Game Semantics

Martin Hyland

Contents

1 Introduction

The aim of these notes is to explain how games can provide an intensional semantics for functional programming languages, and for a theory of proofs. From the point of view of program semantics, the rough idea is that we can move from modelling computable functions (which give the 'extensional' behaviour of programs) to modelling 'intensional' aspects of the algorithms themselves. In proof theory, the tradition has been to consider syntactic representations of (what are presumably intended to be 'intensional') proofs; so the idea is to give a more intrinsic account of a notion of proof.

Three main sections follow this Introduction. Section 2 deals with games and partial strategies; it includes a discussion of the application of these ideas to the modelling of algorithms. Section 3 is about games and total strategies; it runs parallel to the treatment in Section 2, and is quite compressed. Section 4 gives no more than an outline of more sophisticated notions of game, and discusses them as models for proofs. Exercises are scattered through the text.

I very much hope that the broad outline of these notes will be comprehensible on the basis of little beyond an understanding of sequences (lists) and trees. However the statements of some results and some of the exercises presuppose a little knowledge of category theory, of domain theory and of linear logic. The main categorical ideas used in the notes are explained in Appendix A. I have tried to give references for other background information. I ask those unfamiliar with category theory not to be put off by the fact that occasionally category theoretic language is used to give succinct expression to a collection of (hopefully plausible) phenomena.

The ideas of games and strategies are very intuitive, indeed that is a strong point in favour of their use as a basis for semantics. The disadvantage of course is that compelling intuitions can be misleading. There may well be mistakes in this account, and a critical attitude is recommended.

1.1 Precursors

The well-established denotational semantics for (functional) programming languages, which makes use of domain theory, is a theory of functions in extension: the interpretation of programs is via certain 'extensional' functions which they may be regarded as computing. This point of view is already apparent in classical recursion theory: the notion of partial recursive functions is independent of any specific machine, but the notion of effective algorithm is apparently machine-dependent. It would be reasonable to conclude that the notion of algorithm is inevitably machine-dependent (or language-dependent or syntax-dependent). Hence the very idea of modelling algorithms naturally in some sufficiently abstract way is a brave one. The pioneers in this endeavour were Kahn and Plotkin (1978) (in English, Kahn and Plotkin (1993)) and Berry and Curien (1982). A succinct account of the main ideas[1] in the tradition of concrete data structures is given in Curien (1993).

A game theoretic approach to proof (or at least to provability) was suggested by Lorenzen (Lorenzen and Lorenz 1978). His ideas have been developed and made precise by a number of people, and form the basis for a distinctive tradition in philosophical logic. For a good survey of work in this area, see Felscher (1986).

Structured (or compositional) approaches to games and strategies trace their origins back to work of Blass (1972) and Conway (1976), though neither were motivated by semantical questions. Joyal gave a compositional account of Conway's work, defining a compact closed category of games. (For an introduction to compact closed categories see Kelly and Laplaza (1980).) Joyal's observation inspired me to think of games in connection with program semantics; but we still have no good understanding of (or applications of) the category of Conway games, and it will not play a role in these notes. Blass himself drew attention to the semantic possibilities of his ideas (Blass 1992). For a compositional approach see Abramsky and Jagadeesan (1994).

1.2 Categories of Games: Ideas

1.2.1 The protagonists

The games we consider involve two players, P (Player, ∃loise, Left, Us) and O (Opponent, ∀belard, Right, Them) who play moves alternately. I adopt the uncontentious nomenclature: Player vs Opponent. As the contrast between the

[1] Similar ideas were developed by Kleene in a series of papers; Gandy modified Kleene's ideas to the continuous case, and he together with his student Pani have recently obtained a number of interesting partial results which have yet to be published.

respective pairs of names suggests, it is a crucial feature of (the use of) our intuition about games that our attitude to the two participants should be quite different: we favour Player (Us, Left) over Opponent (Them, Right). Some aspects of this preference may be indicated by the following series of dichotomies[2].

Player	Opponent
Strategy	Counterstrategy
Actor	Environment
Programmer	Computer
Operating system	Users
Program	Data
Proof	Refutation
Event	Cell
Output	Input

It is worth stressing that the conceptual tools described here are not intended to deal with interaction between many agents, as considered for example in concurrency theory. There is no obvious generalisation of the theory of two-person games considered here to many-person games.

1.2.2 Perspective of categorical logic

From the point of view of categorical logic, the important aspect is compositionality. Hence our preference for Player and for things on the left. We wish to compose programs and proofs; or in alternative jargon we desire modular tools of program or proof construction. It will turn out that it is Player's strategies for games which we shall be able to compose in an appropriate way, and hence we focus on Player's role. What we shall do is in the mainstream tradition of the categorical interpretation of types theories and of proof systems, and I indicate the connections in the following table:

Object	Map	Categorical Composition
Type	Term	Application in context
Proposition	Proof	Composition via Cut rule
Type	Algorithm	Composition plus hiding
Game	Strategy	Scratchpad Composition

This table incorporates within it both the Curry-Howard isomorphism and basic ideas of categorical logic (here proof theory). Good general background in categorical logic can be found in Lambek and Scott (1986). For a view of categorical type theory see Crole (1993).

[2] Do not get carried away by duality! There are similar seeming dichotomies which it is as well not to put in this list, as they can be used to refer to aspects of games which are independent of the dichotomy Player vs Opponent. Examples are: Question vs Answer, Active vs Passive, Positive vs Negative. Questions and Answers play a substantial role in Section 4.

1.3 Fundamental structures

1.3.1 Simple games

In this introductory survey, we write G, H for the kinds of two-person games in which we are interested. It is a consequence of our interest in composition that in standard play, O starts the play. In the simplest form, the game will be completely determined by the succession of moves with O playing first. I call such games *simple*; they are called *negative* in Abramsky and Jagadeesan (1994).

There is something like a natural duality for games: one interchanges the roles of the two players. Of course the dual of a simple game is not a simple game since P has to start: indeed it is best not to regard it as a game, but rather as a 'co-game'. We write $G^\perp$ for the dual of the game G.[3]

1.3.2 Playing games in parallel

In the cases we are interested in, there are the following two operations on games, each involving interleaved plays from the two component games:

Tensor product $G \otimes H$: in this game, G and H are played in parallel. For simple G and H, it will automatically happen that P can only move in the game in which O has just played; however O is allowed to switch games. There is usually a unit I for the tensor product, namely the 'empty game' in which no player can move.

Linear maps $G \multimap H$: in this game the dual $G^\perp$ of G is played in parallel with H. For simple games G and H, it will automatically happen that O must start in H; P can play in either $G^\perp$ or H, and thereafter P is allowed to switch games; on the other hand O can only now move in the game in which P has just moved.

Note that the switching behaviour in $G \multimap H$ is dual to that in $G \otimes H$. This reflects the duality between *tensor* and *par* in linear logic (Girard 1987).[4]

1.3.3 Strategies

The crucial notion is that of strategy; it will be interpreted computationally as an algorithm, and logically as an argument. Intuitively a strategy is some means for

[3] There are more complex possibilities. In the games which concern Abramsky and Jagadeesan (1994), the possibility that P could start a game G plays a role: while it cannot be realised in the standard play of G, it may be realised in the standard play of games constructed from G. The same phenomenon arises in the context of Conway games. Such situations which will not be treated in these notes; but with them one has a genuine duality.

[4] In the case of more complex games, O may switch in $G \otimes H$, and P may switch in $G \multimap H$. The 'Blass Convention', in the spirit of linear logic, is that the other player cannot switch. The 'Conway Convention' allows the other player to switch as well.

determining the next move in a game, but a natural question arises.

What is the next move determined by?

Here are some possible answers.

- The easy answer is 'everything': this gives rise to what are called history-sensitive strategies.
- A surprising answer is 'the last move'; then one has what might best be described as a game of stimulus and response (a rally in tennis perhaps). This has been studied by Abramsky and his coworkers (Abramsky and Jagadeesan 1994; Abramsky, Jagadeesan, and Malacaria 1994).
- A natural answer is 'the current position', perhaps identified with the succeeding game. This is the basis for much work in traditional logic, but presents problems with regard to compositionality.
- A further (not quite obvious) answer couched in terms of 'views' will be discussed in Section 4.

Note again that the thrust of these notes is that we want a good notion of composition. We want to be able to compose strategies in a disciplined way, so as to be able to argue effectively about the behaviour of composed strategies in a structured fashion. (We are rather far from this ideal!) We give mathematical expression to the idea of a good composition by forming categories of games; then we can exploit their rich structure.

1.3.4 Categorical structure

Let us review the structure which we extract from consideration of the notion of a simple game. (The relevant categorical definitions are given in Appendix A.) We can identify the following significant ideas.

A notion of game Two-person games A, B, C, ... played between Opponent O (who plays first) and Player P. The games will be the objects of a category $\mathcal{G}$ of games.

A notion of strategy in a game A P-strategy $\sigma : A$ for the game A (for O going first) will become a map $\sigma : I \longrightarrow A$ (or element of A) in the category $\mathcal{G}$.

Tensor products of games The tensor product $A \otimes B$ of two games A and B is obtained by playing them in parallel. It gives rise to a symmetric monoidal structure on the category of games. The unit I for the tensor product is the empty game.

Linear function spaces of games The linear function space $A \multimap B$ of 'maps' from A to B is obtained by playing B in parallel with the dual $A^\perp$ of A. It gives rise to the closed structure on the category of games.

Copy-cat strategy For each game A, there is a P-strategy in $A \multimap A$ which simply copies moves by Opponent in A (respectively $A^\perp$) as the corresponding moves for Player in $A^\perp$ (respectively A). This acts as the identity in the category.

Composition of strategies [5] We can compose P-strategies $\sigma : (A \multimap B)$ and $\tau : (B \multimap C)$ to obtain a strategy $\sigma;\tau : (A \multimap C)$. This gives the composition in the category.

From the above we derive the following general result.

Theorem *There is a symmetric monoidal closed category $\mathcal{G}$ of games and strategies.*

The import of this is that the category $\mathcal{G}$ is a model for the multiplicative fragment of intuitionistic linear logic. The models which we shall consider all have the property that the unit I for the tensor product is the terminal object of the category: $I \cong 1$. In the language of linear logic, the multiplicative and additive units coincide, so in fact we have models of affine linear logic.

We have further structure in the cases we shall consider.

