Socially Responsive, Environmentally Friendly Logic

Samson Abramsky

Oxford University Computing Laboratory

Abstract

We consider the following questions: What kind of logic has a natural semantics in multi-player (rather than 2-player) games? How can we express branching quantifiers, and other partial-information constructs, with a properly compositional syntax and semantics? We develop a logic in answer to these questions, with a formal semantics based on multiple concurrent strategies, formalized as closure operators on Kahn-Plotkin concrete domains. Partial information constraints are represented as co-closure operators. We address the syntactic issues by treating syntactic constituents, including quantifiers, as arrows in a category, with arities and co-arities. This enables a fully compositional account of a wide range of features in a multi-agent, concurrent setting, including IF-style quantifiers.

Contents

1 Introduction **2**

2 From 2-person to n-person games **3**
- 2.1 The naturalness of 2-person games . . . 3
- 2.2 Background: 2-person games . . . 3
- 2.3 An Aside . . . 4
- 2.4 From 2-person to n-person games . . . 4

3 Semantics of $\mathcal{L}_{\mathcal{A}}$ **5**
- 3.1 Static Semantics: Concrete Data Structures as Concurrent Games . . . 6
- 3.2 Dynamic Semantics: Concurrent Strategies . . . 10
  - 3.2.1 Inductive Construction of Strategy Sets . . . 10
  - 3.2.2 Local conditions on strategies . . . 11
  - 3.2.3 Global definitions of strategy sets . . . 12
  - 3.2.4 Comparison of the local and global definitions . . . 14
- 3.3 Evaluation of Strategy Profiles . . . 16
- 3.4 Outcomes and Valuations . . . 17

4 Towards environmentally friendly logic **19**
- 4.1 Syntax as a Category . . . 19
- 4.2 Static Semantics Revisited . . . 20
- 4.3 Interlude: Structural Congruence . . . 21
- 4.4 Valuations . . . 22
- 4.5 Dynamic Semantics Revisited . . . 24
- 4.6 Visibility Functions, Occlusion and IF-Quantifiers . . . 24

4.6.1 IF-quantifiers 25

5 Further Directions **26**

1 Introduction

We begin with the following quote from the manifesto of a recent Workshop [AdM7]:

> "Traditionally, logic has dealt with the zero-agent notion of truth and the one-agent notion of reasoning. In the last decades, research focus in logic shifted from these topics to the vast field of "interactive logic", encompassing logics of communication and interaction. The main applications of this move to n-agent notions are logical approaches to games and social software."

However, while there are certainly applications of multi-modal logics to reasoning *about* n-person games (see e.g. [PP03, Paul02]), the more intimate connections between Games and Logic which manifest themselves in various forms of Game Semantics have all, to the best of our knowledge, been based on 2-person games. We are therefore led to consider the following question:

What kind of logic has a natural semantics in multi-player (rather than 2-player) games?

Another topic which has been studied extensively in recent years has been the logical aspects of games of imperfect information, starting with Henkin-style branching quantifiers [Hen61], and Hintikka's game-theoretical interpretation of these, and continuing with the IF-logic of Hintikka and Sandu [HS89, HS95, HS96]. The issue of whether and how a compositional semantics for IF-logic can be given has been studied by several authors, particularly Wilfrid Hodges [Hod97]. However, there is an even more basic question which does not seem to have received much, if any, attention: namely, how to give a properly compositional *syntax* for such constructs. For example, how can we build up a formula with branching quantifiers piece by piece? It might seem that IF-logic sweeps this question aside, since it does on the face of it have a compositional syntax. However, more careful consideration shows that the scope issues raised by the IF quantifiers and connectives do not fit into the usual pattern of variable-binding operators.

Our aim in the present paper is to develop a logical syntax, and an accompanying formal semantics, which addresses both these questions. The semantics is naturally phrased in terms of strategies for n-person games; and it will fit well with our compositional analysis of partial information constructs. Both our syntactical and semantical explorations will bring to light some rather unexpected connections to developments in Theoretical Computer Science.

The ideas in this paper were first presented in a lecture given at the 11th Amsterdam Colloquium in December 1997. Some of the underlying technical notions are developed in rather different contexts in a number of other papers [AJ94, AM99, Abr00a, Abr03]. One motivation for writing this paper is to attempt to communicate some of the techniques and concepts which have been developed within the Theoretical Computer Science semantics community to a broader audience. We have therefore tried to present the ideas in a self-contained fashion, and in a fairly expansive expository style.

2 From 2-person to n-person games

2.1 The naturalness of 2-person games

The basic metaphor of Game Semantics of Logic is that the players stand for **Proponent** and **Opponent**, or **Verifier** and **Falsifier**, or (with Computer Science motivation) for **System** and **Environment**. This 2-agent view sets up a natural *duality*, which arises by interchanging the rôles of the two players. This duality is crucial in defining the key notion of *composition of strategies* in the Game Semantics developed in Computer Science. It stands as the Game-Semantical correlate of the logical notion of *polarity*, and the categorical notions of *domain* and *codomain*, and *co-* and *contra-variance*.

So this link between Logic and 2-person games runs deep, and should make us wary of facile generalization. It can be seen as having the same weight as the binary nature of composition in Categories, or of the Cut Rule in Logic. Are there good multi-ary generalizations of these notions?

Nevertheless ... we shall put forward a simple system which seems to us to offer a natural generalization. We shall validate this notion to the extent of providing a precise and (in our opinion) elegant semantics in n-person games. This at least has the merit of broaching the topic, and putting a clear, if far from comprehensive, proposal on the table. There are undoubtedly many further subtleties to explore, but it is a start.

2.2 Background: 2-person games

As a starting point, we shall briefly review a Hintikka-style 'Game-Theoretical Semantics' of ordinary first-order logic, in negation normal form. Thus formulas are built from literals by conjunction, disjunction, and universal and existential quantification. Given a model $\mathcal{M}$, a game is assigned to each sentence as follows. (We shall be informal here, just as Hintikka invariably is).

- **Literals** $A(a_1, \ldots, a_n)$, $\neg A(a_1, \ldots, a_n)$. The game is trivial in this case. There is a winning move for Verifier if the literal is true in the model, and a winning move for Falsifier otherwise.
- **Conjunction** $\varphi_1 \wedge \varphi_2$. The game $G(\varphi_1 \wedge \varphi_2)$ has a first move by Falsifier which chooses one of the sub-formulas φ_i, $i = 1, 2$. It then proceeds with $G(\varphi_i)$.
- **Disjunction** $G(\varphi_1 \vee \varphi_2)$ has a first move by Verifier which chooses one of the sub-formulas φ_i, $i = 1, 2$. It then proceeds with $G(\varphi_i)$.
- **Universal Quantification** $G(\forall x. \varphi)$ has a first move by Falsifier, which chooses an element a of $\mathcal{M}$. The game proceeds as $G(\varphi[a/x])$.
- **Existential Quantification** Dually, Verifier chooses the element.

The point of this interpretation is that $\mathcal{M} \models \varphi$ in the usual Tarskian sense if and only if Verifier has a winning strategy for $G(\varphi)$.

Note that there is a very natural game-semantical interpretation of negation: $G(\neg\varphi)$ is the same game as $G(\varphi)$, but with the rôles of Verifier and Falsifier interchanged.

2.3 An Aside

In fact, the above Game semantics should really be seen as applying to the *additive fragment of Linear Logic*, rather than to Classical Logic. Note in particular that it fails to yield a proper analysis of *implication*, surely the key logical connective.

Indeed, if we render $\varphi \rightarrow \psi$ as $\neg\varphi \ \vee \ \psi$, then note that $G(\neg\varphi \ \vee \ \psi)$ does not allow for any flow of information between the antecedent and the consequent of the implication. At the very first step, one of $\neg\varphi$ or ψ is chosen, and the other is discarded.[1] In order to have the possibility of such information flow, it is necessary for $G(\neg\varphi)$ and $G(\psi)$ to run *concurrently.* This takes us into the realm of the *multiplicative connectives* in the sense of Linear Logic [Gir87]. The game-theoretical interpretation of negation also belongs to the multiplicative level.

2.4 From 2-person to n-person games

We shall now describe a simple syntax which will carry a natural interpretation in n-agent games. We say that the resulting logic is "socially responsive" in that it allows for the actions of multiple agents.

We fix, once and for all, a set of agents $\mathcal{A}$, ranged over by $\alpha, \beta, \ldots$

We introduce an $\mathcal{A}$-indexed family of binary connectives $\oplus_\alpha$, and an $\mathcal{A}$-indexed family of quantifiers Q_α. Thus we have a syntax:

$$\varphi \quad ::= \quad L \ \mid \ \varphi \oplus_\alpha \psi \ \mid \ Q_\alpha x.\, \varphi.$$

(Here L ranges over literals).

The intended interpretation of $\varphi \oplus_\alpha \psi$ is a game in which agent α initially chooses either φ or ψ, and then play proceeds in the game corresponding to the chosen sub-formula. Similarly, for $Q_\alpha x.\, \varphi$, α initially chooses an instance a for x, and play proceeds as for $\varphi[a/x]$.

2-person games as a special case If we take $\mathcal{A} = \{V, F\}$, then we can make the following identifications:

$$\oplus_V = \vee, \qquad \oplus_F = \wedge, \qquad Q_V = \exists, \qquad Q_F = \forall.$$

Whither negation? In 2-person Game Semantics, negation is interpreted as rôle interchange. This generalizes in the multi-agent setting to *rôle permutation.* Each permutation $\pi \in S(\mathcal{A})$ ($S(X)$ being the symmetric group on the set X) induces a logical operation $\hat{\pi}(\varphi)$ of permutation of the rôles of the agents in the game corresponding to A. In the 2-agent cases, there are two permutations in $S(\{V, F\})$, the identity (a 'no-op'), and the transposition $V \leftrightarrow F$, which corresponds exactly to the usual game-theoretical negation.

Other connectives In the light of our remarks about the essentially *additive* character (in the sense of Linear Logic) of the connectives $\oplus_\alpha$ and their game-semantical interpretation, it is also natural to consider some multiplicative-style connectives. We shall introduce two very basic connectives of this kind, which will prove particularly useful for the compositional analysis of branching quantifiers:

[1] This is of course quite analogous to the "paradoxes of material implication". Our point is that the game structure, if taken seriously as a way of articulating interactive behaviour, rather than merely being used as a carrier for standard model-theoretic notions, opens up new and more interesting possibilities.

1. **Parallel Composition**, $\varphi \| \psi$. The intended semantics is that $G(\varphi \| \psi)$ is the game in which play in $G(\varphi)$ and $G(\psi)$ proceeds in parallel.

2. **Sequential Composition**, $\varphi \cdot \psi$. Here we firstly play in $G(\varphi)$ to a conclusion, and then play in $G(\psi)$.

It will also be useful to introduce a constant $\mathbf{1}$ for a "neutral" or "vacuously true" proposition. The intended semantics is an empty game, in which nothing happens. Thus we should expect $\mathbf{1}$ to be a unit for both sequential and parallel composition.

