) \sim C_4(S, F \times [\,])$. Then, by the induction hypothesis, we have that $R, F \vDash_{C_4(S,[\,]\times G)} \phi$ and $S, G \vDash_{C_4(S,F\times[\,])} \psi$, and hence that $R \circ S, H \vDash_{C_4} \phi * \psi$.

- Case $R, E \vDash_{C_2} \phi \circ\!\!-\!\!* \psi$. Let $C_3 = R, G$. By the induction hypothesis, we have that if $S, F \vDash_{C_2} \phi$, then $S, F \vDash_{C_4} \phi$ and that $R \circ S, G \times F \vDash_{C_4} \psi$. Hence we have that $R, G \vDash_{C_4} \phi \circ\!\!-\!\!* \psi$.
- Case $C_1 \vDash_{C_2} \langle \leq n \rangle \phi$. As there exist C_1', C_2', and m such that $C_1 \xRightarrow[C_2]{C_\emptyset}{}^{m} C_1'$, $C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C_2'$, $C_1' \vDash_{C_2'} \phi$, and $m \leq n$, by the definition of bisimulation, we know that there exist C_3' and C_4' such that $C_3 \xRightarrow[C_4]{C_\emptyset}{}^{m} C_3'$, $C_4 \xRightarrow[C_\emptyset]{C_3}{}^{o} C_4'$, where $C_1' \sim C_3'$ and $C_2' \sim C_4'$. By the induction hypothesis, we know that, as $C_1' \vDash_{C_2'} \phi$ then $C_3' \vDash_{C_4'} \phi$. Hence we have that $C_3 \vDash_{C_4} \langle \leq n \rangle \phi$.
- Case $C_1 \vDash_{C_2} [\leq n]\phi$. We have that, for all C_1' and C_2' such that $C_1 \xRightarrow[C_2]{C_\emptyset}{}^{m} C_1'$, $C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C_2'$, and $m \leq n$, $C_1' \vDash_{C_2'} \phi$. By the definition of bisimulation, we know that, for all C_3' and C_4' such that $C_3 \xRightarrow[C_4]{C_\emptyset}{}^{m} C_3'$ and $C_4 \xRightarrow[C_\emptyset]{C_3}{}^{o} C_4'$, $C_1' \sim C_3'$ and $C_2' \sim C_4'$. By the induction hypothesis, we know that as $C_1' \vDash_{C_2'} \phi$, $C_3' \vDash_{C_4'} \phi$. Hence we have that $C_3 \vDash_{C_4} [a]\phi$.
- Case $C_1 \vDash_{C_2} \langle > n \rangle \phi$. As there exist C_1', C_2', and m such that if $C_1 \xRightarrow[C_2]{C_\emptyset}{}^{m} C_1'$ and $C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C_2'$, then $C_1' \vDash_{C_2'} \phi$, for $m > n$. By the definition of bisimulation, we know that there exist C_3' and C_4' such that $C_3 \xRightarrow[C_4]{C_\emptyset}{}^{m} C_3'$, $C_4 \xRightarrow[C_\emptyset]{C_3}{}^{o} C_4'$, where $C_1' \sim C_3'$ and $C_2' \sim C_4'$. By the induction hypothesis, we know that, as $C_1' \vDash_{C_2'} \phi$, we have $C_3' \vDash_{C_4'} \phi$. Hence we have that $C_3 \vDash_{C_4} \langle \leq n \rangle \phi$.
- Case $C_1 \vDash_{C_2} [> n]\phi$. We have that, for all C_1' and C_2' such that $C_1 \xRightarrow[C_2]{C_\emptyset}{}^{m} C_1'$ and $C_2 \xRightarrow[C_\emptyset]{C_1}{}^{o} C_2'$, and $m > n$, $C_1' \vDash_{C_2'} \phi$. By the definition of bisimulation, we know that there exist C_3' and C_4' such that $C_3 \xRightarrow[C_4]{C_\emptyset}{}^{m} C_3'$ and $C_4 \xRightarrow[C_\emptyset]{C_3}{}^{o} C_4'$, where $C_1' \sim C_3'$ and $C_2' \sim C_4'$. By the induction hypothesis, we know that, as $C_1' \vDash_{C_2'} \phi$, $C_3' \vDash_{C_4'} \phi$. Hence we have that $C_3 \vDash_{C_4} [a]\phi$.

□

So, bisimilar processes can be used interchangeably within a larger system, without changing the logical properties of the larger system.

It is unclear whether a useful converse can be obtained for the given, global, bisimulation relation. By restricting the logic to the fragment without $\circ\!\!-\!\!*$, and taking a different, *local*, equivalence relation, however, it is possible to obtain a converse [1, 2, 14] — the local equivalence fails, however, to be a congruence, and as such its usefulness is limited; it is a strictly local reasoning tool that supports decomposition but not, in general, composition.

To this end, we introduce the *local equivalence relation* $\approx\ \subseteq (OCont \times CCont) \times (OCont \times CCont)$, the largest binary relation such that, if $C_1, D_1 \approx C_2, D_2$ (with C_1' and D_1' , etc., modifying them, as usual), then

(1) for all $C_1' \in OCont, D_1' \in CCont$, $a, b \in$ Act, if $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C_1'$, and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D_1'$, then there exist $C_2' \in OCont$ and $D_2' \in CCont$ such that $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C_2'$, $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D_2'$, and $C_1', D_1' \approx C_2', D_2'$, and, for all $C_1' \in OCont, D_1' \in CCont, m, n$, if $C_1 \xRightarrow[C_\emptyset]{D_1}{}^{m} C_1'$, and $D_1 \xRightarrow[C_1]{e,\mathbf{1}}{}^{n} D_1'$, then there exist $C_2' \in OCont$ and $D_2' \in CCont$ such that $C_2 \xRightarrow[C_\emptyset]{D_2}{}^{m} C_2'$, $D_2 \xRightarrow[C_2]{e,\mathbf{1}}{}^{n} D_2'$, and $C_1', D_1' \approx C_2', D_2'$, and,

(2) for all $C_2' \in OCont$, $D_2' \in CCont$, $a, b \in$ Act, if $C_2' \in OCont$ and $D_2' \in CCont$ such that $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C_2'$, $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D_2'$, then there exist $C_1' \in OCont$ and $D_1' \in CCont$ such that $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C_1'$, and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D_1'$,

and $C_1', D_1' \approx C_2', D_2'$, and for all $C_2' \in \mathit{OCont}$, $D_2' \in \mathit{CCont}$, m, n, if $C_2 \, {\xRightarrow[C_\emptyset]{D_2}}^{m} \, C_2'$, $D_2 \, {\xRightarrow[C_2]{e,\mathbf{1}}}^{n} \, D_2'$, then there exist $C_1' \in \mathit{OCont}$ and $D_1' \in \mathit{CCont}$ such that $C_1 \, {\xRightarrow[C_\emptyset]{D_1}}^{m} \, C_1'$, and $D_1 \, {\xRightarrow[C_1]{e,\mathbf{1}}}^{n} \, D_1'$, and $C_1', D_1' \approx C_2', D_2'$, and

(3) $R = S$.

The union of any set of relations that satisfy these two conditions also satisfies these conditions, so the largest such relation is well-defined.

Essentially, this equivalence relation starts from the view that processes should be considered equivalent whenever they have the same behaviour given the same resources and similar context. The local equivalence relation fails to be a congruence, however, as it is not respected by the product constructor, ×, for processes [13]. Therefore, we do not have an analogue of Theorem 5 for local equivalence. (Note that, in [11, 13, 14], the equivalence corresponding to the equivalence $\sim$ taken here is referred to as the *global equivalence*.)

We can, however, obtain a version of the full Hennessy-Milner theorem, provided we restrict the logic to the fragment without $\multimapdotbothA\!\!*$. The need for this restriction arises from the failure of the local equivalence to be a congruence, because the satisfaction relation for $\circ\!\!-\!\!*$ requires that two subsystems be combined using $\times$.