Products on the category The terminal game 1 is the empty game. (We already noted the problem $I = 1$.) In the product $A \times B$ of games A and B, Opponent gets a choice as to which game to play. (In fact we shall have arbitrary products in $\mathcal{G}$.)

Monoidal comonad A symmetric monoidal functor $! : \mathcal{G} \longrightarrow \mathcal{G}$, and monoidal natural transformations $\varepsilon : ! \longrightarrow 1_{\mathcal{G}}$, $\delta : ! \longrightarrow !!$, forming a comonad.

Comonoid structure Monoidal natural transformations $e : ! \longrightarrow I$ and $d : ! \longrightarrow !\otimes!$ giving (free) !-coalgebras the structure of a symmetric comonoid. This comonoid structure is compatible with the comonad in the sense that it is preserved by coalgebra maps: thus, whenever $f : (!A, \delta_A) \to (!B, \delta_B)$ is a coalgebra map, then f is also a comonoid map.

We make some remarks about this additional structure. First, it is a quite general phenomenon that a symmetric monoidal closed category (SMCC) together with the structure of a monoidal comonad equipped with discard and duplication as above gives rise to a cartesian closed category (CCC) of coalgebras with objects the products of free coalgebras.

Theorem *Suppose that a* SMCC *$\mathcal{C}$ is equipped with a monoidal comonad, itself equipped with a commutative comonoid structure as above. Then the category of Eilenberg-Moore coalgebras has products; and the full subcategory on objects isomorphic to finite products of free coalgebras is cartesian closed.*

[5] As stressed by Abramsky, composition of strategies has a natural reading in process terms as 'parallel composition plus hiding'.

For a fuller discussion of these points see Benton, Bierman, de Paiva, and Hyland (1992), Bierman (1993) and Bierman (1995). The general result does have applications, but the categories of games which we consider have products. It follows that they satisfy Seely's axioms for a model of linear logic extended as in Bierman (1995), that is, they are new-Seely categories. This means that the Kleisli category of the comonad is equivalent to the one introduced in the last theorem. Thus we can give very concrete descriptions of the CCCs derived from our SMCCs of games.

Theorem *The Kleisli category $\mathcal{G}_!$ for the comonad ! on $\mathcal{G}$ forms a cartesian closed category of games and strategies.*

The simply typed λ-calculus lies at the heart of (typed) functional programming, and is more or less the static proof theory (in the sense of Girard) of the $(\Rightarrow)$-fragment of intuitionistic logic (minimal logic). The two interpretations of the λ-calculus give the Curry-Howard correspondence. It is natural and changes little to include products in a typed language, to include conjunction in the logic giving the $(\wedge, \Rightarrow)$-fragment, and so to add pairing to the λ-calculus. (See for example Girard (1989b).) The simply typed λ-calculus under $\beta\eta$-equality corresponds exactly to CCCs. (See Lambek and Scott (1986).) It follows that any CCC models proof theory. The primary aim in these notes is to consider the CCCs arising from games either as models for a functional programming language or as models for a theory of proofs.

1.3.5 Intensionality

We may contrast the notion of an algorithm which has a definite *intensional* component with that of the function which it computes regarded as determined by its *extensional* behaviour (its graph). It seems that this distinction can be reflected to some extent in a standard categorical notion.

Definition *A category $\mathcal{C}$ with terminal object* 1, *is* extensional *if and only if $\mathcal{C}$* has enough points, *that is, if and only if whenever $f, g : A \to B$ in $\mathcal{C}$,*

$$a; f = a; g \text{ for all } a : 1 \longrightarrow A, \text{ implies } f = g.$$

(One also says that 1 is a generator *in $\mathcal{C}$.)*

Suppose that a category $\mathcal{C}$, typically a cartesian closed category (CCC), is being used to model computations or proofs. We shall take the fact that $\mathcal{C}$ is not extensional in this technical sense, as indicating that it has intensional aspects.[6] Caution is indicated here. The converse looks worse than suspect as one can always add points, as in the Capitalisation Lemma of Freyd and Scedrov (1990). (This reference contains an interesting discussion of points in categories; and a forceful treatment of many other aspects of category theory.)

[6] This question of extensionality of models has nothing to do with the question of the η-rule in the λ-calculus. This automatically holds in CCCs.

1.4 Notation and prerequisites

In our formulation, a game A is determined by a tree of moves, so we start with some notation for finite sequences (lists) infinite sequences (streams) and trees.

Sequences For finite (and occasionally for infinite) sequences p and q we write $q \preceq p$ for the *prefix* relation (equivalently p *extends* q or q *is an initial subsequence of* p); $q \prec p$ is the corresponding strict order relation (q is a *proper prefix of* p). We let juxtaposition or an infix dot denote *concatenation* of sequences or elements (moves): pa or $p.a$ (respectively pq or $p.q$) is the sequence p followed by element a (respectively sequence q).

Trees A *tree* T on a set X is a prefix closed collection of finite sequences of elements of X; a *subtree* S of T is a subcollection of finite sequences of T which itself forms a tree. A tree on X is a subtree of the full tree X^* of all finite sequences on X.[7]

Projections Suppose first that $X = Y + Z$ is given as a coproduct (or disjoint union). If $p \in X^*$, then we let *the projection of p on Y*, $p_Y \in Y^*$ be the sequence whose elements in order are those elements in order of the sequence p which lie in (the image of) Y. We extend projections to trees in the obvious way. If T is a tree, then *the projection of T on Y* is

$$T_Y = \{p_Y \mid p \in T\}\,;$$

clearly T_Y is a tree on Y.
Secondly suppose that $X = Y \times \mathbb{N}$ is given as an $\mathbb{N}$-indexed copower (sum) of Y. If $p \in X^*$, then we let *the kth projection of p*, $p_k \in Y^*$ be the sequence whose elements in order are those elements in order of the sequence p which lie in the kth copy of Y. We extend projections to trees in the obvious way. If T is a tree, then *the kth projection of T* is

$$T_k = \{p_k \mid p \in T\}\,;$$

clearly T_k is a tree on Y.

Infinite sequences For a set X, write $X^{\leq\omega} = X^* \cup X^\omega$ for the set of finite and infinite sequences of elements of X.
Suppose that T is a tree on a set X. Write

$$T^\infty = \{p \in X^\omega \mid q \in T \text{ for all } q \prec p\}$$

for the collection of infinite sequences generated by T, and

$$\overline{T} = \{p \in X^{\leq\omega} \mid q \in X^* \text{ implies } q \in T \text{ for all } q \preceq p\}$$

[7] Some readers will be acquainted with concurrency theory, and the use made in that subject of various notions of labelled tree; it may help to stress that our trees correspond to deterministic labelled trees.

for the complete set of all finite and infinite sequences determined by T. (IN game-theoretic terms, this will be the set of all *plays* in T.) Finally define the set of *maximal finite or infinite* sequences from T as follows

$$\widehat{T} = \{p \in \overline{T} \mid \text{whenever } q \in \overline{T} \text{ then } p \preceq q \text{ implies } p = q\}\,.$$

(This will be the set of all *completed plays*.)

1.5 Acknowledgements

Much of the material in these notes was discussed in a series of lectures in Cambridge in Lent Term 1994. I owe a lot to my eclectic audience of computer scientists, logicians, mathematicians and philosophers. I owe a particular debt to Gavin Bierman and Luke Ong, with whom I have discussed models for linear logic in general and games in particular on numerous occasions. (They also allowed me to steal some of their TEXto help create these notes!) Finally I acknowledge use of Paul Taylor's diagram macros.

2 Games and computation

2.1 A monoidal closed category of games

2.1.1 Games and strategies

This section gives simple notions of game and of strategy from which to construct a category. The games are determined by trees of moves.

Definition 2.1 *A* game for fun *or* fun-game *A is given by a set $M = M_A$ of* moves *together with a non-empty tree T_A on M called the* game tree.
The elements p, q, .. of T_A are called positions *or* plays.
In a play $a_0.a_1.....a_n$, the moves a_0, a_2, ... , of even parity *are moves* played by the Opponent *O; and the moves a_1, a_3, ...* , of odd parity *are moves* played by the Player *P.*
If a play $a_0.a_1.....a_n$ is of odd length then O has just moved and it is P to move; we let O_A be the set of such odd positions *in T_A. On the other hand if a play is of even length then P has just moved and it is O to move; we write P_A for the set of such* even positions *in T_A. (Note that in an odd position, an even move has just been played, and vice-versa.)*
T_A is the disjoint union of the odd positions O_A and even positions P_A.

Throughout Section 2 we shall use *games* to refer to fun-games; but later we shall need to distinguish them from other kinds of games.

Player moves second and hence we can compose Player's strategies (P-strategies) to give a category structure on games. We give first a technically smooth definition.

Definition 2.2 *A* P-strategy *in a game A is given by a non-empty subtree S of T_A satisfying the following.*

If $p \in S \cap O_A$ then there is a unique move a with $p.a \in S$.

Remark It is probably more intuitive to present a strategy by means of a partial map[8] $\phi : O_A \longrightarrow M_A$, giving moves when Player is to move. We then stipulate that the domain $\mathrm{dom}(\phi) \subseteq O_A$ is a prefix closed collection of odd positions, and that ϕ satisfies

(i) if $p \in \mathrm{dom}(\phi)$, then $p.\phi(p) \in P_A$, and
(ii) if $p, q \in \mathrm{dom}(\phi)$ and $q \prec p$, then $q.\phi(q) \prec p$.

The two notions of strategy determine one another via bijections $\phi \rightarrow S_\phi$ and $S \rightarrow \phi_S$ given as follows.

$$S_\phi = \{<>\} \cup \{q \mid q \preceq p.\phi(p) \text{ for some } p \in \mathrm{dom}(\phi)\ \};$$

$$\phi_S(p) = a \text{ if and only if } p.a \in S.$$

We refer to the equivalent notions of strategy[9] as being either *in subtree mode* or *in function mode*. Greek letters σ, τ , ... will denote strategies without regard to mode of representation.

Exercises 1

1. *Show that a strategy S given in subtree mode is determined by $S \cap P_A$.*
2. *Show that for any partial map $\psi : O_A \longrightarrow M_A$ there is a maximal partial map $\phi : O_A \longrightarrow M_A$ contained in ψ which is a strategy in function mode.*
3. *Show that a strategy is also determined by a subtree S of T_A such that (i) for $p \in S \cap O_A$, $p.a \in S$ and $p.b \in S$ imply $a = b$, and (ii) for any $p \in S \cap P_A$, $p.a \in O_A$ implies $p.a \in S$.*
4. *Formulate notions of non-deterministic partial strategy, and of deterministic and non-deterministic total strategy as functions and as subtrees along the lines of Definition 2.2.*

2.1.2 Tensor product and linear function space

The idea behind the tensor product and linear function space of games was explained in Section 1.2, so we just give the formal definitions.