Thus the syntax for our multi-agent logic $\mathcal{L}_{\mathcal{A}}$ stands as follows.

$$\varphi \quad ::= \quad \mathbf{1} \;\mid\; A \;\mid\; \varphi \oplus_{\alpha} \psi \;\mid\; Q_{\alpha}x.\,\varphi \;\mid\; \varphi \cdot \psi \;\mid\; \varphi \| \psi \;\mid\; \hat{\pi}(\varphi).$$

Here A ranges over atomic formulas, and $\pi \in S(\mathcal{A})$.

Quantifiers as Particles The syntax of $\mathcal{L}_{\mathcal{A}}$ is more powerful than may first appear. Consider an idea which may seem strange at first, although it has also arisen in Dynamic Game Logic [vBen03] and implicitly in the semantics of non-deterministic programming languages [AP81]. Instead of treating quantifiers as prefixing operators $Q_{\alpha}x.\varphi$ in the usual way, we can consider them as stand-alone particles $Q_{\alpha}x \equiv Q_{\alpha}x.\mathbf{1}$. We should expect to have

$$(Q_{\alpha}x) \cdot \varphi \;\equiv\; (Q_{\alpha}x.\mathbf{1}) \cdot \varphi \;\equiv\; Q_{\alpha}x.(\mathbf{1} \cdot \varphi) \;\equiv\; Q_{\alpha}x.\,\varphi. \qquad (1)$$

Thus this particle view of quantifiers does not lose any generality with respect to the usual syntax for quantification. But we can also express much more.

Example 2.1 *(2-agent case).*

$$[(\forall x \exists y) \;\|\; (\forall u \exists v)] \cdot A(x, y, u, v).$$

*This is the **Henkin quantifier**, expressed compositionally in the syntax of* $\mathcal{L}_{\mathcal{A}}$.

More generally:

Proposition 2.2 *Every partially-ordered quantifier prefix in which the partial order is a series-parallel poset can be expressed in the syntax of* $\mathcal{L}_{\mathcal{A}}$.

We could therefore recast the grammar of $\mathcal{L}_{\mathcal{A}}$ as follows:

$$\varphi \quad ::= \quad A \;\mid\; \varphi \oplus_{\alpha} \psi \;\mid\; Q_{\alpha}x \;\mid\; \varphi \cdot \psi \;\mid\; \varphi \| \psi \;\mid\; \hat{\pi}(\varphi).$$

However, we shall stick to the previous syntax, as this is more familiar, and will provide us with an independent check that the expected equivalences (1) hold.

3 Semantics of $\mathcal{L}_{\mathcal{A}}$

We shall now develop a semantics for this logic. This semantics will be built in two levels:

1. **Static Semantics of Formulas** To each formula $\varphi \in \mathcal{L}_{\mathcal{A}}$, we shall assign a form of game rich enough to allow for concurrent actions, and for rather general forms of temporal or causal dependency between moves. This assignment will be fully compositional.

2. **Dynamic Semantics** We then formulate a notion of *strategy* for these games. For each agent $\alpha \in \mathcal{A}$ there will be a notion of α-strategy. We shall show to build strategies for the games arising from formulas, compositionally from the strategies for the sub-formulas. We shall also define the key notion of how to evaluate *strategy profiles*, *i.e.* a choice of strategy for each agent, interacting with each other to reach a collective outcome. We shall find an elegant mathematical expression for this, *prima facie* very complicated, operational notion.

   We shall also discuss the notion of *valuation of an outcome*, on the basis of which logical notions of *validity*, or game-theoretical notions of *equilibria* can be defined. However, we shall find that a fully compositional account of valuations will require the more refined analysis of syntax to be given in the next Section.

3.1 Static Semantics: Concrete Data Structures as Concurrent Games

The structures we shall find convenient, and indeed very natural, to use as our formal representations of games were introduced by Gilles Kahn and Gordon Plotkin in 1975 (although their paper only appeared in a journal in 1993) [KP78]. They arose for Kahn and Plotkin in providing a representation theory for their notion of *concrete domains.* The term used by Kahn and Plotkin for these structures was *information matrices*; subsequently, they have usually been called *concrete data structures*, and we shall follow this latter terminology (although in some ways, the original name is more evocative in our context of use).

What, then, is a concrete data structure (CDS)? It is a structure

$$M = (C, V, D, \vdash)$$

where:

- C is a set of *cells*, or 'loci of decisions'—places where the agent can make their moves. These cells have a spatio-temporal significance: they both allow the distributed nature of multi-agent interactions to be articulated, and also capture a causal or temporal flow between events, as we shall see.

- V is a set of 'values', which label the choices which can be made by the agents from their menus of possible moves: for example, choosing the first or second branch of a compound formula $\varphi \oplus_\alpha \psi$, or choosing an instance for a quantifier.

- $D \subseteq C \times V$ is a set of *decisions*, representing the possible choices which can be made for how to 'fill' a cell. Note that specifying D allows some primitive typing for cells; only certain choices are appropriate for each given cell. (The more usual terminology for decisions is 'events'; here we have followed Kahn and Plotkin's original terminology, which is very apt for our purposes.)

- The relation $\vdash \subseteq \mathcal{P}_{\mathsf{f}}(D) \times C$ is an *enabling relation* which determines the possible temporal flows of events in a CDS. ($\mathcal{P}_{\mathsf{f}}(X)$ is the set of *finite* subsets of X.)

A *state* or *configuration* over a CDS M is a set $s \subseteq D$ such that:

- s is a partial function, *i.e.* each cell is filled at most once.

- If $(c, v) \in s$, then there is a sequence of decisions

$$(c_1, v_1), \ldots, (c_k, v_k) = (c, v)$$

  in s such that, for all j, $1 \leq j \leq k$, for some $\Gamma_j \subseteq \{(c_i, v_i) \mid 1 \leq i < j\}$:

$$\Gamma_j \vdash c_j.$$

  This is a "causal well-foundedness" condition. (Kahn and Plotkin phrase it as: "*c has a proof in x*".) Note that, in order for there to be any non-empty states, there must be *initial cells* c_0 such that $\varnothing \vdash c_0$.

We write $\mathcal{D}(M)$ for the set of states, partially ordered by set inclusion. This is a concrete domain in the sense of Kahn and Plotkin. In particular, it is closed under directed unions and unions of bounded families, and the finite states form a basis of compact elements, so it is algebraic.

To obtain a structure to represent a multi-agent game, we shall consider a CDS M augmented with a *labelling map*

$$\lambda_M : C_M \longrightarrow \mathcal{A}$$

which indicates which agent is responsible for filling each cell. We call (M, λ_M) an $\mathcal{A}$-*game*.

We are now ready to specify the compositional assignment of an $\mathcal{A}$-game to each formula of $\mathcal{L}_{\mathcal{A}}$. We assume a set $\mathcal{I}$ which will be used as the domain of quantification. We shall use $\uplus$ for the disjoint union of sets.

- **Constant 1**. This is assigned the empty CDS $(\varnothing, \varnothing, \varnothing, \varnothing)$.
- **Atomic formulas** A. These are assigned the empty CDS $(\varnothing, \varnothing, \varnothing, \varnothing)$.
- **Choice connectives** $\varphi \oplus_\alpha \psi$. Let

$$M = [\![\varphi]\!], \qquad N = [\![\psi]\!]$$

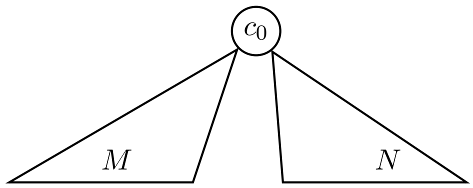
  be the CDS assigned to φ and ψ, with labelling functions λ_M and λ_N. Then the CDS $M \oplus_\alpha N$ is defined as follows:

$$(C_M \uplus C_N \uplus \{c_0\}, V_M \cup V_N \cup \{1, 2\}, D_M \uplus D_N \uplus \{(c_0, 1), (c_0, 2)\}, \vdash_{M \oplus_\alpha N})$$

  where

$$\begin{array}{l} \vdash_{M\oplus_\alpha N} c_0 \\ (c_0, 1), \Gamma \vdash_{M\oplus_\alpha N} c \iff \Gamma \vdash_M c \\ (c_0, 2), \Gamma \vdash_{M\oplus_\alpha N} c \iff \Gamma \vdash_N c. \end{array}$$

  This is the standard *separated sum* construction on CDS as in [KP78]. Pictorially:



Initially, only the new cell c_0 is enabled. It can be filled with either of the values 1 or 2. If it is filled by 1, we can proceed as in $M = [\![\varphi]\!]$, while if it is filled with 2, we proceed as in $N = [\![\psi]\!]$. This makes the usual informal specification precise, in a rather general setting.

To complete the specification of $M \oplus_\alpha N$ as an $\mathcal{A}$-game, we specify the labelling function $\lambda_{M\oplus_\alpha N}$:

$$\begin{array}{lll} c_0 & \mapsto & \alpha \\ c & \mapsto & \lambda_M(c) \quad (c \in C_M) \\ c & \mapsto & \lambda_N(c) \quad (c \in C_N). \end{array}$$

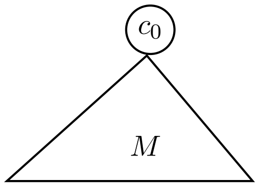
As expected, the initial cell c_0 must be filled by the agent α; other cells are filled as in the corresponding sub-games.

- **Quantifiers** $Q_\alpha x.\, \varphi$. Let $M = [\![\varphi]\!]$.

$$Q_\alpha(M) \quad = \quad (C_M \uplus \{c_0\}, V_M \cup \mathcal{I}, D_M \uplus (\{c_0\} \times \mathcal{I}), \vdash_{Q_\alpha(M)}).$$

$$\begin{array}{l} \vdash_{Q_\alpha(M)} c_0 \\ (c_0, a), \Gamma \vdash_{Q_\alpha(M)} c \iff \Gamma \vdash_M c \quad (a \in \mathcal{I}). \end{array}$$

This is a variant of the standard *lifting* construction on CDS.



Initially, only the new cell c_0 is enabled. It can be filled with any choice of individual a from the domain of quantification $\mathcal{I}$. Subsequently, we play as in M. The labelling function, $\lambda_{Q_\alpha(M)}$:

$$c_0 \mapsto \alpha, \qquad c \mapsto \lambda_M(c) \quad (c \in C_M).$$

Although the interpretation of the quantifier particle $Q_\alpha x \equiv Q_\alpha x.\, \mathbf{1}$ can be derived from the definitions already given, we set it out explicitly to show how simple and natural it is:

$$[\![Q_\alpha x]\!] = (\{c_0\}, \mathcal{I}, \{c_0\} \times \mathcal{I}, \{\vdash c_0\}),$$

with labelling function $c_0 \mapsto \alpha$. Thus the game consists of a single cell, labelled by α, initially enabled, in which any element from $\mathcal{I}$ can be chosen — a true "particle of action" in a multi-agent setting.