Consider the fragment of the logic that excludes $\circ\!\!-\!\!*$. Assume that all atomic propositions are values as sets of contexts that are also closed under $\approx$. Alter the I and $*$ clauses of the interpretation so that

$$\begin{array}{lll} C \models_{C_1} I & \text{iff} & C_1, C \approx C_1, (e, \mathbf{1}) \\ C \models_{C_1} \phi * \psi & \text{iff} & \text{there exist } S, T \text{ and } F, G \text{ such that } C_1, C \approx C_1, (S \circ T, F \times G), \text{ and} \\ & & S, F \models_{C_2} \phi \text{ and } T, G \models_{C_3} \psi, \text{ where } C_2 = C'(T, [\,] \times G) \text{ and} \\ & & C_3 = C'(S, F \times [\,]). \end{array}$$

We also alter the valuation function so that it maps each atomic proposition to a $\approx$-closed set of closed contexts (i.e., if $C_1, D_1 \approx C_2, D_2$ and $C_1(D_1) \in \mathcal{V}$, then $C_2(D_2) \in \mathcal{V}$).

We define two contexts (with accompanying outer contexts) to be logically equivalent if they satisfy exactly the same set of logical statements; that is, $C_1, D_1 \equiv C_2, D_2$ if and only if, for all ϕ, $D_1 \vDash_{C_1} \phi$ iff $D_2 \vDash_{C_2} \phi$.

In the proof of the first part of the Hennessy–Milner property for global bisimulation (Theorem 7) we make explicit use of the fact that global bisimulation is a congruence. We also make implicit use of the sub-property of congruence that global bisimulation is an equivalence relation. We cannot obtain the first part of the Hennessy–Milner property for the full logic. In order to obtain the first part of the Hennessy–Milner property for the remainder of the logic, it is necessary, however, to show that local bisimulation is an equivalence relation.

Lemma 8. *The local equivalence $\approx$ is an equivalence relation; that is, it is reflexive, symmetric, and transitive.*

Proof. By induction on the structure of bisimulations. We give illustrative cases.

(1) Reflexive. Let $\mathcal{R} = \{((C_1, D_1), (C_1, D_1))\}$. The relation $\mathcal{R}$ is a local bisimulation if and only if, for all $(C_1, D_1) \mathcal{R} (C_2, D_2)$, the following holds: for all $C_1' \in \mathit{OCont}, D_1' \in \mathit{CCont}$, $a, b \in \mathrm{Act}$, if $C_1 \, {\xrightarrow[C_\emptyset]{D_1}}^{a} \, C_1'$, and $D_1 \, {\xrightarrow[C_1]{e,\mathbf{1}}}^{b} \, D_1'$, then there exist $C_2' \in \mathit{OCont}$ and $D_2' \in \mathit{CCont}$ such that $C_2 \, {\xrightarrow[C_\emptyset]{D_2}}^{a} \, C_2'$, $D_2 \, {\xrightarrow[C_2]{e,\mathbf{1}}}^{b} \, D_2'$, and $C_1', D_1' \approx C_2', D_2'$, and, for all $C_1' \in \mathit{OCont}, D_1' \in \mathit{CCont}$, m, n, if $C_1 \, {\xRightarrow[C_\emptyset]{D_1}}^{m} \, C_1'$, and $D_1 \, {\xRightarrow[C_1]{e,\mathbf{1}}}^{n} \, D_1'$, then there exist $C_2' \in \mathit{OCont}$ and $D_2' \in \mathit{CCont}$ such that $C_2 \, {\xRightarrow[C_\emptyset]{D_2}}^{m} \, C_2'$, $D_2 \, {\xRightarrow[C_2]{e,\mathbf{1}}}^{n} \, D_2'$, and $C_1', D_1' \approx C_2', D_2'$, and, for all $C_2' \in \mathit{OCont}$, $D_2' \in \mathit{CCont}$, $a, b \in \mathrm{Act}$, if $C_2' \in \mathit{OCont}$ and $D_2' \in \mathit{CCont}$ such that $C_2 \, {\xrightarrow[C_\emptyset]{D_2}}^{a} \, C_2'$, $D_2 \, {\xrightarrow[C_2]{e,\mathbf{1}}}^{b} \, D_2'$, then there exist $C_1' \in \mathit{OCont}$ and $D_1' \in \mathit{CCont}$ such that $C_1 \, {\xrightarrow[C_\emptyset]{D_1}}^{a} \, C_1'$, and $D_1 \, {\xrightarrow[C_1]{e,\mathbf{1}}}^{b} \, D_1'$, and $C_1', D_1' \approx C_2', D_2'$, and for all $C_2' \in \mathit{OCont}, D_2' \in \mathit{CCont}$,

m, n, if $C_2 \xRightarrow[C_\emptyset]{D_2}{}^{m} C'_2$ and $D_2 \xRightarrow[C_2]{e,\mathbf{1}}{}^{n} D'_2$, then there exist $C'_1 \in OCont$ and $D'_1 \in CCont$ such that $C_1 \xRightarrow[C_\emptyset]{D_1}{}^{m} C'_1, D_1 \xRightarrow[C_1]{e,\mathbf{1}}{}^{n} D'_1$, and $C'_1, D'_1 \approx C'_2, D'_2$, and $R = S$.

Consider some $(C_1, D_1) \mathcal{R} (C_2, D_2)$. By the definition of the relation, we know that $C_2 = C_1$ and $D_2 = D_1$. As any evolution of each of the processes C_1 and D_1 is decorated with the same outer and inner contexts ($C_\emptyset$ and D_1, and C_1 and $e, \mathbf{1}$, respectively), we immediately have reflexivity.

(2) Symmetric. Let $\mathcal{R} = \{((C_2, D_2), (C_1, D_1)) \mid C_1, D_1 \approx C_2, D_2\} \cup \approx$. If $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C'_2$, and $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D'_2$, then we need to show that $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C'_1$, and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D'_1$, where $C'_2, D'_2 \mathcal{R} C'_1, D'_1$. As $C_1, D_1 \approx C_2, D_2$, by the definition of local bisimulation, we have that $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C'_1$ and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D'_1$, where $C'_1, D'_1 \approx C'_2, D'_2$, and hence $C'_2, D'_2 \mathcal{R} C'_1, D'_1$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

(3) Transitive. Let $\mathcal{R} = \{((C_1, D_1), (C_3, D_3)) \mid C_1, D_1 \approx C_2, D_2$ and $C_2, D_2 \approx C_3, D_3\}$. If $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C'_1$, and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D'_1$, then we need to show that $C_3 \xrightarrow[C_\emptyset]{D_3}{}^{a} C'_3$, $D_3 \xrightarrow[C_3]{e,\mathbf{1}}{}^{b} D'_3$, where $C'_1, D'_1 \mathcal{R} C'_3, D'_3$. By the definition of local bisimulation, as $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C'_1$ and $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D'_1$, we have that $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C'_2$ and $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D'_2$, where $C'_1, D'_1 \approx C'_2, D'_2$, and, similarly, as $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C'_2$ and $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D'_2$, we have that $C_3 \xrightarrow[C_\emptyset]{D_3}{}^{a} C'_3$ and $D_3 \xrightarrow[C_3]{e,\mathbf{1}}{}^{b} D'_3$, where $C'_2, D'_2 \approx C'_3, D'_3$. As $C'_1, D'_1 \approx C'_2, D'_2$ and $C'_2, D'_2 \approx C'_3, D'_3$, we then have that $(C'_1, D'_1) \mathcal{R} (C'_3, D'_3)$. The other cases are similar. Hence $\mathcal{R}$ is closed and a bisimulation.

