A Substructural Modal Logic of Utility

GABRIELLE ANDERSON AND DAVID PYM, *University College London, Department of Computer Science, London WC1E 6BT, UK*
E-mail: gabrielle.anderson@cantab.net; d.pym@ucl.ac.uk

Abstract

We introduce a substructural modal logic of utility that can be used to reason about optimality with respect to properties of states. Our notion of state is quite general, and is able to represent resource allocation problems in distributed systems. The underlying logic is a variant of the modal logic of bunched implications, and based on resource semantics, which is closely related to concurrent separation logic. We consider a labelled transition semantics and establish conditions under which Hennessy–Milner soundness and completeness hold. By considering notions of cost, strategy and utility, we are able to formulate characterizations of Pareto optimality, best responses, and Nash equilibrium within resource semantics. We also show that our logic is able to serve as a logic for a fully featured process algebra and explain the interaction between utility and the structure of processes.

Keywords: Substructural logic, bunched logic, modal logic, process algebra, resource semantics, Hennessy–Milner logic utility, Pareto optimality, Nash equilibrium.

1 Introduction

Mathematical modelling and simulation modelling are fundamental tools of engineering, science and social sciences such as economics, and provide decision-support tools in management. The components of distributed systems (as described, for example, in [13]) are typically modelled using various algebraic structures for the structural components—location, resource, and process— and probability distributions to represent stochastic interactions with the environment [2, 10–12]. Applications of this approach to systems security modelling have been explored extensively in, for example, [4–7, 11]. A key aspect of modelling distributed systems is resource allocation. For example, when many processes execute concurrently, they compete for resources. A common desire of system designers, managers, and users is to determine, if possible, the optimal allocation of resources required in order to solve a specific problem or deliver a specific service.

We develop a substructural modal predicate logic, MBIU, that can be used to reason about optimality with respect to properties of states. Our notion of state is quite general, and is able to represent resource allocation problems in distributed systems; in particular, it encompasses models of distributed systems in which there is a notion of agent [11, 13, 29]. The preferences of agents among the various outcomes of system evolutions are modelled using numerical payoffs, as formulated in game theory. We use arithmetic predicates to relate states to payoffs, and so are able to give a logical representation of agents' degrees of satisfaction. The payoff of a state is defined via the actions that the state can perform: the logic's modal formulae can then be used to reason about the payoffs of states that are related by the transition system. The logic also includes substructural connectives—as in BI [16, 25, 26], MBI [2, 10, 12], and Separation Logic [18, 28]—which can be used, among other things, to support reasoning about decision-making by concurrent combinations of agents. The notion of optimality of resource allocation is a central topic in economics, where game theory plays a significant role.

In Section 2, we develop MBIU. To do so, we must we introduce actions, and define a notion of transition systems with concurrent structure on their states. We introduce a slight variant on the standard notion of bisimulation for such a transition system, and describe various properties that we require for our results to hold; in particular, that the concurrent composition operator is a congruence with respect to the bisimulation relation. In Section 2.2, we specify our logic, and define its semantics in terms of concurrent transition relations. We obtain the technical result that, provided that bisimulation in the underlying transition system is a congruence with respect to the concurrent composition and any state can only evolve in finitely many ways, full Hennessy–Milner completeness holds for MBIU; that is, bisimulation equivalence of states corresponds exactly to logical equivalence in MBIU.

An agent or process, in a given starting state, makes a choice between possible actions and so evolves, along with its environment, to achieve a new state. Associated with such an action is its value, or utility, which is determined by a *payoff function*. When agents evolve, or multiple agents co-evolve—such as when competing as players in a game for resources — they make sequences of moves, called *strategies*, that determine the outcome of the game and the payoffs for each of the agents. For all elementary notions from economics required for this article, including ideas from utility theory and game theory, a suitable source is [29].

In order to define MBIU, we must introduce actions on states, their transition systems and the associated notion of bisimulation, payoffs, and strategies. We give some basic examples of how the logic is used to express properties of states, and establish the conditions on the operational semantics of actions that are required in order to obtain a Hennessy–Milner soundness and completeness theorem.

In Section 3, we illustrate how the the logical set-up that we have introduced can be used to capture the classical notions of optimality and equilibrium as established in utility theory and game theory. We begin with a classic example from distributed systems modelling: mutual producer–consumer. We then explain how our set-up can be used to express Pareto optimality. This example leads naturally into a discussion of game-theoretic examples and concepts. We consider here the prisoner's dilemma, the best-response property, and Nash equilibria.

In Sections 4 and 5, we show that the framework for modelling distributed systems, as introduced in [11, 12] and improved theoretically in [2], is encompassed by our framework, with the consequence that the treatment of of utility in MBIU extends to the modelling framework given in [2]. We revisit the mutual producer–consumer example from Section 3, explaining the interaction between the process-theoretic structure and utility.

Finally, in Section 6, we discuss a range of challenges for future research, in both logical and utility-theoretic directions.

A short version of this article is [1].

2   A substructural modal logic of utility

In this section, we define a substructural modal predicate logic, MBIU, that can be used to reason about optimality with respect to properties of states. Our notion of state is quite general; in particular, it encompasses models of distributed systems in which there is a notion of agent [11, 13, 29]. The preferences of agents among the various outcomes of system evolutions are modelled using numerical payoffs, as formulated in game theory. We use arithmetic predicates to relate states to payoffs, and so are able to give a logical representation of agents' degrees of satisfaction. The payoff of a state is defined via the actions that the state can perform: the logic's modal formulae can then be used to reason about the payoffs of states that are related by the transition system. The logic also includes substructural connectives — as in BI, MBI, and Separation Logic [18, 28]—which can be used,

among other things, to support reasoning about decision-making by concurrent combinations of agents.

In Section 2.1, we introduce actions, and define a notion of transition systems with concurrent structure on their states. We introduce a slight variant of the standard notion of bisimulation for such a transition system, and describe various properties that we require for our results to hold; in particular, that the concurrent composition operator is a congruence with respect to the bisimulation relation. In Section 2.2, we specify our logic, and define its semantics in terms of concurrent transition relations. We obtain the technical result that, provided that bisimulation in the underlying transition system is a congruence with respect to the concurrent composition and any state can only evolve in finitely many ways, full Hennessy–Milner completeness holds for MBIU. That is, bisimulation equivalence of states corresponds exactly to logical equivalence in MBIU.

2.1 Transition systems

First, we introduce our notion of action. We assume a set, **Act**, of actions, which correspond to the events of the system.

DEFINITION 1 (Action structure)
An action structure **Act** is a structure $(\mathbf{Act}, \cdot, 1)$ such that $(\mathbf{Act}, \cdot)$ is a total magma, and $1 \in \mathbf{Act}$ is a distinguished action.

Note that we do not require that the distinguished action 1 be a unit for $\cdot$, nor do we require $\cdot$ to be commutative, so **Act** is not necessarily a (commutative) monoid. We use $=$ to denote syntactic equality of actions. Let ab denote $a \cdot b$.

We take an additional equivalence relation on actions, $\equiv$, for a given action structure **Act**.

DEFINITION 2 (Action-composition equivalence relation)
An action-composition equivalence relation is an equivalence relation $\equiv \, \subseteq \mathbf{Act} \times \mathbf{Act}$ such that, for all $a, b, a', b' \in \mathbf{Act}$, $a \cdot 1 \equiv a$, $a \cdot b \equiv b \cdot a$, and $a \equiv a'$ and $b \equiv b'$ implies $a \cdot b \equiv a' \cdot b'$.

Note that the syntactic equivalence relation $=$ and the action-composition equivalence relation $\equiv$ are not necessarily the same. Herein, we only consider actions to be interchangeable if they are *syntactically* equal.

Then, we can define transition systems.

DEFINITION 3 (Transition system)
Let **Act** be an action structure. A transition system is a structure

$$(\mathbf{S}, \mathbf{Act}, \rightarrow)$$

with carrier set **S**, action structure **Act**, and transition relation $\rightarrow \, \subseteq \mathbf{S} \times \mathbf{Act} \times \mathbf{S}$.

Let r, s, etc., range over elements of the carrier set of a transition system. We refer to these elements as states.

Where we use partial functions, we employ the standard notations $R \downarrow$ and $R \uparrow$ to denote that the expression R is, respectively, defined or undefined.

Next, we add the concurrent structure of the states of a transition system. Concurrent transition systems (with some well-formedness conditions, defined below) are the core mathematical structure representing system dynamics in this article.

DEFINITION 4 (Concurrent transition system)
A concurrent transition system is a structure $(\mathbf{S}, \mathbf{Act}, \rightarrow, \equiv, \circ, e)$ such that $(\mathbf{S}, \mathbf{Act}, \rightarrow)$ is a transition system, $\equiv$ is an action-composition equivalence relation, $e \in \mathbf{S}$ is a distinguished element of the state space, and $\circ : \mathbf{S} \times \mathbf{S} \rightharpoonup \mathbf{S}$ is a partial operation such that

- for all r and s, if $r \circ s$ is defined, then $s \circ r$ is defined;
- for all r, r', s, s', and all a and b, if $r \circ s$ is defined, $r \xrightarrow{a} r'$, and $s \xrightarrow{b} s'$, then $r' \circ s'$ is defined and $r \circ s \xrightarrow{ab} r' \circ s'$;
- for all r, r'', s and c, there exist r''', s''', and a and b, s.t. if $r \circ s$ is defined and $r \circ s \xrightarrow{c} r''$, then $c = ab$, $r \xrightarrow{a} r'''$, $s \xrightarrow{b} s'''$, and $r'' = r''' \circ s'''$;
- the distinguished action 1 is s.t. $e \xrightarrow{1} e$ and, for all s and a, if $e \xrightarrow{a} s$, then $a = 1$ and $s = e$;
- for all r, $r \circ e$ is defined.

The action 1 is the distinguished action of $\mathbf{Act}$, and can, in general, only be considered a unit with respect to the action equivalence relation $\equiv$ (and not, in general, with respect to syntactic equality). The operation $\circ$ is referred to as the concurrent composition operation. In the sequel, we work with a fixed concurrent transition system at every point. We sometimes refer to the distinguished state as the unit state. We write $r \rightarrow s$ if there exists some a such that $r \xrightarrow{a} s$, $\rightarrow^*$ for the reflexive, transitive closure of $\rightarrow$, and $\rightarrow^+$ for the transitive closure of $\rightarrow$.

We can use the partiality of the concurrent composition, along with a transition system, to model straightforwardly key examples in systems modelling [11, 12], such as the following.

EXAMPLE 5 (Semaphores)
Let $\mathbf{Act}$ be the action structure freely generated by the atomic actions a and 1, where 1 is the distinguished action. Suppose a concurrent transition system $(\{s, e\}, \mathbf{Act}, \rightarrow, \circ, e)$, where $s \xrightarrow{a} s$, $s \circ e = s$, and $s \circ s$ is undefined. Note that $\rightarrow$ is undefined for any values that are neither specified explicitly nor required by properties of Definitions 3 and 4. We then have that no state can perform the action aa; that is, $e \xrightarrow{qa} $ and $s \xrightarrow{qa} $. The concurrent transition system acts like a semaphore, in that only one access action a can be performed at any given time.

The standard notion of bisimulation is that two states in a system are bisimilar if they can perform the same actions, and, after those reductions, remain bisimilar. We weaken that approach slightly, and consider two states in a system to be bisimilar if they can perform actions that are equivalent under $\equiv$, and, following those reductions, remain bisimilar. We define the notion of a (action-composition-equivalence-relation) bisimulation relation between states in a concurrent transition system.

DEFINITION 6 (Bisimulation)
A (action-composition-equivalence relation) bisimulation is a relation $\mathcal{R}$ such that, for all states $r \mathcal{R} s$, then, for all actions $a \in \mathbf{Act}$,

- if $r \xrightarrow{a} r'$, then there exist $b \in \mathbf{Act}$, $s' \in \mathbf{S}$ such that $s \xrightarrow{b} s'$, $a \equiv b$, and $r' \mathcal{R} s'$, and
- if $s \xrightarrow{a} s'$, then there exist $b \in \mathbf{Act}$, $r' \in \mathbf{S}$ such that $r \xrightarrow{b} r'$, $a \equiv b$, and $r' \mathcal{R} s'$.

Let $\sim \subseteq \mathbf{State} \times \mathbf{State}$ be the union of all bisimulations for a given concurrent transition system. The union of any two bisimulations is also a bisimulation. Hence $\sim$ is well defined, and a bisimulation. Note that the usual definition of bisimulation for process calculi is a special case of the above, where $\mathbf{Act}$ is a monoid of actions with commutative operation $\cdot$ and unit 1, and action-composition equivalence is just syntactic equality.

There are various technical properties of bisimulation, that we make use in the remainder of this article. First, the bisimulation relation $\sim$ is an equivalence relation.

LEMMA 7
For all states $s, s', s'' \in \mathbf{S}$, $s \sim s$, $s \sim s'$ implies $s' \sim s$, and $s \sim s'$ and $s' \sim s''$ imply $s \sim s''$.

PROOF. The above are straightforward to observe. ∎

Secondly, in the state space quotiented by the bisimulation relation, the concurrent composition $\circ$ is commutative.

LEMMA 8
For all $r_1, r_2 \in \mathbf{S}$, if $r_1 \circ r_2$ is defined, then $r_1 \circ r_2 \sim r_2 \circ r_1$.

PROOF. The bisimulation $\sim$ relation is the largest bisimulation relation, and contains all other bisimulation relations. In order to show that the above properties hold it is sufficient, therefore, to define a relation $\mathcal{R}$, for which the required properties hold, and show that the relation $\mathcal{R}$ is a bisimulation.

Let

$$\mathcal{R} = \{(r_1 \circ r_2, r_2 \circ r_1) \mid r_1 \circ r_2 \text{ is defined }\}.$$

By Definition 4, as $r_1 \circ r_2$ is defined, $r_2 \circ r_1$ is defined. Suppose that $r_1 \circ r_2 \xrightarrow{c} s$. By Definition 4, we have that there exist $a_1, a_2 \in \mathbf{Act}$, $r_1', r_2' \in \mathbf{S}$ such that $c = a_1 a_2$, $r_1 \xrightarrow{a_1} r_1'$, $r_2 \xrightarrow{a_2} r_2'$, and $s = r_1' \circ r_2'$. Also by Definition 4, we have that $r_2' \circ r_1'$ is defined and $r_2 \circ r_1 \xrightarrow{a_2 a_1} r_2' \circ r_1'$. By Definition 2, $a_1 a_2 \equiv a_2 a_1$. As $r_1' \circ r_2'$ is defined, we have that $(r_1' \circ r_2', r_2' \circ r_1') \in \mathcal{R}$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation. ∎

Thirdly, the distinguished state of a concurrent transition system, e, is a unit with respect to bisimulation.

LEMMA 9
For all states $s \in \mathbf{S}$, $s \circ e \sim s$.

PROOF. Let

$$\mathcal{R} = \{(s \circ e, s) \mid s \in \mathbf{S}\}.$$

By Definition 4, $s \circ e$ is defined. Suppose that $s \circ e \xrightarrow{c} r'$. By Definition 4, we have that there exist $a_1, a_2 \in \mathbf{Act}$, $r_1', r_2' \in \mathbf{S}$ such that $c = a_1 a_2$, $s \xrightarrow{a_1} r_1'$, $e \xrightarrow{a_2} r_2'$, and $r' = r_1' \circ r_2'$. Also by Definition 4, we have that $a_2 = 1$ and $r_2' = e$. By Definition 2, $a_1 1 \equiv a_1$. We straightforwardly have that $(r_1' \circ e, r_1') \in \mathcal{R}$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.
∎

The transition systems can be non-deterministic. Consider the following example, which is externally non-deterministic in the sense that transitions with different actions are defined on individual states [30]:

EXAMPLE 10
Let $\mathbf{Act}$ be the action structure freely generated from the atomic actions p, c, and 1, where 1 is the distinguished action. Suppose a concurrent transition system

$$((\mathbb{N} \times \mathbb{N}) \cup \{e\}, \mathbf{Act}, \rightarrow, =, \circ, e),$$

whose transition relation is the least transition relation such that

$$(m,n) \xrightarrow{p} (m,n+1)$$
$$(m+1,n) \xrightarrow{c} (m,n)$$
$$(m,n) \xrightarrow{1} (m,n),$$

and $(m_1,m_2) \circ (n_1,n_2) = (m_1+n_1, m_2+n_2)$. Then, for the resource $(2,0)$, actions p and c are both defined on the resource and so, in the transition system, there is external non-determinism between the distinct actions, p and c.

When evolving non-deterministic transition systems, it is necessary to have a method to decide between possible options. A strategy can be used to determine, for a given state, which possible transition preferred. If there are no possible transitions, then the strategy returns the non-state symbol $\bullet$.

DEFINITION 11 (Strategy)
A strategy is a total function $\sigma : \mathbf{S} \to ((\mathbf{Act} \times \mathbf{S}) \cup \{\bullet\})$ such that, for all states $r \sim s \in \mathbf{S}$,

- if there exist $a \in \mathbf{Act}$, $r' \in \mathbf{S}$ such that $\sigma(r) = (a,r')$, then there exist $b \in \mathbf{Act}$ and $s' \in \mathbf{S}$ such that $a \equiv b$, $\sigma(s) = (b,s')$, $r \xrightarrow{a} r'$, $s \xrightarrow{b} s'$, and $r' \sim s'$, and
- $\sigma(r) = \bullet$ if and only if $r \not\rightarrow$.

EXAMPLE 12
We can define a strategy to resolve the non-determinism we see in Example 10. Let σ be a function such that

$$\sigma((m,n)) = \begin{cases} (c, (m-1,n)) & \text{if } 1 \le m \\ (1, (m,n)) & \text{otherwise.} \end{cases}$$

This strategy chooses the to consume whenever possible, and to do nothing otherwise.

One property that we immediately obtain is that all strategies map the distinguished state to the pair of the distinguished action and the distinguished state.

LEMMA 13
For all strategies σ, $\sigma(e) = (1,e)$.

PROOF. By Definition 4, $s \xrightarrow{1} e$ and, for all states s and actions a, if $e \xrightarrow{a} s$, then $a=1$ and $s=e$. By Definition 11, there exists some a and s' such that $\sigma(e) = (a,s')$ and $e \xrightarrow{a} s'$. As the transition $e \xrightarrow{1} e$ is the only transition defined on the distinguished state e, $a=1$ and $s'=e$. ∎

The transition system approach to distributed systems modelling abstracts away from the entities that make decisions, and their mechanisms for doing so. A mechanism for resolving choices can be reintroduced into the models through strategies: it does not, however, represent the goals and interests of the entities making the choices. We can model the decision-making entities' preferences concerning the events (or outcomes) of the system through the use of a map from actions to the rationals. These numbers are interpreted as measures of an agent's level of satisfaction with a given action. Let **Act** be an action structure, with distinguished action 1.

DEFINITION 14 (Action payoff function)
An action payoff function is a partial function $v : \mathbf{Act} \to \mathbb{Q}$ such that, for all $a,b \in \mathbf{Act}$:

- If $v(a)$ and $v(b)$ are defined, then $v(ab) = v(a) + v(b)$;

- If $a \equiv b$ and $v(a)$ is defined, then $v(b)$ is defined and $v(a) = v(b)$;
- $v(1) = 0$.

Note that it is possible to have that $v(ab)$ is defined, but that $v(a)$ and $v(b)$ are not defined (c.f., Example 36).

We use different action payoff functions to represent the preferences of different decision-making entities (or, agents). In order to extend payoff functions to states, we must consider what value to give those states that can perform *no* actions. The structure in which we value states is the tropical semiring.

DEFINITION 15 (Tropical semiring)
The tropical semiring

$$(\mathbb{Q} \cup \{-\infty\}, \sup, -\infty, +, 0)$$

comprises the carrier set of the rationals together with negative infinity, supremum as the additive function, addition as the multiplicative function, and negative infinity and zero as the additive and multiplicative units, respectively. Let the elements of the tropical semiring be denoted q, q', etc.. We sometimes refer to such elements as *utility values* or *payoffs*. Let $\mathcal{D} = (\mathbb{Q} \cup -\infty)$.