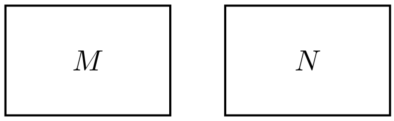
- **Parallel Composition** $\varphi \| \psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$: we define

$$M \| N \quad = \quad (C_M \uplus C_N, V_M \uplus V_N, D_M \uplus D_N, \vdash_M \uplus \vdash_N).$$

The labelling function is defined by:

$$\lambda_{M\|N}(c) = \begin{cases} \lambda_M(c) & (c \in C_M) \\ \lambda_N(c) & (c \in C_N). \end{cases}$$

Pictorially:



Decisions in M and N can be made concurrently, with no causal or temporal constraints between them. This is the standard *product* construction on CDS.

- **Sequential Composition** $\varphi \cdot \psi$. We say that a state $s \in \mathcal{D}(M)$ is *maximal* if

$$\forall t \in \mathcal{D}(M)[s \subseteq t \;\Rightarrow\; s = t].$$

We write $\mathsf{Max}(M)$ for the set of maximal elements of $\mathcal{D}(M)$.

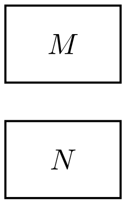
Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$: we define

$$M \cdot N \quad = \quad (C_M \uplus C_N, V_M \uplus V_N, D_M \uplus D_N, \vdash_{M \cdot N}),$$

where

$$\Gamma \vdash_{M \cdot N} c \quad \Longleftrightarrow \quad \Gamma_M \vdash c \quad \vee \quad (\Gamma = s \cup \Delta \;\;\wedge\;\; s \in \mathsf{Max}(M) \;\;\wedge\;\; \Delta \vdash_N c).$$

Pictorially:



The idea is that firstly we reach a maximal state in M—a "complete play"—and then we can continue in N. Note that this construction makes sense for arbitrary CDS M, N. Even if M has infinite maximal states, the finitary nature of the enabling relation means that no events from N can occur in $M \cdot N$ following an infinite play in M.

The labelling function is defined by:

$$\lambda_{M \cdot N}(c) = \begin{cases} \lambda_M(c) & (c \in C_M) \\ \lambda_N(c) & (c \in C_N). \end{cases}$$

Note that the difference between $M \| N$ and $M \cdot N$ is purely one of *temporality* or *causality*: when events can occur (or, in the alternative terminology, decisions can be made) relative to each other.

- **Role Switching** $\hat{\pi}(\varphi)$, $\pi \in S(\mathcal{A})$. The CDS $[\![\hat{\pi}(\varphi)]\!]$ is *the same* as $M = [\![\varphi]\!]$. However:

$$\lambda_{\hat{\pi}(M)} = \pi \circ \lambda_M.$$

3.2 Dynamic Semantics: Concurrent Strategies

We now turn to the task of defining a suitable notion of *strategy* for our games. We shall view a strategy for an agent α on the game M as a function $\sigma : \mathcal{D}(M) \to \mathcal{D}(M)$. The idea is that $\sigma(s)$ shows the moves which agent α would make, in the situation represented by the state s, when following the strategy represented by σ. Some formal features of σ follow immediately from this:

- (S1) Since past moves cannot be undone, we must have $s \subseteq \sigma(s)$, *i.e.* σ is *increasing.*
- (S2) If $(c, v) \in \sigma(s) \setminus s$, it must be the case that $\lambda_M(c) = \alpha$, since α is only able to make decisions in its own cells.

We shall impose two further conditions. While not quite as compelling as the two above, they also have clear motivations.

- (S3) Idempotence: $\sigma(\sigma(s)) = \sigma(s)$. Since the only information in $\sigma(s)$ over and above what is in s is what σ put there, this is a reasonable normalizing assumption. It avoids situations where σ proceeds 'slowly', making decisions in several steps which it could have taken in one step (since the information from the Environment, *i.e.* the other agents, has not changed).
- (S4) Monotonicity: $s \subseteq t \Rightarrow \sigma(s) \subseteq \sigma(t)$. This condition reflects the idea that states only contain *positive* information. The fact that a cell c has *not* been filled yet is not a definite, irrevocable piece of information. Another strategy, working on behalf of another agent, may be running concurrently with us, and just about to fill c.

Finally, there is a more technical point, which is a necessary complement to the above conditions. We take σ to be a function on $\mathcal{D}(M)^\top$, which is obtained by adjoining a top element $\top$ to $\mathcal{D}(M)$. Note that, since σ is increasing, we must have $\sigma(\top) = \top$. The significance of adding a top element to the codomain is that it allows for *partial strategies*, which are undefined in some situations.

The following result is standard.

Proposition 3.1 *$\mathcal{D}(M)^\top$ is an algebraic complete lattice.*

Taking the conditions (S1), (S3), (S4) together says that σ is a *closure operator* on $\mathcal{D}(M)^\top$. A closure operator additionally satisfying (S2) is said to be an α-closure operator. We write $\mathsf{Cl}_\alpha(M)$ for the set of α-closure operators on $\mathcal{D}(M)^\top$.

The full specification of the game based on a CDS M will comprise a set of strategies $S_\alpha(M) \subseteq \mathsf{Cl}_\alpha(M)$ for each agent α. By limiting the set of strategies suitably, we can in effect impose constraints on the information available to agents.

3.2.1 Inductive Construction of Strategy Sets

There are several approaches to defining the strategy sets S_α. We are interested in *compositional definitions* of $S_\alpha(\llbracket\varphi\rrbracket)$. There are two main approaches to such defitions, both of which have been extensively deployed in Game Semantics as developed in Computer Science.

1. We can define the strategy sets themselves directly, by induction on the construction on φ. This is the "global" approach. It is akin to realizability, and in general leads to strategy sets of high logical complexity. See [AM99, Abr00] for examples of this approach.

2. We can use an indirect, more "local" approach, in which we add some structure to the underlying games, and use this to state conditions on strategies, usually phrased as conditions on individual plays or runs of strategies. $S_\alpha(\llbracket\varphi\rrbracket)$ is then defined to be the set of strategies in $\mathsf{Cl}_\alpha(\llbracket\varphi\rrbracket)$ satisfying these conditions. This has in fact been the main approach used in the Game Semantics of programming languages [AJM00, HO00]. However, this approach has not been developed for the kind of concurrent, multi-agent games being considered here.

It seems that both of these approaches may be of interest in the present context. We therefore show how both can be applied to the semantics of $\mathcal{L}_\mathcal{A}$. We begin with the local approach, which is perhaps closer to the intuitions.

3.2.2 Local conditions on strategies

The idea is to capture, as part of the structure of the underlying game, which information about the current state should be available to agent α when it makes a decision at cell c. This can be formalized by a function

$$\gamma_M : C_M \longrightarrow [\mathcal{D}(M) \longrightarrow \mathcal{D}(M)]$$

which for each cell c assigns a function $\gamma_M(c)$ on states. The idea is that, if $\lambda_M(c) = \alpha$, $\gamma_M(c)(s)$ restricts s to the part which should be visible to agent α, and on the basis of which he has to decide how to fill c. It follows that $\gamma_M(c)$ should be *decreasing*: $\gamma_M(c)(x) \subseteq x$. Note the duality of this condition to (S1). We add the assumptions of monotonicity and idempotence, with much the same motivation as for strategies. It follows that $\gamma_M(c)$ is a *co-closure operator*.

Notation Remembering that a state $s \in \mathcal{D}(M)$ is a partial function, we write $s\searrow c$ to mean that s is defined at the cell c, or that "s fills c", as it is usually expressed. Also, we write C_M^α for the set of α-labelled cells in C_M.

Now, given such a function γ_M, we can define the strategy set $S_\alpha(M)$:

$$S_\alpha(M) = \{\sigma \in \mathsf{Cl}_\alpha(M) \mid \forall s \in \mathcal{D}(M).\forall c \in C_M^\alpha.[\sigma(s)\searrow c \;\Rightarrow\; \sigma(\gamma_M(c)(s))\searrow c]\}. \tag{2}$$

Thus the information constraint imposed by γ_M is expressed by the condition that σ can only make a decision at cell c in state s if it would have made the same decision in the smaller (less information) state $\gamma_M(c)(s)$.[2] This is a direct analogue of the use of views in Hyland-Ong style games [HO00] to define 'innocent strategies'. However, apart from the more general format of our games, there is greater flexibility in the provision of the function γ_M as a separate component of the game structure, whereas specific view functions are built into the fabric of HO-games.

We now show how the functions $\gamma_{\llbracket\varphi\rrbracket}$ can be defined, compositionally in φ.

- **Atomic formulas, and constant 1**. This case is trivial, since the set of cells is empty.
- **Choice connectives** $\varphi \oplus_\alpha \psi$. Let $M = \llbracket\varphi\rrbracket$, $N = \llbracket\psi\rrbracket$. We define $\gamma_{M\oplus_\alpha N}$ by:

$$\gamma_{M\oplus_\alpha N}(c)(s) = \begin{cases} \varnothing, & c = c_0 \\ \{(c_0, 1)\} \cup \gamma_M(c)(s \setminus (c, 1)), & (c \in C_M) \\ \{(c_0, 2)\} \cup \gamma_N(c)(s \setminus (c, 2)), & (c \in C_N). \end{cases}$$

[2] It appears that our condition only requires that the cell c be filled somehow in $\gamma_M(c)(x)$; however, monotonicity of σ ensures that if it is filled in $\gamma_M(c)(s)$, then it must be filled *with the same value* in s.

- **Quantifiers** $Q_\alpha.\varphi$. Let $M = [\![\varphi]\!]$.

$$\gamma_{Q_\alpha(M)}(c_0)(s) = \varnothing, \qquad \gamma_{Q_\alpha(M)}(c)(\{(c_0, a)\} \uplus s) = \{(c_0, a)\} \cup \gamma_M(c)(s) \quad (c \in C_M).$$

  Thus the choice initially made by α to decide the value of the quantifier is visible to all the agents.

- **Parallel Composition** $\varphi \| \psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We define $\gamma_{M\|N}$ by:

$$\gamma_{M\|N}(c)(s) = \begin{cases} \gamma_M(c)(\pi_M(s)), & c \in C_M \\ \gamma_N(c)(\pi_N(s)), & c \in C_N \end{cases}$$

  Here π_M, π_N are the *projection functions*; e.g.

$$\pi_M(s) = \{(c, v) \in s \mid c \in C_M\}.$$

  Thus the view at a cell in the sub-game M or N is what it would have been if we were playing only in that sub-game. This implements a complete block on information flow between the two sub-games. It can be seen as corresponding directly to the Linear Logic connective $\otimes$.