□

The following analogue of Theorem 7 then holds for the local equivalence:

Theorem 9. *If $C_1, D_1 \approx C_2, D_2$, then $C_1, D_1 \equiv C_2, D_2$.*

Proof. Straightforward, by induction over the definition of $D_1 \models_{C_1} \phi$, essentially following the proof of Theorem 7. The different cases are as follows:

- Case $D_1 \models_{C_1} p$. This is valid if and only if $C_1(D_1) \in \mathcal{V}(p)$. As $\mathcal{V}$ is $\approx$-closed and $C_1, D_1 \approx C_2, D_2$, we hence have that $C_2(D_2) \in \mathcal{V}(p)$;
- Case $D_1 \models_{C_1} I$. By Lemma 8, we have that, as $\approx$ is transitive, so $C_2, D_2 \approx C_1, (e, \mathbf{1})$, and hence that $D_2 \models_{C_2} I$;
- Case $D_1 \models_{C_1} \phi * \psi$. By the hypothesis, we have that there exist some S, T and F, G such that $C_1, D_1 \approx C_1, (S \circ T, F \times G)$, $S, F \models_{C_2} \phi$, and $T, G \models_{C_3} \psi$, where $C_2 = C'(T, [\,] \times G)$ and $C_3 = C'(S, F \times [\,])$. By Lemma 8, we have that $\approx$ is transitive, and hence that $C_2, D_2 \approx C_1, (S \circ T, F \times G)$, so that $D_2 \models_{C_2} \phi * \psi$.

□

We can now obtain, in the absence of multiplicative implication, a converse to Theorem 9.

Theorem 10. *Consider the modal logic without the multiplicative implication, $\multimap\!\!*$. If $C_1, D_1 \equiv C_2, D_2$, then $C_1, D_1 \approx C_2, D_2$.*

Proof. Suppose, for a contradiction, that the theorem is false. Then there must be some contexts C_1, D_1, C_2, D_2, with $C_1, D_1 \equiv C_2, D_2$ and, without loss of generality, some transition $C_1 \xrightarrow[C_\emptyset]{D_1}{}^{a} C'_1$, and transition $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{b} D'_1$ (or some $C_1 \xRightarrow[C_\emptyset]{D_1}{}^{m} C'_1$, and transition $D_1 \xrightarrow[C_1]{e,\mathbf{1}}{}^{n} D'_1$), for some C'_1 and D'_1 such that there is no C'_2 and D'_2 with both $C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C'_2$ and transition $D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D'_2$, (or $C_2 \xRightarrow[C_\emptyset]{D_2}{}^{m} C'_2$ and transition $D_2 \xRightarrow[C_2]{e,\mathbf{1}}{}^{n} D'_2$), and $C'_1, D'_1 \approx C'_2, D'_2$.

Let $\mathcal{B} = \{(C_2', D_2') \mid C_2 \xrightarrow[C_\emptyset]{D_2}{}^{a} C_2' \text{ and } D_2 \xrightarrow[C_2]{e,\mathbf{1}}{}^{b} D_2'\}$. If $\mathcal{B} = \emptyset$ then know that D_1 can do a b action and D_2 cannot, and we can hence show that $D_1 \vDash_{C_1} \langle b \rangle \top$ and $D_2 \nvDash_{C_2} \langle b \rangle \top$, which contradicts the hypothesis that $C_1, D_1 \equiv C_2, D_2$. Therefore $\mathcal{B}$ must be non-empty. Since D_2 is image finite (in context C_2) then we may enumerate the n elements as $(C_{2_1}', D_{2_1}'), \ldots, (C_{2_n}', D_{2_n}')$. Also, as, for each $i \in 1, \ldots, n$, $C_1', D_1' \not\equiv C_{2_i}', D_{2_i}'$ then there is some ϕ_i such that $D_1' \vDash_{C_1'} \phi_i$ and $D_{2_i}' \nvDash_{C_{2_i}'} \phi_i$. But then we can show that $D_1 \vDash_{C_1} \langle a \rangle (\phi_1 \wedge \ldots \wedge \phi_n)$ and $D_2 \nvDash_{C_2} \langle a \rangle (\phi_1 \wedge \ldots \wedge \phi_n)$, which contradicts the hypothesis that $C_1, D_1 \equiv C_2, D_2$. Therefore, $\mathcal{B}$ cannot be non-empty. A similar approach can be taken for$\Longrightarrow$ transitions. As $\mathcal{B}$ cannot be both empty and non-empty, our supposition must be false and we are done. □

To obtain stronger results of this type, with full congruence properties and a Hennessy–Milner equivalence theorem for the full logic with respect to local equivalence, it seems to be necessary to require a quite significant adjustment of our conceptual set-up.

A key component of our set-up is the co-evolution of resources and processes,

$$R, E \xrightarrow{a} R', E',$$

where the resource elements R are drawn from a resource monoid,

$$\mathbf{R} = (\mathbf{R}, \sqsubseteq, \circ, e).$$

Whilst resource monoids have proved very valuable in range of settings, including the bunched logic BI and its application to program verification via Separation Logic [24, 37], it seems that in the setting of a process algebra with both concurrent product and choice, a richer structure is required if the desired meta-theoretic properties are to be obtained. More specifically, we conjecture that it necessary for the underlying resource semantics to have, correspondingly, two combinators with appropriate properties. This foundational work will be pursued elsewhere.

4. Trust Domains

In our modelling approach, the decision of an agent to trust some other agents (a context) is modelled by a choice of reduction that interacts with the context. The behaviour that the agent performs within that interaction can be characterized using logical properties. We can then describe the notion of whether an agent trusts some context by whether or not it fulfils a given logical property. The decision to trust some context will, however, have some cost or risk associated with the trust, and we want to describe a given agent's risk appetite, at a given choice point. We can describe this cost-bound using the cost modalities, as introduced in Section 3.4.

We define the trust domain associated with an agent E, relative to available (implicitly located) resources R, precondition ϕ, required trust property, ψ, and cost-bound, n, as follows:

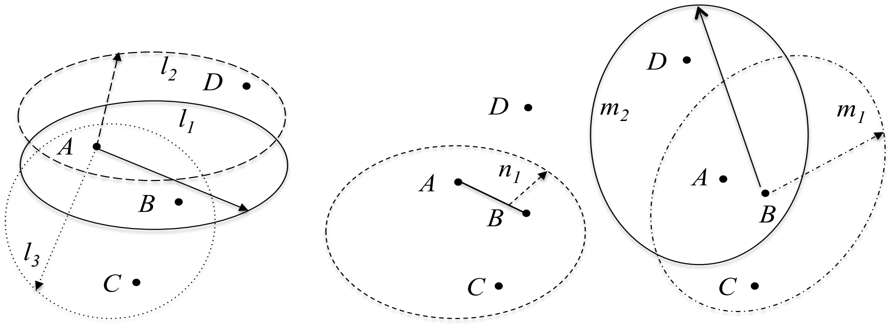
$$TD((R, E), \phi, \psi, n) \quad = \quad \{S, F \mid S, F \vDash_{R,E \times [\,]} \phi \text{ and } R, E \vDash_{S,F \times [\,]} \langle \leq n \rangle \psi\}. \tag{6}$$

The precondition ϕ limits the agent's contexts being considered; for example, imposing a locality condition. This definition enables us to talk about choice decisions that an agent makes under some, usually incomplete, knowledge about its surroundings.

For a brief technical justification of the definition, recall that multiplicative implication is valid if, for any context that fulfils the left hand side of the implication, when it is added to the current agent's context, the agent fulfils the right hand side of the implication. Essentially, a trust domain is the collection of such contexts for the logical property $\phi \multimapinv\!\!*\langle \leq n \rangle \psi$ interpreted with respect to the agent that is doing the trusting.

This approach is depicted schematically in the first diagram of Figure 8. Again, the diagram is intended to be understood in the context of the classical model of distributed systems (see, for example, [15, 14]) in which processes (here, agents) execute relative to collections of resources, located at specified places within the system. The system is understood as residing within an environment from which events are incident upon it and to which it exports events [15, 14].