Fix an action payoff function v, a strategy σ, and let δ be some rational number in the open interval $(0, 1)$. We can then extend the notion of preference over actions to preferences over states. These numbers are interpreted as measures of an agent's level of satisfaction with the given states [29].

DEFINITION 16 (State payoff function)
A state payoff function is a partial function $u_{v,\sigma,\delta} : \mathbf{S} \rightharpoonup (\mathbb{Q} \cup -\infty)$ such that:

$$u_{v,\sigma,\delta}(s) = \begin{cases} v(a) + \delta \times u_{v,\sigma,\delta}(s') & \text{if } \sigma(s) = (a, s'), \text{ and } v(a) \text{ and } u_{v,\sigma,\delta}(s') \text{ are defined} \\ -\infty & \text{if } \sigma(s) = \bullet \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The value that can be obtained from actions performed at states reachable in the future is less than value that can be obtained immediately; that is, δ is a discount factor for future values. Since $\delta \in (0, 1)$, the recursive term in Definition 16 is contractive.

LEMMA 17
For all states s, action payoff functions v, strategies σ, and discount factors δ, if $\sigma(s) = (a, s)$ and $v(a) = 0$, then $u_{v,\sigma,\delta}(s) = 0$.

PROOF. By Definition 16, we have that $u_{v,\sigma,\delta}(s) = 0 + \delta \times u_{v,\sigma,\delta}(s)$. As $(1 - \delta) \neq 0$, we have that $u_{v,\sigma,\delta}(s) = 0$. ∎

We can now determine payoffs for various resources in Example 12 (which relies on Example 10).

EXAMPLE 18
This is a simplification of a distributed systems example, presented fully in Example 34. Let v be an action payoff function such that

$$v(p) = -1 \qquad v(c) = 3 \qquad \delta = 0.8.$$

Recall that

$$\sigma((m, n)) = \begin{cases} (c, (m-1, n)) & \text{if } 1 \leq m \\ (1, (m, n)) & \text{otherwise,} \end{cases}$$

and hence that

$$\sigma((0,0)) = (1,(0,0)) \qquad \sigma((1,0)) = (c,(0,0)) \qquad \sigma((2,0)) = (c,(1,0)).$$

We then have that

$$u_{v,\sigma,\delta}((0,0)) = 0 \qquad\qquad \text{By Lemma 17.}$$

$$u_{v,\sigma,\delta}((1,0)) = 3 + 0.8 \times u_{v,\sigma,\delta}((0,0))$$
$$= 3$$

$$u_{v,\sigma,\delta}((2,0)) = 3 + 0.8 \times u_{v,\sigma,\delta}((1,0))$$
$$= 5.4.$$

Note that with a different strategy, and the same action payoff function, discount factor, and underlying systems model, different payoffs can be achieved.

State payoff functions (Definition 16) specify the value of states in terms of a series of simultaneous equations. In order to solve these straightforwardly, we only consider strategies that generate a finite set of simultaneous equations. We make some auxiliary definitions that we use to reason about the actions and states chosen by repeatedly applying a strategy to a state (and its resulting chosen states). We particularly make use of these definitions in the proof of various equational properties of payoffs of resource–process pairs (Section 5).

DEFINITION 19
If $\sigma(r) = (a,s)$, then $\sigma_{state}^0(r) = s$ and $\sigma_{act}^0(r) = a$, and, for all $n \in \mathbb{N}$, if $\sigma(r) = (a,s)$, $\sigma_{state}^n(s) \downarrow$, and $\sigma_{act}^n(s) \downarrow$, then $\sigma_{state}^{n+1}(r) = \sigma_{state}^n(s)$ and $\sigma_{act}^{n+1}(r) = \sigma_{act}^n(s)$. Let $\sigma_{last}(s,\sigma) = n$ if and only if $\sigma_{state}^n(s) \downarrow$ and, for all $n' > n$, $\sigma_{state}^{n'}(s) \uparrow$.

DEFINITION 20 (Strategy transition closure)
Let $C_\sigma(r)$ be the set of states that can be reached from state r by following the transitions specified by a strategy σ, that is,

$$C_\sigma(r) = \{s \mid n \in \mathbb{N} \text{ and } \sigma_{state}^n(r) = s\} \cup \{r\}.$$

In the case that $C_\sigma(r)$ is *finite*, $u_{v,\sigma,\delta}(r)$ is specified in terms of a *finite set of simultaneous linear equations* (by Definition 16), which can be solved using the methods described in [20]. Henceforth, we consider only strategies σ such that, for all states $s \in \mathbf{S}$, $C_\sigma(s)$ is finite. The background definitions required for the proof of Lemma 21 (below) can also be found in [20].

With these assumptions, we can show a key property: bisimilar states are mapped to the same payoffs. This is used to demonstrate the fulfilment of required properties concerning the interpretation of logical predicates (Definition 26).

LEMMA 21
If $r \sim s$, then, for all v, σ, δ and $u_{v,\sigma,\delta}(r) = u_{v,\sigma,\delta}(s)$.

PROOF. By our assumptions, r and s both have a finite number of successor states. These states, and their relevant transition systems, can be uniquely mapped into the final coalgebra of finite and infinite sequences of actions. In particular, since $r \sim s$ we know that both are uniquely mapped to the same element of the final coalgebra (see Definition 11). The payoff functions of r and s (which are contractive in their recursive terms) can be computed as the solution of a finite linear system of

equations determined by their transition structure. As the elements r and s are mapped to the same element of the final coalgebra, the linear systems that determine the payoff can be shown to have the exact same solution. ∎

We conclude this section with an additional property of our framework, namely, that the unit state always has a payoff of 0.

LEMMA 22
For all action payoff functions v, strategies σ, and discount factors δ, $u_{v,\sigma,\delta}(e)=0$.

PROOF. By Definition 14, we have that $v(1)$ is defined, and equal to 0. By Definitions 4 and 11, we have that $\sigma(e)=(1,e)$ and $e \xrightarrow{1} e$. By Definition 16, we have that $u_{v,\sigma,\delta}(e)=0+\delta \times u_{v,\sigma,\delta}(e)$. As $(1-\delta)\neq 0$, we have that $u_{v,\sigma,\delta}(e)=0$. ∎

2.2 Logic and transition semantics

We define a modal predicate logic, MBIU, for expressing properties of resources and their utility. Building directly on [2, 10, 12], we define a semantics for MBIU in terms a concurrent transition system and its corresponding bisimulation relation.

We assume a two-sorted first-order language Σ, building standard terms t,u, etc., from standard variables x,y,z, etc., and *action terms*, denoted w, w', etc., built from *action variables* α, β, etc.. The predicate symbols of the language, however, may be applied to standard terms *only*.

DEFINITION 23
The action terms of MBIU, denoted d, d', etc., building on actions a, b, c, etc., are formed according to the following grammar:

$$d ::= a \mid \alpha \mid d \diamond d.$$

Let q be a term constant denoting the rational number q, and $\mathrm{v}(d)$ be a constant denoting the rational-valued payoff of the denotation of an action term d according to action payoff function v.

DEFINITION 24 (Terms)
Let the numerical terms, denoted t, t', etc., be formed according to the following grammar:

$$t ::= q \mid x \mid \mathrm{v}(d) \mid t+t \mid t \times t.$$

We assume a set Pred of predicate symbols, each with a given arity n, with elements denoted p, q, etc.. Then, formulae can be defined as follows.

DEFINITION 25 (Predicate formulae)
The predicate formulae of MBIU, denoted p, p′, etc., are given by the following grammar:

$$\begin{aligned}\varphi ::= &\; \mathrm{p}(\mathbf{t}) \mid t=t \mid d \equiv d \mid \bot \mid \top \mid \neg \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid I \mid \varphi * \varphi \mid \varphi \mathbin{-\!\!*} \varphi \mid \\ &\; \langle d \rangle \varphi \mid [d]\varphi \mid \exists \alpha.\varphi \mid \forall \alpha.\varphi \mid \exists x.\varphi \mid \forall x.\varphi,\end{aligned}$$

where t, s, x, and α range over terms, action terms, term variables, and action variables, respectively.

The (additive) modalities are the standard *necessarily* and *possibly* connectives familiar from modal logics, in particular Hennessy–Milner-style logics for process algebras [17, 24]. As such, they implicitly use meta-theoretic quantification to make statements about reachable resources.

Multiplicative modalities can also be defined [11, 12]. The connectives $*$ and $\mathbin{-\!\!*}$ are the multiplicative conjunction (with unit I) and implication (right adjoint to $*$), respectively, familiar from bunched logics [16] and in particular Boolean BI [22].

Now we give a Kripke-style frame semantics for MBIU. A *valuation* is a function mapping standard variables to rational numbers and action variables to actions. Valuations can be extended to arbitrary terms and action terms in the standard way: action constants are mapped to their obvious action, $\diamond$ is mapped to action composition $\cdot$, term constants are mapped to their obvious denotations, and arithmetical functions are mapped to their standard definitions. Let $\rho(\_)$ denote valuation of terms and action terms. Valuations extend to tuples of terms in the straightforward way. We relate the value of states to terms, for each payoff function $v \in \mathbf{V}$, via a distinguished predicate $\mathrm{u}_v(t)$. An interpretation then comprises a model and a valuation.

DEFINITION 26 (MBIU-model)
A *model*, M, of MBIU, together with a valuation ρ of variables, interprets standard terms in the carrier set of the tropical semiring, $\mathcal{D}$, and action terms in a set **Act** of *actions*, denoted a, b, etc., in the manner familiar from first-order logic. We write t^M for the interpretation of term t in model M (extended pointwise to tuples of terms). Models must also contain the following elements:

- a concurrent transition system $(\mathbf{S}, \mathbf{Act}, \rightarrow, \equiv, \circ, e)$;
- a set of payoff functions $\mathbf{V}$, a discount factor $\delta \in (0,1)$, and a strategy σ;
- an interpretation $\mathrm{p}^M \subseteq \mathbf{S} \times \mathcal{D}^k$, for each predicate symbol p of arity k such that the following properties hold:

  - *Predicate $\sim$-closure:* if $s \sim s'$ and $(s, \mathbf{d}) \in \mathrm{p}^M$, then $(s', \mathbf{d}) \in \mathrm{p}^M$;
  - *Distinguished predicates:* $(s, \rho(t)) \in \mathrm{u}_v^M$ if and only if $u_{v,\sigma,\delta}(s) = \rho(t)$.

As bisimilar states have the same payoff, for fixed action payoff function, strategy, and discount factor (Lemma 21), interpretations of the distinguished predicates are $\sim$-closed.

We can then define the semantics of formulas ϕ via the satisfaction relation $s \vDash_{M,\rho} \phi$, where M is a model, s is a state in the concurrent transition system of the model, and ρ is a valuation. Satisfaction in a given model is then denoted $s \vDash_{M,\rho} \varphi$, read as 'for the given model M, with valuation ρ, the state s has property φ'. The definition of the satisfaction relation is given by Figure 1. In the sequel, we drop the model M or the valuation ρ, writing $s \vDash_\rho \phi$ or $s \vDash \phi$, when their definitions are obvious. An alternative formulation of MBIU with intuitionistic additives (cf. [12, 25]) can be taken if desired. Its use in modelling applications remains to be explored in future work.

We can now formally describe payoff properties of states.

EXAMPLE 27
Recall Examples 10, 12 and 18. The formula

$$\phi = \exists x, y.(\langle p \rangle \mathrm{u}_v(x)) \wedge (\langle c \rangle \mathrm{u}_v(y)) \wedge (v(p) + (\delta \times x) < v(c) + \delta \times y)$$

denotes that it is possible to perform actions p and c, and that the payoff obtained by performing p (and the actions that follow from the resulting state) is less than that obtained by performing c (and the actions that follow from the resulting state). Note that $u_{v,\sigma,\delta}((2,1)) = 5.4$ and $u_{v,\sigma,\delta}((1,0)) = 3$, and hence

$$-1 + 0.8 \times 5.4 = 3.32 < 5.4 = 3 + 0.8 \times 3.$$

As a result, we have that $(2,0) \vDash \phi$.

$$
\begin{array}{llll}
s & \vDash_\rho & \mathrm{p}(\mathbf{t}) & \text{iff} & (\rho(\mathbf{t}), s) \in \mathrm{p}^M \\
s & \vDash_\rho & t_1 = t_2 & \text{iff} & \rho(t_1) = \rho(t_2) \\
s & \vDash_\rho & d_1 \equiv d_2 & \text{iff} & \rho(d_1) \equiv \rho(d_2) \\
s & \vDash_\rho & \bot & \text{never} \\
s & \vDash_\rho & \top & \text{always} \\
s & \vDash_\rho & \neg\phi & \text{iff} & s \nvDash_\rho \phi \\
s & \vDash_\rho & \phi_1 \vee \phi_2 & \text{iff} & s \vDash_\rho \phi_1 \text{ or } s \vDash_\rho \phi_2 \\
s & \vDash_\rho & \phi_1 \wedge \phi_2 & \text{iff} & s \vDash_\rho \phi_1 \text{ and } s \vDash_\rho \phi_2 \\
s & \vDash_\rho & \phi_1 \rightarrow \phi_2 & \text{iff} & s \vDash_\rho \phi_1 \text{ implies } s \vDash_\rho \phi_2 \\
s & \vDash_\rho & I & \text{iff} & s \sim e \\
s & \vDash_\rho & \phi_1 * \phi_2 & \text{iff} & \text{there exist } s_1, s_2, \text{ with } s \sim s_1 \circ s_2, \text{ such that} \\
& & & & \quad s_1 \vDash_\rho \phi_1 \text{ and } s_2 \vDash_\rho \phi_2 \\
s & \vDash_\rho & \phi_1 \mathbin{-\!\!*} \phi_2 & \text{iff} & \text{for all } s', s' \vDash_\rho \phi_1 \text{ and } s \circ s' \text{ defined implies} \\
& & & & \quad s \circ s' \vDash_\rho \phi_2 \\
s & \vDash_\rho & \langle d \rangle \phi & \text{iff} & \text{there exist } a \in \mathbf{Act},\, s' \in \mathbf{S} \text{ such that } s \xrightarrow{a} s', \\
& & & & \quad \rho(d) \equiv a, \text{ and } s' \vDash_\rho \phi \\
s & \vDash_\rho & [d]\phi & \text{iff} & \text{for all } a \in \mathbf{Act},\, s' \in \mathbf{S},\, s \xrightarrow{a} s' \text{ and } \rho(d) \equiv a \text{ implies} \\
& & & & \quad s' \vDash_\rho \phi \\
s & \vDash_\rho & \exists\alpha.\phi & \text{iff} & \text{there exists } a \in \mathbf{Act} \text{ such that } s \vDash_{\rho[\alpha:=a]} \phi \\
s & \vDash_\rho & \forall\alpha.\phi & \text{iff} & \text{for all } a \in \mathbf{Act},\, s \vDash_{\rho[\alpha:=a]} \phi \\
s & \vDash_\rho & \exists x.\phi & \text{iff} & \text{there exists } q \in \mathcal{D} \text{ such that } s \vDash_{\rho[x:=q]} \phi \\
s & \vDash_\rho & \forall x.\phi & \text{iff} & \text{for all } q \in \mathcal{D},\, s \vDash_{\rho[x:=q]} \phi
\end{array}
$$

FIGURE 1. Satisfaction relation for MBIU.

To obtain some key theoretical results concerning of our modelling framework, we require some additional properties.

When we perform a composition of states, it is necessary to take account of the partiality of the composition operator. As a result, we shall also require the following $\circ$-$\sim$-*closed* property of concurrent transition systems.

DEFINITION 28 ($\circ$-$\sim$-closed)
If $r_1 \sim s_1$, $r_2 \sim s_2$, and $r_1 \circ r_2$ is defined, then $s_1 \circ s_2$ is defined.

Henceforth, all concurrent transitions systems are assumed to be $\circ$-$\sim$-closed. An immediate result is that concurrent compositions of bisimilar states are bisimilar. This is a key result, which is used in the proof of the soundness direction of the Hennessy–Milner correspondence (Theorem 32, Case $\varphi = \varphi_1 \mathbin{-\!\!*} \varphi_2$).

LEMMA 29 (Composition congruence)
If $r_1 \sim s_1$, $r_2 \sim s_2$, and $r_1 \circ r_2$ and $s_1 \circ s_2$ are defined, then $r_1 \circ r_2 \sim s_1 \circ s_2$.

PROOF. Let
$$
\mathcal{R} = \{(r_1 \circ r_2, s_1 \circ s_2) \mid r_1 \sim s_1, r_2 \sim s_2, r_1 \circ r_2 \downarrow, \text{ and } s_1 \circ s_2 \downarrow\}.
$$

Suppose that $r_1 \circ r_2 \xrightarrow{c} r'$. By Definition 4, there exist a_1, a_2, r'_1, r'_2 such that $c = a_1 a_2$, $r' = r'_1 \circ r'_2$, $r_1 \xrightarrow{a_1} r'_1$ and $r_2 \xrightarrow{a_2} r'_2$. By Definition 6, there exist b_1, b_2, s'_1, s'_2 such that $a_1 \equiv b_1$, $a_2 \equiv b_2$, $s_1 \xrightarrow{b_1} s'_1$,

$s_2 \xrightarrow{b_2} s'_2$, $r'_1 \sim s'_1$ and $r'_2 \sim s'_2$. By Definition 2, $a_1 a_2 \equiv b_1 b_2$. By Definition 4, $s_1 \circ s_2 \xrightarrow{b_1 b_1} s'_1 \circ s'_2$. We immediately have that $(r_1 \circ r_2)\mathcal{R}(s_1 \circ s_2)$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation. ∎

When we describe states logically, it is necessary to take account of the number of successor states that can be reached. As a result, we shall also require the following image-finiteness property of concurrent transition systems.

DEFINITION 30 (Image-finite)
A state s is image-finite if it has finitely many derivatives.

From this point onwards, all states are assumed to be image-finite.

With this set-up, we can prove the Hennessy–Milner soundness and completeness theorem. While this result is standard within the field of process calculi, in the multi-world framework of resource–process calculi, this result has only recently been obtained in adequate generality (the issue was raised in [12] and conclusively addressed in [2]). A key contribution in [2] was restrictions on resource–process calculi sufficient to prove congruence results, which are required to prove the soundness direction of the Hennessy–Milner result for the standard notion of bisimulation and a logic that included multiplicative implication and multiplicative modalities. In this article, we modify those requirements to permit the handling of predicate logics which can be used to represent notions of optimality (given a suitable notion of payoff over worlds).

We define the notion of logical equivalence as follows:

DEFINITION 31 (Logical equivalence)
Fix some model M. Then, $r \equiv_{\text{MBIU}} s$ if and only if, for all valuations ρ and formulae ϕ, $r \vDash_{M,\rho} \phi$ if and only if $s \vDash_{M,\rho} \phi$.

With this set-up, we can prove the soundness direction of the Hennessy–Milner completeness theorem — operational equivalence implies logical equivalence. This proof requires the congruence property (Lemma 29).

THEOREM 32
If $r \sim s$, then $r \equiv_{\text{MBIU}} s$.

PROOF. Fix a model M. We show that, for all states r and s, valuations ρ, and formulae ϕ, if $r \vDash_{M,\rho} \phi$ and $r \sim s$, then $s \vDash_{M,\rho} \phi$. This property is sufficient to prove logical equivalence. We proceed by induction over the structure of the satisfaction relation, $r \vDash_{M,\rho} \varphi$.

Case $\varphi = \text{p}(\mathbf{t})$. By Definition 26, as $(\rho(\mathbf{t}), r) \in \text{p}^M$ and $r \sim s$, we have that $(\rho(\mathbf{t}), s) \in \text{p}^M$. Hence we have that $s \vDash_{M,\rho} \text{p}(\mathbf{t})$.

Case $\varphi = t_1 = t_2$. By the hypothesis, we have that $\rho(t_1) = \rho(t_2)$. Hence, we have that $s \vDash_\rho t_1 = t_2$.

Case $\varphi = d_1 \equiv d_2$. By the hypothesis, we have that $\rho(d_1) \equiv \rho(d_2)$. Hence we have that $s \vDash_\rho d_1 \equiv d_2$.