- **Sequential Composition** $\varphi \cdot \psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We define $\gamma_{M\cdot N}$ by:

$$\gamma_{M\cdot N}(c)(s) = \begin{cases} \gamma_M(c)(\pi_M(s)), & c \in C_M \\ \pi_M(s) \cup \gamma_N(c)(\pi_N(s)), & c \in C_N. \end{cases}$$

  Thus while we are playing in M, visibility is at it was in that sub-game. When we have finished a complete play s in M and start to play in N, we can see the whole completed play s, together with what is visible in the sub-game N.

- **Role Permutation** $\hat{\pi}(\varphi)$. We set $\gamma_{[\![\hat{\pi}(\varphi)]\!]} = \gamma_{[\![\varphi]\!]}$. The same information is available from each cell; but, for example, if agent α had more information available than agent β in M, that advantage will be transferred to β in $\hat{\pi}(M)$ if π interchanges α and β.

3.2.3 Global definitions of strategy sets

We define the strategy sets $S_\alpha([\![\varphi]\!])$ compositionally from the construction of φ. The main point to note is the constructions on strategies which arise in making these definitions; these show the *functorial character* of the game-semantical interpretation of the connectives, and point the way towards a *semantics of proofs*—or indeed, in the first instance, to what the proof system should be— for the logic.

- **Atomic formulas and constant 1**. These cases are trivial, since the set of cells is empty. Thus $\mathcal{D}([\![A]\!]) = \mathcal{D}([\![\mathbf{1}]\!]) = \{\varnothing\}$. We set $S_\alpha([\![A]\!]) = S_\alpha([\![\mathbf{1}]\!]) = \{\mathsf{id}_{\{\varnothing\}}\}$.

- **Choice connectives** $\varphi \oplus_\alpha \psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We firstly define some constructions on closure operators:

$$\mathsf{in}_1 : \mathsf{Cl}_\alpha(M) \longrightarrow \mathsf{Cl}_\alpha(M \oplus_\alpha N), \quad \mathsf{in}_2 : \mathsf{Cl}_\alpha(N) \longrightarrow \mathsf{Cl}_\alpha(M \oplus_\alpha N)$$

$$\oplus : \mathsf{Cl}_\beta(M) \times \mathsf{Cl}_\beta(N) \longrightarrow \mathsf{Cl}_\beta(M \oplus_\alpha N) \ \ (\beta \neq \alpha).$$

For $i = 1, 2$:

$$\mathsf{in}_i(\sigma)(s) = \begin{cases} \{(c_0, i)\} \cup \sigma(s \setminus \{(c_0, i)\}), & (c_0, j) \in s \Rightarrow j = i \\ \top & \text{otherwise.} \end{cases}$$

$$\begin{array}{lcl} \sigma \oplus \tau(\varnothing) & = & \varnothing \\ \sigma \oplus \tau(\{(c_0, 1)\} \uplus s) & = & \{(c_0, 1)\} \cup \sigma(s) \\ \sigma \oplus \tau(\{(c_0, 2)\} \uplus t) & = & \{(c_0, 2)\} \cup \tau(t). \end{array}$$

Thus $\mathsf{in}_1(\sigma)$ is the strategy for α in $M \oplus_\alpha N$ which firstly decides to play in M, and then subsequently plays like σ, which is ("inductively") assumed to be a strategy for α in M. Similarly for $\mathsf{in}_2(\tau)$. Note that both these strategies are "non-strict": that is, $\mathsf{in}_i(\sigma)(\varnothing)$ will at least contain (c_0, i). This reflects the idea that agent α must play the first move in $M \oplus_\alpha N$, and nothing can happen until he does. The strategy $\sigma \oplus \tau$ for another agent $\beta \neq \alpha$ must on the other hand "wait" at the initial state $\varnothing$ until α has made its decision. Once this has happened, it plays according to σ if the decision was to play in M, and according to τ if the decision was to play in N.

Note how the definitions of in_1, in_2 show why strategies must in general be partial; there is a "self-consistency" condition that we should only be confronted with situations in which the decisions ascribed to us are indeed those we actually made.

We now define the strategy sets for $M \oplus_\alpha N$:

$$\begin{array}{lcl} S_\alpha(M \oplus_\alpha N) & = & \{\mathsf{in}_1(\sigma) \mid \sigma \in S_\alpha(M)\} \cup \{\mathsf{in}_2(\tau) \mid \tau \in S_\alpha(N)\} \\ S_\beta(M \oplus_\alpha N) & = & \{\sigma \oplus \tau \mid \sigma \in S_\beta(M) \ \wedge \ \tau \in S_\beta(N)\} \quad (\beta \neq \alpha). \end{array}$$

- **Quantifiers** $Q_\alpha(\varphi)$. Let $M = [\![\varphi]\!]$. We define operations

$$\oplus_{a \in \mathcal{I}} : \mathsf{Cl}_\beta(M)^{\mathcal{I}} \longrightarrow \mathsf{Cl}_\beta(Q_\alpha(M)), \quad (\beta \neq \alpha),$$

and, for each $a \in \mathcal{I}$:

$$\mathsf{up}_a : \mathsf{Cl}_\alpha(M) \longrightarrow \mathsf{Cl}_\alpha(Q_\alpha(M)).$$

$$\mathsf{up}_a(\sigma)(s) = \begin{cases} \{(c_0, a)\} \cup \sigma(s \setminus \{(c_0, a)\}), & (c_0, b) \in s \Rightarrow b = a \\ \top & \text{otherwise.} \end{cases}$$

$$\begin{array}{lcl} (\oplus_{a \in \mathcal{I}} \sigma_a)(\varnothing) & = & \varnothing \\ (\oplus_{a \in \mathcal{I}} \sigma_a)(\{(c_0, b)\} \uplus s) & = & \{(c_0, b)\} \cup \sigma_b(s). \end{array}$$

Note the similarity of these operations to those defined for the choice connectives. (In fact, the separated sum $M \oplus N$ can be seen as the composite $(M + N)_\perp$ of disjoint sum and lifting constructions.) Note also, in the definition of $\oplus_{a \in \mathcal{I}} \sigma_a$, the dependence of the strategy σ_b used to continue the play on the element $b \in \mathcal{I}$ initially chosen by α.

We define the strategy sets as follows:

$$\begin{array}{lcl} S_\alpha(Q_\alpha(M)) & = & \{\mathsf{up}_a(\sigma) \mid a \in \mathcal{I}, \sigma \in S_\alpha(M)\} \\ S_\beta(Q_\alpha(M)) & = & \{\oplus_{a \in \mathcal{I}} \sigma_a \mid \forall a \in \mathcal{I}.\, \sigma_a \in S_\beta(M)\} \quad (\beta \neq \alpha). \end{array}$$

- **Parallel Composition** $\varphi\|\psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We define an operation

$$\| : \mathsf{Cl}_\alpha(M) \times \mathsf{Cl}_\alpha(N) \longrightarrow \mathsf{Cl}_\alpha(M\|N)$$

$$\sigma\|\tau(s) = \sigma(\pi_M(s)) \cup \tau(\pi_N(s)).$$

This is just the functorial action of the product, and gives the "information independence" of play in the two sub-games. Now we define the strategy sets:

$$S_\alpha(M\|N) = \{\sigma\|\tau \mid \sigma \in S_\alpha(M) \ \wedge \ \tau \in S_\alpha(N)\}.$$

- **Sequential Composition** $\varphi \cdot \psi$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. Given $\sigma \in \mathsf{Cl}_\alpha(M)$, and a family $(\tau_s)_{s\in\mathsf{Max}(M)} \subseteq \mathsf{Cl}_\alpha(N)$ indexed by maximal states in M, we define:

$$(\sigma \cdot (\tau_s)_s)(t) = \begin{cases} \sigma(t), & t \in \mathcal{D}(M), \sigma(t) \notin \mathsf{Max}(M) \\ \sigma(t) \cup \tau_{\sigma(t)}(\varnothing), & t \in \mathcal{D}(M), \sigma(t) \in \mathsf{Max}(M) \\ s \cup \tau_s(u), & t = s \cup u, s \in \mathsf{Max}(M), u \in \mathcal{D}(N). \end{cases}$$

This allows for arbitrary dependency of agent α's play in N on what previously occurred in M. Note that in the case that σ completes a maximal play s in M, control passes immediately to τ_s to continue in N. We then define

$$S_\alpha(M \cdot N) = \{\sigma \cdot (\tau_s)_s \mid \sigma \in S_\alpha(M) \ \wedge \ \forall s \in \mathsf{Max}(M).[\tau_s \in S_\alpha(N)]\}.$$

- **Role Permutation** $\hat{\pi}(\varphi)$. Let $M = [\![\varphi]\!]$. Here we simply set $S_\alpha(\hat{\pi}(M)) = S_{\pi^{-1}(\alpha)}(M)$.

3.2.4 Comparison of the local and global definitions

Having defined strategy sets in these two contrasting fashions, we must compare them. Let φ be a formula of $\mathcal{L}_\mathcal{A}$. We write $S^{\mathsf{g}}_\alpha([\![\varphi]\!])$ for the strategy set for $[\![\varphi]\!]$ defined according to the global construction, and similarly $S^{\mathsf{l}}_\alpha([\![\varphi]\!])$ for the local definition (2) using the visibility function $\gamma_{[\![\varphi]\!]}$.

Proposition 3.2 *For all* $\varphi \in \mathcal{L}_\mathcal{A}$, $S^{\mathsf{g}}_\alpha([\![\varphi]\!]) \subseteq S^{\mathsf{l}}_\alpha([\![\varphi]\!])$.

Proof A straightforward induction on φ. Note in particular that $\sigma\|\tau$ satisfies the local condition (2) with respect to the parallel composition. □

The converse is false in general. The strategies in $S^{\mathsf{g}}_\alpha([\![\varphi]\!])$ have two important global properties which strategies satisfying the local condition (2) need not possess.

Safety We define the *domain* $\mathsf{dom}(\sigma)$ of a closure operator $\sigma \in \mathsf{Cl}_\alpha(M)$ to be the least subset of $\mathcal{D}(M)^\top$ satisfying the following conditions:

$(D1)$ $\varnothing \in \mathsf{dom}(\sigma)$

$(D2)$ $s \in \mathsf{dom}(\sigma) \ \Rightarrow \ \sigma(s) \in \mathsf{dom}(\sigma)$

$(D3)$ $S \subseteq \mathsf{dom}(\sigma),\ S$ directed $\ \Rightarrow \ \bigcup S \in \mathsf{dom}(\sigma)$

$(D4)$ $s \in \mathsf{dom}(\sigma), s \subseteq t \in \mathcal{D}(M), [(c, v) \in t \setminus s \ \Rightarrow \ \lambda_M(c) \neq \alpha] \ \Rightarrow \ t \in \mathsf{dom}(\sigma)$

Note that $(D1)$–$(D3)$ are the usual inductive definition of the set of iterations leading to the least fixpoint of σ. The condition $(D4)$ gives this definition its game-theoretic or multi-agent character.