In Figure 8 (for now, we assume a fixed trust property), the left-hand diagram shows the cost-bounds for agent A, which trusts agent B at cost-bound l_1, trusts agent D at cost-bound l_2, and trusts agent C at cost-bound l_3. The right-hand diagram shows the cost-bounds for agent B, which trusts agent A and C at the cost-bound m_1. The centre diagram shows the cost-bound n_1 for a combined agent $A \times B$, where we require that combined cost-bound $l_3 + m_1$ be less than the cost-bound n_1. In this example, D is trusted by



Left Centre Right

Figure 8. Cost Bounds and Trust Domains

A at cost-bound l_2, though not at l_1, and not trusted by B at m_1. D is not trusted by $A \times B$ at $l_1 + m_1$. Here, B accepts a cost (and uses its higher cost-bound) in order to 'convince' A to trust C. In order to do so, B, however, has to give up trusting D (as it cannot use its other higher m_2 cost-bound for D as well). This corresponds to the notion that $A \times B$ trusts C, provided B is accepts the risk with respect to trusting C. Similarly, in order to trust C together with B, agent A must use its cost bound l_3, giving up trusting either D or C on their own.

5. Trust Domain Combinators

Given the formal definition of trust domains, stated in the previous section, we can now explore how trust domains can be constructed and deconstructed. We introduce a running example. We then discuss several ways to construct trust domains with more complex logical properties, from those with simpler logical properties.

We consider negation, and then go on to simple combinators of additive conjunction and disjunction. We discuss additive implication, and show that it has the standard interpretation as the disjunction between the negation of the premiss, and the conclusion. We describe how to determine what is trusted by the combination of two agents together, with respect to what each trusts individually. Finally, we describe substitution for trust domains, with cases for the substitution of logical properties, cost-bounds, and agents.

5.1. **Running Example: Contract Choices.** To aid the discussion we make use of a running example, that of a mergers and acquisitions (M&A) deal team. As part of a M&A process, a bank provides a series of cost valuations of different parts of the companies that are being merged/acquired. The task requires access to very confidential details of the companies being valued. There are many specialized aspects of the valuation, however, and often appropriate analysts are not employed within the bank. In these cases the bank makes use of external specialist contractors. Given that the contractors have varying levels of security infrastructure, which are generally less efficacious than that of the bank itself, there is a very real risk of loss of data that is shared to contractors.

Consider a scenario (portrayed by the first diagram in Figure 8) where Alice's bank (A) has three potential contractors, each of which could be used to perform the analysis: Big Corp (B), Dad and Son's (D), and Fly By Night Ltd. (C). Big Corp can provide a generic valuation (logically represented by the property $\psi_1 = \langle \text{gen\_val} \rangle \top$), at a lower cost l_1. Dad and Son's can provide a specialized valuation (logically represented by the property $\psi_2 = \langle \text{spec\_val} \rangle \top$), but at a slightly higher cost l_2. Finally, Fly By Night Ltd. can also provide a specialized valuation (and is similarly logically represented). C are not trusted on its own at own at any cost level, as they are a new business and there is a risk they will renege on their commitments. Big Corp is a large enough company that it can provide dedicated consultants to go work on site at the bank, but the other two have very few consultants, and work from their offices, on multiple contracts at the same time. Hence Alice's bank must share its information on the company being valued off-site, when working with D or C (logically represented by the property $\psi_3 = \langle \text{share} \rangle \top$). In order to mitigate the risks associated with sharing information, the bank can coerce D and C into contracts with punitive repercussions if the data is leaked (logically represented by the property $\psi_4 = \langle \text{contract} \rangle \top$), but is unable to do so with B.

Note that, for these examples, agents A, B, and so on are contexts, and hence consist of both a process and a resource component. We now proceed to describe the combinator meta-theory, making use of this running example.

5.2. **Negation.** A trust domain describes whether an agent is *able* to fulfil a property within a cost-bound. Conversely, we may wish to ensure that an agent is *not able* to fulfil a property within a cost-bound

$$\neg TD((R,E),\phi,\psi,n) = \{V,I \mid V,I \vDash_{R,E\times[\,]} \phi_1 \text{ and } R,E \vDash_{U,H\times[\,]} \neg\langle \leq n\rangle\psi\}. \tag{7}$$

Typically, classical possibility and necessity modalities are dual: in our present setting, having an agent fulfilling the property $\neg\langle \leq k\rangle\psi$ would be equivalent to having that an agent fulfilling the property $[< k]\neg\psi$. In order to obtain this duality, we define negation on trust domains as

$$TDN((R,E),\phi,\psi,n) = \{S,F \mid S,F \vDash_{R,E} \phi \text{ and } R,E \vDash_{S,F} [< n]\,\psi\}. \tag{8}$$

Trust domain negation is used to reason about properties that we want to always hold; such as that an individual is never trusted, or that an agent always avoids data leaks.

We can represent that an agent is not able to fulfil a property within a given cost-bound, as in Equation 7, using a trust domain negation, as in Equation 8, and vice versa.

Lemma 11.
(1) $\neg TD((R,E),\phi,\psi,n) = TDN((R,E),\phi,\neg\psi,n)$.
(2) $\neg TDN((R,E),\phi,\psi,n) = TD((R,E),\phi,\neg\psi,n)$.

Proof.
(1)
$$\begin{aligned}\neg TD((R,E),\phi,\psi,n) &= \{V,I \mid V,I \vDash_{R,E\times[\,]} \phi_1 \text{ and } R,E \vDash_{U,H\times[\,]} \neg\langle \leq n\rangle\psi\}\\ &= \{V,I \mid V,I \vDash_{R,E\times[\,]} \phi_1 \text{ and } R,E \vDash_{U,H\times[\,]} [\leq n]\neg\psi\}\\ &= TDN((R,E),\phi,\neg\psi,n)\end{aligned}$$
(2)
$$\begin{aligned}\neg TDN((R,E),\phi,\psi,n) &= \{S,F \mid S,F \vDash_{R,E} \phi \text{ and } R,E \vDash_{S,F} [\leq n]\,\psi\}\\ &= \{S,F \mid S,F \vDash_{R,E} \phi \text{ and } R,E \vDash_{S,F} \neg\langle \leq n\rangle\,\neg\psi\}\\ &= TD((R,E),\phi,\neg\psi,n).\end{aligned}$$

□

Lemma 12. *A trust domain and its negation are disjoint.*

Proof.

$$\begin{aligned}&TD((R,E),\phi,\psi,n) \cap \neg TD((R,E),\phi,\psi,n)\\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } \ R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle\,\psi\} \cap \{V,I \mid V,I \vDash_{R,E\times[\,]} \phi_1 \text{ and } R,E \vDash_{U,H\times[\,]} \neg\langle \leq n\rangle\psi\}.\\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } \ R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle\,\psi \text{ and } R,E \vDash_{S,F\times[\,]} \neg\langle \leq n\rangle\psi\}\\ &\quad = \emptyset.\end{aligned}$$

□

An example of trust domain negation is in the discussion of additive implication below. We recover the expected property of the conjunct of a property with its negation

$$TD((R,E),\phi,\psi,n) \cap \neg TD((R,E),\phi,\psi,n) = \emptyset. \tag{9}$$

5.3. **Additive Disjunction.** The bank may be interested in either the generalized valuation, or it may be interested in the specialized valuation. The former can be represented using the trust domain $TD_1 = TD(A,\top,\langle \text{gen\_val}\rangle\top,l_1)$, and the latter using $TD_2 = TD(A,\top,\langle \text{spec\_val}\rangle\top,l_2)$. In this case, Alice will trust B to do the generalized valuation ($TD_1 = \{\ B \times [\,]\ \}$). If $l_2 < l_3$, then Alice will trust D but not C to do the specialized valuation ($TD_2 = \{\ D \times [\,]\ \}$).

The bank may, however, be indifferent to whether or not the valuation is specialized. This can be represented by the logical property $\langle \text{gen\_val}\rangle\top \vee \langle \text{spec\_val}\rangle\top$. If $l_1 = l_2$, the trust domain for this disjunction $TD_3 = TD(A,\top,\psi_1 \vee \psi_2,l_1) = \{\ B \times [\,], D \times [\,]\ \}$ is simply the set union of trust domains TD_1 and TD_2.

In general, we have the following, with symmetric properties for disjunction of the preconditions:

Lemma 13.

(1) $TD((R,E),\phi,\psi_1 \vee \psi_2, n) = TD((R,E),\phi,\psi_1,n) \ \cup \ TD((R,E),\phi,\psi_2,n)$.
(2) $TDN((R,E),\phi,\psi_1 \vee \psi_2, n) = TDN((R,E),\phi,\psi_1,n) \ \cup \ TDN((R,E),\phi,\psi_2,n)$.