[8] Note that since strategies are given by partial functions, we are here dealing with partial but deterministic strategies. Other choices are clearly possible.

[9] Glynn Winskel drew to my attention that one can also think about strategies in terms of Petri nets. Regard the positions in O_A as conditions (or places), and those in P_A as events, where the initial position is a unique starting event. Then a strategy corresponds exactly to a possible state of the net.

Definition 2.3

The unit game I is defined by setting $M_I = \varnothing$. This determines the game tree which consists only of the empty sequence. (Thus I is a game in which no moves can be made.)

Given games A and B, the games $A \otimes B$ and $A \multimap B$ are defined as follows.
Moves *The moves are $M_{A\otimes B} = M_{A \multimap B} = M_A + M_B$, the coproduct (disjoint sum) of M_A and M_B.*
Game tree *The game tree $T_{A\otimes B}$ is the subtree of $(M_{A\otimes B})^*$ consisting of those sequences p whose projections $p_A = p_{M_A}$, and $p_B = p_{M_B}$ preserve parity of moves and are in T_A, T_B, respectively.*
The game tree $T_{A\multimap B}$ is the subtree of $(M_{A\multimap B})^*$ consisting of those sequences p such that the projection $p_A = p_{M_A}$ reverses, while $p_B = p_{M_B}$ preserves parity of moves and the projections are in T_A, T_B, respectively.

It is helpful to think of $A \multimap B$ as the result of playing the cogame $A^\perp$, which is A with the parity of moves reversed, in parallel with B. Suppose that we write $p_{A^\perp}$ for the sequence p_A with the parities of moves notionally reversed. Then in the definition of $A \multimap B$, we would say that $p_{A^\perp}$ and p_B preserve parity of moves and are in $T_{A^\perp}$, T_B respectively. This reformulation is helpful conceptually and technically (particularly in the case that $A = B$).

Exercises 2

1. *Show that it is indeed a design feature of these definitions that*
   - *in $A \otimes B$, O may switch between the games A and B, but P may not;*
   - *in $A \multimap B$, P may switch between the cogame $A^\perp$ and the game B, but O may not.*
2. *In a game of the form $(A \multimap B) \multimap C$, which of O and P can switch between which of the (co-)games (and when)? Do the same for a game of the form $A \multimap (B \otimes C)$. Explore a few more complicated examples!*
3. *Using the obvious intuitive notion of isomorphism of games, establish the following isomorphisms.*[10]
   - $A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$.
   - $I \otimes A \cong A \cong A \otimes I$.
   - $A \otimes B \cong B \otimes A$.
   - $A \multimap (B \multimap C) \cong (A \otimes B) \multimap C$.
4. *Write S for the game which ends after a unique initial move.*
   (i) Give concrete descriptions of the games $S \otimes S$ and $S \multimap S$.

[10] Why do we not get identities in place of isomorphisms? In what sense is the coproduct + of sets associative and commutative?

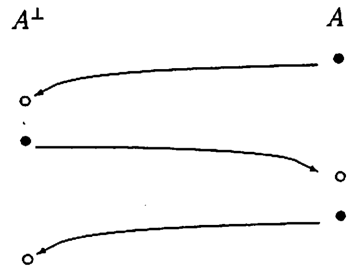

Figure 1: Identity strategy.

(ii) Suppose that a game has just one initial move. Show that it is isomorphic to a game of the form $B \multimap S$ for some game B.

5. *(i) For which games A do we have $A \cong A \otimes A$?*
   (ii) For which games A do we have $A \cong A \multimap A$?

2.1.3 The category $\mathcal{LFG}$ of linear games of fun

As explained earlier, the form of compositionality which we expect will give rise to the structure of a category with games as objects and strategies as maps. We now describe the relevant structure on the strategies.

Definition 2.4 *For any game A we define a strategy ι_A in $A \multimap A$ in subtree mode as follows.*

$$p \in \iota_A \cap P_A \text{ if and only if for all even } q \preceq p, \quad q_{A^{\perp}} = q_A.$$

Suppose that we have strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$, in subtree mode. Their composite $\sigma;\tau$ is defined as follows.

$$p \in \sigma;\tau \cap P_{A \multimap B} \text{ if and only if}$$

$$\exists q \in \sigma \cap P_{A \multimap B}. \exists r \in \tau \cap P_{B \multimap C}. q_{A^{\perp}} = p_{A^{\perp}} \text{ and } q_B = r_{B^{\perp}} \text{ and } r_C = p_C.$$

The definitions of the identity maps and of composition are based on very simple ideas. The identity strategy simply copies moves from the copy of A to that of $A^{\perp}$ and vice-versa as in Figure 1. Composition can be understood by imagining that when playing $\sigma;\tau$ in $A \multimap C$, P keeps a scratchpad on which to record (corresponding pairs of) moves in B and $B^{\perp}$. A representative play is shown in Figure 2. The opponent starts the game $A \multimap C$ in C, and P plays in $B \multimap C$ a move according to τ which happens to be in $B^{\perp}$. (The situation where the move is in C is clearly straightforward.) P copies this over to B and regards that move as the start of a play in $A \multimap B$. Playing according to σ gives a move in B, which is copied over to $B^{\perp}$, where it is regarded as a move of O. P responds to that according to

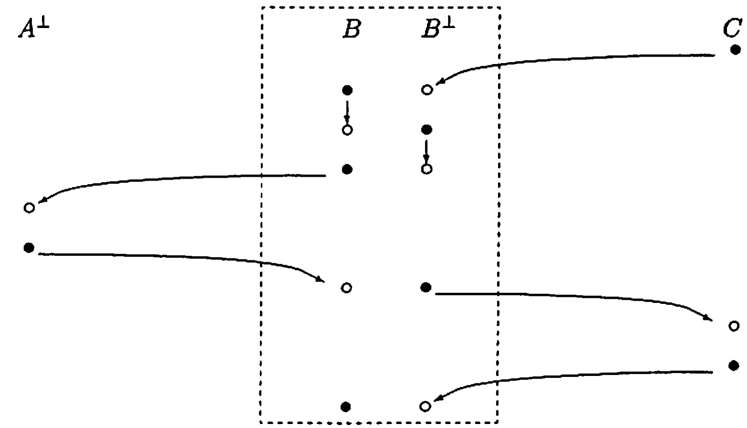

Figure 2: Composition of strategies.

τ getting in the case illustrated a move in $B^{\perp}$. This in its turn is copied over to B where it is regarded as a response of O. P responds to that according to σ, getting a move in $A^{\perp}$. At last P is able to play a move 'for real' in the game $A \multimap C$. In the illustration O responds (necessarily) in $A^{\perp}$, and the game continues. (Further 'real moves' in C are shown.)

Let us now record the properties which say that we have a category.

Proposition 2.1 *Suppose that A and B are games with $\sigma : A \multimap B$; then $\iota_A;\sigma = \sigma = \sigma;\iota_B$.*
Suppose that A, B, C and D are games with $\rho : A \multimap B$, $\sigma : B \multimap C$ and $\tau : C \multimap D$; then $\rho;(\sigma;\tau) = (\rho;\sigma);\tau$. □

In view of this proposition, we can define a category of games.

Definition 2.5 *The category $\mathcal{LFG}$ of* linear fun-games *has as objects, games as defined in definition 2.1; as maps from A to B, strategies in $A \multimap B$ as defined in definition 2.2; and identities and composition as in definition 2.4.*

We let $\mathcal{LFG}(A)$ denote the set of strategies in the game A and $\mathcal{LFG}(A,B)$ the set of maps from A to B in $\mathcal{LFG}$. So $\mathcal{LFG}(A,B) = \mathcal{LFG}(A \multimap B)$.

Exercises 3

1. *Show that the strategy ι_A can be defined in function mode by stipulating that $\iota_A(p.a) = c(a)$ for all $p.a \in O_{A \multimap A}$. ($c : M_{A \multimap A} \longrightarrow M_{A \multimap A}$ is the twist isomorphism on $M_{A \multimap A} = M_{A^{\perp}} + M_A = M_A + M_A$ which interchanges*

$M_{A^\perp}$ and M_A.)[11]

2. Show that the composite $\sigma;\tau$ can be defined in function mode as follows.

   $\sigma;\tau(p) = d$ if and only if there are (necessarily unique) positions q consistent with σ and r consistent with τ with $q_B = r_B$ and with either $\sigma(q) = d$ or $\tau(r) = d$.

3. Suppose that games A and B are isomorphic in the categorical sense that there are maps $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow A$ such that $\sigma;\tau = \iota_A$ and $\tau;\sigma = \iota_B$. Are they isomorphic in the naive sense described above? (Just stop to think a minute. Maybe it would be worth going back and reformulating something?)

4. Characterise the monomorphisms and epimorphisms in the category $\mathcal{LFG}$.

2.1.4 Elementary structure on $\mathcal{LFG}$

This section contains a brief sketch of some elementary categorical structure on $\mathcal{LFG}$.

Symmetric monoidal closed structure on $\mathcal{G}$ Here is a brief indication of why $\mathcal{LFG}$ supports the structure of a symmetric monoidal closed category (SMCC).[12] First we have a choice of 'unit' I; and we have the construction $A \otimes B$ which is functorial in A and B. Secondly the isomorphisms from Exercises 2 give natural isomorphisms: associativity, $a_{ABC} : (A \otimes B) \otimes C \longrightarrow A \otimes (B \otimes C)$; identities $l_A : I \otimes A \longrightarrow A$ and $r_A : A \otimes I \longrightarrow A$; and symmetry $c_{AB} : A \otimes B \longrightarrow B \otimes A$. These satisfy the coherence conditions detailed in Appendix A, so that $\mathcal{G}$ has the structure of a symmetric monoidal category. Finally we have a natural isomorphism $\mathcal{LFG}(\mathcal{A} \otimes \mathcal{B}, \mathcal{C}) \cong \mathcal{LFG}(\mathcal{A}, \mathcal{B} \multimap \mathcal{C})$ derived from the isomorphism $(A \otimes B) \multimap C \cong A \multimap (B \multimap C)$ and hence a closed structure on $\mathcal{LFG}$.