We say that σ is *safe* if $\top \notin \mathsf{dom}(\sigma)$. Thus if σ is never confronted by α-moves that it would not itself have made, then whatever the other agents do it will not "crash" or "abort" (which is how we think of $\top$).

Proposition 3.3 *For all $\varphi \in \mathcal{L}_{\mathcal{A}}$, every strategy in $S^{\mathsf{g}}_{\alpha}([\![\varphi]\!])$ is safe.*

Progress We consider the following assumptions on strategies $\sigma \in \mathsf{Cl}_{\alpha}(M)$:

- (WP) If $s \in \mathcal{D}(M)$ contains an enabling of some α-cell c which is not filled in s, then $\sigma(s) \neq s$. In other words, σ does *something* (makes at least one decision) whenever it can.

- (MP) For all $s \in \mathcal{D}(M)$, if $\sigma(s)$ contains an enabling of some α-cell c, then it fills c. Thus σ decides *every* α-cell as soon as it becomes accessible.

We call (WP) the *weak progress assumption* and (MP) the *maximal progress assumption.* Clearly (MP) implies (WP).

Lemma 3.4 *The weak progress assumption implies the maximal progress assumption, and hence the two conditions are equivalent.*

Proof Let σ be an α-strategy not satisfying (MP). Then there must be a state s such that some α-cells are accessible but not filled in $\sigma(s)$. By idempotence, $\sigma(\sigma(s)) = \sigma(s)$, and hence σ does not satisfy (WP). □

Proposition 3.5 *For all $\varphi \in \mathcal{L}_{\mathcal{A}}$, every strategy in $S^{\mathsf{g}}_{\alpha}([\![\varphi]\!])$ satisfies the maximal progress assumption (MP).*

Given an $\mathcal{A}$-game M, we define $S^{\mathsf{lsp}}_{\alpha}(M)$ to be the set of all strategies in $S^{\mathsf{l}}_{\alpha}(M)$ which are safe and satisfy the weak progress assumption.

Theorem 3.6 (Characterization Theorem) *For all $\varphi \in \mathcal{L}_{\mathcal{A}}$, $S^{\mathsf{g}}_{\alpha}([\![\varphi]\!]) = S^{\mathsf{lsp}}_{\alpha}([\![\varphi]\!])$.*

Proof By induction on φ. We indicate some cases.

1. Parallel composition. For a strategy $\sigma \in S^{\mathsf{lsp}}_{\alpha}(M \| N)$, the visibility condition implies that $\sigma = \sigma_1 \| \sigma_2$. The safety of σ implies that of σ_1 and σ_2, and similarly (MP) for σ implies (MP) for σ_1 and σ_2. Thus $\sigma_1 \in S^{\mathsf{lsp}}_{\alpha}(M)$ and $\sigma_2 \in S^{\mathsf{lsp}}_{\alpha}(N)$, and we can apply the induction hypothesis. The case for sequential composition is similar.

2. Choice connectives. Here the progress assumption implies that the strategy holding the initial cell must fill it. Safety implies that strategies for other players must have the form $\sigma \oplus \tau$. Play after the initial cell is filled reduces to play in the chosen sub-game, and we can apply the induction hypothesis.

□

Thus we explicitly characterize the "immanent" properties of the strategy sets $S^{\mathsf{g}}_{\alpha}([\![\varphi]\!])$ in terms of local conditions on information visibility, plus safety and liveness properties.

3.3 Evaluation of Strategy Profiles

Consider a CDS M with a strategy set S_α for each agent $\alpha \in \mathcal{A}$. A *strategy profile* is an $\mathcal{A}$-tuple

$$(\sigma_\alpha)_{\alpha \in \mathcal{A}} \in \prod_{\alpha \in \mathcal{A}} S_\alpha$$

which picks out a choice of strategy for each $\alpha \in \mathcal{A}$. The key operation in "bringing the semantics to life" is to define the result or *outcome* of playing these strategies off against each other. Given the concurrent nature of our games, and the complex forms of temporal dependency and information flow which may arise in them, it might seem that a formal definition of this operation will necessarily be highly complex and rather messy. In fact, our mathematical framework allows for a very elegant and clean definition. We shall use the notation $\langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}}$ for this operation. It maps strategy profiles to *states of M*. The idea is that the state arising from $\langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}}$ will be that reached by starting in the initial state $\varnothing$, and repeatedly playing the strategies in the profile until no further moves can be made. The formal definition is as follows.

Definition 3.7 *We define $\langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}}$ to be the* least common fixpoint *of the family of closure operators $(\sigma_\alpha)_{\alpha \in \mathcal{A}}$; i.e. the least element s of $\mathcal{D}(M)^\top$ such that $\sigma_\alpha(s) = s$ for all $\alpha \in \mathcal{A}$.*

The following Proposition (which is standard) guarantees that this definition makes sense.

Proposition 3.8 *Any family of closure operators C on a complete lattice L has a common least fixpoint. In case the lattice has finite height, or $C = \{c_1, \ldots, c_n\}$ is finite and the closure operators in C are continuous (i.e. preserve directed joins) this common least fixpoint can be obtained constructively by the expression*

$$\bigvee_{k \in \omega} c^k(\bot), \qquad \textit{where } c = c_1 \circ \cdots \circ c_k.$$

Any permutation of the order of the composition in defining c, or indeed any "schedule" which ensures that each closure operator is applied infinitely often, will lead to the same result.

We recall the notion of *safety* from the previous Section.

Proposition 3.9 *Let $(\sigma_\alpha)_{\alpha \in \mathcal{A}}$ be a strategy profile in which σ_α is safe for all $\alpha \in \mathcal{A}$. Then $\langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}} \neq \top$.*

Proof We prove by transfinite induction on the iterations towards the least fixpoint that every state which is reached is in the domain of every strategy in the profile. The base case is $(D1)$, and the limit ordinal case is $(D3)$. Applying some σ_α to the current state stays in the domain of σ_α by $(D2)$, and in the domain of every other strategy in the profile by $(D4)$. □

In particular, we know by Proposition 3.3 that this applies to our setting, where we have a formula $\varphi \in \mathcal{L}_{\mathcal{A}}$, with corresponding CDS $M = [\![\varphi]\!]$ and strategy sets $S_\alpha([\![\varphi]\!])$, $\alpha \in \mathcal{A}$. Furthermore, we have:

Proposition 3.10 *For all $\varphi \in \mathcal{L}_{\mathcal{A}}$, and every strategy profile $(\sigma_\alpha)_{\alpha \in \mathcal{A}} \in S^{\mathsf{g}}_{\alpha}([\![\varphi]\!]) = S^{\mathsf{lsp}}_{\alpha}([\![\varphi]\!])$:*

$$\langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}} \in \mathsf{Max}([\![\varphi]\!]).$$

Proof Firstly, we know by Proposition 3.9 that $\langle\sigma_\alpha\rangle_{\alpha\in\mathcal{A}} \neq \top$. Let $s = \langle\sigma_\alpha\rangle_{\alpha\in\mathcal{A}}$. If s is not maximal, some cell c must be accessible but not filled in s. Suppose c is an α-cell. Since σ_α satisfies (WP), we must have $\sigma_\alpha(s) \neq s$, contradicting the definition of $\langle\sigma_\alpha\rangle_{\alpha\in\mathcal{A}}$ as a common fixpoint of all the strategies in the profile. □

Thus the outcome of evaluating a strategy profile in the game arising from any formula is always a well-defined maximal state.

Remark We pause to mention another connection with Theoretical Computer Science. Our use of closure operators as strategies, and the definition of the evaluation of strategy profiles as least common fixpoints, builds on ideas which arose originally in the semantics of *dataflow* [JPP89] and *concurrent constraint programming* [SRP]. They have also been applied extensively to constraint programming and constraint propagation algorithms over the past decade [Apt97]. Our own previous development of these ideas appears in a number of papers [AM99, Abr00a, Abr03].

3.4 Outcomes and Valuations

One ingredient which has been missing thus far in our account of the semantics of $\mathcal{L}_\mathcal{A}$ has been any notion of *payoff* or *utility* in game-theoretic terms, or of *truth-valuation* in logical terms, which may serve as a basis for game-theoretical notions of *equilibria* or logical notions such as *validity*. The status of the standard notions on the logical side is far from clear when we pass to multi-agent games. However, we can certainly provide support for studying equilibrium notions in our framework, in such a way that these specialize to the usual logical notions in the two-agent case. We shall only enter into a preliminary discussion here, simply to indicate some of the possibilities.

As we have just seen, for the games arising from formulas in $\mathcal{L}_\mathcal{A}$, evaluation of strategy profiles always leads to maximal states. Moreover, the CDS corresponding to any formula has only finitely many *cells* (although if $\mathcal{I}$ is infinite, so also will be the sets of values, decisions and states). Hence any state consists of only finitely many decisions.

Proposition 3.11 *For any closed formula $\varphi \in \mathcal{L}_\mathcal{A}$, a maximal state in $[\![\varphi]\!]$ corresponds to a combination of atomic sentences, built by series and parallel composition from (instances of) atomic subformulas of φ. If φ is built from atomic formulas using only the choice connectives and quantifiers, maximal states will correspond exactly to single instances of atomic subformulas.*

Proof Given a maximal state s, we argue by induction on φ. We indicate some cases:

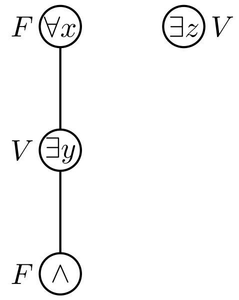
- $\varphi \oplus_\alpha \psi$. Then s contains (c_0, i). If $i = 1$, we continue with the formula φ and the maximal state of $[\![\varphi]\!]$ obtained by removing $(c_0, 1)$ from s. Similarly if $i = 2$.
- $Q_\alpha x.\, \varphi$. Then s contains (c_0, a). We continue inductively with $\varphi[a/x]$ and the maximal state of $[\![\varphi[a/x]]\!]$ obtained by removing (c_0, a) from s.
- $\varphi \| \psi$. We continue inductively with φ and $\pi_M(s)$, and with ψ and $\pi_N(s)$, and glue the results back together with parallel composition.
- $\varphi \cdot \psi$. Essentially the same as for parallel composition.

□

Example We take $\mathcal{A} = \{V, F\}$, and use standard notation for choice connectives and quantifiers. Consider the formula

$$\forall x. \exists y. [A(x, y) \ \wedge \ B(y)] \parallel \exists z. C(z).$$

The corresponding CDS has four cells:



In a maximal state of this CDS, these cells are all filled. If the $\forall x$ cell is filled with $a \in \mathcal{I}$, $\exists y$ with $b \in \mathcal{I}$, $\wedge$ with 1, and $\exists z$ with $c \in \mathcal{I}$, then the state will correspond to the following parallel composition of atomic sentences:

$$A(a, b) \parallel C(c).$$

In the usual Hintikka-style Game semantics, a model $\mathcal{M}$ is used to evaluate atomic sentences. We can see that in our setting, this work can equivalently be done by a *Boolean valuation function*

$$\mathsf{val} : \mathsf{Max}([\![\varphi]\!]) \longrightarrow \{0, 1\}.$$

So for 2-agent games, we could simply use such a valuation to give a notion of winning strategy for Verifier, and hence of logical validity.