Proof.

(1)

$$\begin{aligned} TD((R,E),\phi,\psi_1 \vee \psi_2,n) &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_1 \vee \psi_2\} \\ &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } (R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_1 \text{ or } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_2)\} \\ &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_2\} \cup \\ &\ \ \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_1\} \\ &= TD((R,E),\phi,\psi_1,n) \ \cup \ TD((R,E),\phi,\psi_2,n) \end{aligned}$$

(2)

$$\begin{aligned} & TDN((R,E),\phi,\psi_1 \vee \psi_2,n) \\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} [\leq n] \psi_1 \vee \psi_2\} \\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and for all } C_1',C_2',m,o.{C_1 \xRightarrow[C_2]{e,\mathbf{1}}}^{m} C_1', {C_2 \xRightarrow[C_\emptyset]{C_1}}^{o} C_2', \text{ and } m \leq n \\ &\qquad\quad C_1' \vDash_{C_2'} \psi_1 \vee \psi_2\} \\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and for all } C_1',C_2',m,o.{C_1 \xRightarrow[C_2]{e,\mathbf{1}}}^{m} C_1', {C_2 \xRightarrow[C_\emptyset]{C_1}}^{o} C_2', \text{ and } m \leq n, \\ &\qquad\quad C_1' \vDash_{C_2'} \psi_1 \text{ or } C_1' \vDash_{C_2'} \psi_2\} \\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and for all } C_1',C_2',m,o.{C_1 \xRightarrow[C_2]{e,\mathbf{1}}}^{m} C_1', {C_2 \xRightarrow[C_\emptyset]{C_1}}^{o} C_2', \text{ and } m \leq n, \\ &\qquad\quad C_1' \vDash_{C_2'} \psi_1\} \cup \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and for all } C_1',C_2',m,o.{C_1 \xRightarrow[C_2]{e,\mathbf{1}}}^{m} C_1', {C_2 \xRightarrow[C_\emptyset]{C_1}}^{o} C_2', \\ &\qquad\quad \text{and } m \leq n, C_1' \vDash_{C_2'} \psi_2\} \\ &\quad = \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} [\leq n] \psi_1\} \cup \\ &\qquad\quad \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} [\leq n] \psi_2\} \\ &\quad = TDN((R,E),\phi,\psi_1,n) \ \cup \ TDN((R,E),\phi,\psi_2,n), \end{aligned}$$

□

5.4. **Additive Implication.** If the data necessary for the valuation is analysed off site, then there is a higher chance of it being leaked. As such, we may wish to ensure that the logical property $\psi_3 = \langle \text{share} \rangle \top$ does not hold. The risk can be mitigated, however, by using contracts (represented by $\psi_4 = \langle \text{contract} \rangle \top$ holding) that include punitive costs in the case that information is leaked; this money can then be used to cover legal costs, work to reduce reputational damage, and so on. This requirement can be represented using the logical implication $\psi_3 \rightarrow \psi_4$. If $l_1 = l_2$, the trust domain for this property, $TD_4 = TD(A, \top, \psi_3 \rightarrow \psi_4, l_2) = \{ \ B \times [\,], D \times [\,] \ \}$, can be interpreted as those contexts where either there is no choice within the cost-bound such that the information is shared (a trust domain negation), combined by set union with those where there is a contract available.

In general, we have the following:

Lemma 14.

$$TD((R,E),\phi,\psi_1 \rightarrow \psi_2,n) = TDN((R,E),\phi,\neg\psi_1,n) \ \cup \ TD((R,E),\phi,\psi_2,n).$$

Proof.

$$\begin{aligned} TD((R,E),\phi,\psi_1 \rightarrow \psi_2,n) &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_1 \rightarrow \psi_2\} \\ &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \neg\psi_1 \vee \psi_2\} \\ &= \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \neg\psi_1\} \cup \\ &\ \ \{S,F \mid S,F \vDash_{R,E\times[\,]} \phi \text{ and } R,E \vDash_{S,F\times[\,]} \langle \leq n\rangle \psi_2\} \\ &= TD((R,E),\phi,\neg\psi_1,n) \ \cup \ TD((R,E),\phi,\psi_2,n). \end{aligned}$$

□

5.5. **Additive Conjunction.** The bank is interested in contractors that can both perform a relevant analysis, and can either perform it without sharing (i.e., on site) or admits a strong contract. This can be represented by the logical property $\langle \text{spec\_val} \rangle \top \wedge (\psi_3 \to \psi_4)$. We make use of trust domains TD_1 and TD_4 from the previous sections. If $l_1 = l_2$, the trust domain for this conjunction $TD_5 = TD(A, \top, \psi_1 \wedge (\psi_3 \to \psi_4), l_2) = \{ \ B \times [\,] \ \}$ is simply the set intersection of trust domains TD_1 and TD_4.

In general, we have the following, with symmetric properties for conjunction of the preconditions:

Lemma 15.

(1) $TD((R, E), \phi, \psi_1 \wedge \psi_2, n) \subseteq TD((R, E), \phi, \psi_1, n) \ \cap \ TD((R, E), \phi, \psi_2, n)$

(2) $TDN((R, E), \phi, \psi_1 \wedge \psi_2, n) = TDN((R, E), \phi, \psi_1, n) \ \cap \ TDN((R, E), \phi, \psi_2, n)$.

Proof.

(1)

$$\begin{aligned}
&TD((R, E), \phi, \psi_1 \wedge \psi_2, n) \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and } R, E \vDash_{S,F\times[\,]} \langle \leq n \rangle \psi_1 \wedge \psi_2\} \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for some } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1 \wedge \psi_2\}\} \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for some } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1 \text{ and } C'_1 \vDash_{C'_2} \psi_2\} \\
&\quad \subseteq \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for some } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1\} \cap \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for some } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \\
&\qquad \text{and } m \leq n, C'_1 \vDash_{C'_2} \psi_2\} \\
&\quad = TD((R, E), \phi, \psi_1, n) \ \cap \ TD((R, E), \phi, \psi_2, n)
\end{aligned}$$

(2)

$$\begin{aligned}
&TDN((R, E), \phi, \psi_1 \wedge \psi_2, n) \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and } R, E \vDash_{S,F\times[\,]} [\leq n] \psi_1 \wedge \psi_2\} \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for all } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1 \wedge \psi_2\}\} \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for all } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1 \text{ and } C'_1 \vDash_{C'_2} \psi_2\} \\
&\quad = \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for all } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \text{ and } m \leq n, \\
&\qquad\quad C'_1 \vDash_{C'_2} \psi_1\} \cap \{S, F \mid S, F \vDash_{R,E\times[\,]} \phi \text{ and, for some } C'_1, C'_2, m, o \text{ such that } C_1 \overset{e,\mathbf{1}}{\underset{C_2}{\Longrightarrow}}{}^{m} C'_1, C_2 \overset{C_1}{\underset{C_\emptyset}{\Longrightarrow}}{}^{o} C'_2, \\
&\qquad \text{and } m \leq n, C'_1 \vDash_{C'_2} \psi_2\} \\
&\quad = TDN((R, E), \phi, \psi_1, n) \ \cap \ TDN((R, E), \phi, \psi_2, n),
\end{aligned}$$

□

The former is not equality because TD_1 denotes that agent A can make a choice and fulfil property ψ_1, and TD_2 denotes that an agent A can make a choice and fulfil property ψ_2. There is no guarantee they are the *same* choice.

5.6. **Multiplicative Conjunction.** So far our analysis has not been very favourable towards Fly By Night Ltd. The risk associated with them, by Alice's bank, is high, and as such they haven't managed to make it into any of the bank's trust domains. If Fly By Night Ltd. is a pet project of a senior individual of Big Corp, however, B may have a much lower cost associated with interacting with C than A does. In fact, Big Corp may be willing to accept the tender from Alice's bank and subcontract part of it out to Fly By Night Ltd.