Case $\varphi = \bot$. As the premises assume $r \vDash_{M,\rho} \bot$, we have a contradiction and can disregard this case.

Case $\varphi = \top$. We have that $s \vDash_{M,\rho} \top$, straightforwardly.

Case $\varphi = \varphi_1 \vee \varphi_2$. By the hypotheses, we have that $r \vDash_{M,\rho} \varphi_1$ or $r \vDash_{M,\rho} \varphi_2$. By the induction hypothesis, we have that $s \vDash_{M,\rho} \varphi_1$ or $s \vDash_{M,\rho} \varphi_2$. Hence we have that $s \vDash_{M,\rho} \varphi_1 \vee \varphi_2$.

Case $\varphi = \varphi_1 \wedge \varphi_2$. By the hypotheses, we have that $r \vDash_{M,\rho} \varphi_1$ and $r \vDash_{M,\rho} \varphi_2$. By the induction hypothesis, we have that $s \vDash_{M,\rho} \varphi_1$ and $s \vDash_{M,\rho} \varphi_2$. Hence we have that $s \vDash_{M,\rho} \varphi_1 \wedge \varphi_2$.

Case $\varphi = \varphi_1 \rightarrow \varphi_2$. By the induction hypothesis, we have that $s \vDash_{M,\rho} \varphi_1$ whenever $r \vDash_{M,\rho} \varphi_1$, and $s \vDash_{M,\rho} \varphi_2$ whenever $r \vDash_{M,\rho} \varphi_2$. Hence we have that $s \vDash_{M,\rho} \varphi_1 \rightarrow \varphi_2$.

Case $\varphi = I$. By Lemma 7, as $r \sim e$ and $r \sim s$, we have that $s \sim e$. Hence we have that $s \vDash_{M,\rho} I$.

Case $\varphi = \varphi_1 * \varphi_2$. By the hypotheses, we have that $r \sim (r_1 \circ r_2)$, $r_1 \vDash_{M,\rho} \varphi_1$, and $r_2 \vDash_{M,\rho} \varphi_2$. By Lemma 7, we have that $s \sim r_1 \circ r_2$. Hence we have that $s \vDash_{M,\rho} \varphi_1 * \varphi_2$.

Case $\varphi = \varphi_1 \mathbin{-\!*} \varphi_2$. Suppose some r' such that $r' \vDash_{M,\rho} \varphi_1$ and $s \circ r'$ is defined. By Definition 28, we have that $r \circ r'$ is defined. By Lemma 29, we have that $r \circ r' \sim s \circ r'$. By the hypotheses, we have that $r \circ r' \vDash_{M,\rho} \varphi_2$. By the induction hypothesis, we have that $s \circ r' \vDash_{M,\rho} \varphi_2$. Hence we have that $s \vDash_{M,\rho} \varphi_1 \mathbin{-\!*} \varphi_2$.

Case $\varphi = \langle d \rangle \psi$. By the hypothesis, there exist a, r' such that $r \xrightarrow{a} r'$, $\rho(d) \equiv a$, and $r' \vDash_{M,\rho} \psi$. By the definition of bisimulation, we have that there exist $b \in \mathbf{Act}$, $s' \in \mathbf{S}$ such that $s \xrightarrow{b} s'$, $a \equiv b$, and $r' \sim s'$. By Definition 2, $\rho(d) \equiv b$. By the induction hypothesis, we have that $s' \vDash_{M,\rho} \psi$. Hence we have that $s \vDash_{M,\rho} \langle d \rangle \psi$.

Case $\varphi = [d]\psi$. Suppose that $s \xrightarrow{b} s'$ and $\rho(d) \equiv b$. By the definition of bisimulation, we have that there exist $a \in \mathbf{Act}$, $r' \in \mathbf{S}$ such that $r \xrightarrow{a} r'$, $a \equiv b$, and $r' \sim s'$. By Definition 2, $a \equiv \rho(d)$. By the hypotheses, we have that $r' \vDash_{M,\rho} \psi$. By the induction hypothesis, we have that $s' \vDash_{M,\rho} \psi$. Hence we have that $s \vDash_{M,\rho} [d]\psi$.

Case $\varphi = \exists \alpha. \psi$. By the hypotheses, there exists $a \in \mathbf{Act}$ such that $r \vDash_{M,\rho[\alpha:=a]} \psi$. By the induction hypothesis, we have that $s \vDash_{M,\rho[\alpha:=a]} \psi$. Hence we have that $s \vDash_{M,\rho} \exists \alpha. \psi$.

Case $\varphi = \forall \alpha. \psi$. We have, for all $a \in \mathbf{Act}$, that $r \vDash_{M,\rho[\alpha:=a]} \psi$. By the induction hypothesis, we have, for all $a \in \mathbf{Act}$, that $s \vDash_{M,\rho[\alpha:=a]} \psi$. Hence we have that $s \vDash_{M,\rho} \forall \alpha. \psi$.

Case $\varphi = \exists x. \psi$. By the hypotheses, there exists $q \in \mathcal{D}$ such that $r \vDash_{M,\rho[x:=q]} \psi$. By the induction hypothesis, we have that $s \vDash_{M,\rho[x:=q]} \psi$. Hence we have that $s \vDash_{M,\rho} \exists x. \psi$.

Case $\varphi = \forall x. \psi$. We have, for all $q \in \mathcal{D}$, that $r \vDash_{M,\rho[x:=q]} \psi$. By the induction hypothesis, we have, for all $q \in \mathcal{D}$, that $s \vDash_{M,\rho[x:=q]} \psi$. Hence we have that $s \vDash_{M,\rho} \forall x. \psi$. ■

The reverse direction of the Hennessy–Milner completeness theorem relies on image-finiteness (Definition 30).

THEOREM 33
If $r \equiv_{\mathrm{MBIU}} s$, then $r \sim s$.

PROOF. Fix some model M. Supposing that $r \equiv_{\mathrm{MBIU}} s$, we require to show that $r \sim s$. As $\sim$ is the *largest* relation closed under the conditions in Definition 6, it suffices to show that $\equiv_{\mathrm{MBIU}}$ is a bisimulation.

Let

$$\mathcal{R} = \{(r,s) \mid r \equiv_{MBIU} s\}.$$

Suppose some $a \in \mathbf{Act}$ and $r' \in \mathbf{State}$ such that $r \xrightarrow{a} r'$. Suppose, for a contradiction, that, there exist no $b \in \mathbf{Act}$ and $s' \in \mathbf{State}$, such that $s \xrightarrow{b} s'$, $a \equiv b$, and $r' \mathcal{R} s'$.

Let $\mathscr{F} = \{s' \mid s \xrightarrow{b} s' \text{ and } a \equiv b\}$. If $\mathscr{F}$ is empty, then $r \vDash \langle a \rangle \top$ and $s \nvDash \langle a \rangle \top$, contradicting $r \equiv_{\mathrm{MBIU}} s$. Hence $\mathscr{F}$ is non-empty. By Definition 30, $\mathscr{F} = \{s_i' \mid 1 \le i \le n\}$, for some finite n. By our supposition, for all $1 \le i \le n$, $(r', s_i') \notin \mathcal{R}$, hence $r' \not\equiv_{\mathrm{MBIU}} s_i'$. Thus there exist formulae $\phi_1, \ldots, \phi_n$ such that $r' \vDash \phi_i$ but $s_i' \nvDash \phi_i$, for all i. Hence $r \vDash \langle a \rangle (\phi_1 \wedge \ldots \wedge \phi_n)$ and $s \nvDash \langle a \rangle (\phi_1 \wedge \ldots \wedge \phi_n)$, again contradicting $r \equiv_{\mathrm{MBIU}} s$.

Hence our supposition must be false, and there must exist $b \in \mathbf{Act}$ and $s_i' \in \mathscr{F}$ such that $s \xrightarrow{b} s_i'$, $a \equiv b$, and $r' \mathcal{R} s_i'$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation. ■

The notion of attaching payoffs or weights to actions exists in the literature. Markov chains support reasoning about complex notions such as average utility with a given time discount, but do not provide compositionality results over model structures [19]. 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 [15]. That calculus has an associated modal logic, where the action modalities are also modalities on the weights of the actions. The notion of payoff of a process state is not directly represented, and cannot be reasoned over in the logic.

3 Examples and optimality

To illustrate the logical set-up we have introduced, we begin with a classic example from distributed systems modelling: mutual producer–consumer. We then explain how our set-up can be used to express Pareto optimality. This example leads naturally into a discussion of game-theoretic examples and concepts. We consider here the prisoner's dilemma, the best-response property, and Nash equilibrium.

EXAMPLE 34 (Mutual producer–consumer)
A classic example of distributed systems modelling is distributed coordination without mutual exclusion, the most common form of which is that of the producer–consumer system [11, section 2.3.5]. In such a scenario, one entity generates work that another entity can handle at a later point. We modify this slightly to the scenario with two entities, where each entity can generate work for, and consume work from, the other.

We extend Example 10. Suppose a concurrent transition system

$$((\mathbb{N} \times \mathbb{N}) \cup \{e\}, \mathbf{Act}, \rightarrow, =, \circ, e),$$

where

$$(m,n) \xrightarrow{p_1} (m,n+1)$$
$$(m+1,n) \xrightarrow{c_1} (m,n)$$
$$(m,n) \xrightarrow{p_2} (m+1,n)$$
$$(m,n+1) \xrightarrow{c_2} (m,n)$$
$$(m,n) \xrightarrow{1} (m,n),$$

and $(m_1,m_2) \circ (n_1,n_2) = (m_1+n_1, m_2+n_2)$.

The states of the concurrent transition system are pairs of natural numbers, where the first element of the pair denotes the number of work packages that the first entity can consume and the second element of the pair denotes the number of work packages that the second entity can consume. The action p_1 denotes production of a work package by the first entity for the second entity, and the action c_1 denotes the consumption of a work package by the first entity. The actions p_2 and c_2 have the obvious converse denotations.

Consider the situation where the entities 'profit' from the consumption of work packages, and must 'pay' to create work packages. A pair of possible payoff functions v_1 and v_2, for the two entities, which represents this situation is

$$v_1(p_1) = -1 \ v_1(c_1) = 3 \ v_1(p_2) = 0 \ \ v_1(c_2) = 0$$
$$v_2(p_1) = 0 \ \ \ v_2(c_1) = 0 \ v_2(p_2) = -2 \ v_2(c_2) = 4.$$

Note that each entity has no direct preferences over the actions of the other entity.

Let the discount factor δ be 0.8, and the strategy σ be a function such that each entity consumes if possible, does nothing if the other is consuming, and produce together when there are no resources for either to consume:

$$\sigma((m,n)) = \begin{cases} (c_1 c_2, (m-1, n-1)) & \text{if } 1 \le m \text{ and } 1 \le n \\ (c_1, (m-1, n)) & \text{if } 1 \le m \text{ and } n = 0 \\ (c_2, (m, n-1)) & \text{if } m = 0 \text{ and } 1 \le n \\ (p_1 p_2, (m+1, n+1)) & \text{if } m = 0 \text{ and } n = 0. \end{cases}$$

Consider the resource $(10,0)$. As there are only work packages available for the first entity, the actions defined on the resource are the consume action c_1, the produce actions p_1 and p_2, and the distinguished action 1. Each entity incurs a negative payoff when performing a produce action, which only benefits the other entity. The payoffs that can be obtained by performing the p_1 and c_1 actions, in the state $(10,0)$, are as follows,

$$v_1(p_1) + \delta \times u_{v_1, \sigma, \delta}(10, 1) \approx -1 + \delta \times 13.4 \approx 9.7$$
$$v_1(c_1) + \delta \times u_{v_1, \sigma, \delta}(9, 0) \approx 13.4$$
$$v_2(p_1) + \delta \times u_{v_2, \sigma, \delta}(10, 1) = 0 + 0.8 \times 4 \quad = 3.2$$
$$v_2(c_1) + \delta \times u_{v_2, \sigma, \delta}(9, 0) = 0.$$

In state $(10,0)$, the action c_1 gains the most for the first entity and p_1 gains the most for the second.

For either action, it is not possible to swap to an alternative action that makes one of the entities better off, without making the other entity worse off. This notion is called *Pareto optimality*.

DEFINITION 35 (Pareto optimality)
A state s is Pareto optimal if there exists an action a such that, for all other actions b, if some entity weakly prefers that action b be performed, then there is some other agent that strongly prefers that action a be performed. Formally, the state s is Pareto optimal if, for entities with payoff functions v_1, ..., v_n,

$$R \vDash \exists \alpha . \forall \beta . (\neg(\beta \equiv \alpha)) \rightarrow$$
$$\begin{pmatrix} \forall x, x'. \exists y, y'. \\ \left(((\langle \alpha \rangle \mathsf{u}_{v_1}(x)) \wedge (\langle \beta \rangle \mathsf{u}_{v_1}(x')) \wedge (x < x')) \rightarrow \right. \\ \left(((\langle \alpha \rangle \mathsf{u}_{v_2}(y)) \wedge (\langle \beta \rangle \mathsf{u}_{v_2}(y')) \wedge (y' < y)) \right. \\ \vee \dots \vee \\ \left. ((\langle \alpha \rangle \mathsf{u}_{v_n}(y)) \wedge (\langle \beta \rangle \mathsf{u}_{v_n}(y')) \wedge (y' < y)) \right) \end{pmatrix}$$
$$\vee \dots \vee$$
$$\begin{pmatrix} \forall x, x'. \exists y, y'. \\ \left(((\langle \alpha \rangle \mathsf{u}_{v_n}(x)) \wedge (\langle \beta \rangle \mathsf{u}_{v_n}(x')) \wedge (x < x')) \rightarrow \right. \\ \left(((\langle \alpha \rangle \mathsf{u}_{v_1}(y)) \wedge (\langle \beta \rangle \mathsf{u}_{v_1}(y')) \wedge (y' < y)) \right. \\ \vee \dots \vee \\ \left. ((\langle \alpha \rangle \mathsf{u}_{v_{n-1}}(y)) \wedge (\langle \beta \rangle \mathsf{u}_{v_{n-1}}(y')) \wedge (y' < y)) \right) \end{pmatrix}.$$

We abbreviate the above formula as $PO(v_1, \dots, v_n)$. In Example 34, the resource $(10,0)$ is Pareto optimal, witnessed by both the actions p_1 and c_1, and $(10,0) \vDash PO(v_1, v_2)$ holds. Note that optimality is defined in terms of actions; this is because, here, we take seriously the representation of actions that *perform* allocations. A transition is then an (actively performed) state allocation.

One field in which notions of optimality have been studied significantly is that of games and decision theory. We can model games in our resource semantics. A classic decision-making example from game theory is the prisoner's dilemma.

EXAMPLE 36 (Prisoner's dilemma)

Two individuals have been arrested, and are kept separately, so that they cannot collude in their decision-making. Each is offered the choice of attempting to 'defect', and give evidence against their partner, or to 'collaborate', and say nothing. If one person collaborates and the other defects, then the collaborating partner goes to jail for a long time, and the defecting partner goes free. If both people defect, then they both go to jail for a moderate time. If both people collaborate, then they both go to jail for a short time.

Let **Act** be the action structure freely generated by the actions c_1, d_1, c_2, d_2 and 1, where 1 is the distinguished action. Let $\mathbf{S} = \{r_1, r_2, r_{1,2}, e\}$ be the state space. The state r_1 denotes a resource where the first person can make a choice, the r_2 resource denotes a resource where the second person can make a choice and the $r_{1,2}$ resource denotes a resource where both people can make a choice at the same time. Let $r_1 \circ r_2 = r_{1,2}$ be defined, and

$$r_1 \xrightarrow{c_1} e \qquad r_1 \xrightarrow{d_1} e \qquad r_2 \xrightarrow{c_2} e \qquad r_2 \xrightarrow{d_2} e.$$

The c_1 action denotes collaboration by the first person, and the d_1 action denotes defection by the person. The c_2 and d_2 actions have the obvious denotations for the second person. Then, $(\mathbf{S}, \mathbf{Act}, \rightarrow, =, \circ, e)$ is a concurrent transition system.

We make use of the trivial strategy, for all states $s \in \mathbf{S}$, $\sigma(s) = (1, s)$. The action payoff functions v_1 and v_2 for the two people are

$$v_1(c_1 c_1) = -2 \ v_1(c_1 d_2) = -6 \ v_1(d_1 c_2) = 0 \quad v_1(d_1 d_2) = -4$$
$$v_2(c_1 c_1) = -2 \ v_2(c_1 d_2) = 0 \quad v_2(d_1 c_2) = -6 \ v_2(d_1 d_2) = -4.$$

So, if the first person collaborates and the second defects, then the first person receives six years in prison (cost $v_1(c_1 d_2) = -6$), while the second receives no time in prison (cost $v_2(c_1 d_2) = 0$).

We can define notions of *best response* and *Nash equilibrium*.

DEFINITION 37 (Best response)

An action a is a best response for a given entity to a particular choice of action b by another entity, at a given resource, if the (former) entity has no other action c available to it such that the action cb is defined on the resource and the entity strongly prefers cb to ab. Formally, a is the best response to action b at resource s if

$$s \vDash \forall \alpha. \exists x, y. \Big(\big((\langle a \rangle \top \wedge \langle \alpha \rangle \top) * (\langle b \rangle \top) \big) \wedge \big([a \diamond b](\mathsf{u}_v(x)) \wedge [\alpha \diamond b](\mathsf{u}_v(y)) \big) \Big)$$
$$\rightarrow \big((v(\alpha \diamond b) + \delta \times y) \leq (v(a \diamond b) + \delta \times x) \big).$$

We abbreviate the above formula, denoting that a is the best response to action b for the agent whose payoff function is v, as $BR(a, b, v)$. In the prisoner's dilemma example, the best response for the first agent to the action c_2 is d_1, and $r_{1,2} \vDash BR(d_1, c_2, v_1)$ holds.

We generalize this notation slightly, so that we write $BR(a, b_1, \ldots, b_n, v)$ to denote that a is the best response to the composite action $b_1 \ldots b_n$, for the payoff function v. Formally,

$$R \vDash \forall \alpha. \exists x, y.$$
$$\begin{pmatrix} \big((\langle a \rangle \top \wedge \langle \alpha \rangle \top) * (\langle b_1 \diamond (\ldots \diamond b_n) \rangle \top) \big) \wedge \\ \big([a \diamond (b_1 \diamond (\ldots \diamond b_n))](\mathsf{u}_v(x)) \big) \wedge \\ \big([\alpha \diamond (b_1 \diamond (\ldots \diamond b_n))](\mathsf{u}_v(y)) \big) \end{pmatrix}$$
$$\rightarrow$$
$$\big((v(\alpha \diamond (b_1 \diamond (\ldots \diamond b_n))) + \delta \times y) \leq (v(a \diamond (b_1 \diamond (\ldots \diamond b_n))) + \delta \times x) \big).$$

Now we can express Nash equilibrium.

DEFINITION 38 (Nash equilibrium)

A state s is a Nash equilibrium for a set of entities $I = \{1,\ldots,n\}$ if there is a collection of actions a_1, $\ldots$, a_n such that, for each entity $i \in I$ with payoff function v_i, the action a_i is the best response to the composition of actions a_j, where $j \in I \setminus \{i\}$. Formally, the state s is a Nash equilibrium if

$$s \vDash \exists \alpha_1 \ldots \alpha_n . BR(\alpha_1, \alpha_2, \ldots, \alpha_n, v_1) \wedge \ldots \wedge BR(\alpha_n, \alpha_1, \ldots, \alpha_{n-1}, v_n).$$

We abbreviate the above formula as $NE(v_1, \ldots, v_n)$. In the prisoner's dilemma example, the Nash equilibrium is the state $r_{1,2}$, witnessed by the actions d_1 and d_2, for payoff functions v_1 and v_2, and $r_{1,2} \vDash NE(v_1, v_2)$ holds.