The multi-agent case More generally, in the multi-agent case we should consider valuations

$$\mathsf{val}_\alpha : \mathsf{Max}([\![\varphi]\!]) \longrightarrow \mathcal{V}_\alpha$$

for each agent α, into some set of *preferences* or *utilities*. Given an outcome

$$o = \langle \sigma_\alpha \rangle_{\alpha \in \mathcal{A}} \in \mathsf{Max}([\![\varphi]\!]),$$

we can evaluate it from the perspective of each agent α as $\mathsf{val}_\alpha(o)$, and hence formulate notions such as Nash equilibrium and other central game-theoretical notions.

Compositionality? Until now our semantics has been fully compositional. However, as things stand, valuation functions *cannot* be described in a compositional fashion. The problem becomes clear if we consider our treatment of atomic formulas. Their current representation in our semantics is vacuous — the empty CDS. This carries no information which can be used by a valuation function to express the dependence of an outcome on the values previously chosen for the variables appearing in the atomic formula. We can recover this correspondence globally, as in Proposition 3.11, but not *compositionally*, by gluing together a valuation function defined for an atomic formula with those arising from the context in which it occurs.

We shall now give a reformulation of the syntax of $\mathcal{L}_\mathcal{A}$ which will allow us to take full account of the role of variables and variable-binding in our semantics, and hence provide the basis for a compositional treatment both of valuation functions, and of IF-quantifiers and other partial-information constructs.

4 Towards environmentally friendly logic

It is, or should be, an aphorism of semantics that:

> **The key to compositionality is parameterization**.

Choosing the parameters aright allows the meaning of expressions to be made sensitive to their contexts, and hence defined compositionally. While this principle could—in theory—be carried to the point of trivialization, in practice the identification of the right form of parameterization does usually represent some genuine insight into the structure at hand.

We shall now describe an approach to making the syntax of quantifier particles, including IF-quantifiers, fully compositional. This can then serve as a basis for a fully compositional account of valuations on outcomes.

4.1 Syntax as a Category

Note firstly a certain kind of quasi-duality between quantifiers and atomic formulas. Quantifiers $Q_\alpha x$ project the scope of x inwards over sequential compositions (but not across parallel compositions). Atomic formulas $A(x_1, \ldots, x_n)$ depend on variables coming from an outer scope.

Now consider IF-quantifiers $\forall x/y$ which bind x, but also declare that it does *not* depend on an outer quantification over y. This is a peculiar binding construct, quite apart from its semantic interpretation. The bidirectional reach of the scope—inwards for x, outwards for y—is unusual, and difficult to make sense of in isolation from a given context of use. So in fact, it seems hard to give a decent compositional *syntax* for IF-quantifiers, before we even start to think about semantics.

Once again, there is work coming from Theoretical Computer Science which is suggestive: namely the π-calculus [MPW92, MPW92a], with its *scope restriction* and *extrusion*. The action calculi subsequently developed by Milner [Mil93] are even more suggestive, although only certain features are relevant here.

We shall reformulate our view of logical syntax as follows. Each syntactic constituent will have an *arity* and a *co-arity*. Concretely, we shall take these arities and co-arities to be finite sets of variables, although algebraically we could just take them to be natural numbers. We shall write a syntactic expression as

$$\varphi : X \longrightarrow Y$$

where X is the arity, and Y is the co-arity. The idea is that the arity specifies the variables that φ expects to have bound by its outer environment, while the co-arity represents variables that it is binding with respect to its inner environment.

The quantifier particle $Q_\alpha x$ can be described in these terms as

$$Q_\alpha x : \varnothing \longrightarrow \{x\} \tag{3}$$

or more generally as

$$Q_\alpha x : X \longrightarrow X \uplus \{x\}.$$

An atom $A(x_1, \ldots, x_n)$ will have the form

$$A(x_1, \ldots, x_n) : \{x_1, \ldots, x_n\} \longrightarrow \varnothing,$$

so we indeed see a duality with (3).

We specify "typed" versions of sequential and parallel composition with respect to these arities and co-arities:

$$\frac{\varphi : X \longrightarrow Y \quad \psi : Y \longrightarrow Z}{\varphi \cdot \psi : X \longrightarrow Z} \qquad \frac{\varphi : X_1 \longrightarrow Y_1 \quad \psi : X_2 \longrightarrow Y_2}{\varphi \| \psi : X_1 \uplus X_2 \longrightarrow Y_1 \uplus Y_2}$$

The constant **1** has the form

$$\mathbf{1} : X \longrightarrow \varnothing$$

for any X.

We take these syntactic expressions modulo a notion of *structural congruence*, as in the π-calculus and action calculi. We impose the axioms

$$\varphi \cdot (\psi \cdot \theta) \equiv (\varphi \cdot \psi) \cdot \theta, \qquad \mathbf{1} \cdot \varphi \equiv \varphi \equiv \varphi \cdot \mathbf{1}$$

wherever these expressions make sense with respect to the typing with arities and co-arities.

Thus we are in fact describing a *category* $\mathbf{C}(\mathcal{L}_{\mathcal{A}})$. The objects are the arities—"co-arities" are simply arities appearing as the codomains of arrows in the category. The arrows are the syntactic expressions modulo structural congruence; and the composition in the category is sequential composition.

To complete the picture: for the choice connectives, we have

$$\frac{\varphi : X \longrightarrow \varnothing \quad \psi : X \longrightarrow \varnothing}{\varphi \oplus_\alpha \psi : X \longrightarrow \varnothing}$$

and for role interchange

$$\frac{\varphi : X \longrightarrow Y}{\hat{\pi}(\varphi) : X \longrightarrow Y}.$$

For the IF-quantifier we have

$$\forall x/y : X \uplus \{y\} \longrightarrow X \uplus \{x\} \uplus \{y\},$$

which makes explicit the fact that y occurs free in $\forall x/y$.

The arrows in $\mathbf{C}(\mathcal{L}_{\mathcal{A}})$ will be the well-formed formulas (both open and "co-open") of our logic. In particular, the sentences or closed formulas will be the arrows of the form $\varphi : \varnothing \longrightarrow \varnothing$.

4.2 Static Semantics Revisited

We consider the static semantics of a syntactic constituent $\varphi : X \to Y$. The $\mathcal{A}$-game $[\![\varphi]\!]$ defined as in Section 3.1 remains unchanged. In particular, atomic formulas are still assigned the empty $\mathcal{A}$-game. The new ingredient in the static semantics will reflect the intentions behind the arities and coarities, which we now set out in greater detail. The arity X is the set of variables being imported (as "free variables") from the outer environment by φ. Thus an "open formula" in the usual sense will be an arrow of type $X \to \varnothing$. The novel feature in our approach to logical syntax, following Milner's action calculi, are the co-arities. In $\varphi : X \to Y$, it is useful to write

$$Y = (X \cap Y) \uplus (Y \setminus X).$$

Now:

- $X \cap Y$ represents those variables imported from the outer environment which we simply "pass on through" to be imported in turn by the inner environment. (The variables in $X \setminus Y$ are hidden from the inner environment).
- $Y \setminus X$ represents the variables which are being *defined* by φ, and exported to the inner environment, where they will bind free occurrences of those variables.

As we have seen, variables bound by quantifiers are interpreted in our semantics by cells where localized decisions can be made. Hence the act of defining a variable amounts to binding it to a cell. Thus the single new component in the static semantics of $\varphi : X \to Y$ will be a function

$$\mathsf{bind}_M : Y \setminus X \longrightarrow C_M$$

where $M = [\![\phi]\!]$. We now show how bind is defined compositionally.

- **Atomic formulas, constant 1, choice connectives** $\oplus_\alpha$. These cases are all trivial, since the types are of the form $X \to \varnothing$, and $\varnothing \setminus X = \varnothing$.
- **Quantifiers** $Q_\alpha x : X \to X \uplus \{x\}$. As we saw in Section 3.1, $[\![Q_\alpha x]\!]$ has a single cell c_0. We set

$$\mathsf{bind}_{[\![Q_\alpha x]\!]}(x) = c_0.$$

- **Parallel Composition** $\varphi \| \psi : X_1 \uplus X_2 \longrightarrow Y_1 \uplus Y_2$, where $\varphi : X_1 \to Y_1$, $\psi : X_2 \to Y_2$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We define

$$\mathsf{bind}_{M\|N}(y) = \begin{cases} \mathsf{bind}_M(y), & y \in Y_1 \setminus (X_1 \cup X_2) \\ \mathsf{bind}_N(y), & y \in Y_2 \setminus (X_1 \cup X_2). \end{cases}$$

- **Sequential Composition** $\varphi \cdot \psi : X \to Z$, where $\varphi : X \to Y$ and $\psi : Y \to Z$. This is the key case. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. We can write

$$Z \setminus X \quad = \quad (Z \setminus (X \cup Y)) \uplus (Z \cap (Y \setminus X)).$$

  Hence we can define:

$$\mathsf{bind}_{M \cdot N}(z) = \begin{cases} \mathsf{bind}_M(z), & z \in Z \cap (Y \setminus X) \\ \mathsf{bind}_N(z), & z \in Z \setminus (X \cup Y). \end{cases}$$

- **Role Interchange** $\hat{\pi}(\varphi)$. Let $M = [\![\varphi]\!]$.

$$\mathsf{bind}_{\hat{\pi}(M)} = \mathsf{bind}_M.$$

4.3 Interlude: Structural Congruence

We have already introduced a notion of structural congruence $\equiv$ in defining the sysntactic category $\mathbf{C}(\mathcal{L}_\mathcal{A})$. We now consider the interpretation of the structural congruence using the static semantics, and the issue of axiomatization.

We say that our semantics validates a structural congruence

$$\varphi \equiv \psi : X \longrightarrow Y$$

if $[\![\varphi]\!] \cong [\![\psi]\!]$, that is if the $\mathcal{A}$-games they denote are isomorphic (in the usual sense of isomorphism for relational structures).