Recall the cost-bounds introduced in the second and third diagrams of Figure 8; let these represent the bounds of the joint system of Alice's Bank working with Big Corp and the bounds of Big Corp, respectively.

Consider the case where $l_3 + m_2 < n_1$. In the previously considered trust domains, TD_1 and TD_2, C was never trusted. Here we can trade off the higher cost associated with A trusting C with the lower cost associated with B trusting C.

Consider a trust domain for Alice's bank $TD_6 = TD(A, \top, (\psi_1 \vee \psi_2), l_3) = \{\ D \times [\,], B \times [\,], B \times C \times [\,]\ \}$, where $\psi_1 \vee \psi_2$ denotes that the bank is interested in either a general or a specialized analysis, but doesn't care which. Consider also a trust domain for Big Corp $TD_7 = TD(B, \top, \psi_5, m_1) = \{\ A \times C \times [\,]\ \}$, where ψ_5 denote some logical property that Big Corp is interested in. We want to show that the trust domain for the combined agents is $TD_8 = \{\ C \times [\,]\ \}$. Note that TD_6 contains the context $B \times F \times [\,]$, and that TD_7 contains the context $A \times C \times [\,]$. When constructing the trust domain of a combined agent, from the trust domains of the sub agents, we can do so by finding the contexts that differ only in the first agent's context containing the second agent, and the second agent's context containing the first agent. If we strip each of the agents from the context (B from $B \times C \times [\,]$, A from $A \times C \times [\,]$) then we get the expected context $C \times [\,]$.

In general, where $A = R, E$, where $B = S, F$, and where $C = R \circ S, E \times F$, we have the following:

Lemma 16.

$$
\begin{aligned}
TD(C, \phi, (\psi_1 * \psi_2), n) \quad = \quad \bigcup_{n_1+n_2=n} & \{W, J \mid S \circ W, F \times J \in TD(A, (\phi * p_1), \psi_1, n_1) \\
& \textit{and}\ R \circ W, E \times J \in TD(B, (\phi * p_2), \psi_2, n_2)\}
\end{aligned}
$$

Proof.

$$
\begin{aligned}
TD(R \circ S, E \times F, \phi, (\psi_1 * \psi_2), n) &= \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and } R \circ S, E \times F \vDash_{V, I \times [\,]} \langle \leq n \rangle\, \psi_1 * \psi_2\} \\
&= \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and, for some } G', C'_2, m, o \text{ such that } R \circ S, E \times F \xRightarrow[V,I]{e,\mathbf{1}}{}^{m}\ R \circ S, G' \\
&\qquad \text{and } V, I \xRightarrow[C_\emptyset]{R \circ S, E \times F}{}^{o}\ C'_2, \text{ and } m \leq n, R \circ S, G' \vDash_{C'_2} \psi_1 * \psi_2\} \\
&= \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and, for some } G', C'_2, m, o \text{ such that } R \circ S, E \xRightarrow[V,I]{e,\mathbf{1}}{}^{m}\ R \circ S, G', \\
&\qquad V, I \xRightarrow[C_\emptyset]{R \circ S, E \times F}{}^{o}\ C'_2, \text{ and } m \leq n, \text{ there exist } E', F' \text{ such that } G' \sim E' \times F', \\
&\qquad R, E' \vDash_{C'_2(S, [\,] \times F')} \psi_1 \text{ and } S, F' \vDash_{C'_2(R, E' \times [\,])} \psi_2\} \\
&= \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and, for some } E', F', C'_2, m_1, m_2, o \text{ such that} \\
&\qquad R, E \xRightarrow[C_2(S, [\,] \times F)]{e,\mathbf{1}}{}^{m_1}\ R, E', \quad S, F \xRightarrow[C_2(R, E \times [\,])]{e,\mathbf{1}}{}^{m_2}\ S, F', \quad m_1 + m_2 \leq n, \\
&\qquad V, I \xRightarrow[C_\emptyset]{R \circ S, E \circ F}{}^{o}\ C'_2, \quad R, E' \vDash_{C'_2(S, [\,] \times F')} \psi_1 \text{ and } S, F' \vDash_{C'_2(R, E' \times [\,])} \psi_2\} \\
&= \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and, for some } m_1, m_2 \text{ such that } m_1 + m_2 \leq n, \\
&\qquad R, E \vDash_{S \circ V, F \times I \times [\,]} \langle \leq m_1 \rangle \psi_1 \text{ and } S, F \vDash_{R \circ V, E \times I \times [\,]} \langle \leq m_2 \rangle \psi_2\} \\
&= \bigcup_{m_1 + m_2 \leq n} \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and } R, E \vDash_{S \circ V, F \times I \times [\,]} \langle \leq m_1 \rangle \psi_1 \text{ and } S, F \vDash_{R \circ V, E \times I \times [\,]} \langle \leq m_2 \rangle \psi_2\} \\
&= \bigcup_{m_1 + m_2 \leq n} \{V, I \mid V, I \vDash_{R \circ S, E \times F \times [\,]} \phi \text{ and } S, F \vDash_{R \circ V, E \times I \times [\,]} p_1, \quad R, E \vDash_{S \circ V, F \times I \times [\,]} p_2, \\
&\qquad\qquad R, E \vDash_{S \circ V, F \times I \times [\,]} \langle \leq m_1 \rangle \psi_1, \text{ and } S, F \vDash_{R \circ V, E \times I \times [\,]} \langle \leq m_2 \rangle \psi_2\} \\
&= \bigcup_{m_1 + m_2 \leq n} \{V, I \mid T \circ V, G \times I \in \{R \circ V, E \times I \mid R, E \vDash_{S \circ T, F \times G \times [\,]} \phi, \ R, E \vDash_{S \circ V, F \times I \times [\,]} p_2, \\
&\qquad\qquad \text{and } S, F \vDash_{R \circ V, E \times I \times [\,]} \langle \leq n_2 \rangle\, \psi_2\} \text{ and} \\
&\qquad\qquad U \circ V, H \times I \in \{R \circ V, E \times I \mid R, E \vDash_{S \circ T, F \times G \times [\,]} \phi \text{ and } R, E \vDash_{S \circ V, F \times I \times [\,]} p_2 \\
&\qquad\qquad \text{and } S, F \vDash_{R \circ V, E \times I \times [\,]} \langle \leq n_2 \rangle\, \psi_2\}\}. \\
&= \bigcup_{m_1 + m_2 \leq n} \{V, I \mid T \circ V, G \times I \in TD(A, (\phi * p_1), \psi_1, n_1) \text{ and } U \circ V, H \times I \in TD(B, (\phi * p_2), \psi_2, n_2)\}.
\end{aligned}
$$

□

There are a few technicalities of note in this definition. Firstly, we consider any cost-bounds n_1 and n_2, associated with the agents R, E and S, F, that sum to the cost-bound n of the joint agents' trust domain. Secondly, in the sub trust domains $TD((R, E), (\phi\ p_1), \psi_1, n_1)$ and $TD((S, F), (\phi * p_2), \psi_2, n_2)$ are defined with extended preconditions $(\phi * p_1)$ and $(\phi * p_2)$ respectively. The atomic proposition p_1 characterizes agent S, F, and p_2 characterizes agent R, E; this is used to permit us to strip off the second agent from the first agent's context, and vice versa, as described above.

$$\frac{}{\bot \preccurlyeq \phi} \qquad \frac{}{\phi \preccurlyeq \top} \qquad \frac{}{\phi \wedge \psi \preccurlyeq \phi} \qquad \frac{}{\phi \preccurlyeq \phi} \qquad \frac{\psi_i \preccurlyeq \phi_i}{o(\overrightarrow{\psi_i}) \preccurlyeq o(\overrightarrow{\phi_i})}$$

where $o(\overrightarrow{\psi_i})$ generalizes the logical operators in the language as functions of vectors of sub-formulæ.

Figure 9. Logical Strengthening Relation

There does not appear to be any obvious trust domain combinator based on multiplicative implication. We hypothesize that this is because the trust domain definition itself is essentially that of a multiplicative implication, and hence no additional information can be brought to bear through the inclusion of another.