4 Resource semantics and modelling

In this section, we take the first step towards using MBIU as a logic of state for a fully featured process algebra. To this end, we recall our theory of distributed systems modelling, as presented in, for example, [2, 11, 12]. Building on the classical distributed systems theory [13], the structural components of this modelling framework are location, resource, and process, together with a stochastically modelled environment. In this article, we make no further use of stochastically modelled environments.

Mathematically, we capture the structural components as follows:

- *Location*. In general, locations can be conveniently modelled using a range of graph-theoretic and topological structures [9, 11], with directed graphs being the key example for most practical modelling work. For simplicity, we make no further use of locations in this article. The reader might think of them either as implicitly present, or consider them to be rolled up into the definition of resources (see [11] for relevant technical support):
- *Resource*. In general, resources are assumed to form a preordered partial commutative resource monoid, in which resource elements can be combined, using the monoid operation, compared, using the preorder. The partiality ensures that not all combinations need be considered 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 [11, 16, 25]. 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: $(\mathbb{N}, \leq, +, 0)$. For this article, we work in the simpler setting in which we omit the preorder (see Definition 39, below):
- *Process*. In general, our treatment of process is based on Milner's synchronous calculus of communicating systems (SCCS) [23], as developed as a basis for systems modelling in [11, 12]. Note that asynchronous calculi can be encoded within such synchronous calculi [23].

The key idea is that resources and processes co-evolve, according to one of the following judgement: $R, E \xrightarrow{a} R', E'$, 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''. The operational semantics that defines such a transition system relies on a (partial) *modification function* (see Definition 40, below) that specifies how a given action modifies a given resource. This approach is known as *resource semantics*.

A simple way to describe distributed systems — neglecting for now the process-theoretic structure — is using resource semantics for the state space and concurrent composition, and using a modification function as the dynamics of the transition system. This family of systems are concurrent transition systems, and have all the properties that we described in Section 2. Later, in Section 5, we develop the theory in the process of a fully featured process algebra.

For now, we begin with the notion of resource from Boolean BI [18].

DEFINITION 39 (Resource monoid)

A resource monoid is a structure $\mathbf{R} = (\mathbf{R}, \circ, e)$ with carrier set $\mathbf{R}$, commutative partial binary operation $\circ : \mathbf{R} \times \mathbf{R} \rightharpoonup \mathbf{R}$, and unit $e \in \mathbf{R}$.

Let $\mathbf{Act}$ be a commutative monoid of actions, freely generated from a set of atomic actions, with operation $\cdot$ and unit 1. The actions correspond to the events of the system. The dynamics of the system is then given by the modification function, which describes how actions transform resources.

DEFINITION 40 (Modification function)

A modification function is a partial function $\mu : \mathbf{R} \times \mathbf{Act} \rightharpoonup \mathbf{R}$ such that

- for all R and S, and all a and b, if $\mu(R,a)$, $\mu(S,b)$, and $R \circ S$ are all defined, then $\mu(R,a) \circ \mu(S,b)$ and $\mu(R \circ S, ab)$ are both defined, and $\mu(R \circ S, ab) = \mu(R,a) \circ \mu(S,b)$;
- for all R, $\mu(R,1) = R$;
- for all R and S, and all c, if $R \circ S$ and $\mu(R \circ S, c)$ are defined, then there exist a and b such that $c = ab$, and $\mu(R,a)$ and $\mu(S,b)$ are both defined; and
- the distinguished action 1 is s.t. $\mu(e,1) = e$ and, for all s and a, if $\mu(e,a) = s$, then $a = 1$ and $s = e$.

If $\mu(R,a)$ is defined, then we say that action a is defined on resource R. We refer to a structure $(\mathbf{R}, \mathbf{Act}, \mu, =, \circ, e)$ as a resource monoid model.

A key systems modelling example, seen previously in Example 5, is that of semaphores. Note that Example 41 is essentially the same as Example 5, excepting that here we use the modification function as the transition relation.

EXAMPLE 41 (Semaphores)

Let $\mathbf{Act}$ be the free monoid generated by the atomic actions a and 1, where 1 is the distinguished action. Let $\mathbf{R}$ be the resource monoid $(\mathbf{R}, \circ, e)$ such that $s \circ e = s$ and $s \circ s$ is undefined. We use a modification function such that $\mu(sa) = s$. We then have that no resource can perform the action aa, that is, $\mu(e, aa) \uparrow$ and $\mu(s, aa) \uparrow$. The resource monoid model acts like a semaphore, in that only one access action a can be performed at any given time.

The mutual producer–consumer model (Example 34) and the prisoner's dilemma model (Example 36) are also resource semantics models. In fact, all resource models (as specified in this section) are concurrent transitions systems (as specified in Definition 4).

PROPOSITION 42

A resource monoid model $(\mathbf{R}, \mathbf{Act}, \mu, =, \circ, e)$ is a concurrent transition system.

PROOF. By Definition 40, we have that $(\mathbf{R}, \mathbf{Act}, \mu)$ is a transition system.

Suppose some $a, b, c \in \mathbf{Act}$. By the definition of a commutative monoid, we have that $a1 = a$, $ab = ba$, and, if $a = a'$ and $b = b'$, $ab = a'b'$. Hence $=$ is an action-composition equivalence relation.

Suppose some states $r, s, r', s', r'' \in \mathbf{S}$ and actions $a, b \in \mathbf{Act}$. As $\circ$ is commutative, if $r \circ s$ is defined, then $s \circ r$ is defined. The other required properties of $\circ$ follow straightforwardly from Definition 40. ∎

If the modification function is defined for an action a on a resource R, and $\mu(R,a) = S$, then we say that there exists a transition $R \xrightarrow{a} S$, and that S is a successor of R. The notion of *bisimulation* in Definition 6 is immediately applicable to resource models.

In order to use of resource models as a semantics for MBIU, we restrict ourselves to those resource models that conform to Definitions 28 and 30. With those restrictions in place, we can then use resource monoid models as a semantics for MBIU. So we can make use of logical characterizations of notions of optimality, such as were described in Section 3, over distributed systems modelled using resource monoid models.

We conclude this section with a property of payoff functions for resource monoid models which is not true of payoff functions for generic concurrent transition systems; namely, that if a strategy chooses the unit action in some state, then the payoff of that state is always 0.

LEMMA 43
For all action payoff functions v, strategies σ, and discount factors δ, if $\sigma(s) = (1, s')$, then $u_{v,\sigma,\delta}(s) = 0$.

PROOF. By Definition 14, we have that $v(1) = 0$. By Definitions 11 and 40, we have that $s \xrightarrow{1} s$ and $s' = s$. By Definition 16, we have that $u_{v,\sigma,\delta}(s) = 0 + \delta \times u_{v,\sigma,\delta}(s)$. As $(1 - \delta) \neq 0$, we have that $u_{v,\sigma,\delta}(s) = 0$. ∎

5 Resource–process systems modelling

One modelling approach, which might be expected to form the basis of an example of our methodology in Section 2, is that based on the resource–process calculi, as given in [11, 12] and introduced in Section 4. These calculi consist of two components: resources, which describe objects that can be created, moved, and consumed; and processes, which describe the dynamics of systems, and have a more complex, algebraic structure, including sequencing, non-deterministic choice, and fixed points. Each component has a notion of composition, and so resource–process pairs have the obvious composition pairwise on the components. An action-indexed transition system can be defined in terms of a structural operational semantics over the structure of processes, so that resources and processes (i.e., the state) co-evolve: $R, E \xrightarrow{a} R', E'$.

Unfortunately, in such calculi (e.g., in [11, 12]), bisimulation fails to be a congruence for concurrent composition. As a result, the soundness direction of the Hennessy–Milner property holds only for fragments of the logic that exclude multiplicative implication ($-\!\ast$). Bisimulation fails to be a congruence for concurrent composition because of the way in which the resource semantics interacts with the resource–process operational semantics. Resources can be viewed as being 'capabilities', which enable behaviour in the process components of the pairs. When performing concurrent composition, these 'capabilities' can be exchanged between the process components of the pairs, enabling different behaviour in different compositions. This clearly violates the required congruence property.

This problem has been solved, in [2], by changing the resource semantics to ensure that 'capabilities' cannot be exchanged between process components in the operational semantics. Additional structure is added to the resource model, beyond that in [11, 12] and Section 4. The key structural modification is the introduction of additional combinatorial structure to the resource semantics — resources are bunched, being combined using either $\otimes$, corresponding to the monoidal composition $\circ$, or $\oplus$, which builds in choice — with the key property being injectivity of concurrent composition.

In this section, we review the resource–process calculi as set up in [2] and show that they are indeed examples of our methodology. In particular, we show that our analysis of utility extends to these resource–process calculi, and provide an extended example (Example 64, below) based on the 'mutual producer–consumer' introduced in Example 34, comprising distributed coordination without mutual exclusion: a mutual producer–consumer system, where each 'agent' can generate work for, and consume work from, the other. In Example 34, the 'agents' performing the production

and consumption are represented indirectly. For example, it is not possible to consider one agent's behaviour on its own; as the dynamics are directly encoded via the resource semantics, both agents are always 'present' in any given resource. Using the richer resource–process framework that we introduce in this section, we can represent the dynamics of the different agents more directly. Specifically, we represent these agents as processes. We can then demonstrate how, for example, the first entity cannot make progress when it only possesses resources that the second process can consume available to it.

The set-up of the required process calculi — henceforth known as Calculi of Bunched Resources and Processes, or CBRP — assumes the provision of certain additional data pertaining to some semantic structure $(\mathbf{Act}, \mathbf{R}, \mu, \Gamma, \mathbf{H})$ — of actions, resources, modification function, a set *redistribution functions*, and a set *hiding functions*, respectively — over which we work and which we define in the development below. The actions, resources (excepting the injective bunching structure), and modification are defined as they are in Section 4, the redistribution functions are used to specify how combinations of resources defined using $\otimes$ and $\oplus$ can be rearranged, and the hiding functions are used to bind resources to processes locally (see Definition 50, below). The modification function, the redistribution functions, and the hiding functions are all essential parts of the operational semantics (see Figure 2). Thus we should properly refer to the calculus as $(\mathbf{Act}, \mathbf{R}, \mu, \Gamma, \mathbf{H})$-CBRP. In this section, however, we suppress the prefix as, at every stage, we work with a fixed such structure.

We begin with a notion of resource which can be seen as restricting the combinatorial structure taken in Section 4 in that it considers choices between resources, and it requires the notions of composition to be injective. Let $\mathbf{R}$ be a set of resources, equipped with an 'empty' element $e \in \mathbf{R}$. We write R, S, etc. to denote resources. We consider unique (partial) concurrent composition of, and non-deterministic choice between, resources. In [11, 12, 25, 27], and other works in the relevant logic tradition, *bunches* are trees with leaves labelled by atomic resources, and internal nodes labelled by either $\oplus$ or $\otimes$. We implement bunching through the use of two injective functions; a resource is a node of a particular type if there exists some (unique) pair of resources that are mapped to the initial resource by the relevant function.

DEFINITION 44 (Resource models)
A *resource model* $(\mathbf{R}, e, \otimes, \oplus)$ is a structure consisting of a set of resources $\mathbf{R}$ with a distinguished 'empty' resource $e \in \mathbf{R}$, and two injective, partial functions $\otimes, \oplus : \mathbf{R} \times \mathbf{R} \rightharpoonup \mathbf{R}$, such that, for all $R, S, T \in \mathbf{R}$ and $\odot \in \{\oplus, \otimes\}$,

(1) $R \odot S$ is defined if and only if $S \odot R$ is defined,
(2) $R \odot (S \odot T)$ is defined if and only if $(R \odot S) \odot T$ is defined,
(3) $R \otimes e$ is defined,
(4) $R \oplus R$ is defined, and
(5) $R \otimes (S \oplus T)$ is defined if and only if $(R \otimes S) \oplus (R \otimes T)$ is defined.

Note that properties 2, 4, and 5 are only required to obtain the algebraic results (Proposition 56) and are not necessary to obtain the Hennessy–Milner correspondence via the approach in Section 2.

In the sequel, when we write an expression of the form $R \otimes S$ or $R \oplus S$, we assume that the result of the application of the partial function to its arguments is defined. Actions correspond to the events of a system. In resource–process algebra as set up in [11, 12], actions are used to determine how resources evolve. This necessitates a relationship between the concurrent structure of actions and the concurrent structure of resources. To obtain an analogous relationship in our setting (formally stated in Definition 47), we also require action composition to be injective.

DEFINITION 45 (Action model)
An *action model* (**Act**, ·, 1) is a structure consisting of a set of actions with a distinguished unit action $1 \in \textbf{Act}$, and an injective, total function ·.

In many process algebras, such as SCCS and SCRP, the commutative monoid structure of actions is used to prove various algebraic properties of states. In this section, unlike resource monoid models (Section 4), as we do not require that 1 be a unit for · with respect to syntactic equality, the actions do not form a (commutative) monoid with respect to syntactic equality. We establish that the CBRP notion of actions (Definition 45) is an action structure (as in Definition 1), a property that we will use when we demonstrate that CBRP are instances of concurrent transition systems (Proposition 54).

LEMMA 46
The structure (**Act**, ·, 1) is an action structure.

PROOF. As **Act** is closed under pairing, (**Act**, ·) is a total magma. ∎

The semantics of resources is then given by a modification function from action-resource pairs to resources.

DEFINITION 47 (Modification function)
A modification function is a partial function $\mu : \mathbf{R} \times \mathbf{Act} \rightharpoonup \mathbf{R}$ such that

- for all R and S, and all a and b, if $\mu(R,a)$, $\mu(S,b)$, and $R \otimes S$ are all defined, then $\mu(R,a) \otimes \mu(S,b)$ and $\mu(R \otimes S, ab)$ are both defined, and $\mu(R \otimes S, ab) = \mu(R,a) \otimes \mu(S,b)$ holds;
- for all R, $\mu(R,1) = R$;
- for all R and S, and all c, if $R \otimes S$ and $\mu(R \otimes S, c)$ are defined, then there exist a and b such that $c = ab$, and $\mu(R,a)$ and $\mu(S,b)$ are both defined;
- the distinguished action 1 is s.t. $\mu(e,1) = e$ and, for all s and a, if $\mu(e,a) = s$, then $a = 1$ and $s = e$.

Note that the action 1 is a unit with respect to μ's action on resources. Note also that a modification function is one of the parameters to the calculus.

Modification functions are homomorphisms with respect to the concurrent product structure of resource bunches. As a result, we cannot use the modification function to 'move' resources from one side of a concurrent product to another (such a move corresponds to changing the process to which the resources are allocated; for example, passing an object from producer to consumer). Using a modification function, we can only add or remove resources to each side of a product independently of what is on the other side of the concurrent product.

As we cannot use a modification function for redistribution of resources, instead, we make use of *redistribution functions*. In Figure 2, the rules for the operational semantics of sequential composition are

$$\frac{R, E \xrightarrow{a} R', E' \rightarrow \quad \gamma \in \Gamma}{R, E :_\gamma F \xrightarrow{a} R', E' :_\gamma F} \text{ PREFIXONE} \qquad \frac{R, E \xrightarrow{a} R', E' \nrightarrow \quad \gamma \in \Gamma}{R, E :_\gamma F \xrightarrow{a} \gamma(R'), F} \text{ PREFIXTWO.}$$

The resource–process pair $R, E :_\gamma F$ consists of a resource bunch and a sequential composition. The sequential composition consists of two processes, E and F, and a redistribution function γ. If the prefix E can evolve with the resources R to a non-blocked state, then the sequential composition evolves similarly (the PREFIXONE rule). If the prefix E can evolve with the resources R to a blocked state, then the redistribution function is applied to the resulting resources R', and the pair that consists of the redistributed resources and the suffix, $\gamma(R'), F$, is the result of the transition (the PREFIXTWO

$$\frac{}{R, a \xrightarrow{a} \mu(R,a), \mathbf{0}} \;\textsc{Act} \qquad \frac{R_i, E_i \xrightarrow{a} R_i', E_i'}{R_1 \oplus R_2, E_1 + E_2 \xrightarrow{a} R_i', E_i'} \;\textsc{Sum}, \, i \in \{1, 2\}$$

$$\frac{R_1, E_1 \xrightarrow{a_1} R_1', E_1' \quad R_2, E_2 \xrightarrow{a_2} R_2', E_2'}{R_1 \otimes R_2, E_1 \times E_2 \xrightarrow{a_1 \cdot a_2} R_1' \otimes R_2', E_1' \times E_2'} \;\textsc{Prod}$$

$$\frac{R, E \xrightarrow{a} R', E' \rightarrow \quad \gamma \in \Gamma}{R, E :_\gamma F \xrightarrow{a} R', E' :_\gamma F} \;\textsc{PrefixOne} \qquad \frac{R, E \xrightarrow{a} R', E' \nrightarrow \quad \gamma \in \Gamma}{R, E :_\gamma F \xrightarrow{a} \gamma(R'), F} \;\textsc{PrefixTwo}$$

$$\frac{h(R), E \xrightarrow{a} h(R'), E' \quad h \in \mathbf{H}}{R, \nu h.E \xrightarrow{\nu h.a} R', \nu h.E'} \;\textsc{Hide} \qquad \frac{R, E[\text{fix } X.E/X] \xrightarrow{a} R', E'}{R, \text{fix } X.E \xrightarrow{a} R'.E'} \; FV(E) \subseteq \{X\} \;\textsc{Rec}$$

FIGURE 2. Operational semantics.

rule). The redistribution function is applied to the resources so that the structure of the resulting resources will match the structure of the suffix process. Redistribution functions are total so that the evolution of a sequential composition can only be blocked by the behaviour of the prefixing process, not the redistribution of resources.

DEFINITION 48 (Redistribution functions)
A redistribution function is a total function $\gamma : \mathbf{R} \to \mathbf{R}$. Let there be a set of redistribution functions Γ whose elements are written γ, γ', etc..

Let Γ, which is one of the parameters to the calculus, include the identity function. From a modelling perspective, we argue that the use of redistribution functions encourages good discipline with respect to making decisions about how resources are allocated to processes within a system. In [9, 11, 12], following a transition, all possible allocations are possible, and a system can non-deterministically choose between them. In the resource–process modelling methodology used in this section, whenever resources are to be reallocated (i.e., following each reduction step, within a sequential composition), a conscious modelling decision is required as to where the resources should be allocated.

In classical process calculi, *restriction* is used to ensure that certain behaviour is only visible, or accessible, in certain parts of a system. A similar feature can be incorporated into resource–process modelling [12]. The *hiding* operator on processes associates additional resources with the process to which it is applied. If a resource–process pair is allocated additional resources, it may be able to perform additional actions. This behaviour must then be restricted, however; only actions that could be performed without the additional resources must be visible beyond the process where the hidden resources are available. First, we define a notion of action containment, so that we can formalize the notion of 'additional behaviour'.

DEFINITION 49 (Action-containment order)
We define $\le$ to be the least reflexive-transitive relation on actions such that $1 \le \alpha$, for any atomic action α, and if $a \le a'$ and $b \le b'$, then $a \cdot b \le a' \cdot b'$.

Then, we define hiding functions on actions and resources. In Figure 2, the rule for the operational semantics of hiding functions is

$$\frac{h(R), E \xrightarrow{a} h(R'), E' \quad h \in \mathbf{H}}{R, \nu h.E \xrightarrow{\nu h.a} R', \nu h.E'} \text{ HIDE.}$$

A resource–process pair $R, \nu h.E$ evolves by stripping the hiding operator $\nu h.$ from the process component and applying the hiding function h to the resource component, resulting in the resource–process pair $h(R), E$. Following the evolution of the transformed state, the resulting pair $h(R'), E'$ is modified by applying the inverse of the hiding function to the resource component and adding the hiding operator to the process component, resulting in the resource–process pair $R', \nu h.E'$. To ensure that a hiding function and its inverse can be uniquely applied, hiding functions on resources are bijections. Moreover, the action performed in the evolution of the transformed state must be suitably transformed to restrict external visibility of actions that can only be performed with the additional resources.