Proposition 4.1 *The following axioms for structural congruence are valid in the static semantics:*

$$\begin{aligned}
\varphi \cdot (\psi \cdot \theta) &\equiv (\varphi \cdot \psi) \cdot \theta \\
\mathbf{1} \cdot \varphi &\equiv \varphi \\
\varphi &\equiv \varphi \cdot \mathbf{1} \\
\varphi \| (\psi \| \theta) &\equiv (\varphi \| \psi) \| \theta \\
\mathbf{1} \| \varphi &\equiv \varphi \\
\varphi &\equiv \varphi \| \mathbf{1} \\
\varphi \| \psi &\equiv \psi \| \varphi \\
\varphi \oplus_\alpha \psi &\equiv \psi \oplus_\alpha \varphi \\
\hat{\pi}_1(\hat{\pi}_2(\varphi)) &\equiv \widehat{\pi_1 \circ \pi_2}(\varphi) \\
\hat{\pi}(\varphi \oplus_\alpha \psi) &\equiv \hat{\pi}(\varphi) \oplus_{\pi(\alpha)} \hat{\pi}(\psi) \\
\hat{\pi}(Q_\alpha x) &\equiv Q_{\pi(\alpha)} x \\
\hat{\pi}(\varphi \cdot \psi) &\equiv \hat{\pi}(\varphi) \cdot \hat{\pi}(\psi) \\
\hat{\pi}(\varphi \| \psi) &\equiv \hat{\pi}(\varphi) \| \hat{\pi}(\psi)
\end{aligned}$$

Remark 4.2 *The following are in general* not *valid in the static semantics (which we write colourfully if imprecisely as $\not\equiv$):*

$$\begin{aligned}
\varphi \oplus_\alpha \varphi &\not\equiv \varphi \\
\varphi \oplus_\alpha \mathbf{1} &\not\equiv \varphi \\
\varphi \oplus_\alpha (\psi \oplus_\alpha \theta) &\not\equiv (\varphi \oplus_\alpha \psi) \oplus_\alpha \theta \\
(\varphi \oplus_\alpha \psi) \cdot \theta &\not\equiv (\varphi \cdot \theta) \oplus_\alpha (\psi \cdot \theta) \\
\varphi \cdot (\psi \oplus_\alpha \theta) &\not\equiv (\varphi \cdot \psi) \oplus_\alpha (\varphi \cdot \theta) \\
\varphi \| (\psi \oplus_\alpha \theta) &\not\equiv (\varphi \| \psi) \oplus_\alpha (\varphi \| \theta) \\
(\varphi_1 \| \psi_1) \cdot (\varphi_2 \| \psi_2) &\not\equiv (\varphi_1 \cdot \varphi_2) \| (\psi_1 \cdot \psi_2)
\end{aligned}$$

Remark 4.3 *If we weaken the notion of validity of $\varphi \equiv \psi$ to $\mathcal{D}([\![\varphi]\!]) \cong \mathcal{D}([\![\varphi]\!])$ (order-isomorphism), then the only one of the above non-equivalences which becomes valid is*

$$(\varphi \oplus_\alpha \psi) \cdot \theta \equiv (\varphi \cdot \theta) \oplus_\alpha (\psi \cdot \theta).$$

Conjecture 4.4 *The axioms listed in Proposition 4.1 are complete for validity in the static semantics.*

4.4 Valuations

We are now in a position to give a compositional definition of valuation functions on outcomes. For each agent $\alpha \in \mathcal{A}$ we shall fix a set $\mathcal{V}_\alpha$ of values (utilities, payoffs, truth-values . . .). What structure should $\mathcal{V}_\alpha$ have? In order to express preferences between outcomes, and hence to capture the classical game-theoretic solution concepts such as Nash equilibrium, we would want $\mathcal{V}_\alpha$ to carry an order structure. For our present purposes, we need only that $\mathcal{V}_\alpha$ carries two binary operations

$$\odot, \otimes : \mathcal{V}_\alpha^2 \longrightarrow \mathcal{V}_\alpha$$

and an element $1 \in \mathcal{V}_\alpha$, such that $(\mathcal{V}_\alpha, \odot, 1)$ is a monoid, and $(\mathcal{V}_\alpha, \otimes, 1)$ is a commutative monoid.[3]

Now let $M : X \to Y$ be an $\mathcal{A}$-game, e.g. $M = [\![\varphi]\!]$, for $\varphi : X \to Y$ in $\mathbf{C}(\mathcal{L}_\mathcal{A})$. Then for each $\alpha \in \mathcal{A}$, an α-valuation function will have the form

$$\mathsf{val}_{M,\alpha} : \mathcal{I}^X \times \mathsf{Max}(M) \longrightarrow \mathcal{V}_\alpha. \tag{4}$$

Note firstly that we are only considering valuations applied to *maximal* states, which is reasonable since by Proposition 3.10, these are the only possible outcomes of evaluating strategy profiles,. However, in a more general setting where infinite plays are possible, e.g. in the games arising from fixpoint extensions of $\mathcal{L}_\mathcal{A}$, one should consider *continuous valuations* defined on the whole domain of states $\mathcal{D}(M)$.

The form of $\mathsf{val}_{M,\alpha}$ expresses the dependency of the valuation both on the values of the variables being imported from the environment, and on the final state of play. There are two extremal cases:

1. If $X = \varnothing$, then the valuation simply reduces to a function $\mathsf{Max}(M) \longrightarrow \mathcal{V}_\alpha$ on maximal states. In particular, this will be the case for closed formulas.
2. If $C_M = \varnothing$, the valuation reduces to a function $\mathcal{I}^X \longrightarrow \mathcal{V}_\alpha$. This is exactly the usual kind of function from assignments to variables to (truth)-values induced by an atomic formula evaluated in a first-order model $\mathcal{M}$.

By allowing a range of intermediate cases between these two extremes, we can give a compositional account which, starting with given assignments of the form (2) for atomic formulas, ends with valuations of the form (1) for sentences.

We now give the compositional definition of the valuation function. We use $\eta \in \mathcal{I}^X$ to range over assignments to variables.

- **Atomic formulas**. We take the valuation functions $\mathcal{I}^X \longrightarrow \mathcal{V}_\alpha$ as given. Thus atomic formulas are operationally void—no moves, no plays—but valuationally primitive—generating the entire valuation function of any complex formula. This seems exactly right.
- **Constant 1**. We set $\mathsf{val}_{[\![\mathbf{1}]\!],\alpha}(\eta, \varnothing) = 1$.
- **Choice connectives** $\varphi \oplus_\beta \psi : X \to \varnothing$. Let $M = [\![\varphi]\!]$, $N = [\![N]\!]$.

$$\begin{aligned} \mathsf{val}_{M \oplus_\beta N,\alpha}(\eta, \{(c_0, 1)\} \uplus s) &= \mathsf{val}_{M,\alpha}(\eta, s) \\ \mathsf{val}_{M \oplus_\beta N,\alpha}(\eta, \{(c_0, 2)\} \uplus s) &= \mathsf{val}_{N,\alpha}(\eta, s). \end{aligned}$$

[3]A plausible general suggestion is to identify these two algebraic structures, and to take $\mathcal{V}_\alpha$ to be a (commutative) *quantale*, *i.e.* a sup-lattice-enriched monoid. These structures have been used fairly extensively in Computer Science over the past 15 years [AV93, BCS05].

- **Quantifiers** $Q_\beta x : X \rightarrow X \uplus \{x\}$.

$$\mathsf{val}_{[\![Q_\beta x]\!],\alpha}(\eta, \{(c_0, a)\}) = 1.$$

- **Parallel Composition** $\varphi \| \psi : X_1 \uplus X_2 \rightarrow Y_1 \uplus Y_2$, where $\varphi : X_1 \rightarrow Y_1$, $\psi : X_2 \rightarrow Y_2$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$. Note that $\eta \in \mathcal{I}^{X_1 \uplus X_2}$ can be written as $\eta = (\eta_1, \eta_2)$, where $\eta_1 \in \mathcal{I}^{X_1}$, $\eta_2 \in \mathcal{I}^{X_2}$.

$$\mathsf{val}_{M\|N,\alpha}(\eta, s) = \mathsf{val}_{M,\alpha}(\eta_1, \pi_M(s)) \otimes \mathsf{val}_{N,\alpha}(\eta_2, \pi_N(s)).$$

- **Sequential Composition** $\varphi \cdot \psi : X \rightarrow Z$, where $\varphi : X \rightarrow Y$ and $\psi : Y \rightarrow Z$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$.

$$\mathsf{val}_{M\cdot N,\alpha}(\eta, s) = \mathsf{val}_{M,\alpha}(\eta, \pi_M(s)) \odot \mathsf{val}_{N,\alpha}(\eta', \pi_N(s)$$

where $\eta' \in \mathcal{I}^Y$ is defined as follows:

$$\eta'(y) = \begin{cases} \eta(y), & y \in X \\ s(\mathsf{bind}_M(y)), & y \in Y \setminus X. \end{cases}$$

This is the key case—the only one where the bind function is used.

- **Role Interchange** $\hat{\pi}(\varphi)$. Let $M = [\![\varphi]\!]$.

$$\mathsf{val}_{\hat{\pi}(M),\alpha} = \mathsf{val}_{M,\pi^{-1}(\alpha)}.$$

4.5 Dynamic Semantics Revisited

Given an $\mathcal{A}$-game $M : X \rightarrow Y$, an α-strategy is a family $(\sigma_\eta)_{\eta \in \mathcal{I}^X}$, where $\sigma_\eta \in \mathsf{Cl}_\alpha(M)$ for all η. The definition of evaluation of strategy profiles is simply carried over pointwise to these families, so that we get an outcome for each $\eta \in \mathcal{I}^X$. The global definition of the strategy sets $S_\alpha(M)$ can also be carried over pointwise in a straightforward fashion. However, the explicit dependence on the values assigned to free variables also creates some new possibilities, in particular for giving semantics to IF-quantifiers. This is best discussed in terms of the local definition of information constraints via visibility functions, to which we now turn.

4.6 Visibility Functions, Occlusion and IF-Quantifiers

We recall that the visibility function for an $\mathcal{A}$-game $M : X \rightarrow Y$ has the form

$$\gamma_M : C_M \longrightarrow [\mathcal{D}(M) \longrightarrow \mathcal{D}(M)]$$

and assigns a co-closure operator to each cell, specifying the information which is visible in any state to the agent wishing to fill the cell. We now augment this with an assignment

$$\mathsf{Occ}_M : C_M \longrightarrow \mathcal{P}(X).$$

The idea is that $\mathsf{Occ}_M(c) \subseteq X$ is the set of variables which are *occluded* at the cell c; hence the decision made at c cannot depend on the values of these variables. This leads to the following refinement of the information constraint (2) on a family of strategies $(\sigma_\eta)_\eta$. Note firstly that, given $c \in C_M$, with $X_1 = X \setminus \mathsf{Occ}_M(c)$ and $X_2 = \mathsf{Occ}_M(c)$, we can write $\eta \in \mathcal{I}^X$

as $\eta = (\eta_c, \eta_{\neg c})$, where $\eta_c \in \mathcal{I}^{X_1}$, $\eta_{\neg c} \in \mathcal{I}^{X_2}$. Now we can write the condition on $(\sigma_\eta)_\eta$ as follows:

$$\forall \eta, \eta' \in \mathcal{I}^X, c \in C_M, s \in \mathcal{D}(M).[\sigma_\eta(s)(c) = \sigma_{\eta_c, \eta'_{\neg c}}(\gamma_M(c)(s))(c)]. \tag{5}$$

(Here equality of partial functions is intended: either both states are undefined at c, or both are defined and have the same value.)