Categorical products The definition for finite products in $\mathcal{LFG}$ is given below; the extension to infinite products is obvious.

Definition 2.6

First define the terminal game 1 by stipulating that $M_1 = \varnothing$. Thus the terminal game 1 is just the unit game I. (The game tree consists only of the empty sequence, and no moves can be made in 1.)

Now suppose that A and B are games; the product game $A \times B$ is defined as follows:

Moves *The moves are $M_{A\times B} = M_A + M_B$, the coproduct or disjoint union of M_A and M_B.*

Game tree *The game tree $T_{A\times B}$ is the subtree of $(M_{A\times B})^*$ consisting of those sequences p with either $p = p_{M_A} \in T_A$ or $p = p_{M_B} \in T_B$.*

[11] In matrix notation one can write the map c in the definition of ι_A as

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Those who know the Geometry of Interaction (Girard 1989a) will recognise this matrix.

[12] This notion is explained in Appendix A, but if the reader just has in mind that the various isomorphisms given in Exercises 2 are intuitively natural, that will probably be enough.

Proposition 2.2 *The object 1 is a terminal object in $\mathcal{LFG}$. For games A and B, there are maps* **fst** $: A \times B \longrightarrow A$, **snd** $: A \times B \longrightarrow B$ *exhibiting $A \times B$ as a product in $\mathcal{LFG}$.*

Enrichment in domains We can regard $\mathcal{LFG}$ as a category enriched in some category of domains. We make do with an intuitive account of enrichment, but for more on this important topic see Kelly (1982). Here we presuppose some knowledge of domain theory. It is clear that the collection of all P-strategies $\mathcal{LFG}(A)$ in a game A forms a poset under inclusion (in either mode). We expect it to be some kind of domain or complete partial order (CPO).

Given an even position $p \in P_A$ of a game A, we define the least strategy $\sigma[p]$ giving rise to the play p by $\sigma[p] = \{q | q \preceq p\}$. (In function mode $\sigma[p](q) = a$ if and only if $q.a \preceq p$.)

Lemma 2.3 *(i) For any subset X of $\mathcal{LFG}(A)$, the supremum $\bigsqcup X$ exists if and only if $\bigcup X$ is a (partial) function, (in which case $\bigsqcup X = \bigcup X$). Hence each $\mathcal{LFG}(A)$ is a consistently complete* CPO.
(ii) A strategy $\sigma \in \mathcal{LFG}(A)$ is compact if and only if σ is a finite partial function if and only if σ is a finite subtree. Hence each $\mathcal{LFG}(A)$ is algebraic and satisfies the finiteness axiom (I).
(iii) A strategy $\sigma \in \mathcal{LFG}(A)$ is prime if and only if it is $\sigma[p]$ for some position $p \in P_A$ (that is, if and only if as a subtree, it consists of a single path). Hence each $\mathcal{LFG}(A)$ is prime algebraic.
It follows at once that the collection $\mathcal{LFG}(A)$ of P-strategies in A ordered by inclusion is a dI-domain.

Now we know that each $\mathcal{LFG}(B, C) = \mathcal{LFG}(B \multimap C)$ carries the structure of a dI-domain. To show that $\mathcal{LFG}$ is enriched in some category of domains, we should consider how the operations of $\mathcal{LFG}$ as a structured category are reflected in maps of dI-domains. Examples of such operations are the composition $\mathcal{LFG}(A, B) \times \mathcal{LFG}(B, C) \to \mathcal{LFG}(A, C)$ and tensor product $\mathcal{LFG}(A, B) \times \mathcal{LFG}(C, D) \to \mathcal{LFG}(A \otimes C, B \otimes D)$. Since strategies are determined by the prime strategies which they contain, it follows that all the relevant maps are affine (that is, preserve all non-empty sups) or biaffine as in the examples above. Hence we have the main result of this section.

Theorem 2.4 *The category $\mathcal{LFG}$ of games and strategies is a* SMCC *with products, enriched over the* SMCC *of dI-domains and affine maps.*

This result on enrichment suggests that we think of the category of games as a kind of generalised (linear) domain theory. This perspective is useful in a number of ways. For example, one can present $\mathcal{LFG}$ as a category indexed over itself in such a way as to obtain a model for polymorphism (along the lines of Girard (1986), Coquand, Gunter, and Winskel (1989), Taylor (1986), and others). For another approach to polymorphism see Abramsky (1997).

Exercises 4

1. *Recall the game S with just one initial move after which the game is over.*
(i) Show that any game is isomorphic to a (possibly infinite) product of games with just one initial move.
(ii) Establish the following decomposition.[13] *For any game A we can find games B_i for $i \in I$ with $A \cong \prod\{B_i \multimap S | i \in I\}$.*
(iii) Identify the least collection $\mathcal{W}$ of games containing S, closed under isomorphism and under finite products, and such that if $W \in \mathcal{W}$ then $W \multimap S \in \mathcal{W}$. Show that the full subcategory on such objects is a SMCC.

2. *Establish the following facts.*
(i) The contravariant functor $(-) \multimap S$ is self dual on $\mathcal{LFG}$: we have

$$\mathcal{LFG}(A, B \multimap S) \cong \mathcal{LFG}(B, A \multimap S)$$

naturally in A and B.
(ii) The game $A \multimap S$ can be identified with $S(A)$, the game which is obtained from A by adding a fresh initial move, $\bullet$ say, for Opponent, and then letting play continue as in $A^\perp$.
(iii) The self duality is enriched in the closed structure:

$$A \multimap S(B) \cong S(A \otimes B) \cong B \multimap S(A)$$

3. *A map $\rho : A \longrightarrow B$ in $\mathcal{LFG}$ is* strict *if and only if $\perp; \rho = \perp : I \longrightarrow B$. Write $\mathcal{LFG}_s(A, B)$ for the set of strict maps from A to B. On the basis of this definition establish the following facts.*
(i) A map $\rho : A \longrightarrow B$ is strict just when ρ's response to an initial move (necessarily) in B is always a move in $A^\perp$.
(ii) Let $L = S^2$ be the square of the functor S above. Show that L is the lift for our notion of strict map: that is, there is a natural isomorphism $\mathcal{LFG}(A, B) \cong \mathcal{LFG}_s(LA, B)$.
(iii) Let $\eta_A : LA \to A$ be the strict map corresponding to the identity ι_A in the natural isomorphism. Show that $\rho : A \longrightarrow B$ is strict if and only if $L\rho; \eta_B = \eta_A; \rho$.

4. *Show that $\mathcal{LFG}$ does not have a coproduct, but that it does have a weak coproduct.*

[13] One can read this decomposition as follows. A game can do an O-move and become a cogame. Similarly a cogame can do a P-move and become a game. You may detect echoes of The Expansion Theorem for CCS in this question.

5. *Consider the category of games with history-free strategies in the sense of Abramsky and Jagadeesan (1994). Does this category have finite products?*

2.2 A cartesian closed category of games

2.2.1 The exponential comonad

The simple intuition behind the so-called exponential $!A$ of a game A is that it is an infinite ordered tensor product of (versions of) A.[14] We imagine that we are given instances $A_0, A_1, A_2, \cdots$ of the game A, and we play their infinite tensor product subject to the stipulation that O may not open (that is make the first move in) an instance A_{k+1} until all the A_i for $i \leq k$ have been opened. The formal definition is as follows.

Definition 2.7 *Suppose that A is a game for fun. Then the game $!A$ is defined as follows:*
Moves *The moves are $M_{!A} = M_A \times \mathbb{N}$, the countable copower of M_A.*
Game tree *$T_{!A}$ is the subtree of $(M_{!A})^*$ consisting of those sequences p such that (i) all the projections p_k are in T_A, and (ii) the first move in the $k+1$st copy is made after the first move in the kth.*

We devote this section to an explanation of that structure associated with the exponential which gives rise to a cartesian closed category of games.

! as a monoidal comonad It is routine to check that $!A$ is functorial in A; so that we have a functor $! : \mathcal{LFG} \longrightarrow \mathcal{LFG}$. In addition we can define mediating natural transformations $m_I : I \longrightarrow !I$ and $m_{AB} : !A \otimes !B \longrightarrow !(A \otimes B)$ making $!$ a *monoidal (endo)functor* on $\mathcal{LFG}$. The axioms for monoidal functors are explained in Appendix A; here we just describe the maps involved.

The map m_I This is uniquely determined since $!I \cong I \cong 1$.

The map m_{AB} This is more interesting; we have $(!A)^\perp$ and $(!B)^\perp$ in parallel with $!(A \otimes B)$. Player does the natural thing; moves in the successive versions of $A \otimes B$ are copied to moves in the successive versions of $A^\perp$ or $B^\perp$ as appropriate, and then also vice-versa. (This requires careful bookkeeping as which version corresponds to which is not determined in advance of a play. Figure 3 illustrates the case where O starts by playing in A in the first version of $A \otimes B$, then plays in B in the second version and then continues in A in the second version.)

We can further define the counit $\varepsilon_A : !A \longrightarrow A$ and the comultiplication $\delta_A : !A \longrightarrow !!A$ for a comonad as follows.

[14] Curien has considered in detail a more sophisticated exponential, which is already implicit in his early work on sequentiality, and which gives rise to the category of sequential algorithms. This underlies the recent treatment of full abstraction for extensions of PCF given in Cartwright, Curien, and Felleisen (1994). Curien's exponential is a retract of our crude exponential.

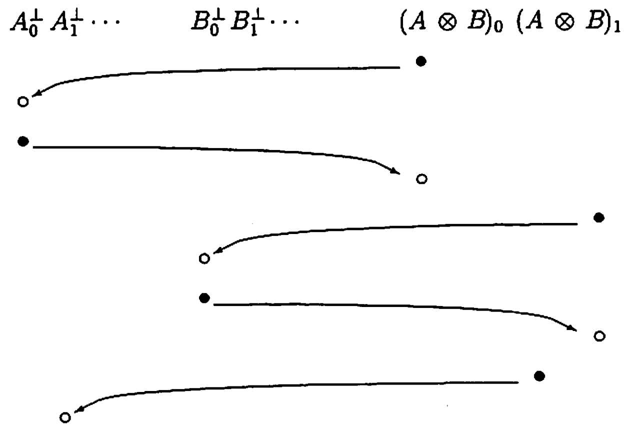

Figure 3: Mediating strategy.