DEFINITION 50 (Hiding functions)
Let $(\mathbf{R}, e, \otimes, \oplus)$ be a resource model and μ be a modification function. A function $h : \mathbf{R} \to \mathbf{R}$ on a resource model is a hiding function if it is a bijection. Let there be a set of hiding functions $\mathbf{H}$ whose elements are written h, h', etc.. Define $A : \mathbf{H} \to \mathbf{Act} \to \mathcal{P}(\mathbf{Act})$ such that

$$A(h, a) = \{b \leq a \mid \text{for all } R, S \in \mathbf{R}, \mu(h(R), a) = h(S) \text{ implies } \mu(R, b) = S\}.$$

Then, a *hiding function on actions* $\nu : \mathbf{H} \to \mathbf{Act} \to \mathbf{Act}$ is defined as

$$\nu h.a = \begin{cases} \sup(A(h, a)) & \text{if } \sup(A(h, a)) \text{ is defined and unique} \\ 1 & \text{otherwise.} \end{cases}$$

Let $\mathbf{H}$, which is one of the parameters to the calculus, include the identity function. Next, we define processes formally.

DEFINITION 51 (Processes)
Processes are formed according to the following grammar:

$$E ::= \mathbf{0} \mid X \mid a \mid E + E \mid E \times E \mid E :_\gamma E \mid \nu h.E \mid \text{fix} X.E.$$

Here, $\mathbf{0}$ is the null process, X is a process variable, a is an action, $\gamma \in \Gamma$ is a redistribution function, and $h \in \mathbf{H}$ is a hiding function. Let $\mathbf{Proc}$ be the set of all processes, and E, F etc. denote processes. The process $\mathbf{1}$, which performs the action 1 infinitely, is denoted as $\mu X.1 :_{id} X$.

Closed processes are those processes that contain no free variables. A *state* is a pair consisting of a resource and a closed process. Let $\mathbf{State}$ be the set of all states, and $\mathbf{CState}$ be the set of all closed states.

The operational behaviour of a closed state is defined by a labelled family of transition relations

$$\xrightarrow{a} \quad \subseteq \quad \mathbf{CState} \times \mathbf{Act} \times \mathbf{CState},$$

The family is defined recursively using the derivation rules in Figure 2.

An action process reduces according to the modification function μ. Nondeterminism is introduced solely through the presence of sums. There, a choice must be made both in the process component and the resource component. Product processes distribute the resources according to the multiplicative structure in the resources.

Sequential composition behaves slightly counter-intuitively. If the prefix is reduced to a non-blocking state, then the sequential composition follows similarly. If the prefix process is reduced to a blocking state, then the sequential composition reduces to the resource that results from applying the redistribution function to the residual resources from evolving the prefix, and the suffix. The redistribution function is used to redistribute the resources between the process components, following a reduction that moves to the second part of a sequential composition. It should be noted that the use of process prefixing, rather than action prefixing, is a deliberate design decision, made so that models can more intuitively reflect the structure of the system they abstract.

We can then show that all CBRP, equipped with a suitable notion of equivalence of actions and composition of states, are concurrent transition systems (as specified in Definition 4).

DEFINITION 52 (resource–process action equivalence)
Let the relation $\equiv$ be the least action equivalence relation such that, for all actions a, b, c, $a \cdot (b \cdot c) \equiv (a \cdot b) \cdot c$.

DEFINITION 53 (Concurrent composition of resource–process states)
The concurrent composition of resource–process states $\circ$ is the partial function $(R_1, E_1) \circ (R_2, E_2) = (R_1 \otimes R_2, E_1 \times E_2)$, that is defined if and only if $R_1 \otimes R_2$ is defined.

PROPOSITION 54
The structure $(\mathbf{CState}, \mathbf{Act}, \rightarrow, \equiv, \circ, (e, \mathbf{1}))$ is a concurrent transition system.

PROOF. By Lemma 46, $\mathbf{Act}$ is an action structure. Then, we straightforwardly have that $(\mathbf{CState}, \mathbf{Act}, \rightarrow)$ is a transition system, and $\equiv$ is an action-composition equivalence relation.

Suppose some states $(R, E), (S, F), (R', E'), (S', F'), (T, G) \in \mathbf{CState}$ and actions $a, b \in \mathbf{Act}$. By Definition 44, we have that, if $R \otimes S$ is defined, then $S \otimes R$ is defined, and hence if $(R, E) \circ (S, F)$ is defined, then $(S, F) \circ (R, E)$ is defined. Suppose that $(R, E) \circ (S, F)$ is defined, $R, E \xrightarrow{a} R', E'$ and $S, F \xrightarrow{b} S', F'$. By the PROD rule, we have that $R \otimes S, E \times F \xrightarrow{ab} R' \otimes S', E' \times F'$. Suppose that $R \otimes S, E \times F \xrightarrow{c} T, G$. By the PROD rule, we have that there exist a, b, R', S', E', F' such that $c = ab$, $T = R' \otimes S'$, $G = E' \times F'$, $R, E \xrightarrow{a} R', E'$ and $S, F \xrightarrow{b} S', F'$. Suppose that $e, \mu X.1 :_{id} X \xrightarrow{a} S', F'$. By the FIX, PREFIXTWO and ACT rules, and Definition 47, we have that $S' = e$, $F' = \mu X.1 :_{id} X$ and $a = 1$. By Definition 44, we have that $R \otimes e$ is defined, and hence $(R, E) \circ (e, \mathbf{1})$ is defined. ∎

In order to use CBRP transition systems as a semantics for MBIU, we must restrict ourselves to those calculi that conform to Definitions 28 and 30. In order to obtain the property specified in Definition 28, it is sufficient to restrict ourselves to those calculi that have the following property:

DEFINITION 55 ($\sim$-resource-closed CBRP)
A calculus is $\sim$-resource-closed if, for all R_1, E_1, S_1, F_1, R_2, E_2, S_2, F_2, if $R_1, E_1 \sim S_1, F_1$ and $R_2, E_2 \sim S_2, F_2$, then $R_1 \otimes R_2$ (respectively, $R_1 \oplus R_2$) is defined if and only if $S_1 \otimes S_2$ (respectively, $S_1 \oplus S_2$) is defined.

Henceforth, we consider only CBRP that are $\sim$-resource-closed. An immediate result is that concurrent compositions of bisimilar resource–process pairs are bisimilar (Lemma 29), and that we have the Hennessy–Milner completeness result (Theorems 32 and 33).

In order to reason equationally about resource–process states, it is also useful to establish various algebraic properties concerning concurrent composition and choice. Notable standard algebraic properties of process calculi are commutativity and associativity of concurrent composition. We obtain such properties for CBRP.

PROPOSITION 56 (Algebraic properties)
For all bunched resources $R, S, T \in \mathbf{R}$ and closed processes E, F, G,

(Commutativity of choice)	$R \oplus S, E + F \sim S \oplus R, F + E$
(Unit of choice)	$R \oplus R, E + \mathbf{0} \sim R, E$
(Associativity of choice)	$R \oplus (S \oplus T), E + (F + G) \sim$
	$\qquad (R \oplus S) \oplus T, (E + F) + G$
(Commutativity of product)	$R \otimes S, E \times F \sim S \otimes R, F \times E$
(Unit of product)	$R \otimes e, E \times \mathbf{1} \sim R, E$
(Zero property of product)	$R \otimes e, E \times \mathbf{0} \sim e, \mathbf{0}$
(Associativity of product)	$R \otimes (S \otimes T), E \times (F \times G) \sim$
	$\qquad (R \otimes S) \otimes T, (E \times F) \times G$
(Distribution of product	$R \otimes (S \oplus T), E \times (F + G) \sim$
over choice)	$\qquad (R \otimes S) \oplus (R \otimes T), (E \times F) + (E \times G)$.

PROOF. **Commutativity of choice.** Let

$$\mathcal{R} = \{((R \oplus S, E + F), (S \oplus R, F + E)) \mid E \text{ and } F \text{ are closed}\} \cup \sim.$$

Suppose that $R \oplus S, E + F \xrightarrow{a} T, G$. By the SUM rule, either $R, E \xrightarrow{a} T, G$ or $S, F \xrightarrow{a} T, G$. As $R \oplus S$ is defined, by Definition 44[1], $S \oplus R$ is defined. By the SUM rule, $S \oplus R, F + E \xrightarrow{a} T, G$. By Definition 52, $a \equiv a$. By Lemma 7, $T, G \sim T, G$, and hence $(T, G)\mathcal{R}(T, G)$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.

Unit of choice. Let
$$\mathcal{R} = \{((R \oplus R, E \oplus \mathbf{0}), (R, E)) \mid E \text{ is closed}\} \cup \sim.$$

Suppose that $R \oplus R, E \oplus \mathbf{0} \xrightarrow{a} T, G$. By the SUM rule, either $R, E \xrightarrow{a} T, G$ or $R, \mathbf{0} \xrightarrow{a} T, G$. By Figure 2, $R, \mathbf{0} \not\rightarrow$, and hence $R, E \xrightarrow{a} T, G$. By Definition 52, $a \equiv a$. As simulation is an equivalence relation, $T, G \sim T, G$, and hence $(T, G)\mathcal{R}(T, G)$.

Suppose that $R, E \xrightarrow{a} T, G$. By Definition 44[3], $R \oplus R$ is defined. By the SUM rule, $R \oplus R, E \oplus \mathbf{0} \xrightarrow{a} T, G$. By Lemma 7, $T, G \sim T, G$, and hence $(T, G)\mathcal{R}(T, G)$.

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

Associativity of choice. Let

$$\mathcal{R} = \{((R \oplus (S \oplus T), E + (F + G)), ((R \oplus S) \oplus T, (E + F) + G)) \mid E, F, G \text{ are closed}\} \cup \sim.$$

Suppose that $R \oplus (S \oplus T), E + (F + G) \xrightarrow{a} U, H$. By repeated application of the SUM rule, either $R, E \xrightarrow{a} U, H, S, F \xrightarrow{a} U, H$, or $T, G \xrightarrow{a} U, H$. As $R \oplus (S \oplus T)$ is defined, by Definition 44 (2), $(R \oplus S) \oplus T$ is defined. Suppose that $R, E \xrightarrow{a} U, H$. By repeated application of the SUM rule, $R \oplus S, E + F \xrightarrow{a} U, H$ and $(R \oplus S) \oplus T, (E + F) + G \xrightarrow{a} U, H$. Suppose that $S, F \xrightarrow{a} U, H$. By repeated application of the SUM

rule, $R \oplus S, E + F \xrightarrow{a} U, H$ and $(R \oplus S) \oplus T, (E + F) + G \xrightarrow{a} U, H$. Suppose that $T, G \xrightarrow{a} U, H$. By the SUM rule, $(R \oplus S) \oplus T, (E + F) + G \xrightarrow{a} U, H$. By Definition 52, $a \equiv a$. By Lemma 7, $U, H \sim U, H$, and hence $(U, H) \mathcal{R} (U, H)$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.

Commutativity of product. Let

$$\mathcal{R} = \{((R \otimes S, E \times F), (S \otimes R, F \times E)) \mid E \text{ and } F \text{ are closed}\}.$$

Suppose that $R \otimes S, E \times F \xrightarrow{c} T, G$. By the PROD rule, there exist a, b, R', S', E', and F', such that $c = ab$, $T = R' \otimes S'$, $G = E' \times F'$, $R, E \xrightarrow{a} R', E'$, and $S, F \xrightarrow{b} S', F'$. By Definition 44 (2), $S \otimes R$ and $S' \otimes R'$ are defined. By the PROD rule, $S \otimes R, F \times E \xrightarrow{ba} S' \otimes R', F' \times E'$. By Definition 52, $ab \equiv ba$. We immediately have that $(R' \otimes S', E' \times F') \mathcal{R} (S' \otimes R', F' \times E')$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.

Unit of product. Let

$$\mathcal{R} = \{((R \otimes e, E \times \mathbf{1}), (R, E)) \mid E \text{ is closed}\}.$$

Suppose that $R \otimes e, E \times \mathbf{1} \xrightarrow{c} T, G$. By the PROD rule, there exist a, b, R', E', S', F' such that $c = ab$, $T = R' \otimes S'$, $G = E' \times F'$, $R, E \xrightarrow{a} R', E'$, and $e, \mathbf{1} \xrightarrow{b} S', F'$. As $\mathbf{1} = \text{fix } X.(1 : X)$, by the REC, PREFIXTWO and ACT rules, $b = 1$, $S' = \mu(e, 1)$, and $F' = \mathbf{1}$. By Definition 40, $S' = e$. By Definition 52, $a1 \equiv a$. We immediately have that $(R' \otimes e, E' \times \mathbf{1}) \mathcal{R} (R', E')$.

Suppose that $R, E \xrightarrow{a} R', E'$. As $\mathbf{1} = \text{fix } X.(1 : X)$, by the REC, PREFIXTWO and ACT rules, $e, \mathbf{1} \xrightarrow{1} \mu(e, 1), \mathbf{1}$. By Definition 40, $S' = e$. By Definition 44[3], $R \otimes e$ and $R' \otimes e$ are defined. By the PROD rule, $R \otimes e, E \times \mathbf{1} \xrightarrow{a1} R' \otimes e, E' \otimes \mathbf{1}$. By Definition 52, $a1 \equiv a$. We immediately have that $(R' \otimes e, E' \times \mathbf{1}) \mathcal{R} (R', E')$.

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

Zero property of product. Let

$$\mathcal{R} = \{((R \otimes e, E \times \mathbf{0}), (e, \mathbf{0})) \mid E \text{ is closed}\}.$$

Suppose that $R \otimes e, E \times \mathbf{0} \rightarrow$. By the PROD rule, $R, E \rightarrow$ and $e, \mathbf{0} \rightarrow$. This is a contradiction as, by Figure 2, $e, \mathbf{0} \nrightarrow$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.

Associativity of product. Let

$$\mathcal{R} = \{((R \otimes (S \otimes T), E \times (F \times G)), ((R \otimes S) \otimes T, (E \times F) \times G)) \mid E, F, G \text{ are closed}\}.$$

Suppose that $R \otimes (S \otimes T), E \times (F \times G) \xrightarrow{d} U, H$. By repeated application of the PROD rule, there exist a, b, c, R', S', T', E', F', G', such that $d = a(bc)$, $U = R' \otimes (S' \otimes T')$, $H = E' \times (F' \times G')$, $R, E \xrightarrow{a} R', E'$, $S, F \xrightarrow{b} S', F'$. and $T, G \xrightarrow{c} T', G'$. By Definition 44 (2), we have that $(R \otimes S) \otimes T$ is defined. By further application of the PROD rule, $(R \otimes S) \otimes T, (E \times F) \times G \xrightarrow{(ab)c} (R' \otimes S') \otimes T', (E' \times F') \times G'$. By Definition 52, $a(bc) \equiv (ab)c$. We then have that $(R' \otimes (S' \otimes T'), E' \times (F' \times G')) \mathcal{R} ((R' \otimes S') \otimes T', (E' \times F') \times G')$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation.

Distribution of product over choice. Let

$$\mathcal{R} = \left\{ \begin{array}{c} (R \otimes (S \oplus T), E \times (F+G)), \\ ((R \otimes S) \oplus (R \otimes T), (E \times F) + (E \times G)) \end{array} \right) \mid E, F, G \text{ are closed} \right\}.$$
$$\cup \sim$$

Suppose that $(R \otimes (S \oplus T), E \times (F+G)) \xrightarrow{c} U, H$. By Definition 44[4], $(R \otimes S) \oplus (R \otimes T)$ is defined. By the Prod rule, there exists a, b, R', E', V', I' such that $c = ab, U = R' \otimes V', H = E' \times I', R, E \xrightarrow{a} R', E'$ and $S \oplus T, F+G \xrightarrow{b} V', I'$. By the Sum rule, either $S, F \xrightarrow{b} V', I'$ or $T, G \xrightarrow{b} V', I'$. Suppose the former. By the Prod rule, $R \otimes S, E \times F \xrightarrow{ab} R' \otimes V', E' \times I'$. By the Sum rule, $(R \otimes S) \oplus (R \otimes T), (E \times F) + (E \times G) \xrightarrow{ab} R' \otimes V', E' \times I'$. Suppose the latter. By the Prod rule, $R \otimes T, E \times G \xrightarrow{ab} R' \otimes V', E' \times I'$. By the Sum rule, $(R \otimes S) \oplus (R \otimes T), (E \times F) + (E \times G) \xrightarrow{ab} R' \otimes V', E' \times I'$. By Definition 52, $ab \equiv ab$. By Lemma 7, $U, H \sim U, H$, and hence $(U, H)\mathcal{R}(U, H)$.

The other case is similar. Hence $\mathcal{R}$ is closed and a bisimulation. ∎

COROLLARY 57

For all bunched resources $R, S, T \in \mathbf{R}$ and closed processes E, F, G,

$$R \oplus R, E+F \sim R \oplus R, F+E \quad R \oplus S, E+E \sim S \oplus R, E+E$$
$$R \otimes R, E \times F \sim R \otimes R, F \times E \quad R \otimes S, E \times E \sim S \otimes R, E \times E.$$

Moreover, it is possible to reason equationally about the *payoffs* of resource–process pairs. We define a class of strategies that generate payoffs functions whose output will follow the structure of the resource–process pairs (Proposition 62). These strategies are known as *elementary strategies*.

DEFINITION 58

An elementary CBRP strategy is a strategy σ such that, for all actions a, b, c, resources R, S, R', S', T and closed processes E, F, E', F', G,

(1) if $\sigma(R, E) = (a, (R', E'))$ and $(R', E') \not\rightarrow$, then $\sigma(R, E :_\gamma F) = (a, (\gamma(R'), F))$,
(2) if $\sigma(R, E) = (a, (R', E'))$ and $(R', E') \rightarrow$, then $\sigma(R, E :_\gamma F) = (a, (R', E' :_\gamma F))$,
(3) $\sigma(R_1 \oplus R_2, E_1 + E_2) = \sigma(R_i, E_i)$, for some $i \in \{1, 2\}$,
(4) if $R \otimes S \downarrow$, then $c = ab, T = R' \otimes S', G = E' \times F', \sigma(R, E) = (a, (R', E'))$, and $\sigma(S, F) = (b, (S', F'))$ if and only if

$$\sigma(R \otimes S, E \times F) = (c, (T, G)),$$

and
(5) $\sigma(R, \text{fix } X.E) = \sigma(R, E[\text{fix } X.E/X])$.

In order to obtain the equational result for concurrent composition (Proposition 62[8]), we establish two auxiliary lemmas. First, we show that if a strategy can be applied to a state at least n times, then we can unroll the definition of state payoff functions (Definition 16) n times, and the payoff of the state is the sum of the discounted payoffs of the action chosen at each step and the discounted payoff of the state reached after n steps.

LEMMA 59

For all states (R, E) and natural numbers $n \in \mathbb{N}$, if $u_{v, \sigma, \delta}(R, E)$ and $\sigma_{state}^n(R, E)$ are defined, then

$$u_{v, \sigma, \delta}(R, E) = \left(\Sigma_{i=0}^n \delta^i \times v(\sigma_{act}^i(R, E)) \right) + \delta^{n+1} \times u_{v, \sigma, \delta}(\sigma_{state}^n(R, E)).$$

PROOF. By induction over n.