We now give the compositional definition of the occlusion function (non-trivial cases only).

1. **Choice connectives**.

$$\mathsf{Occ}_{M \oplus_\alpha N}(c) = \begin{cases} \varnothing, & c = c_0 \\ \mathsf{Occ}_M(c), & c \in C_M \\ \mathsf{Occ}_N(c), & c \in C_N \end{cases}$$

2. **Quantifiers** $Q_\alpha x : X \to X \uplus \{x\}$.

$$\mathsf{Occ}_{Q_\alpha x}(c_0) = \varnothing.$$

3. **Parallel Composition** $\varphi \| \psi : X_1 \uplus X_2 \to Y_1 \uplus Y_2$, where $\varphi : X_1 \to Y_1$, $\psi : X_2 \to Y_2$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$.

$$\mathsf{Occ}_{M \| N}(c) = \begin{cases} \mathsf{Occ}_M(c) \cup X_2, & c \in C_M \\ \mathsf{Occ}_N(c) \cup X_1, & c \in C_N. \end{cases}$$

4. **Sequential Composition** $\varphi \cdot \psi : X \to Z$, where $\varphi : X \to Y$ and $\psi : Y \to Z$. Let $M = [\![\varphi]\!]$, $N = [\![\psi]\!]$.

$$\mathsf{Occ}_{M \cdot N}(c) = \begin{cases} \mathsf{Occ}_M(c), & c \in C_M \\ (X \cap \mathsf{Occ}_N(c)) \cup (X \setminus Y), & c \in C_N. \end{cases}$$

5. **Role Interchange**. $\mathsf{Occ}_{\hat{\pi}(M)} = \mathsf{Occ}_M$.

The only case in the definition of the visibility function which needs to be revised to take account of the occlusion function is that for sequential composition:

$$\gamma_{M \cdot N}(c)(s) = \begin{cases} \gamma_M(c)(\pi_M(s)), & c \in C_M \\ (\pi_M(s) \setminus S) \cup \gamma_N(c)(\pi_N(s)) & c \in C_N \end{cases}$$

where

$$S = \{(c', v) \in D_M \mid \exists y \in (\mathsf{Occ}_N(c) \cap (Y \setminus X)).\, \mathsf{bind}_M(y) = c'\}.$$

4.6.1 IF-quantifiers

It is now a simple matter to extend the semantics to multi-agent versions of the IF-quantifiers. We consider a quantifier of the form $Q_\alpha x/Y : X \uplus Y \to X \uplus Y \uplus \{x\}$. Thus agent α is to make the choice for x, and must do so independently of what has been chosen for the variables in Y. The $\mathcal{A}$-game $M = [\![Q_\alpha x/Y]\!]$ is the same as for the standard quantifier $Q_\alpha x$, as are the bind and val functions. The difference is simply in the occlusion function:

$$\mathsf{Occ}_M(c_0) = Y.$$

This is then propagated by our compositional definitions into larger contexts in which the quantifier can be embedded, and feeds into the partial information constraint (5) to yield exactly the desired interpretation.

5 Further Directions

There are numerous further directions which it seems interesting to pursue. We mention a few.

- Some extensions are quite straightforward. In particular, an extension of $\mathcal{L}_\mathcal{A}$ with *fixpoints*

  $$\mu P(x_1, \ldots, x_n).\, \varphi(P)$$

  can be considered. The standard theory of solutions of domain equations over CDS [KP78] can be used to extend the static semantics to such fixpoint formulas. Moreover, our semantic constructions work for arbitrary CDS with infinite states. The only point which needs to be reconsidered in this setting is how the valuation functions are defined. The best idea seems to be to define valuations on all states, not only maximal ones. The value space should itself include partial values and form a domain, and the valuation function should be continuous. For example, we could take $\mathcal{V}_\alpha$ to be the *interval domain* $\mathbb{I}[0, 1]$ on the unit interval.

  An extension to full second-order logic, although technically more demanding, is also possible [AJ05].

- What is the full spectrum of possible connectives which can be used to explore the resources of our semantics? The logic $\mathcal{L}_\mathcal{A}$ we have introduced is quite natural, but this question remains wide open. Here is one precise version:

  Question 5.1 *Which set of connectives and quantifiers is* descriptively complete*, in the sense that every finite CDS is the denotation of a formula built from these quantifiers and connectives?*

  Another dimension concerns the information-flow structures and constraints expressible in the logic. The multiplicative connectives for sequential and parallel composition which we have studied are very basic. The parallel composition corresponds to the Linear Logic $\otimes$. The Linear Logic $⅋$ does allow for information flow between the parallel components; and there are surely a whole range of possibilities here.

  Problem 5.2 *Classify the possibilities for multiplicative connectives and information-flow constraints in the semantic space of concurrent games and strategies.*

  As one illustration, consider a connective $M \lhd N$ which combines features of sequential and parallel composition. The $\mathcal{A}$-game is defined as for $M \| N$, while the visibility function $\gamma_{M \lhd N}$ is defined as for $M \cdot N$. Play can proceed concurrently in both sub-games; there is information flow from M to N, but not *vice versa*.

- We have only considered *deterministic* strategies in this paper. Mixed, non-deterministic, probabilistic, and perhaps even quantum strategies should also be considered.

- The whole question of proof theory for the logic $\mathcal{L}_\mathcal{A}$ has been left open. In a sense, we have given a *semantics of proofs* without having given a syntax! How multi-agent proof theory should look is conceptually both challenging and intriguing.

- Viewed from a model-theoretic perspective, IF-logic seems dauntingly complex. Our more intensional and operational view may offer some useful alternative possibilities. Just as the shift from validity to model-checking often replaces an intractable problem by an efficiently solvable one, so the shift from model-theoretic validity or definability of IF-formulas to constructing, reasoning about and running strategies for concurrent games described by proofs of formulas seems a promising approach to making this rich paradigm computationally accessible.
- The logic and semantics we have developed appears to flow from very natural intuitions. These should be supported by a range of convincing examples and applications.

References

[AdM7] Seventh Augustus De Morgan Workshop: "Interactive Logic: Games and Social Software", King's College, London, November 4–7 2005. URL: `http://www.illc.uva.nl/ADMW05/`.

[AV93] Samson Abramsky, Steven Vickers: Quantales, Observational Logic and Process Semantics. Mathematical Structures in Computer Science 3(2): 161-227 (1993)

[AJ94] S. Abramsky, R. Jagadeesan, New foundations for the Geometry of Interaction, *Information and Computation,* 111(1):53-119, 1994. Conference version appeared in LiCS '92.

[AM99] S. Abramsky and P.-A. Melliès. Concurrent games and full completeness. *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*, 431–44, 1999.

[Abr00] S. Abramsky, Process Realizability, in Foundations of Secure Computation: Proceedings of the 1999 Marktoberdorf Summer School, F. L. Bauer and R. Steinbruggen, eds. (IOS Press) 2000, 167–180.

[Abr00a] S. Abramsky, Concurrent Interaction Games, in Millennial Perspectives in Computer Science, J. Davies, A. W. Roscoe and J. Woodcock, eds. (Palgrave) 2000, 1–12.

[AJM00] Samson Abramsky, Radha Jagadeesan, Pasquale Malacaria: Full Abstraction for PCF. Inf. Comput. 163(2): 409-470 (2000)

[Abr03] Samson Abramsky: Sequentiality vs. Concurrency In Games And Logic. Mathematical Structures in Computer Science 13(4): 531-565 (2003)

[AJ05] S. Abramsky and R. Jagadeesan, A Game Semantics for Generic Polymorphism, in Annals of Pure and Applied Logic, vol 133, 3–37, 2005.

[AP81] Krzysztof R. Apt, Gordon D. Plotkin: A Cook's Tour of Countable Nondeterminism. ICALP 1981: 479-494

[Apt97] Krzysztof R. Apt: From Chaotic Iteration to Constraint Propagation. ICALP 1997: 36-55

[BCS05] Alexandru Baltag, Bob Coecke, Mehrnoosh Sadrzadeh: Algebra and Sequent Calculus for Epistemic Actions. Electr. Notes Theor. Comput. Sci. 126: 27-52 (2005) [vBen03] Johan van Benthem: Logic Games are Complete for Game Logics. Studia Logica 75(2): 183-203 (2003)

[Gir87] J.-Y. Girard. Linear Logic. *Theoretical Computer Science,* 50:1–102, 1987.

[Hen61] L. Henkin, Some remarks on infinitely long formulas, *Infinitistic Methods*, Pergamon, 1961.

[HS89] Jaakko Hintikka, Gabriel Sandu: Informational independence as a semantical phenomenon, Logic, Methodology and Philosophy of Science VIII, J. E. Fenstad et al., eds, Elsevier, Amsterdam, 1989:571–589.

[HS95] Jaakko Hintikka, Gabriel Sandu: What is the Logic of Parallel Processing? Int. J. Found. Comput. Sci. 6(1): 27-49 (1995)

[HS96] Jaakko Hintikka, Gabriel Sandu: Game-theoretical Semantics, in van Benthem and ter Meulen, Handbook of Logic and Language, Elsevier 1996.

[Hod97] Wilfrid Hodges: Compositional Semantics for a Language of Imperfect Information. Logic Journal of the IGPL 5(4): (1997)

[HO00] J. M. E. Hyland, C.-H. Luke Ong: On Full Abstraction for PCF: I, II, and III. Inf. Comput. 163(2): 285-408 (2000)

[JPP89] Radha Jagadeesan, Prakash Panangaden, Keshav Pingali: A Fully Abstract Semantics for a Functional Language with Logic Variables LICS 1989: 294-303

[KP78] G. Kahn and G. Plotkin. Concrete Domains. *Theoretical Computer Science*, 121:187–277, 1993. Appeared as TR IRIA-Laboria 336 in 1978.

[MPW92] Robin Milner, Joachim Parrow, David Walker: A Calculus of Mobile Processes, I Inf. Comput. 100(1): 1-40 (1992)

[MPW92a] Robin Milner, Joachim Parrow, David Walker: A Calculus of Mobile Processes, II Inf. Comput. 100(1): 41-77

[Mil93] Robin Milner: Action Calculi, or Syntactic Action Structures. MFCS 1993: 105-121

[Paul02] Marc Pauly: A Modal Logic for Coalitional Power in Games. J. Log. Comput. 12(1): 149-166 (2002)

[PP03] Marc Pauly, Rohit Parikh: Game Logic - An Overview. Studia Logica 75(2): 165-182 (2003)

[Res01] Pedro Resende: Quantales, finite observations and strong bisimulation. Theor. Comput. Sci. 254(1-2): 95-149 (2001)

[SRP] Vijay A. Saraswat, Martin C. Rinard, Prakash Panangaden: Semantic Foundations of Concurrent Constraint Programming. POPL 1991: 333-352