Counit ε_A In the game $!A \multimap A$ we play $(!A)^\perp$ in parallel with A. In the strategy ε_A Player only makes use of the first copy of $A^\perp$ in $(!A)^\perp$, and simply copies moves across as in the identity (copy-cat) strategy. (The picture is like Figure 1, but with further unused copies of $A^\perp$ on the left.)

Comultiplication δ_A In the game $!A \multimap !!A$, we have $(!A)^\perp$ in parallel with $!!A$. In $!!A$, Opponent has in effect an $\mathbb{N} \times \mathbb{N}$-indexed collection of copies of A in which he may choose to play (of course with restriction on the legitimate order). In the strategy δ_A, Player makes use of his $\mathbb{N}$-indexed copies of $A^\perp$ in $(!A)^\perp$, to imitate the behaviour of Opponent; whenever O opens a new copy of A, P opens a corresponding new copy of $A^\perp$, thereby instituting a link between the respective copies; and thereafter whenever O plays in the one, P copies in the other. The idea is indicated in Figure 4. Moves from the first two versions of A in the first version of $!A$ and moves from the first version of A in the second version of $!A$ are shown copied.

It is routine to check that that ε_A and δ_A are the components of natural transformations $\varepsilon : ! \longrightarrow 1_{\mathcal{LFG}}$ and $\delta : ! \longrightarrow !!$; and that they are monoidal natural transformations in the sense explained in Appendix A. We can sum up this discussion in the following proposition.

Proposition 2.5 *The data $(!, \varepsilon, \delta)$ together with the ancillary structure m_I and m_{AB} form a monoidal comonad on $\mathcal{LFG}$.*

Comonoid structure on ! Now let us see how ! supports the operations of weakening and contraction associated with the 'exponential' of linear logic. The discard maps $e_A : !A \longrightarrow I$ and duplication maps $d_A : !A \longrightarrow !A \otimes !A$ are as follows.

Discard e_A The game $!A \multimap I$ is rather disappointing; it has no starting moves for O and so is isomorphic to I. Thus we have to let e_A be the unique 'empty strategy' in $!A \multimap I$.

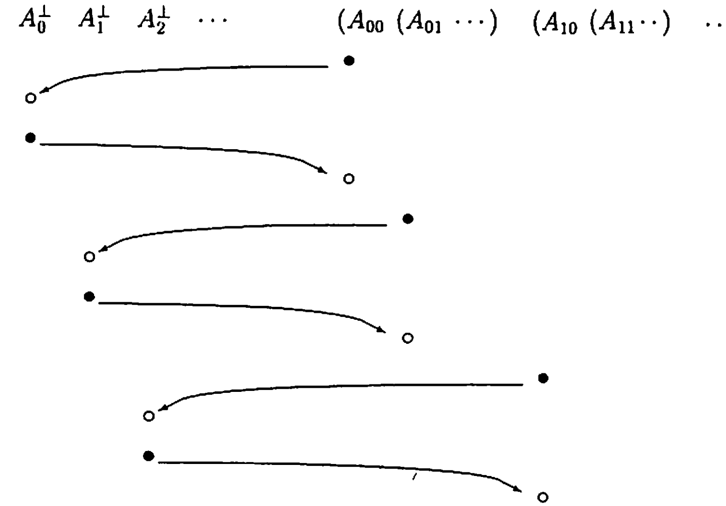

Figure 4: Comultplication strategy.

Duplication d_A The game $(!A \multimap !A \otimes !A)$ is more interesting. It amounts to playing $(!A)^\perp$ in parallel with two copies of $!A$. Opponent can switch between the copies of $!A$, while Player can switch between $(!A)^\perp$ and whichever copy of $!A$ is current. In the strategy d_A, Player systematically sets up a correspondence between fresh versions of A started by O in either of the two copies of $!A$, and fresh versions of $A^\perp$ in the copy of $(!A)^\perp$; he simply copies O's moves from any version to the corresponding one. So the picture is as in Figure 4 save that there are just the two (rather than countably many) copies of A on the right.

It is straightforward to check the desirable properties of discard and duplication.

Proposition 2.6 *The components of the natural transformations $e_A : !A \to I$ and $d_A : !A \to !A \otimes !A$ give each free coalgebra $\delta_A : !A \to !!A$ the structure of a commutative comonoid. This structure is compatible with the comonad in that it is preserved by coalgebra maps: whenever $f : (!A, \delta_A) \to (!B, \delta_B)$ is a coalgebra, then f is also a comonoid morphism.*

We could deduce at once that the category of Eilenberg-Moore coalgebras has products; and that the full subcategory on objects isomorphic to finite products of free coalgebras is cartesian closed. Fortunately, the category $\mathcal{LFG}$ has products. It follows that it satisfies Seely's axioms for a model of linear logic extended as in Bierman (1995), that is, it is a new-Seely category. Hence the Kleisli category of the comonad is the CCC in which we are interested.

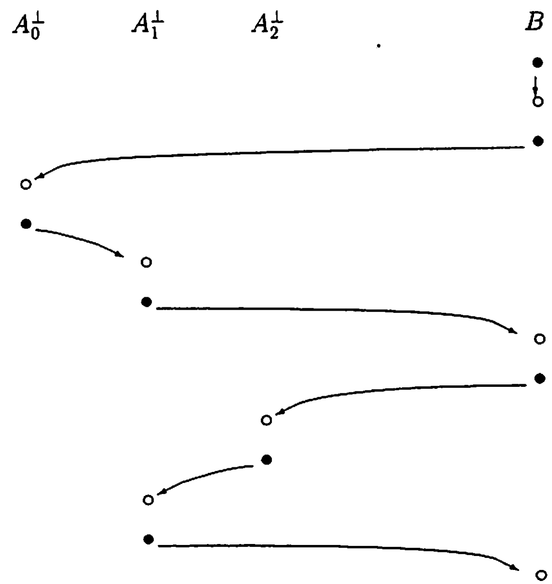

Figure 5: Play in the function space $A \Rightarrow B$.

Exercises 5

1. *Is the counit $\varepsilon_A : !A \to A$ ever an isomorphism? Is the comultiplication $\delta_A : !A \to !!A$ ever an isomorphism?*
2. *For which games A do we have an isomorphism $A \cong !A$? For which games A do we have an isomorphism $!A \cong !!A$?*
3. *Give an example to show that $!A \otimes !B$ and $!(A \otimes B)$ need not be isomorphic.*[15]

2.2.2 Description of the CCC $\mathcal{FG}$

Definition 2.8 *The category $\mathcal{FG}$ of* fun-games *(without qualification) is the Kleisli category for the comonad $!$ on $\mathcal{LFG}$*

The comonoid structure on the comonad $(!, \varepsilon, \delta)$ together with the existence of products in our category $\mathcal{LFG}$ ensures that the Kleisli category for $!$ is cartesian closed.

[15]The question why these objects are not isomorphic arose at the summer school. Nick Benton immediately gave a computational intuition. In $!(A \otimes B)$ there are always the same number of versions of A and B in which Opponent could be playing without having to 'call up' a fresh copy. But that is not true of $!A \otimes !B$.

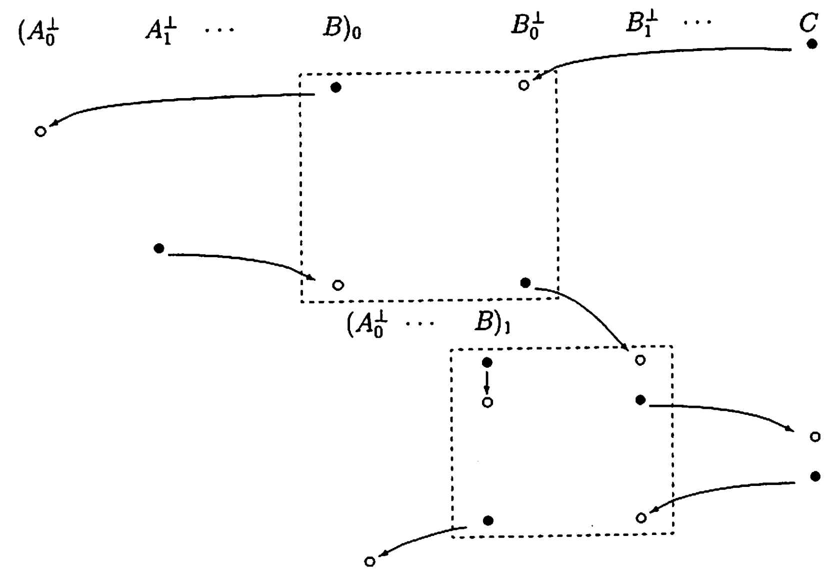

Figure 6: Composition of $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$.

Proposition 2.7 *The category $\mathcal{FG}$ is cartesian closed. The categorical product is the product $A \times B$ and the function space $B \Rightarrow C$ is defined by*

$$B \Rightarrow C = !B \multimap C.$$

The standard adjunction induced by the comonad $!$ is a monoidal adjunction.

We write $\mathcal{FG}(A) = \mathcal{LFG}(A)$ for the elements (points) of A in $\mathcal{FG}$; then $\mathcal{FG}(A, B) = \mathcal{FG}(A \Rightarrow B) = \mathcal{LFG}(!A \multimap B)$.

Explicit description of $\mathcal{FG}$ The category $\mathcal{FG}$ of fun-games and (non-linear) strategies has the following description. The objects of the category are games as in Definition 2.1. The maps from A to B are strategies in the game $!A \multimap B = A \Rightarrow B$ obtained by playing a countable sequence of versions of $A^\perp$ in parallel with B. A typical play is illustrated in Figure 5, with a strategy for Player indicated by the arrows. The identity on a game A is given by the map $\varepsilon_A : !A \to A$. Finally we describe how to compose two maps $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ in $\mathcal{FG}$. The strategies $\sigma : !A \multimap B$ and $\tau : !B \multimap C$ in $\mathcal{LFG}$ compose to give the composite

$$!A \xrightarrow{\delta} !!A \xrightarrow{!\sigma} !B \xrightarrow{\tau} C.$$

A play according to a composite strategy is indicated in Figure 6. Opponent opens in C and τ gives Player a move in the first copy $B_0^\perp$. This opens a scratchpad, and a few of Player's moves are shown given by σ. Eventually Player copies over a move as an O-move in $B_0^\perp$, and τ gives a response in $B_1^\perp$. This opens a fresh scratchpad, and calls up a fresh version of σ, and P-moves are made in accord with σ then with τ and then with σ again. In Figure 6 two scratchpads are shown, but more could be opened, and Player may well return to earlier scratchpads.