Suppose $n=0$. We immediately have that

$$u_{v,\sigma,\delta}(R,E) = v(\sigma_{act}^0(R,E)) + \delta \times u_{v,\sigma,\delta}(\sigma_{state}^0(R,E)).$$
$$= v(\sigma_{act}^n(R,E)) + \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^n(R,E)).$$

Suppose $n>0$. Let $\sigma(R,E)=(a,(R',E'))$. By Definition 19, $\sigma_{state}^{n-1}(R',E')=S,F$. By Definition 16, $v(a)$ and $u_{v,\sigma,\delta}(R',E')$ are defined. By the induction hypothesis,

$$u_{v,\sigma,\delta}(R',E') = \Sigma_{j=0}^{n-1} \delta^j \times v(\sigma_{act}^j(R,E)) + \delta^n \times u_{v,\sigma,\delta}(\sigma_{state}^{n-1}(R',E')).$$

By Definition 16,

$$\begin{aligned}
u_{v,\sigma,\delta}(R',E') &= v(a) + \delta \times u_{v,\sigma,\delta}(R',E') \\
&= v(\sigma_{act}^0(R,E)) + \delta \times \left(\begin{array}{l} \Sigma_{j=0}^{n-1} \delta^j \times v(\sigma_{act}^j(R',E')) + \\ \delta^n \times u_{v,\sigma,\delta}(\sigma_{state}^{n-1}(R',E'))) \end{array} \right) \\
&= v(\sigma_{act}^0(R,E)) + (\Sigma_{j=0}^{n-1} \delta^{j+1} \times v(\sigma_{act}^j(R',E')) + \\
&\qquad \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^{n-1}(R',E'))) \\
&= v(\sigma_{act}^0(R,E)) + (\Sigma_{i=1}^{n} \delta^i \times v(\sigma_{act}^{i-1}(R',E')) + \\
&\qquad \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^{n-1}(R',E'))) \\
&= v(\sigma_{act}^0(R,E)) + (\Sigma_{i=1}^{n} \delta^i \times v(\sigma_{act}^i(R,E)) + \\
&\qquad \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^{n-1}(R',E'))) \\
&= \left(\Sigma_{i=0}^{n} \delta^i \times v(\sigma_{act}^i(R,E)) \right) + \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^n(R,E)). \quad \blacksquare
\end{aligned}$$

Second, we show that if a strategy can be applied to a concurrent composition of states at least n times, for all number of applications of the strategy up to n, the action and state chosen by the strategy is the concurrent composition of the actions and states chosen by the strategy on the states of the concurrent composition. This provides a way to compositionally reason about actions and states that are chosen by a strategy for a sequence of transitions of a concurrent composition of states.

LEMMA 60
For all elementary strategies σ, natural numbers n, resources R,S,T, and closed processes E,F,G, if $\sigma_{state}^n(R \otimes S, E \times F) = T, G$, then, there exist resources R',S' and closed processes E',F' such that $T = R' \otimes S'$ and $G = E' \times F'$, and for all $0 \le i \le n$,

$$\begin{aligned}
\sigma_{act}^i(R \otimes S, E \times F) &= \sigma_{act}^i(R,E) \cdot \sigma_{act}^i(S,F) \\
\sigma_{state}^i(R \otimes S, E \times F) &= \sigma_{state}^i(R,E) \circ \sigma_{state}^i(S,F).
\end{aligned}$$

PROOF. By induction on n.

Suppose $n=0$. By Definition 19, there exists an action c such that

$$\sigma(R \otimes S, E \times F) = (c,(T,G)).$$

By Definition 11, $R \otimes S, E \times F \xrightarrow{c} T, G$. As the PROD rule is the only operational semantics rule to evolve concurrent compositions, we have that there exist resources R',S' and closed processes E',F' such that $T = R' \otimes S'$ and $G = E' \times F'$. By Definition 58[4], there exist a, b, such that $c=ab$, $T=$

$R' \otimes S'$, $G = E' \times F'$, $\sigma(R,E) = (a,(R',E'))$ and $\sigma(S,F) = (b,(S',F'))$. As $0 \le i \le n$, the only possible value of i is 0. By Definitions 19 and 53,

$$\sigma^0_{act}(R \otimes S, E \times F) = \sigma^0_{act}(R,E) \cdot \sigma^0_{act}(S,F)$$
$$\sigma^0_{state}(R \otimes S, E \times F) = \sigma^0_{state}(R,E) \circ \sigma^0_{state}(S,F).$$

Suppose $n > 0$. By Definition 19, there exist c, U, H such that

$$\sigma(R \otimes S, E \times F) = (c,(U,H))$$

and $\sigma^{n-1}_{state}(U,H) = T,G$. By Definition 11, $R \otimes S, E \times F \xrightarrow{c} U, H$. As the PROD rule is the only operational semantics rule to evolve concurrent compositions, we have that there exist resources R'', S'' and closed processes E'', F'' such that $U = R'' \otimes S''$ and $H = E'' \times F''$. By the induction hypothesis, there exist resources R', S' and closed processes E', F' such that $T = R' \otimes S'$ and $G = E' \times F'$, and for all $0 \le i \le (n-1)$,

$$\sigma^i_{act}(R'' \otimes S'', E'' \times F'') = \sigma^i_{act}(R'',E'') \cdot \sigma^i_{act}(S'',F'')$$
$$\sigma^i_{state}(R'' \otimes S'', E'' \times F'') = \sigma^i_{state}(R'',E'') \circ \sigma^i_{state}(S'',F'').$$

By Definition 58[4], there exist a, b, such that $c = ab$, $\sigma(R,E) = (a,(R'',E''))$ and $\sigma(S,F) = (b,(S'',F''))$, and hence

$$\sigma^0_{act}(R \otimes S, E \times F) = \sigma^0_{act}(R,E) \cdot \sigma^0_{act}(S,F)$$
$$\sigma^0_{state}(R \otimes S, E \times F) = \sigma^0_{state}(R,E) \circ \sigma^0_{state}(S,F).$$

By Definition 19, $\sigma^i_{act}(R'' \otimes S'', E'' \times F'') = \sigma^{i+1}_{act}(R \otimes S, E \times F)$ and $\sigma^i_{state}(R'' \otimes S'', E'' \times F'') = \sigma^{i+1}_{state}(R \otimes S, E \times F)$. Hence, for all $0 \le i \le n$,

$$\sigma^i_{act}(R \otimes S, E \times F) = \sigma^i_{act}(R,E) \cdot \sigma^i_{act}(S,F)$$
$$\sigma^i_{state}(R \otimes S, E \times F) = \sigma^i_{state}(R,E) \circ \sigma^i_{state}(S,F). \qquad \blacksquare$$

We define the payoff of a n-length prefix of a trace, for use when considering sequencing.

DEFINITION 61 (Bounded utility calculation)

$$u_{v,\sigma,\delta,n}(R,E) = \begin{cases} v(a) & \text{if } \sigma(R,E) = (a,(R',E')), v(a) \downarrow, \\ & \quad \text{and } n = 0 \\ v(a) + \delta \times u_{v,\sigma,\delta,(n-1)}(R',E') & \text{if } \sigma(R,E) = (a,(R',E')), \\ & \quad v(a) \downarrow, u_{v,\sigma,\delta,(n-1)}((R',E')) \downarrow, \\ & \quad \text{and } 0 < n \\ -\infty & \text{if } \sigma(R,E) = \bullet \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Now we can show that payoffs of states, determined using elementary strategies, have intuitive equational properties over the structure of states, notably, that the payoff of a non-deterministic choice is the payoff of one of the possible choices and that the payoff of a concurrent composition is the sum of the payoffs of the concurrent components.

PROPOSITION 62

For all valuation functions v, elementary strategies σ, discount factors δ, resources R, R_1, R_2, R', and closed processes E, E', F, E_1, E_2, we have the following:

(1) $u_{v,\sigma,\delta}(R, \mathbf{0}) = -\infty$;

(2) $u_{v,\sigma,\delta}(R, \mathbf{1}) = 0$;

(3) If $R, E \nrightarrow$, then $u_{v,\sigma,\delta}(R, E :_\gamma F) = -\infty$;

(4) If $\sigma_{last}(R, E) = n$, $\sigma_{state}^n(R, E) = R', E'$, and $u_{v,\sigma,\delta}(R, E :_\gamma F)$ is defined, then

$$u_{v,\sigma,\delta}(R, E :_\gamma F) = u_{v,\sigma,\delta,n}(R, E) + \delta^{(n+1)} \times u_{v,\sigma,\delta}(\gamma(R'), F);$$

(5) If $u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + E_2)$ is defined, then

$$u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + E_2) = u_{v,\sigma,\delta}(R_i, E_i),$$

   for some $i \in \{1, 2\}$;

(6) If $u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + \mathbf{0})$ is defined, then

$$u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + \mathbf{0}) = u_{v,\sigma,\delta}(R_1, E_1);$$

(7) If v is total, then

$$u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = u_{v,\sigma,\delta}(R_1, E_1) + u_{v,\sigma,\delta}(R_2, E_2);$$

(8) If $u_{v,\sigma,\delta}(R, \text{fix } X . E)$ is defined, then

$$u_{v,\sigma,\delta}(R, \text{fix } X . E) = u_{v,\sigma,\delta}(R, E[\text{fix } X . E / X]).$$

PROOF.

   1. This follows from Definitions 11 and 16.

   2. This follows similarly to Lemma 22.

   3. As $R, E \nrightarrow$, by PREFIXONE and PREFIXTWO rules, $R, E :_\gamma F \nrightarrow$. By Definition 16, $u_{v,\sigma,\delta}(R, E :_\gamma F) = -\infty$.

   4. By induction over n.

   Suppose that $n = 0$. By Definition 19, we have that there exists an action a such that $\sigma(R, E) = (a, (R', E'))$ and $\sigma(R', E') = \bullet$. By Definition 11, $R', E' \nrightarrow$. By Definition 63, $u_{v,\sigma,\delta,0}(R, E) = v(a)$. By Definition 58 (1), $\sigma(R, E :_\gamma F) = (a, (\gamma(R'), F))$. By Definition 16, $u_{v,\sigma,\delta}(R, E :_\gamma F) = v(a) + \delta \times u_{v,\sigma,\delta}(\gamma(R'), F)$, and hence

$$u_{v,\sigma,\delta}(R, E :_\gamma F) = u_{v,\sigma,\delta,0}(R, E) + \delta^{(n+1)} \times u_{v,\sigma,\delta}(\gamma(R'), F).$$

   Suppose that $n < 0$. By Definition 19, we have that there exists an action a such that $\sigma(R, E) = (a, (R'', E''))$ and $\sigma_{state}^{n-1}(R'', E'') = R', E'$. By Definition 63, $u_{v,\sigma,\delta,n}(R, E) = v(a) + \delta \times u_{v,\sigma,\delta,(n-1)}(R'', E'')$. By the induction hypothesis,

$$u_{v,\sigma,\delta}(R'', E'' :_\gamma F) = u_{v,\sigma,\delta,(n-1)}(R'', E'') + \delta^n \times u_{v,\sigma,\delta}(\gamma(R'), F).$$

By Definition 58 (1), $\sigma(R, E :_\gamma F) = (a, (R'', E'' :_\gamma F))$. By Definition 16,

$$
\begin{aligned}
u_{v,\sigma,\delta}(R, E :_\gamma F) &= v(a) + \delta \times u_{v,\sigma,\delta}(R'', E'' :_\gamma F) \\
&= v(a) + \delta \times (u_{v,\sigma,\delta,(n-1)}(R'', E'') + \delta^n \times u_{v,\sigma,\delta}(\gamma(R'), F)) \\
&= v(a) + \delta \times (u_{v,\sigma,\delta,(n-1)}(R'', E'')) + \delta^{(n+1)} \times u_{v,\sigma,\delta}(\gamma(R'), F) \\
&= u_{v,\sigma,\delta,n}(R, E) + \delta^{(n+1)} \times u_{v,\sigma,\delta}(\gamma(R'), F).
\end{aligned}
$$

5. By Definition 58 (2), $\sigma(R_1 \oplus R_2, E_1 + E_2) = \sigma(R_i, E_i)$, for some $i \in \{1, 2\}$. By Definition 16, we immediately have that $u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + E_2) = u_{v,\sigma,\delta}(R_i, E_i)$.

6. Suppose that $R_1 \oplus R_2, E_1 + \mathbf{0} \to$. By the Sum rule, $R_1, E_1 \to$. By Definition 11, $\sigma(R_1, E_1) \neq \bullet$, $\sigma(R_2, \mathbf{0}) = \bullet$, and $\sigma(R_1 \oplus R_2, E_1 + \mathbf{0}) \neq \bullet$. By Case 5, $\sigma(R_1 \oplus R_2, E_1 + \mathbf{0}) = \sigma(R_1, E_1)$. By Definition 16, we immediately have that

$$
u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + \mathbf{0}) = u_{v,\sigma,\delta}(R_1, E_1).
$$

Suppose that $R_1 \oplus R_2, E_1 + \mathbf{0} \not\to$. By the Sum rule, $R_1, E_1 \not\to$. By Definition 11, $\sigma(R_1, E_1) = \bullet$ and $\sigma(R_2, \mathbf{0}) = \bullet$. By Proposition 62[5], $\sigma(R_1 \oplus R_2, E_1 + \mathbf{0}) = \bullet$. By Definition 16, we immediately have that

$$
u_{v,\sigma,\delta}(R_1 \oplus R_2, E_1 + \mathbf{0}) = u_{v,\sigma,\delta}(R_1, E_1) = -\infty.
$$

7. To prove this property, we must consider three cases. The first is where the concurrent composition can make no transitions. The second is where the concurrent composition can make a finite number of transitions. The third is where the concurrent composition can make an infinite number of transitions.

For the first case, suppose that $R_1 \otimes R_2, E_1 \times E_2 \not\to$. By Definition 11, $\sigma(R_1 \otimes R_2, E_1) = \bullet$. By Definition 16, $u_{v,\delta,\sigma}(R_1 \otimes R_2, E_1 \times E_2) = -\infty$. By Definition 58[4], there exists $i \in \{1, 2\}$ such that $\sigma(R_i, E_i) = \bullet$. By Definition 16, $u_{v,\delta,\sigma}(R_i, E_i) = -\infty$. Hence $u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = u_{v,\sigma,\delta}(R_1, E_1) + u_{v,\sigma,\delta}(R_2, E_2)$.

For the second case, suppose that $R_1 \otimes R_2, E_1 \times E_2 \to$ and that there exists some $n \in \mathbb{N}$ such that $\sigma_{last}(R_1 \otimes R_2, E_1 \times E_2) = n$. Let $\sigma_{state}^n(R_1 \otimes R_2, E_1 \times E_2) = (T, G)$. By Definition 19, $T, G \not\to$. By Definition 11, $\sigma(T, G) = \bullet$. By Definition 16, $u_{v,\delta,\sigma}(T, G) = -\infty$. By Lemma 60, there exist R_1', R_2', E_1', E_2' such that $T = R_1' \otimes R_2'$, $G = E_1' \times E_2'$, and, for all $0 \le i \le n$,

$$
\sigma_{act}^i(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{act}^i(R_1, E_1) \cdot \sigma_{act}^i(R_2, E_2) \tag{1}
$$

$$
\sigma_{state}^i(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{state}^i(R_1, E_1) \circ \sigma_{state}^i(R_2, E_2). \tag{2}
$$

In particular, we have that $\sigma_{act}^n(R_1, E_1) = R_1', E_1'$ and $\sigma_{act}^n(R_2, E_2) = R_2', E_2'$. By Lemma 59, as v is total,

$$
u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = \tag{3}
$$

$$
\left(\Sigma_{i=0}^n \delta^i \times v(\sigma_{act}^i(R_1 \otimes R_2, E_1 \times E_2)) \right) \tag{4}
$$

$$
+
$$

$$
\delta^{n+1} \times u_{v,\sigma,\delta}(T, G) \tag{5}
$$

$$u_{v,\sigma,\delta}(R_1, E_1) = \tag{6}$$

$$\left(\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^i(R_1, E_1))\right) \tag{7}$$

$$+$$

$$\delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{act}^n(R_1, E_1)) \tag{8}$$

$$u_{v,\sigma,\delta}(R_2, E_2) = \tag{9}$$

$$\left(\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^i(R_2, E_2))\right) \tag{10}$$

$$+$$

$$\delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{act}^n(R_1, E_1)). \tag{11}$$

By the PROD rule, either $R_1', E_1' \nrightarrow$ or $R_2', E_2' \nrightarrow$. So, by Definitions 11 and 16, either $u_{v,\sigma,\delta}(R_1', E_1') = -\infty$ or $u_{v,\sigma,\delta}(R_1', E_1') = -\infty$. As a result, $u_{v,\sigma,\delta}(T, G) = u_{v,\sigma,\delta}(R_1', E_1') + u_{v,\sigma,\delta}(R_2', E_2')$; that is, (5) = (8) + (11). By Definition 14, for all $0 \le i \le n$,

$$v(\sigma_{act}^i(R_1 \otimes R_2, E_1 \times E_2)) = v(\sigma_{act}^i(R_1, E_1)) + v(\sigma_{act}^i(R_2, E_2)), \tag{12}$$

and hence (4) = (7) + (10). By rearranging the arithmetic, we obtain our desired result.

$$\begin{aligned} u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) &= (4) + (5) \\ &= ((7) + (10)) + ((8) + (11)) \\ &= ((7) + (8)) + ((10) + (11)) \\ &= (6) + (9) \\ &= u_{v,\sigma,\delta}(R_1, E_1) + u_{v,\sigma,\delta}(R_2, E_2). \end{aligned}$$

For the third case, suppose that $R_1 \otimes R_2, E_1 \times E_2 \rightarrow$ and that there exists no $n \in \mathbb{N}$ such that $\sigma_{last}(R_1 \otimes R_2, E_1 \times E_2) = n$. As we assume that $C_\sigma(R_1 \otimes R_2, E_1 \times E_2)$ is finite, then, by Definition 20, there exist a state $S, F \in C_\sigma(R_1 \otimes R_2, E_1 \times E_2)$ and a natural number n such that $\sigma_{state}^n(S, F) = S, F$. Without loss of generality, let n be the least number such that the above holds. By Lemma 59, as $(1 - \delta^{n+1}) \ne 0$,

$$u_{v,\sigma,\delta}(S, F) = \left(\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^i(S, F))\right) + \delta^{n+1} \times u_{v,\sigma,\delta}(\sigma_{state}^n(S, F))$$
$$(1 - \delta^{n+1}) \times u_{v,\sigma,\delta}(S, F) = (\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^i(S, F))).$$

Suppose that $S, F = R_1 \otimes R_2, E_1 \times E_2$. By Lemma 60, for all $0 \le i \le n$:

$$\sigma_{act}^i(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{act}^i(R_1, E_1) \cdot \sigma_{act}^i(R_2, E_2) \tag{13}$$

$$\sigma_{state}^i(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{state}^i(R_1, E_1) \circ \sigma_{state}^i(R_2, E_2), \tag{14}$$

and $\sigma_{act}^n(R_1, E_1) = R_1, E_1$ and $\sigma_{act}^n(R_2, E_2) = R_2, E_2$. By Definition 14, for all $0 \le i \le n$,

$$v(\sigma_{act}^i(R_1 \otimes R_2, E_1 \times E_2)) = v(\sigma_{act}^i(R_1, E_1)) + v(\sigma_{act}^i(R_2, E_2)).$$

Then

$$(1 - \delta^{n+1}) \times u_{v,\sigma,\delta}(S, F) = (\Sigma_{i=0}^{n}\delta^i \times (v(\sigma_{act}^i(R_1, E_1)) + v(\sigma_{act}^i(R_2, E_2)))) \tag{15}$$

$$= (\Sigma_{i=0}^{n}\delta^i \times (v(\sigma_{act}^i(R_1, E_1)))) \tag{16}$$

$$+$$

$$(\Sigma_{i=0}^{n}\delta^i \times (v(\sigma_{act}^i(R_2, E_2)))). \tag{17}$$

By Lemma 59,

$$u_{v,\sigma,\delta}(R_1,E_1) = \left(\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^{i}(R_1,E_1))\right) +$$
$$\delta^{n+1} \times u_{v,\sigma,\delta}(R_1,E_1)$$
$$(1-\delta^{n+1}) \times u_{v,\sigma,\delta}(R_1,E_1) = (\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^{i}(R_1,E_1))) \tag{18}$$

$$u_{v,\sigma,\delta}(R_2,E_2) = \left(\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^{i}(R_2,E_2))\right) +$$
$$\delta^{n+1} \times u_{v,\sigma,\delta}(R_1,E_1)$$
$$(1-\delta^{n+1}) \times u_{v,\sigma,\delta}(R_2,E_2) = (\Sigma_{i=0}^{n}\delta^i \times v(\sigma_{act}^{i}(R_2,E_2))). \tag{19}$$

So, as $S,F = R_1 \otimes R_2, E_1 \times E_2$,

$$(15) \qquad = (16)+(17)$$
$$= (18)+(19)$$
$$(1-\delta^{n+1}) \times u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = (1-\delta^{n+1}) \times u_{v,\sigma,\delta}(R_1,E_1)+$$
$$(1-\delta^{n+1}) \times u_{v,\sigma,\delta}(R_2,E_2)$$
$$u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) \qquad = u_{v,\sigma,\delta}(R_1,E_1)+u_{v,\sigma,\delta}(R_2,E_2).$$