6. Trust Domain Substitution

The notion of substitution is used when moving from a more skeletal to a more fleshed out model, using more detailed descriptions of trust domains in place of more general ones, without having to reconsider the results obtained from the more abstract model.

In order to explain notions of substitution, we make use of a running example based around hospital staffing. We consider several different notions of substitution. For some trust domain $TD((R, E), \phi, \psi, n)$, we describe how substitution of trust domains corresponds to substituting each of the parameters of its definition, namely: substituting logical properties; substituting cost-bounds; and substituting agents. We describe how substitution interacts with the combinators discussed in the previous section.

6.1. Running Example: Hospital Staffing.

To aid the discussion we make use of a running example, that of staffing within a hospital. We consider a small hospital, with limited staff. The surgeon is supported in the operating theatre by a generalist nurse. We denote the skill of the surgeon by ψ_{su}, and those of the nurse by ϕ_{nu}.

Consider the situation when the nurse is absented, with little notice. The hospital administrators have several possible choices: they can continue with reduced staff, which has financial cost associated with cancelling appointments; they can transfer patients to other hospitals, which has reputational cost; or they can employ substitute a nurse from the general workforce.

Consider further the case where the local workforce consists of specialized nurses, but no generalized nurses. These specialized nurses will cost more to employ, and hence the choice to employ replacement nurses has associated financial cost, beyond that normally incurred by employing the generalist nurse. The specialist nurses' skills are denoted by the logical properties ϕ_{sn}.

6.2. Logical Property Substitution.

The surgeon A, if supported by a suitably qualified nurse (irregardless of the rest of the environment), can perform their job within certain cost-bounds. This defines a trust domain $TD(A, \top * \phi_{nu}, \psi_{su}, n_{su})$, referred to as TD_1. As a specialist nurse also has the skills of a generalist nurse, the nurse fulfils both the specialized property ϕ_{sn} and the generalized property ϕ_{nu}. Intuitively, a trust domain based on the general nursing skills should be replaceable by a trust domain based on the heart and lungs nursing skills, which include the general skills. Alternatively, we can write this as

$$TD(A, \top * \phi_{sn} \wedge \phi_{nu}, \psi_{su}, n_{su}) \subseteq TD(A, \top * \phi_{nu}, \psi_{su}, n_{su}). \tag{10}$$

If either of the logical properties used in the definition of a trust domain are strengthened, whilst the other trust domain parameters are kept constant, then the resulting trust domain is a subset of the original trust domain. We formalize what we mean by strengthening in Figure 9; intuitively, when strengthening a formula, any sub-formula may be replaced by a conjunction including that sub-formula, and top may be replaced by any other formula. We can show that, for all formulæ ϕ and ψ, and for all contexts C_1 and C_2, that if $\psi \preccurlyeq \phi$ then $C_1 \vDash_{C_2} \psi$ implies $C_1 \vDash_{C_2} \phi$. Then we have that, if $\phi' \preccurlyeq \phi$

$$\begin{array}{l} TD((R, E), \phi', \psi, n) \subseteq TD((R, E), \phi, \psi, n) \\ \text{and} \\ TDN((R, E), \phi', \psi, n) \subseteq TDN((R, E), \phi, \psi, n). \end{array} \tag{11}$$

As any instance of $\top$ can be strengthened to any other logical formula, we can replace top with a multiplicative formula. This allows us to specify the presence and properties of additional agents in the trusted ecosystem.

When we build a model of hospital staffing decisions, we can model the administrators to require the skills of the surgeons, but to make no requirements with respect to the support staff; the necessary skills can be more effectively specified by the relevant surgeons.

The administrator AD's trust domain can be written as

$$TD_2 = TD(AD, (\top * \langle \leq n_{su} \rangle \phi_{su}) \,,\ \phi_{ad}, n_{ad}).$$

This states that, in the presence of a surgeon who can make a trust decision and then perform the relevant duties, the administrators can achieve their goals. The properties of the agents that can be trusted by the surgeon are left loosely defined.

If the surgeon is composed with any context in the surgeon's trust domain TD_1, then together they satisfy the administrators' precondition $\phi_{ad} = (\top * \langle \leq n_{su} \rangle \phi_{su})$. The administrator's trust domain can hence be defined in terms of a loosely specified surgeon's trust domain $TD_3 = TD(A, \top, \psi_{su}, n_{su})$, here

$$TD(AD, \phi_{ad}, \psi_{ad}, n_{ad}) \ \supseteq\ \{C \times A \mid C \in TD_3 \text{ and } AD \vDash_{C \times A} \langle n_{ad} \rangle \psi_{ad}\}. \tag{12}$$

Our previous trust domain for the heart and lungs surgeon TD_1 was defined as $TD(A, \top * \phi_{nu}, \psi_{su}, n_{su})$. According to Equation 11 we can substitute contexts in TD_1 for those in TD_3, and hence show that

$$TD(AD, \phi_{ad}, \psi_{ad}, n_{ad}) \ \supseteq\ \{C \times A \mid C \in TD_1 \quad \text{and} \quad AD \vDash_{C \times A} \langle n_{ad} \rangle \psi_{ad}\}. \tag{13}$$

6.3. **Cost-bound Substitution.** Two possible options, when the general nurse is absented, are the transfer of patients to other hospitals, which has reputational cost, and the employment of substitute nurses. We represent the transfer option with the formula ψ^1_{ad}, and the substitute option with the formula ψ^2_{ad}. The acceptable cost for the different cases may, however, be different.

In Section 5.3 we describe how to combine two trust domains with the same pre-condition and cost-bound, and different trust properties. Similarly, we can consider trust domains with different trust bounds. If the first option is held to a tighter cost-bound (i.e., that $n < n_{ad}$) then we can show that

$$TD(AD, \phi_{ad}, \psi^1_{ad}, n) \ \cup\ TD(AD, \phi_{ad}, \psi^2_{ad}, n_{ad}) \ \subseteq\ TD(AD, \phi_{ad}, \psi^1_{ad} \vee \psi^2_{ad}, n_{ad}) \tag{14}$$

In general, if we strengthen the cost-bound associated of a trust domain, then the resulting trust domain is a subset of the original. Formally, if $n' \leq n$ then

$$\begin{gathered} TD((R, E), \phi, \psi, n') \subseteq TD((R, E), \phi, \psi, n) \\ \text{and} \\ TDN((R, E), \phi, \psi, n') \subseteq TDN((R, E), \phi, \psi, n). \end{gathered} \tag{15}$$

6.4. **Agent Substitution.** One of the advantages of our mathematical modelling approach is the ability to reason equationally about processes' behaviour. Bisimulation describes processes that can make the same actions, and the same choices with the same costs, but that are structurally different. If two processes are bisimilar, then, by Theorem 7, they fulfil the same logical properties. Given this fact, we can substitute bisimilar processes ($E \sim F$) in the trust domain definitions, and obtain equivalent trust domains:

$$\begin{gathered} TD((R, E), \phi, \psi, n) \ =\ TD((R, F), \phi, \psi, n) \\ \text{and} \\ TDN((R, E), \phi, \psi, n) \ =\ TDN((R, F), \phi, \psi, n). \end{gathered} \tag{16}$$

7. Discussion

We have considered how to model trust in complex systems of interacting agents, describing the underlying mathematical framework, based in models or location, resource, and process, and a substructural modal process logic of utility. The logic we consider contains both action and cost modalities, and additive and multiplicative connectives. Using this framework, we define the notion of trust domain in terms of the boundary established by the combination of a requisite logical property and an associated cost. As we have explained, in Section 3.4, theoretical improvements to the relationship between local equivalence of processes and logical equivalence should be obtained.

We have defined several combinators for trust domains corresponding to connectives of the process logic, giving set-theoretic operations on trust domains. Using these combinators, we show how to represent notions of transitivity and trade-offs through use of the multiplicative conduction operator. Additionally, we consider how to perform substitution of trust domains, in terms of each of the possible components of a trust domain. We illustrate the meta-theory of trust domains using two examples: contract choices in a mergers and acquisitions process; and staffing concerns in a hospital environment. A richer logic — perhaps with multiplicative modalities and both additive and multiplicative quantifiers, as described in [11, 13, 14] — might yield useful additional combinators. This possibility should be explored.