Enrichment The category $\mathcal{LFG}$ is enriched in the SMCC of dI-domains and affine maps. What about $\mathcal{FG}$? As $\mathcal{FG}(A,B) \cong \mathcal{LFG}(!A,B)$, we can again regard the hom-sets as dI domains. So we need to look at the maps giving the structure of the cartesian closed category. It is easy to see (for example in the case of composition) that we lose affineness; but the maps remain continuous. (The category of dI domains and continuous maps has products, and thus a symmetric monoidal structure; but it is not cartesian closed.)

Proposition 2.8 *The category $\mathcal{FG}$ of fun-games is a* CCC *enriched over the category of dI-domains and continuous maps.*

One can regard the CCC $\mathcal{FG}$ as as an intensional model of the typed λ-calculus in the sense explained in the Introduction: it does not have enough points. As indicated in the Exercises, one can see this by considering the object Σ whose unique maximal play consists of an O-move $\bullet$ followed by a P-move $\circ$. In the next sections we shall consider $\mathcal{FG}$ as a model PCF, and on the basis of that try to tease out the intuition that the category models intensional behaviour.

Exercises 6

1. *Recall the game S with just an initial O-move. Show that the linear and general function spaces coincide: $S \Rightarrow S \cong S \multimap S$.*
2. *(i) Show that $(S \Rightarrow S) \cong \Sigma$, the game with a unique maximal play consisting of an O-move $\bullet$ followed by a P-move $\circ$.*
   (ii) Show that Σ has just two points.
   (iii) Show that there are infinitely many strategies in $\mathcal{FG}(\Sigma,\Sigma)$, and deduce that such strategies are not distinguished by the points of Σ.
3. *Can you construct games A with*
   (i) $A \cong A \times A$;
   (ii) $A \cong A \Rightarrow A$;
   (iii) $A \cong A \times A \cong A \Rightarrow A$?
4. *If games A and B are isomorphic in $\mathcal{FG}$, are they isomorphic in $\mathcal{LFG}$?*
5. *Recall the lift functor $L = S^2$ on $\mathcal{LFG}$. Does L act as a lift in the category $\mathcal{FG}$?*

2.3 Games as a model for intensional computation

We show how strategies for games provide a notion of algorithm[16] by explaining how our category provides a model for PCF. (A brief overview of PCF is given

[16] A better case is made by recent work of Abramsky modelling idealised ALGOL.

in Appendix B.) What is commonly called a denotational semantics for PCF is essentially an interpretation of (or model for) the type theory presented in Appendix B. The usual form of a model for PCF is that the types are interpreted as domains and the terms as continuous (or stable continuous) maps between domains. Here we describe a model in which the types are interpreted as games and the terms as strategies.

2.3.1 Interpretation of PCF types

First we define games $\mathbb{B}$ and $\mathbb{N}$ to model the ground types. These games are of a simple kind. In each game there is a unique opening move for O; in $\mathbb{B}$, P has two possible responses t and f, and then the game is over; in $\mathbb{N}$, P has a countable number of possible responses n for each natural number n, and again then the game is over. The following is a formal definition.

Definition 2.9 *The* boolean game *$\mathbb{B}$ has moves $M_{\mathbb{B}} = \{\bullet, t, f\}$, and game tree $T_{\mathbb{B}} = \{p | p \preceq \bullet.t \text{ or } p \preceq \bullet.f\}$.*
The natural number *game $\mathbb{N}$ has moves $M_{\mathbb{N}} = \{\bullet, 0, 1, 2, 3, \cdots\}$, and game tree $T_{\mathbb{N}} = \{p | p \preceq \bullet.n \text{ for some natural number } n\}$.*

Now for any PCF-type A we define the interpretation $[\![A]\!]$ as a game recursively as follows:

$$\begin{aligned} [\![o]\!] &\stackrel{\text{def}}{=} \mathbb{B}, \\ [\![\iota]\!] &\stackrel{\text{def}}{=} \mathbb{N}, \\ [\![A_1 \Rightarrow A_2]\!] &\stackrel{\text{def}}{=} [\![A_1]\!] \Rightarrow [\![A_2]\!]; \end{aligned}$$

where $\mathbb{B}$ and $\mathbb{N}$ are the boolean and natural number games just defined.

Exercises 7

1. *Show that the domains of strategies $\mathcal{FG}(\mathbb{B})$ and $\mathcal{FG}(\mathbb{N})$ are the traditional flat domains of booleans and natural numbers respectively. (Thus, our interpretation of* PCF *is standard in the sense of Plotkin (1977).)*
2. *Show that the function space domain $\mathcal{FG}(\mathbb{B} \Rightarrow \mathbb{B})$ is infinite.*
3. *What possibilities are there for the first four moves in a strategy in the domain $\mathcal{FG}(\mathbb{B} \times \mathbb{B} \Rightarrow \mathbb{B})$?*
4. *Construct a game whose domain of strategies is the traditional domain of lazy natural numbers.*

2.3.2 Interpretation of PCF: arithmetic and conditionals

Arithmetic The basic arithmetic constants are straightforwardly interpreted as strategies. The interpretations of $n : \iota$ and $\mathtt{t}, \mathtt{f} : o$ are the corresponding total strategies in the games $\mathbb{B}$ and $\mathbb{N}$. We consider the interpretation of the successor function succ for illustration. The game in question $[\![(\iota,\iota)]\!] = (\mathbb{N} \Rightarrow \mathbb{N})$ has as set of moves

$$M_{\mathbb{N}} + M_{\mathbb{N}} = \{(0,\bullet),(0,n) \mid n \in \mathbb{N}\} \cup \{(1,\bullet),(1,n) \mid n \in \mathbb{N}\}$$

and an infinite game tree which we need not describe in detail. The strategy $[\![\mathsf{succ} : (\iota,\iota)]\!]$ is best defined by giving the maximal plays in the subtree; these are

$$\{(1,\bullet).(0,\bullet).(0,n).(1,n+1) \mid n \in \mathbb{N}\}\,.$$

Conditionals We deal with the conditional at ι. The game in question $[\![(o,\iota,\iota,\iota)]\!] = (\mathbb{B} \Rightarrow \mathbb{N} \Rightarrow \mathbb{N} \Rightarrow \mathbb{N})$ has as set of moves

$$M_{\mathbb{B}} + M_{\mathbb{N}} + M_{\mathbb{N}} + M_{\mathbb{N}} = \{(0,\bullet),(0,t),(0,f)\} \cup$$

$$\{(1,\bullet),(1,n) \mid n \in \mathbb{N}\} \cup \{(2,\bullet),(2,n) \mid n \in \mathbb{N}\} \cup \{(3,\bullet),(3,n) \mid n \in \mathbb{N}\}$$

and an infinite game tree which we need not describe in detail. Again the strategy $[\![\mathsf{cond}^\iota : (o,\iota,\iota,\iota)]\!]$ is defined by giving the maximal plays in the subtree; these are

$$\{(3,\bullet).(0,\bullet).(0,\mathrm{t}).(1,\bullet).(1,n).(3,n) \mid n \in \mathbb{N}\} \cup$$

$$\{(3,\bullet).(0,\bullet).(0,\mathrm{f}).(2,\bullet).(2,n).(3,n) \mid n \in \mathbb{N}\}\,.$$

Exercises 8

1. *Define strategies for predecessor,* pred, *and test for zero,* zero?, *by giving the maximal plays in the subtree as above.*
2. *(A misleading question!) What is wrong with the strategy whose maximal moves are*

$$\{(1,\bullet).(0,\bullet).(0,n).(0,\bullet).(0,n).(1,n+1) \mid n \in \mathbb{N}\}\,?$$

3. *Show inductively that the conditionals* cond^ι *and* cond^o *enable one to define conditionals at all types. The resulting conditional* $\mathsf{cond}^{(\iota,\iota)}$ *is (probably) given by*

$$\mathsf{cond}^{(\iota,\iota)} = \lambda x : o.\lambda f, g : (\iota,\iota).\lambda w : \iota.\mathsf{cond}^\iota(x, f(w), g(w))\,.$$

Describe the strategy which is defined by this.

4. *(i) Define a strategy for a non-standard function test for one,* one?, *and compare what you give with the interpretation of*

$$\lambda x : \iota.\mathsf{cond}^\iota(\mathsf{zero?}(\mathsf{pred}(x)), \mathsf{cond}^\iota(\mathsf{zero?}(x), \mathrm{f}, \mathrm{t}), \mathrm{f})\,.$$

(ii) Describe the strategy which interprets the term

$$\lambda x : \iota.\mathsf{cond}^\iota(\mathsf{zero?}(x), \mathsf{cond}^\iota(\mathsf{one?}(x), 0, \Omega^\iota), \mathsf{cond}^\iota(\mathsf{zero?}(x), 0, \Omega^\iota))\,.$$

2.3.3 Fixed points

For any game (not just those which interpret PCF-types) A we shall describe a strategy in $(A \Rightarrow A) \Rightarrow A$ to interpret $\mathbf{Y}$. First we need to understand the structure of the game $(A \Rightarrow A) \Rightarrow A$. The game amounts to a sequence of cogames $(A \Rightarrow A)_i^\perp$ played in parallel with A, in such a way that P can switch games, but O cannot. We call the indicated version of A (in which O starts the game) the *main O-component*. Each $(A \Rightarrow A)_i^\perp$ amounts to playing a sequence A_{ij} in parallel with $A_i^\perp$, in such a way that O can switch games, but P cannot. We call the $A_i^\perp$, in which P starts, the *P-components*, and the games A_{ij} the *subsidiary O-components*. We list the components which we use to structure a discussion of play in $(A \Rightarrow A) \Rightarrow A$.

- *The main O-component.*
- *The P-components.*
- *Subsidiary O-components.*

We now proceed to describe a strategy $\mathbf{Y}$ in $(A \Rightarrow A) \Rightarrow A$. In a play according to the strategy we describe there will be a correspondence between O- and P-components. The first P-component to occur $A_0^\perp$ is the dual of the main O-component. The others in order they are started are the duals of the subsidiary O-components in order they are started. At any even position the duals will be copies of each other. The strategy can be succinctly described as follows: suppose O has just moved:

- *Case 1.* Opening move: P copies this to start the first P-component.
- *Case 2.* O starts a new subsidiary component: P copies this to start a new P-component.
- *Case 3.* O moves in some existing O-component (P-component): P copies the move in the dual P-component (O-component).

Arguing inductively it is easy to see this makes sense as a strategy.