Suppose that $S,F \neq R_1 \otimes R_2, E_1 \times E_2$. By Definition 20, there exists some natural number m such that $\sigma_{state}^{m}(R_1 \otimes R_2, E_1 \times E_2) = S,F$. By Lemma 60, there exist R_1', R_2', E_1', E_2' such that $S = R_1' \otimes R_2'$, $F = E_1' \times E_2'$, and, for all $0 \leq i \leq m$,

$$\sigma_{act}^{i}(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{act}^{i}(R_1,E_2) \cdot \sigma_{act}^{i}(R_2,E_2) \tag{20}$$
$$\sigma_{state}^{i}(R_1 \otimes R_2, E_1 \times E_2) = \sigma_{state}^{i}(R_1,E_1) \circ \sigma_{state}^{i}(R_2,E_2), \tag{21}$$

and hence $\sigma_{state}^{m}(R_1,E_1) = R_1',E_1'$ and $\sigma_{state}^{m}(R_2,E_2) = R_2',E_2'$. By Lemma 59,

$$u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = \left(\Sigma_{i=0}^{m}\delta^i \times v(\sigma_{act}^{i}(R_1 \otimes R_2, E_1 \times E_2))\right) +$$
$$\delta^{m+1} \times u_{v,\sigma,\delta}(R_1' \otimes R_2', E_1' \times E_2'),$$

$$u_{v,\sigma,\delta}(R_1,E_1) = \tag{22}$$
$$\left(\Sigma_{i=0}^{m}\delta^i \times v(\sigma_{act}^{i}(R_1,E_1))\right) \tag{23}$$
$$+$$
$$\delta^{m+1} \times u_{v,\sigma,\delta}(R_1',E_1') \tag{24}$$

$$u_{v,\sigma,\delta}(R_2,E_2) = \tag{25}$$
$$\left(\Sigma_{i=0}^{m}\delta^i \times v(\sigma_{act}^{i}(R_2,E_2))\right) \tag{26}$$
$$+$$
$$\delta^{m+1} \times u_{v,\sigma,\delta}(R_1',E_1'). \tag{27}$$

By Definition 14, for all $0 \leq i \leq m$,

$$v(\sigma_{act}^{i}(R_1 \otimes R_2, E_1 \times E_2)) = v(\sigma_{act}^{i}(R_1,E_1))+v(\sigma_{act}^{i}(R_2,E_2)),$$

and hence

$$u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) = \tag{28}$$

$$\left(\Sigma_{i=0}^m \delta^i \times v(\sigma_{act}^i(R_1, E_1)) + v(\sigma_{act}^i(R_2, E_2)) \right) \tag{29}$$
$$+$$
$$\delta^{m+1} \times u_{v,\sigma,\delta}(R_1' \otimes R_2', E_1' \times E_2') \tag{30}$$

We can use the same technique as in the case where $S, F = R_1 \otimes R_2, E_1 \times E_2$ (above) to prove that

$$u_{v,\sigma,\delta}(R_1' \otimes R_2', E_1' \times E_2') = u_{v,\sigma,\delta}(R_1', E_1') + u_{v,\sigma,\delta}(R_2', E_2'),$$

and hence $(30) = (24) + (27)$. By straightforward rearrangement of arithmetic, we have that $(29) = ((23) + (26))$. Hence,

$$\begin{aligned}
u_{v,\sigma,\delta}(R_1 \otimes R_2, E_1 \times E_2) &= (29) + (30) \\
&= ((23) + (26)) + ((24) + (27)) \\
&= ((23) + (24)) + ((26) + (27)) \\
&= (22) + (25) \\
&= u_{v,\sigma,\delta}(R_1, E_1) + u_{v,\sigma,\delta}(R_2, E_2).
\end{aligned}$$

8. By Definition 58[5], $\sigma(R, \text{fix } X.E) = \sigma(R, E[\text{fix } X.E/X])$. By Definition 16, we immediately have that $u_{v,\sigma,\delta}(R, \text{fix } X.E) = u_{v,\sigma,\delta}(R, E[\text{fix } X.E/X])$. ∎

We can also show similar results for bounded utility calculations.

PROPOSITION 63
For all valuation functions v, elementary strategies σ, discount factors δ, resources R, R_1, R_2, R', processes E, E', F, E_1, E_2, and natural numbers m, n, we have the following:

(1) $u_{v,\sigma,\delta,n}(R, \mathbf{0}) = -\infty$;
(2) $u_{v,\sigma,\delta,n}(R, \mathbf{1}) = 0$;
(3) If $R, E \nrightarrow$, then $u_{v,\sigma,\delta,n}(R, E :_\gamma F) = -\infty$;
(4) If $\sigma_{last}(R, E) = m$, $\sigma_{state}^m(R, E) = R', E'$, $u_{v,\sigma,\delta,n}(R, E :_\gamma F) \downarrow$, and $m < n$, then

$$u_{v,\sigma,\delta,n}(R, E :_\gamma F) = u_{v,\sigma,\delta,m}(R, E) + \delta^{(m+1)} \times u_{v,\sigma,\delta,(n-(m+1))}(\gamma(R'), F);$$

(5) If $\sigma_{last}(R, E) = m$, $\sigma_{state}^m(R, E) = R', E'$, $u_{v,\sigma,\delta,n}(R, E :_\gamma F) \downarrow$, and $n \leq m$, then

$$u_{v,\sigma,\delta,n}(R, E :_\gamma F) = u_{v,\sigma,\delta,n}(R, E);$$

(6) If $u_{v,\sigma,\delta,n}(R_1 \oplus R_2, E_1 + E_2)$ is defined, then

$$u_{v,\sigma,\delta,n}(R_1 \oplus R_2, E_1 + E_2) = u_{v,\sigma,\delta,n}(R_i, E_i),$$

for some $i \in \{1, 2\}$;
(7) If $u_{v,\sigma,\delta,n}(R_1 \oplus R_2, E_1 + \mathbf{0})$ is defined, then

$$u_{v,\sigma,\delta,n}(R_1 \oplus R_2, E_1 + \mathbf{0}) = u_{v,\sigma,\delta,n}(R_1, E_1);$$

(8) If v is total, then

$$u_{v,\sigma,\delta,n}(R_1 \otimes R_2, E_1 \times E_2) = u_{v,\sigma,\delta,n}(R_1, E_1) + u_{v,\sigma,\delta,n}(R_2, E_2);$$

(9) If $u_{v,\sigma,\delta,n}(R, \text{fix } X.E)$ is defined, then

$$u_{v,\sigma,\delta,n}(R, \text{fix } X.E) = u_{v,\sigma,\delta,n}(R, E[\text{fix } X.E/X]).$$

PROOF. This follows Proposition 62. We demonstrate with the cases that differ the most.

4. By induction on m.

Suppose that $m = 0$. By Definition 19, we have that there exists an action a such that $\sigma(R,E) = (a,(R',E'))$ and $\sigma(R',E') = \bullet$. By Definition 11, $R',E' \not\rightarrow$. By Definition 63, $u_{v,\sigma,\delta,n}(R,E :_\gamma F) = v(a) + \delta \times u_{v,\sigma,\delta,n-1}(\gamma(R'),F)$ and $u_{v,\sigma,\delta,0}(R,E) = v(a)$. By Definition 58 (1), $\sigma(R,E :_\gamma F) = (a,(\gamma(R'),F))$. As $m = 0$,

$$u_{v,\sigma,\delta,n}(R,E :_\gamma F) = u_{v,\sigma,\delta,m}(R,E) + \delta \times u_{v,\sigma,\delta,(n-1)}(\gamma(R'),F)$$
$$u_{v,\sigma,\delta,m}(R,E) + \delta^{(m+1)} \times u_{v,\sigma,\delta,(n-(m+1))}(\gamma(R'),F).$$

Suppose that $m > 0$. By Definition 19, we have that there exists an action a such that $\sigma(R,E) = (a,(R'',E''))$ and $\sigma_{state}^{m-1}(R'',E'') = R',E'$. By Definition 63,

$$u_{v,\sigma,\delta,n}(R,E :_\gamma F) = v(a) + \delta \times u_{v,\sigma,\delta,(n-1)}(R'',E'' :_\gamma F).$$

and

$$u_{v,\sigma,\delta,m}(R,E) = v(a) + \delta \times u_{v,\sigma,\delta,(m-1)}(R'',E'').$$

By the induction hypothesis, we have that

$$u_{v,\sigma,\delta,(n-1)}(R'',E'' :_\gamma F) = u_{v,\sigma,\delta,(m-1)}(R'',E'') + \delta^m \times u_{v,\sigma,\delta,((n-1)-((m-1)+1))}(\gamma(R'),F).$$

By straightforward arithmetic, $(n-1) - ((m-1)+1) = (n-(m+1))$. By Definition 58 (1), $\sigma(R,E :_\gamma F) = (a,(R'',E'' :_\gamma F))$. Hence

$$u_{v,\sigma,\delta,n}(R,E :_\gamma F) = v(a) + \delta \times \left(\begin{array}{c} u_{v,\sigma,\delta,(m-1)}(R'',E'') + \\ \delta^m \times u_{v,\sigma,\delta,(n-(m+1))}(\gamma(R'),F)) \end{array} \right)$$
$$= (v(a) + \delta \times u_{v,\sigma,\delta,(m-1)}(R'',E'')) +$$
$$\delta^{(m+1)} \times u_{v,\sigma,\delta,(n-(m+1))}(\gamma(R'),F)$$
$$= u_{v,\sigma,\delta,m}(R,E) + \delta^{(m+1)} \times u_{v,\sigma,\delta,(n-(m+1))}(\gamma(R'),F).$$

5. By induction on m.

Suppose that $m = 0$. As $n \leq m$, $n = 0$. By Definition 19, we have that there exists an action a such that $\sigma(R,E) = (a,(R',E'))$. By Definition 63, $u_{v,\sigma,\delta,0}(R,E :_\gamma F) = v(a)$. By Definition 63, $u_{v,\sigma,\delta,0}(R,E) = v(a)$. By Definition 58[1], $\sigma(R,E :_\gamma F) = (a,(\gamma(R'),F))$. Hence

$$u_{v,\sigma,\delta,n}(R,E :_\gamma F) = u_{v,\sigma,\delta,n}(R,E).$$

Suppose that $0 < m$. By Definition 19, we have that there exists an action a such that $\sigma(R,E) = (a,(R'',E''))$ and $\sigma_{state}^{m-1}(R'',E'') = R',E'$. By Definition 63

$$u_{v,\sigma,\delta,n}(R,E :_\gamma F) = v(a) + \delta \times u_{v,\sigma,\delta,(n-1)}(R'',E'' :_\gamma F)$$

By Definition 63, $u_{v,\sigma,\delta,n}(R,E) = v(a) + \delta \times u_{v,\sigma,\delta,(n-1)}(R'',E'')$. By the induction hypothesis, we have that $u_{v,\sigma,\delta,(n-1)}(R'',E'':_{\gamma} F) = u_{v,\sigma,\delta,(m-1)}(R'',E'')$. By Definition 58[1], $\sigma(R,E:_{\gamma} F) = (a,(R'',E'':_{\gamma} F))$. Hence

$$u_{v,\sigma,\delta,n}(R,E:_{\gamma} F) = v(a) + \delta \times u_{v,\sigma,\delta,(m-1)}(R'',E'')$$
$$= u_{v,\sigma,\delta,n}(R,E).$$

∎

We demonstrate the use of elementary strategies, and their equational payoffs over resource–process structure, in the following example.

EXAMPLE 64 (Mutual producer–consumer)
In Example 34, we introduce an example of distributed coordination without mutual exclusion: a mutual producer-consumer system, where each 'agent' can generate work for, and consume work from, the other. There, the agents performing the production and consumption are represented indirectly. Using a resource–process framework, we can represent the dynamics of the different agents more directly. Specifically, we represent these agents as processes. We can then demonstrate how, for example, the first entity cannot make progress when it only possesses resources that the second process can consume available to it.

Suppose a resource model $(\mathbf{R},e,\otimes,\oplus)$ such that, for all resources $r,s,t \in \mathbf{R}$ and for all natural numbers $n_1,n_2 \in \mathbb{N}$,

- $(n_1,n_2) \in \mathbf{R}$,
- $(r,s) \in \mathbf{R}$ if and only if $r \otimes s = (r,s)$, and
- $r \oplus s = t$ if and only if $r = s = t$.

Intuitively, a pair of natural numbers denotes the resources, or work packages, that could be consumed by the two agents in the system (should they have access to them): the first number denotes the resources that could be consumed by the first entity, and the second number denotes the resources that could be consumed by the second entity. The p_1 action denotes production of a work package by the first entity for the second entity, and the c_1 action denotes the consumption of a work package by the first entity. Note that a process cannot perform a consume action if there are zero resources that it can consume available to it. The p_2 and c_2 actions have the obvious converse denotations. This is represented formally in the modification function. Let μ be a modification function such that

$$\begin{aligned}
\mu((m,n),p_1) &= (m,n+1) \\
\mu((m+1,n),c_1) &= (m,n) \\
\mu((m,n),p_2) &= (m+1,n) \\
\mu((m,n+1),c_2) &= (m,n).
\end{aligned}$$

We represent the first agent with a process, E_1,

$$E_1 = \text{fix } Y_1.\Big((p_1:Y_1)+(c_1:Y_1)+1\Big).$$

The process is a fixed point which consists of three possibilities. The process may either: produce a resource (for the second process), using p_1, and recurse; consume a resource from the other process (if available), using c_1, and recurse; or, perform the tick action and terminate. When combined with the resource $(1,0)$, it can perform any of its three possible actions, as demonstrated by the following

derivations.

$$\frac{\dfrac{\mu((1,0),p_1)=(1,1)}{(1,0),p_1 \xrightarrow{p_1} (1,1),\mathbf{0}}}{\dfrac{(1,0),p_1:E_1 \xrightarrow{p_1} (1,1),E_1}{(1,0),(p_1:E_1)+(c_1:E_1)+1 \xrightarrow{p_1} (1,1),E_1}}$$

$$\frac{\dfrac{\mu((1,0),c_1)=(0,0)}{(1,0),c_1 \xrightarrow{c_1} (0,0),\mathbf{0}}}{\dfrac{(1,0),(c_1:E_1) \xrightarrow{c_1} (0,0),E_1}{(1,0),(p_1:E_1)+(c_1:E_1)+1 \xrightarrow{c_1} (0,0),E_1}} \qquad \frac{\dfrac{\mu((1,0),1)=(1,0)}{(1,0),1 \xrightarrow{1} (1,0),\mathbf{0}}}{(1,0),(p_1:E_1)+(c_1:E_1)+1 \xrightarrow{1} (1,0),\mathbf{0}}$$

When combined with the resource $(0,0)$, it can only produce (and recurse) or terminate; it cannot perform the c_1 action. The process $E_2 = \text{fix } Y_2.((p_2:Y_2)+(c_2:Y_2)+1)$, which represents the second agent, behaves similarly.

In order to transfer the produced resources from one process to another, we make use of a redistribution function γ such that

$$\gamma(z) = \begin{cases} ((n_1+m_1,0),(0,n_2+m_2)) & \text{if } z=((n_1,n_2),(m_1,m_2)) \\ z & \text{otherwise.} \end{cases}$$

This redistribution function takes all of the work packages for the first process, including those that were previously allocated to the second process, and gives them all to the first process, and takes all of the work packages for the second process, including those that were previously allocated to the first process, and gives them all to the second process.

The dynamics of the full system can then be defined by the process E:

$$E = \text{fix } X.(E_1 \times E_2){:}_\gamma X.$$

Suppose that the agents 'profit' from the consumption of work packages, and must 'pay' to create work packages. We can represent this situation via a pair of total payoff functions v_1 and v_2 for the two entities such that

$$v_1(p_1)=-1 \;\; v_1(c_1)=3 \;\; v_1(p_2)=0 \;\;\; v_1(c_2)=0$$
$$v_2(p_1)=0 \;\;\; v_2(c_1)=0 \;\; v_2(p_2)=-2 \;\; v_2(c_2)=4.$$

We make use of a strategy where each entity consumes, if able; if not, and there are no resources for the other entity, it produces; otherwise, it terminates. This is represented via an elementary strategy σ such that

$$(1)\; \sigma\Big((n_1,n_2),(p_1:E_1)+(c_1:E_1)+1\Big) = \Big(c_1,((n_1-1,n_2),E_1)\Big) \text{ if } 1\leq n_1$$
$$(2)\; \sigma\Big((0,0),(p_1:E_1)+(c_1:E_1)+1\Big) \;\;\; = \Big(p_1,((0,1),E_1)\Big)$$
$$(3)\; \sigma\Big((0,n_2),(p_1:E_1)+(c_1:E_1)+1\Big) \;\;\; = \Big(1,((0,n_2),\mathbf{0})\Big) \qquad \text{if } 1\leq n_2$$

$$(4)\ \sigma\Big((n_1,n_2),(p_2:E_2)+(c_2:E_2)+1\Big) = \Big(c_2,((n_1,n_2-1),E_2)\Big)\ \text{if}\ 1\le n_2$$
$$(5)\ \sigma\Big((0,0),(p_2:E_2)+(c_2:E_2)+1\Big)\ = \Big(p_2,((1,0),E_2)\Big)$$
$$(6)\ \sigma\Big((n_1,0),(p_2:E_2)+(c_2:E_2)+1\Big)\ = \Big(1,((n_1,0),\mathbf{0})\Big)\qquad \text{if}\ 1\le n_1.$$

As σ is an elementary strategy, we can the derive the payoff of the resource–process pairs over their structure, via Propositions 62 and 63, rather than via Definition 16. Suppose that $u_{v_1,\delta,\sigma}(((0,0),(0,0)),E)$ is defined. Let us consider the payoff of the state $((0,0),(0,0)),E$, from the perspective of the first agent.

Let the discount factor δ be 0.8. By Proposition 62[8], the payoff of the fixed point of $((0,0),(0,0)),E$ is the payoff of the unfolding of the fixed point:

$$u_{v_1,\delta,\sigma}(((0,0),(0,0)),E) = u_{v_1,\delta,\sigma}(((0,0),(0,0)),(E_1\times E_2):_\gamma E).$$

In order to further proceed in the equational handling of utility, we need to determine whether or not $\sigma_{last}(((0,0),(0,0)),(E_1\times E_2))$ is defined. To establish this, we can apply the strategy σ repeatedly. As the strategy σ is elementary, $\sigma(e,E_1) = \Big(p_1,((0,1),E_1)\Big)$, and $\sigma(e,E_2) = \Big(p_2,((1,0),E_2)\Big)$, we have that

$$\sigma(((0,0),(0,0)),(E_1\times E_2)) = (p_1 p_2,(((0,1),(1,0)),(E_1\times E_2))).$$

Similarly, as $\sigma((0,1),E_1) = \Big(1,((0,1),\mathbf{0})\Big)$ and $\sigma((1,0),E_2) = \Big(1,((1,0),\mathbf{0})\Big)$,

$$\sigma(((0,1),(1,0)),(E_1\times E_2)) = (1\cdot 1,(((0,1),(1,0)),(\mathbf{0}\times\mathbf{0}))).$$

As $((0,1),(1,0)),\mathbf{0}\times\mathbf{0}\nrightarrow$, we have that $\sigma_{last}(((0,0),(0,0)),(E_1\times E_2)) = 1$. Let $R',E' = \sigma^1_{state}(((0,0),(0,0)),E_1\times E_2) = ((0,1),(1,0)),\mathbf{0}\times\mathbf{0}$.