Using multiplicative conjunction, we have shown how a conjoined system's costs can be split up between its various sub-systems. This provides preliminary insight into how the logical formulæ are associated with costs. The relationship between the structure of resource–process models and the costs associated with them, and the relationship between the structure of logical formulæ and the costs associated with them, is of significant interest, and warrants significant further study.

Markov chains support reasoning about complex notions such as average utility with a given time discount (e.g., [26]), but do not provide compositionality results over model structures. The calculus presented here could be extended include probabilistic evolution and expected cost [28], thereby enriching the notion of trust domain with the ability to accommodate stochastic environments. We aim to use our modelling approach in larger scale case studies, such as [6, 9], where costly activities such as hardware-supported and managed security are considered.

Acknowledgements. We are grateful to Matthew Collinson for his advice and support in the writing of this paper.

References

[1] G. Anderson, M. Collinson, and D. Pym. Utility-based Decision-making in Distributed Systems Modelling. In Proc. TARK 2013, Burkhard C. Schipper (editor), Chennai, 2013. Computing Research Repository (CoRR): http://arxiv.org/corr/home. ISBN: 978-0-615-74716-3.

[2] G. Anderson, M. Collinson, and D. Pym. Trust Domains: An Algebraic, Logical, and Utility-theoretic Approach. In *Proc. 6th TRUST*, 2013. *Trust and Trustworthy Computing*, Michael Huth, N. Asokan, Srdjan Capkun, Ivan Flechais, Lizzie Coles-Kemp (editors), LNCS 7904: 232–249, 2013.

[3] S. Andova. Probabilistic process algebra. Ph.D. thesis, Technische Universiteit Eindhoven, 2002. ISBN: 90-386-0592-7.

[4] J. Barwise. Situations, Facts, and True Propositions. In: *The Situation in Logic*. CSLI Lecture Notes 17, 1989.

[5] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. CUP, 1997.

[6] A. Beautement, R. Coles, J. Griffin, C. Ioannidis, B. Monahan, D. P. C. Author), A. Sasse, and M. Wonham. Modelling the Human and Technological Costs and Benefits of USB Memory Stick Security. In M. E. Johnson, editor, *Managing Information Risk and the Economics of Security*, 141–163. Springer, 2008.

[7] J. van Benthem, P. Girard, and O. Roy. Everything Else Being Equal: A Modal Logic for Ceteris Paribus Preferences *Journal of Philosophical Logic* 38(1):83–125, 2009.

[8] J. van Benthem, S. van Otterloo, and O. Roy. *Preference logic, conditionals, and solution concepts in games*. ILLC, 2005.

[9] Y. Beresnevichiene, D. Pym, and S, Shiu. Decision Support for Systems Security Investment. *Proc. Business-driven IT Management (BDIM) 2010*. IEEE Xplore, 2010.

[10] G. Clark, S. Gilmore, J. Hillston, and M. Ribaudo. Exploiting modal logic to express performance measures. In B.R. Haverkort, H.C. Bohnenkamp, and C.U. Smith, editors, *Computer Performance Evaluation: Modelling Techniques and Tools, Proceedings of the 11th International Conference*. LNCS 1786: 211–227, 2000.

[11] M. Collinson and D. Pym. Algebra and logic for resource-based systems modelling. *Mathematical Structures in Computer Science*, 19:959–1027, 2009. doi:10.1017/S0960129509990077.

[12] Matthew Collinson, Brian Monahan, and David Pym. Semantics for structured systems modelling and simulation. In *Proc. Simutools 2010*. ACM Digital Library, ISBN 78-963-9799-87-5, 2010.

[13] M. Collinson, B. Monahan, and D. Pym. A logical and computational theory of located resource. *Journal of Logic and Computation*, 19(b):1207–1244, 2009.

[14] M. Collinson, B. Monahan, and D. Pym. *A Discipline of Mathematical Systems Modelling*. College Publications, 2012.

[15] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems: Concepts and Design*. Addison Wesley; 3rd edition, 2000.

[16] Y. Deng, R. van Glabbeek, M. Hennessy, C. Morgan, and C. Zhang. Remarks on Testing Probabilistic Processes. *Electronic Notes in Theoretical Computer Science* 172:359–397, 2007.

[17] Y. Deng, Yuxin and M. Hennessy. Compositional Reasoning for Weighted Markov Decision Processes. *Science of Computer Programming* 78(12):2537–2579, 2013. doi = 10.1016/j.scico.2013.02.009,

[18] P. Girard. Modal Logic for Belief and Preference Change. Ph.D. thesis, Stanford University, 2008.

[19] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[20] M. Hennessy and G. Plotkin. On observing nondeterminism and concurrency. In *Proceedings of the 7th ICALP*. LNCS 85: 299–308, 1980.

[21] J. Hillston. Compositional Markovian modelling using a process algebra. In W. Stewart (editor), Proceedings of the Second International Workshop on Numerical Solution of Markov Chains: Computations with Markov Chains. Kluwer Academic Press, 1995.

[22] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[23] J. Hillston. Process algebras for quantitative analysis. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS '05)*, 239–248, Chicago, June 2005. IEEE Computer Society Press.

[24] S. Ishtiaq and P. O'Hearn. **BI** as an assertion language for mutable data structures. 28th ACM-SIGPLAN Symposium on Principles of Programming Languages, London, 14–26. Association for Computing Machinery, 2001.

[25] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley & Sons, 1991.

[26] W. Jamroga. A temporal logic for Markov chains. In *Proc. AAMAS 2008*, 607–704. ACM Digital Library, 2008.

[27] M. Kalech and A. Pfeffer. In *AAMAS '10: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*: Volume 1, 267–274. ACM Digital Library, 2010. ISBN: 978-0-9826571-1-9

[28] R. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs*. Wiley, 1976.

[29] F. Knight. *Risk, Uncertainty, and Profit*. No. 16 in *Series of Reprints of Scarce Tracts in Economics*. London School of Economics,1933.

[30] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Model Checking for Performance and Reliability Analysis. *ACM SIGMETRICS Performance Evaluation Review* 36(4): 40–45, 2009.

[31] J. McCarthy. Formalizing context. In *IJCAI*, 555–562, 1993.

[32] R. Milner. Calculi for synchrony and asynchrony. *Theoret. Comp. Sci.*, 25(3):267–310, 1983.

[33] R. Milner. *Communication and Concurrency*. Prentice Hall, New York, 1989.

[34] P. O'Hearn and D. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 1999.

[35] D. Osherson and S. Weinstein. Preference Based on Reasons. *The Review of Symbolic Logic* 5(1):122–147, 2012.

[36] S. Ramchurn, D. Huynh, and N. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[37] J. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proc. 17th LICS*, 55–74, IEEE, 2002.

[38] S. Ross. *Introduction to Probability Models (Ninth Editiion)*. Academic Press, 2006.

[39] Y. Shoham and K. Leyton-Brown. *Multiagent Systems*. Cambridge University Press, 2009.

[40] M. Singh. Trust as Dependence: A Logical Approach. Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011), Tumer, Yolum, Sonenberg and Stone (eds.), May, 26, 2011, 863–870.

[41] A. Sokolova and E. De Vink. Probabilistic Automata: System Types, Parallel Composition and Comparison. LNCS 2925: 1–43, 2004.

[42] The Trust Domains Project. `www.hpl.hp.com/research/cloud_security/TrustDomains.pdf`, `http://www0.cs.ucl.ac.uk/staff/D.Pym/TD.html`

[43] C. Tofts. Processes with Probability, Priority, and Time. *Formal Aspects of Computing*, 6(5):536–564.

[44] G. von Wright. The Logic of Preference. Edinburgh University Press, 1963.

[45] G. von Wright. The Logic of Preference Reconsidered. *Theory and Decision* 3: 140169, 1972.

University College London
E-mail address: `gabrielle.anderson@ucl.ac.uk`

University College London
E-mail address: `d.pym@ucl.ac.uk`