Now we aim to show that $\mathbf{Y}$ is a fixed point operator. (In what follows, we shall not bother to distinguish between the strategy $\mathbf{Y} \in \mathcal{FG}((A \Rightarrow A) \Rightarrow A)$, the map $\mathbf{Y} : 1 \longrightarrow ((A \Rightarrow A) \Rightarrow A)$ in $\mathcal{FG}$, and its exponential transpose $\mathbf{Y} : (A \Rightarrow A) \longrightarrow A$.) Externally (or pointwise), that Y is a fixed point means just that the equation

$$\sigma(\mathbf{Y}(\sigma)) = \mathbf{Y}(\sigma)$$

holds for all $\sigma : A \Rightarrow A$. But as $\mathcal{FG}$ does not have enough points, this is not sufficient to provide a good model for PCF. Rather we need to show that

$$f : (A \Rightarrow A) \models f(\mathbf{Y}(f)) = \mathbf{Y}$$

holds in $\mathcal{FG}$. The expression $f(\mathbf{Y}(f))$ is interpreted as the composition

$$(A \Rightarrow A) \xrightarrow{\Delta} (A \Rightarrow A) \times (A \Rightarrow A) \xrightarrow{1 \times \mathbf{Y}} (A \Rightarrow A) \times A \xrightarrow{\mathrm{ev}} A\,,$$

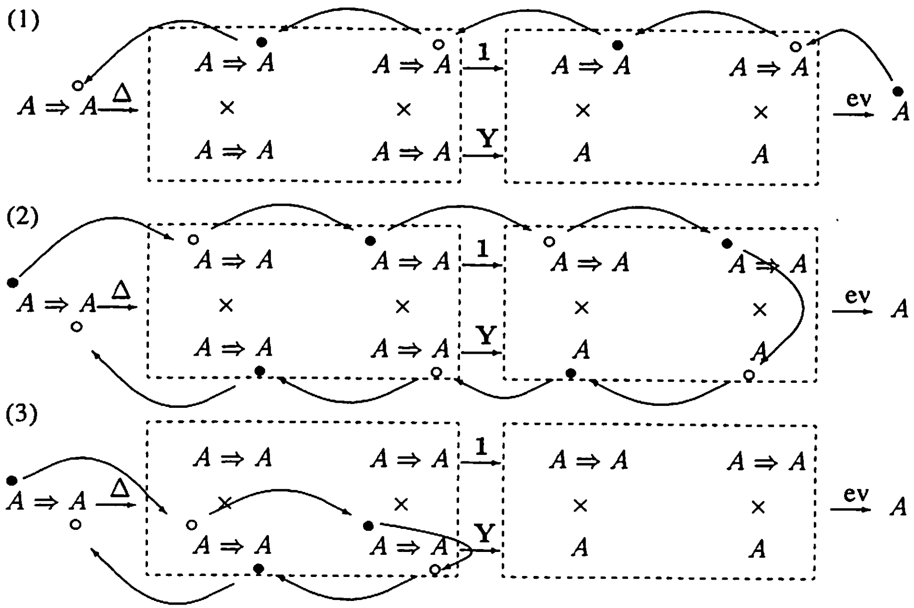

Figure 7: Copying paths in the composite $\Delta; \mathbf{1} \times \mathbf{Y}; \mathrm{ev}$.

where $\mathbf{1}$ is $\iota_{A\Rightarrow A}$. This composite is pictured (three times) in Figure 7 where "copying paths" are drawn to help explain the behaviour of the composite.[17]

In our discussion of the composite $\Delta; \mathbf{1} \times \mathbf{Y}; \mathrm{ev}$ we use the terminology (main O-component, P-components, subsidiary components, the dual of a component) already introduced. We can tie down the behaviour of the composed strategy by making the following observations.

1. After the opening O-move in the main O-component, P copies along path (1) in Figure 7 to give a reply in the first P-component $A_0^\perp$ (the dual of the main O-component). Thereafter any O-move in either of the two components is answered by copying along path (1) (in either direction) to give a P-reply (which is just a "copy") in the dual component.

2. After the first O-move (if any) in the first subsidiary O-component A_{00}, P copies along path (2) to give a reply in the second P-component $A_1^\perp$ (its dual). Thereafter any O-move in either of these two components is answered by copying along path (2) (in either direction) to give a P-reply (just a "copy") in the dual component. Furthermore exactly the same applies *mutatis mutandis* in the case of any subsidiary O-component A_{0j}; after an opening move in A_{0j}, P opens the next available $A_k^\perp$, say, (its dual). [Perhaps it is worth noting that each of these further O-components involves P opening a fresh version of $\mathbf{Y}$ on the hidden scratch pad *i.e.* the "returning" portion of path (2) involves a fresh version of $\mathbf{Y}$ in each new case.]

[17] It does not seem accidental that the three paths in Figure 7 "cover" the whole picture.

3. After the first O-move (if any) in a subsidiary O-component $A_{(i+1)j}$, P copies along path (3) to give a reply in the next available P-component $A_l^\perp$, say,(its dual). Thereafter any O-move in either of these two components is answered by copying along path (3) (in either direction) to give P-reply (just a "copy") in the dual component.

It follows from the three observations above that the composed strategy behaves exactly like $\mathbf{Y}$: that is

$$\begin{tikzcd}
(A \Rightarrow A) \times (A \Rightarrow A) \arrow[r, "\mathbf{1} \times \mathbf{Y}"] & (A \Rightarrow A) \times A \arrow[d, "\mathrm{ev}"] \\
A \Rightarrow A \arrow[u, "\Delta"] \arrow[r, "\mathbf{Y}"'] & A
\end{tikzcd}$$

commutes.

Exercises 9

1. *Show that*

$$f : A \Rightarrow B, g : B \Rightarrow A \vDash f(\mathbf{Y}(\lambda a.g(f(a))) = \mathbf{Y}(\lambda b.f(g(b)))\,.$$

2. *Show that*

$$f : A \times A \Rightarrow A \vDash \mathbf{Y}(\lambda a.f(a,a)) = \mathbf{Y}(\lambda a.\mathbf{Y}(\lambda a'.f(a,a'))\,.$$

3. *Show that*

$$f : A \Rightarrow A \vDash \mathbf{Y}(f) = \mathbf{Y}(\lambda a.f(f(a)))\,.$$

4. *Recall that $\mathcal{FG}$ is enriched in a category of* CPOs. *For a game A consider the map $\mathcal{FG}((A \Rightarrow A) \Rightarrow A) \to \mathcal{FG}((A \Rightarrow A) \Rightarrow A)$ taking a strategy ρ to the interpretation of $\lambda f : A \Rightarrow A.f(\rho(f))$. Let $\tilde{\mathbf{Y}} : (A \Rightarrow A) \Rightarrow A$ be the least fixed point of this map.*
(i) Show that $\tilde{\mathbf{Y}}$ is a fixed point operator.
(ii) (For the more experienced.) Show that $\tilde{\mathbf{Y}} = \mathbf{Y}$.

2.3.4 Catch

We wish to consider a revealing strategy in the game $(\mathrm{N} \times \mathrm{N} \Rightarrow \mathrm{N}) \Rightarrow \mathrm{N}$. In the spirit of Cartwright, Curien, and Felleisen (1994) we extend PCF by a constant catch : $((\iota\times\iota,\iota),\iota)$, whose interpretation will be this strategy. The game in question has as set of moves

$$M_{\mathrm{N}} + M_{\mathrm{N}} + M_{\mathrm{N}} + M_{\mathrm{N}} = \{(0,\bullet),(0,n)\,|\, n \in \mathrm{N}\} \cup$$

$$\{(1,\bullet),(1,n) \mid n \in \mathbb{N}\} \cup \{(2,\bullet),(2,n) \mid n \in \mathbb{N}\} \cup \{(3,\bullet),(3,n) \mid n \in \mathbb{N}\}$$

and an infinite game tree which we need not describe in detail. The strategy $[\![\mathsf{catch} : ((\iota \times \iota, \iota), \iota)]\!]$ is again defined by giving the maximal plays in the subtree; these are

$$\{(3,\bullet).(2,\bullet).(2,k).(3,k+2) \mid k \in \mathbb{N}\} \cup$$
$$\{(3,\bullet).(2,\bullet).(0,\bullet),(3,0)\} \cup \{(3,\bullet).(2,\bullet).(1,\bullet).(3,1)\}\,.$$

This algorithm inspects its argument, the algorithm $\sigma : \mathbb{N} \times \mathbb{N} \Rightarrow \mathbb{N}$ say, in a non-extensional way. If σ outputs a value k at once, then the algorithm returns $k+2$. Otherwise if σ starts by 'looking at its first argument' it returns 0, and if it starts by 'looking at its second argument' it returns 1. If σ does nothing, then the algorithm returns nothing.

Let us explain the sense in which $[\![\mathsf{catch}]\!]$ is a non-extensional strategy. Consider the following two algorithms $+_l : \mathbb{N} \times \mathbb{N} \Rightarrow \mathbb{N}$ and $+_r : \mathbb{N} \times \mathbb{N} \Rightarrow \mathbb{N}$ for addition. The game in question has moves $M_{\mathbb{N}} + M_{\mathbb{N}} + M_{\mathbb{N}}$, and the maximal plays are for $+_l$,

$$\{(2,\bullet).(0,\bullet).(0,n).(1,\bullet).(1,m).2,n+m) \mid n,m \in \mathbb{N}\},$$

and for $+_r$,

$$\{(2,\bullet).(1,\bullet).(1,m).(0,\bullet).(0,n).2,n+m) \mid n,m \in \mathbb{N}\}.$$

The two algorithms $+_l$ and $+_r$ are extensionally equivalent: the composites with any algorithms $n : \mathbb{N}$ and $m : \mathbb{N}$ are the same. On the other hand, $[\![\mathsf{catch}]\!](+_l) = 0 : \mathbb{N}$ while $[\![\mathsf{catch}]\!](+_r) = 1 : \mathbb{N}$, so catch is sensitive to the intensional difference.

One of the central results of Cartwright, Curien, and Felleisen (1994) is that the category of sequential algorithms (Curien 1993) is fully abstract for an extension of PCF by catch. One can obtain the category of sequential algorithms as a kind of retract of $\mathcal{FG}$, but the details are not simple. In any case the algorithms which are the strategies of $\mathcal{FG}$ are of much wider scope; in functional terms, they allow notice to be taken of arguments which vary according to the circumstances under which they are called.