The payoff of the sequential composition is the payoff of the prefix plus the discounted payoff of the suffix

$$u_{v_1,\sigma,\delta}(((0,0),(0,0)),(E_1\times E_2):_\gamma E) = u_{v_1,\sigma,\delta,1}(((0,0),(0,0)),E_1\times E_2)$$
$$+\delta^2\times u_{v_1,\sigma,\delta}(\gamma(R'),E).$$

The payoff of the concurrent composition is the sum of the payoffs of the concurrent components:

$$u_{v_1,\sigma,\delta,1}(((0,0),(0,0)),E_1\times E_2) = u_{v_1,\sigma,\delta,1}((0,0),E_1)+u_{v_1,\sigma,\delta,1}((0,0),E_2).$$

The payoff of the fixed point of $(0,0),E_1$ is the payoff of the unfolding of the fixed point:

$$u_{v_1,\sigma,\delta,1}((0,0),E_1) = u_{v_1,\sigma,\delta,1}((0,0),(p_1:E_1)+(c_1:E_1)+1).$$

The payoff of the non-deterministic choice is the payoff of the branch chosen by the strategy; that is, $\sigma\Big((0,0),(p_1:E_1)+(c_1:E_1)+1\Big) = \Big(p_1,((0,1),E_1)\Big)$:

$$u_{v_1,\sigma,\delta,1}((0,0),(p_1:E_1)+(c_1:E_1)+1) = u_{v_1,\sigma,\delta,1}((0,0),p_1:E_1).$$

The payoff of the sequential composition is the payoff of the prefix plus the discounted payoff of the suffix. Let $\sigma^0_{state}((0,0),p_1) = (R'_1,E_1)$:

$$u_{v_1,\sigma,\delta,1}((0,0),p_1:E_1) = u_{v_1,\sigma,\delta,0}((0,0),p_1)+\delta\times u_{v_1,\sigma,\delta,0}(R'_1,E_1).$$

We straightforwardly have that $R'_1 = \mu((0,0), p_1) = (0,1)$ and $u_{v_1,\sigma,\delta,0}((0,0), p_1) = v_1(p_1) = -1$. The payoff of the fixed point of $(0,1), E_1$ is the payoff of the unfolding of the fixed point:

$$u_{v_1,\sigma,\delta,0}((0,1), E_1) = u_{v_1,\sigma,\delta,1}((0,0), (p_1:E_1) + (c_1:E_1) + 1).$$

The payoff of the non-deterministic choice is the payoff of the branch chosen by the strategy; that is, $\sigma\Big((0,1), (p_1:E_1) + (c_1:E_1) + 1\Big) = \Big(1, ((0,1), \mathbf{0})\Big)$:

$$u_{v_1,\sigma,\delta,0}((0,1), (p_1:E_1) + (c_1:E_1) + 1) = u_{v_1,\sigma,\delta,0}((0,1), 1).$$

The payoff of the trivial action process is $u_{v_1,\sigma,\delta,0}((0,1), 1) = 0$. Putting this all together, we have that

$$u_{v_1,\sigma,\delta,1}((0,0), E_1) = -1 + \delta \times 0 = -1.$$

We can use the same approach to show that $u_{v_1,\sigma,\delta,1}((0,0), E_2) = 0$, as the first agent has no (direct) payoffs from the actions performed by the second agent. Hence we have that $u_{v_1,\sigma,\delta,1}(((0,0),(0,0)), E_1 \times E_2) = -1 + 0 = -1$.

We then consider the payoff of the suffix, $\gamma(R'), E$. Note that when we apply the redistribution function γ, the work packages are moved to the process that can consume them; that is,

$$\gamma(((0,1),(1,0))) = ((1,0),(0,1)).$$

The payoff of the fixed point $((1,0),(0,1)), E$ is the payoff of the unfolding of the fixed point:

$$u_{v_1,\delta,\sigma}((1,0),(0,1), E) = u_{v_1,\delta,\sigma}((1,0),(0,1),(E_1 \times E_2):_\gamma E).$$

In order to further proceed, we need to determine whether or not $\sigma_{last}(((1,0),(0,1)), (E_1 \times E_2))$ is defined. By repeated application of the strategy σ, we determine that

$$\sigma_{last}(((1,0),(0,1)),(E_1 \times E_2)) = 2.$$

Let $\sigma^2_{state}(((1,0),(0,1)),(E_1 \times E_2)) = (R'', E'') = ((0,1),(1,0)), \mathbf{0} \times \mathbf{0}$. The payoff of the sequential composition is the payoff of the prefix plus the discounted payoff of the suffix:

$$\begin{aligned}u_{v_1,\sigma,\delta}(((1,0),(0,1)),(E_1 \times E_2):_\gamma E) = \\ u_{v_1,\sigma,\delta,2}(((1,0),(0,1)), E_1 \times E_2) + \delta^3 \times u_{v_1,\sigma,\delta}(\gamma(R''), E).\end{aligned}$$

Following the above approach, $u_{v_1,\sigma,\delta,2}(((1,0),(0,1)), E_1 \times E_2)$ can be determined in terms of the payoff of the concurrent components. For each component $i \in \{1,2\}$, the strategy choses to perform the action c_i, then p_i, then 1. So

$$u_{v_1,\sigma,\delta,2}((1,0), E_1) = 3 + 0.8 \times -1 + 0.8^2 \times 0 = 2.2,$$

and $u_{v_1,\sigma,\delta,2}((0,1), E_2) = 0$. As a result, we have that

$$u_{v_1,\sigma,\delta}(((1,0),(0,1)),(E_1 \times E_2):_\gamma E) = (2.2 + 0) + \delta^3 \times u_{v_1,\sigma,\delta}(((1,0),(0,1)), E).$$

We can solve this simultaneous equation for $u_{v_1,\sigma,\delta}(((1,0),(0,1)),E)$, and so

$$u_{v_1,\sigma,\delta}(((1,0),(0,1)),E) \approx 4.5.$$

The payoff of the whole system is then the payoff of the original prefix and the discounted payoff of the original suffix:

$$\begin{aligned}
u_{v_1,\sigma,\delta}(((0,0),(0,0)),(E_1 \times E_2):_\gamma E) &= u_{v_1,\sigma,\delta,1}(((0,0),(0,0)),E_1 \times E_2)+ \\
&\quad \delta^2 \times u_{v_1,\sigma,\delta}(\gamma(R'),E) \\
&\approx -1+0.8^2 \times 4.5 \\
&\approx 1.9.
\end{aligned}$$

Recall the notion of Pareto optimality from Definition 35; that is, that a state s is Pareto optimal if there exists an action a such that, for all other actions b, if some entity weakly prefers (the relation is not strict) that action b be performed, then there is some other agent that strongly prefers (the relation is not strict) that action a be performed. Here we have that the state $((0,0),(0,0)),E$ is Pareto optimal, witnessed by the action $p_1 \cdot p_2$. The only other action that can be performed by $((0,0),(0,0)),E$ is $1 \cdot 1$. Note that $((0,0),(0,0)),E \xrightarrow{1 \cdot 1} ((0,0),(0,0)),E$. The payoff for the first agent obtained by performing the action $1 \cdot 1$ is

$$\begin{aligned}
v_1(1 \cdot 1)+\delta \times u_{v_1,\delta,\sigma}(((0,0),(0,0)),E) &\approx 0+0.8 \times 1.9 \\
&\approx 1.5.
\end{aligned}$$

So, for the first agent, switching from the action $p_1 \cdot p_2$ to $1 \cdot 1$ results in a loss of payoff, so the state is Pareto optimal.

6   Discussion

In this article, we motivate our development from a richly expressive modal logic for resource semantics and distributed systems modelling, MBIU. This logic includes both additive and multiplicative propositional connectives and also additive action modalities, as well as certain first-order quantifiers. We employ an abstract formulation of MBIU that is based on a semantics that employs a labelled transition system, a notion of concurrent composition of states, and an equivalence relation on actions. Following the approach in [2], we establish Hennessy–Milner soundness and completeness for our abstract formulation. This framework and logic is sufficient to model classic examples from distributed systems modelling and game theory, and to express game-theoretic concepts, including Pareto optimality, the best-response property, and Nash equilibrium. The key role of the multiplicative conjunction, $*$, in the formulae representing best response should be noted. Used with the additives, it allows the separation of the states performing different actions (the as and bs) to be enforced when required, whilst allowing payoff properties of the overall system to be expressed relative to the overall resources, as required. We then describe two instantiations of our abstract formulation. First, monoidal resource semantics: this can be utilized to provide a simple way to model distributed systems. Many of our early examples in the abstract formulation turned out to be of this class. Secondly, resource–process modelling: this can be utilized to model scenarios in more structural detail. Using this approach, we should be able to incorporate the analysis of utility and optimality presented here into the widely deployed systems and security modelling tools established in, for example, [10–12], with deployments described in, for example, [3, 5–7, 21].

Some conceptual and technical issues, beyond our present scope, remain to be addressed.

Multiplicative modalities, logical formulae that are often included in multiplicative logics such as MBI, can be used to reason about transitions in the situations where additional components are concurrently composed with the state at which a formula is evaluated for satisfiability. With these modalities, it is possible to provide a natural description of various agent based scenarios, including the notion that achieving some goal is within an agent's capabilities, were it to be given additional resources, and the notion that achieving some goal is never within an agent's capabilities, no matter how much additional resource it is given [14]. This can be further extended to represent security examples where attacks can occur through introduction of racy concurrent behaviour.

There are various possible choices of how to interpret the multiplicative components of a logic in the case where the states have a multi-dimensional structure [11, 12, 14]. We present and contrast different possible interpretations (defined informally) of multiplicative implication and multiplicative modalities.

We can add multiplicative modalities into our system straightforwardly. For example, the multiplicative modality $\langle d \rangle_v$ can be specified as

$$R, E \vDash_\rho \langle d \rangle_v \phi \text{ iff there exist } a, S, F, R', E' \text{ such that } R \otimes S, E \times F \xrightarrow{a} R', E',$$
$$|d| \equiv a \text{ and } R', E' \vDash_\rho \phi.$$

Note that this formulation adds both a process and a resource component, following the interpretation of multiplicative implication:

$$R, E \vDash_\rho \phi_1 \multimap \phi_2 \text{ iff for all } S, F, S, F \vDash_\rho \phi_1 \text{ implies } R \otimes S, E \times F \vDash_\rho \phi_2.$$

As a result, this multiplicative modality can be defined in terms of the multiplicative implication and the additive fragment of the logic [2].

By contrast, in [11, 12], multiplicative implication composes both a resource and a process component, while the multiplicative modalities compose solely a resource component. An interpretation of multiplicative implication, following [11, 12], in our resource–process calculus, would be as above, but an interpretation of the multiplicative modality $\langle a \rangle_v \phi$, following [11, 12], in our resource–process calculus, would be

$$R, E \vDash_\rho \langle d \rangle_v \phi \text{ iff there exist } a, S, R', E' \text{ such that } R \otimes S, E \xrightarrow{a} R', E',$$
$$|d| \equiv a, \text{ and } R', E' \vDash_\rho \phi.$$

In further contrast, in [14], one of us has considered a generalization of resource semantics to admit multi-dimensional satisfaction relations of the form, for example, $w, r \vDash \phi$, in which $w \in W$ are taken to be Kripke worlds (ordered by $\sqsubseteq$, say) in the sense of classical modal logic and $r \in R$ are interpreted as resources, where R carries monoidal structure (with composition $\circ$, say). In this set-up, we can define a multiplicative modality $\Diamond_s$ as

$$w, r \vDash \Diamond_s \phi \text{ iff there is a world } w \sqsubseteq v \text{ such that } v, r \circ s \vDash \phi.$$

Such a modality is highly expressive and, among other things, generalizes the usual S4 modality [8, 14]. This multiplicative modality can be defined in terms of the multiplicative implication and the additive fragment of the logic.

Thus, there are various approaches taken in terms of which components are augmented by multiplicative implication and multiplicative modalities. We believe that an investigation into the

comparative properties of these approaches would be valuable. Furthermore, we conjecture that all of the above are examples of a more general treatment of multiplicative connectives within a generalized multi-dimensional handling of concurrent transition systems and that such a handling would have natural resource interpretations.

Another multiplicative possible extension to the logic is multiplicative quantifiers [11, 12]. Multiplicative quantifiers reason about actions in the presence of hiding; their inclusion of multiplicative quantifiers into our system is a complex prospect. A rendering of multiplicative existential quantification, $\exists_\nu \alpha.\phi$, for our resource–process calculus, following [11, 12], would be

$$R, E \vDash_\rho \exists_\nu \alpha.\phi \text{ iff there exists } S, F, a \in \mathbf{Act}, h \in \mathbf{H} \text{ such that } R, E \sim S, \nu h.F$$
$$\text{and } h(S), F \vDash_{\rho[\alpha:=a]} \phi.$$

So, multiplicative quantification is closely related to the notion of hiding and to the multi-dimensional world structure in resource–process calculi. There is no immediately apparent generalization of such an approach to arbitrary concurrent transition systems that do not have a multi-dimensional world structure.

It does not appear possible, within the current framework, to handle of the payoff of the hiding operator equationally. It is relatively straightforward to determine the payoff of a resource–process pair with hiding in terms of the *derivation* of the payoff of the relevant resource–process pair without hiding. This can be done as follows. Extend the notion of elementary strategy (Definition 58), for all hiding functions h, with

$$\text{if } \sigma(h(R), E) = (a, (h(R'), E')), \text{ then } \sigma(R, \nu h.E) = (\nu h.a, (R', \nu h.E')).$$

The payoff of the state $h(R), E$ is specified by a finite set of linear simultaneous equations,

$$u_{v,\sigma,\delta}(h(R), E) = v(\sigma^0_{act}(h(R), E)) + \delta \times u_{v,\sigma,\delta}(\sigma^0_{state}(h(R), E))$$
$$\dots$$
$$u_{v,\sigma,\delta}(\sigma^{n-1}_{state}(h(R), E)) = v(\sigma^n_{act}(h(R), E)) + \delta \times u_{v,\sigma,\delta}(\sigma^n_{state}(h(R), E)).$$

The payoff of the state $R, \nu h.E$, with respect to an elementary strategy σ, can then be specified by the modified finite set of linear simultaneous equations,

$$u_{v,\sigma,\delta}(R, \nu h.E) = v(\nu h.(\sigma^0_{act}(h(R), E))) + \delta \times u_{v,\sigma,\delta}(\sigma^0_{state}(R, \nu h.E))$$
$$\dots$$
$$u_{v,\sigma,\delta}(\sigma^{n-1}_{state}(R, \nu h.E)) = v(\nu h.(\sigma^n_{act}(h(R), E))) + \delta \times u_{v,\sigma,\delta}(\sigma^n_{state}(R, \nu h.E)).$$

It does not appear possible to render this result so that the payoff $u_{v,\sigma,\delta}(R, \nu h.E)$ is determined equationally in terms of the value of the payoff $u_{v,\sigma,\delta}(h(R), E)$. One possibility is to modify our definition of state payoff functions to include action transformations of the form seen above. Let an action transformation function be a total function $f : \mathbf{Act} \to \mathbf{Act}$ such that, for all action payoff functions $v \in \mathbf{V}$ and actions $a \in \mathbf{Act}$, if $v(a) \downarrow$, then $v(f(a)) \downarrow$. We define a transformative payoff function as

$$u_{v,\sigma,\delta,f}(s) = \begin{cases} v(f(a)) + \delta \times u_{v,\sigma,\delta,f}(s') & \text{if } \sigma(s) = (a, s'), v(f(a)) \downarrow, \text{ and } u_{v,\sigma,\delta,f}(s') \downarrow \\ -\infty & \text{if } \sigma(s) = \bullet \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Let us restrict the set of hiding functions $\mathbf{H}$ so that, for all $h \in \mathbf{H}$ and $v \in \mathbf{V}$, $v(a) \downarrow$ implies $v(\nu h.a) \downarrow$. Then, the payoff of $(R, \nu h.E)$, $u_{v,\sigma,\delta}(R, \nu h.E)$, is simply the (action transformed) payoff of $(h(R), E)$ with respect to the hiding function νh, $u_{v,\sigma,\delta,(\nu h)}(h(R), E)$.

Further research is required to determine how hiding can be used in practice in modelling scenarios that consider payoff, the extent to which the lack of equational theory is a concern, and our alternative derivation of payoff in such circumstances.

Finally, while it is possible to define an operational semantics for open states, an appropriate notion of substitution, and an appropriate notion of bisimulation for open states, in arbitrary concurrent transition systems are open problems.

Acknowledgements

References

[1] G. Anderson, J. Brotherston and D. Pym. Hennessy-Milner completeness in transition systems with synchronous concurrent composition. Technical report, RN/15/05, University College London, 2015. Manuscript at http://www.cs.ucl.ac.uk/fileadmin/UCL-CS/research/Research_Notes/RN-15-05.pdf.

[2] G. Anderson and D. Pym. A calculus and logic of bunched resource processes. *Theoretical Computer Science*, **614**, 63–96, 2016.

[3] Y. Beres, D. Pym and S. Shiu. Decision support for systems security investment. In *Proceedings of the 5th Workshops on Network Operations and Management Symposium*, pp. 118–125. IEEE Xplore, 2010.

[4] Y. Beresnevichiene D. Pym and S. Shiu. Decision support for systems security investment. In *Network Operations and Management Symposium Workshops*, pp. 118–125. IEEE Xplore, 2010.

[5] T. Caulfield and D. Pym. Improving security policy decisions with models. *IEEE Security and Privacy*, **13**, 34–41, 2015.

[6] T. Caulfield and D. Pym. Modelling and simulating systems security policy. In *Proceedings of the 8th Conference on Simulation Tools and Techniques*, pp. 9–18. ACM Digital Library, 2015.

[7] T. Caulfield, D. Pym, and J. Williams. Compositional security modelling: structure, economics, and behaviour. *Lecture Notes in Computer Science*, **8533**, 233–245, 2014.

[8] B. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

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

[10] M. Collinson, B. Monahan and D. Pym. Semantics for structured systems modelling and simulation. In *Proceedings of the 3rd Conference on Simulation Tools and Techniques*, pp. 34:1–34:10, 2010.

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

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

[13] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed systems: concepts and design*. Addison Wesley, 3rd edn, 2000.

[14] J.-R. Courtault, D. Galmiche, and D. Pym. A logic of separating modalities. *Theoretical Computer Science*, **637**, 30–58, 2016.

[15] Y. Deng and M. Hennessy. Compositional reasoning for weighted markov decision processes. *Science of Computer Programming*, **78**, 2537–2679, 2013.

[16] D. Galmiche, D. Méry, and D. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, **15**, 1033–1088, 2015.

[17] M. Hennessy and G. Plotkin. On observing nondeterminism and concurrency. Vol. 85 of *Lecture Notes in Computer Science*, pp. 299–308, 1980.

[18] S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *Proceedings of the 28th Symposium on Principles of Programming Languages*, pp. 14–26. ACM SIGPLAN Notices, 2001.

[19] W. Jamroga. A temporal logic for markov chains. In *Proceedings of the 7th Conference on Autonomous Agents and Multiagent Systems*, pp. 697–704. ACM Digital Library, 2008.

[20] J.-B. Jeannin, D. Kozen and A. Silva. Language constructs for non-well-founded computation. Vol. 7792 of *Lecture Notes in Computer Science*, pp. 61–80, 2013.

[21] Hewlett-Packard Laboratories. Towards a science of risk analysis. http://www.hpl.hp.com/news/2011/oct-dec/security_analytics.html (accessed 16 October 2015).

[22] D. Larchey-Wendling and D. Galmiche. Exploring the relation between intuitionistic BI and Boolean BI: an unexpected embedding. *Mathematical Structures in Computer Science*, **19**, 435–500, 2009.

[23] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, **25**, 267–310, 1983.

[24] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[25] P. O'Hearn and D. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, **5**, 215–244, 1999.

[26] D. Pym, P. O'Hearn and H. Yang. Possible worlds and resources: the semantics of BI. *Theoretical Computer Science*, **315**, 257–305, 2003.

[27] S. Read. *Relevant Logic: A Philosophical Examination of Inference*. Basil Blackwell, 1989.

[28] J. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th Conference on Logic in Computer Science*, pp. 55–75. IEEE, 2002.

[29] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[30] A. Sokolova and E. Vink. Probabilistic automata: system types, parallel composition and comparison. Vol. 2925 of *Lecture Notes in Computer Science*, pp. 1–43, 2004.