Exercises 10

1. *Define an extended form of* catch, $\mathsf{catch}_r : ((\iota^r, \iota).\iota)$ *so that* $\mathsf{catch} = \mathsf{catch}_2$.
   (i) What are catch_0 *and* catch_1*? Are they extensional?*
   (ii) Show that catch_1 *can not be modelled in the standard Scott domain model for* PCF*, and hence is not* PCF*-definable. Is* catch_1 *stable?*
   (iii) Is catch_r PCF*-definable from* catch*?*

2. *Consider the following strategy* $[\![\mathsf{lin0?}]\!]$ *in* $(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow \mathbb{B}$. $[\![\mathsf{lin0?}]\!]$ *inspects the algorithm* $\sigma : \mathbb{N} \Rightarrow \mathbb{N}$ *which is its argument. If* σ *outputs a value at once, then* $[\![\mathsf{lin0?}]\!]$ *returns* f*. Otherwise if* σ *starts by 'looking at its argument',* $[\![\mathsf{lin0?}]\!]$ *asks for the value of* σ *at* 0*; then if* σ *at once gives a value* $[\![\mathsf{lin0?}]\!]$ *returns* t*; on the other hand if* σ *continues by again 'asking about its argument',* $[\![\mathsf{lin0?}]\!]$ *returns* f*. If at any point* σ *does nothing, then the algorithm returns nothing.*
   (i) Write down the maximal plays for $[\![\mathsf{lin0?}]\!]$.
   (ii) Describe what $[\![\mathsf{lin0?}]\!]$ *does? In what sense is it intensional?*
   (iii) Is lin0? PCF*-definable from* catch*?*

3. *Consider the strategy in the game* $(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow \mathbb{N}$ *whose maximal plays are of the form*

   $$(2,\bullet).(1,\bullet).(0,\bullet).(0,0).(0,\bullet).(0,1)\cdots(0,\bullet).(0,n).(1,m).(2,n+m)\,.$$

   What does this strategy do?

3 Games and logic

In the previous section we presented a category of games as an intensional model for computation. Now we want to present a corresponding model for constructive proof theory.

It is worth reflecting briefly on the contrast between traditional models of computable (programmable) functions and models for proofs. The point is that the Curry-Howard isomorphism between types and propositions (programs and proofs) has its limitations. A CCC which models the most general forms of computation will in some way involve partial functions. Thus in a category of domains, all the domains are inhabited (by the bottom element $\perp$). So under the Curry-Howard isomorphism, all propositions are provable. To avoid this, we think of CCCs which model proof theory as involving total functions, and accept that the Curry-Howard isomorphism is most convincing in the context of programming in a language (such as Martin-Löf Type Theory) in which all functions are total.

This suggests at once that in order to model constructive proof theory we should consider total strategies. But care is needed to get a composition. In terms of the discussion of Section 2, the problem is that one may stay in the scratchpad of Figure 2 for ever, and so the next move may be undefined. Somehow one has to force a conclusion to the computation of the next move. This is fine if all games are 'well-founded'; but such a restriction presents problems for any simple-minded form of the exponential.[18] The good general way round the problem is to have a notion of winning, and only to compose winning strategies. In the simplest formulation we incorporate into the structure of a game a decision, for each infinite play of the game, as to whether that play is winning for Player or for Opponent. (Our presentation is definitely non-constructive. A constructive version of the material is possible, but its formulation requires care.)

[18] For finite games, an exponential in the style of Curien can overcome this problem.

3.1 A monoidal category of games and total strategies

In this section we define a specific category of games and total strategies.

3.1.1 Games and total strategies

We need to consider infinite plays, and shall use the notation for infinite sequences given in the Introduction.

Definition 3.1 *A* game to win *or* win-game *A is given by a set $M = M_A$ of* moves, *a non-empty tree T_A on M called the* game tree, *and a function $|-|_A = |-| : T_A{}^\infty \longrightarrow \{W, L\}$ giving for all infinite plays in T_A the* result *which is either W,* a win for Player, *or L,* a loss for Player, *in the game A.*
Extend the function $|-|$ at once from $T_A{}^\infty$ to $\overline{T_A}$ by setting

$$|p| = \begin{cases} W & \text{if } p \in O_A; \\ L & \text{if } p \in P_A. \end{cases}$$

The elements $p, q, ..$ of T_A are again called positions *or* plays, *and we adopt from Definition 2.1 the notions of moves of* even and odd parity *and of the sets O_A of* odd positions *and P_A of* even positions *in T_A.*

The definition of the function $|-|$ for all finite plays is the *normal play* convention in Conway (1976). Variations lead to interesting pathologies.

As before, we aim to compose strategies for Player, the P-strategies, but now we want total winning strategies. In this section we refer to these simply as P strategies. As before, we give the definition in subtree mode, just adding clauses to ensure that the strategy is total and wins.

Definition 3.2 *A* P-strategy *in a game A is a non-empty subtree S of T_A satisfying the following three conditions.*

(1) If $p \in S \cap O_A$ then there is a unique move a with $p.a \in S$.
(2) If $p \in S \cap P_A$ then for any a, $p.a \in O_A$ implies $p.a \in S$.
(3) If $p \in \widehat{S}$, then $|p| = W$.

Remark Again we can present a strategy as a partial map $\phi : O_A \longrightarrow M_A$, with domain $\mathrm{dom}(\phi) \subseteq O_A$ a prefix closed collection of odd positions, that is, $\mathrm{dom}(\phi) = \{q \in O_A | \exists p \in \mathrm{dom}(\phi).\, q \preceq p\}$. We require that

(i) if $p \in \mathrm{dom}(\phi)$, then $p.\phi(p) \in P_A$,
(ii) if $p, q \in \mathrm{dom}(\phi)$ and $q \prec p$, then $q.\phi(q) \prec p$,
(iii) if $\phi(p) = a$ and $p.a.b \in O_A$ then $p.a.b \in \mathrm{dom}(\phi)$, and
(iv) if $p \in \widehat{T}$ and if $q \prec p$ odd implies $q.\phi(q) \preceq p$, then $|p| = W$.

The two notions of strategy[19] determine one another as in 2.1.1.

[19] Our strategies are now total and deterministic.

Exercises 11

1. *Show that one can replace the total condition in the tree definition of strategy by the condition $\widehat{S} \subseteq \widehat{T}$.*
2. *Suppose that S is a set of positive integers and t a positive integer. Consider the following game.*
Opponent starts by choosing a number $n_0 \in \mathbb{N}$ with $n_0 \geq t$; Player chooses $n_1 \in \mathbb{N}$ with $s_1 = (n_0 - n_1) \in S$; Opponent then chooses $n_2 \in \mathbb{N}$ with $s_2 = (n_1 - n_2) \in S$; and so on.
(i) Show that while the game is infinite, there are no infinite plays.
(ii) Who wins the game?
(iii) Vary the game by allowing $0 \in S$ but insisting that we never have $s_{k-1} = s_k$ (where conventionally $s_1 = 0$). What happens now?
3. *Consider the strategy for the fixed point operation $\mathbf{Y}$ from Section 2. This is a total strategy as is the identity strategy $1 : A \Rightarrow A$.*[20] *Now consider the interpretation of $\mathbf{Y}(1)$ which is the composite*

$$1 \xrightarrow{(1,\mathbf{Y})} (A \Rightarrow A) \times ((A \Rightarrow A) \Rightarrow A) \xrightarrow{\mathrm{ev}} A$$

(where ev is the evaluation map). What is this strategy?

3.1.2 Tensor product and linear function space

Inevitably conditions on wins must come into the definition of operations on games. We first give full definitions of the tensor product and linear function space of games.

Definition 3.3

The unit game I is defined by setting $M_I = \varnothing$; this determines the game tree which consists only of the empty sequence. (Thus I is a game in which no moves can be made.) There is a unique maximal position in I, which is automatically a win for Player.

Given games A and B, the games $A \otimes B$ and $A \multimap B$ are defined as follows.
Moves *The moves are $M_{A\otimes B} = M_{A\multimap B} = M_A + M_B$, the coproduct of M_A and M_B.*
Game tree *The game tree $T_{A\otimes B}$ is the subtree of $(M_{A\otimes B})^*$ consisting of those sequences p whose projections $p_A = p_{M_A}$ and $p_B = p_{M_B}$ preserve parity of moves and are in T_A, T_B, respectively.*
The game tree $T_{A\multimap B}$ is the subtree of $(M_{A\multimap B})^*$ consisting of those sequences p such that the projection $p_A = p_{M_A}$ reverses, while $p_B = p_{M_B}$ preserves parity of moves and the projections are in T_A, T_B, respectively.

[20] However $(A \Rightarrow A) \Rightarrow A$ is not a logical truth, though $A \Rightarrow A$ is one!

Winning positions *The (maximal) winning positions*[21] *in* $T_{A\otimes B}$ *are determined by*

$$|p| = W \text{ if and only if } |p_A| = W \text{ and } |p_B| = W .$$

The (maximal) winning positions in $T_{A\multimap B}$ *are determined by*

$$|p| = W \text{ if and only if } |p_A| = W \text{ implies } |p_B| = W .$$

Again it may be helpful to think of $A \multimap B$ as the result of playing a cogame $A^\perp$ in parallel with B. The cogame $A^\perp$ is A with the parity of moves reversed, and with outcomes W and L interchanged. So we write $p_{A^\perp}$ for the sequence p_A with the polarities of moves notionally reversed, and with $|p_{A^\perp}| = W$ if and only if $|p_A| = L$. Then in the definition of $A \multimap B$ we could say that $p_{A^\perp}$ and p_B preserve parity of moves and are in $T_{A^\perp}$, T_B respectively; and that $|p| = W$ if and only if $|p_{A^\perp}| = W$ or $|p_B| = W$.

Exercises 12

1. *Establish the following isomorphisms.*
   - $A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$.
   - $I \otimes A \cong A \cong A \otimes I$.
   - $A \otimes B \cong B \otimes A$.
   - $A \multimap (B \multimap C) \cong (A \otimes B) \multimap C$.

   (The fresh point is that the winning positions correspond.)

2. *(i) Show that a maximal play p in* $A \otimes B$ *need not have both its projections* p_A *and* p_B *maximal. Do the same for* $A \multimap B$.
   (ii) Suppose that p is a maximal position in $T_{A\otimes B}$, *but that* p_A *is not maximal in* T_A. *Show that* p_B *is maximal in* T_B, *and that* p_A *is even (so that necessarily* $|p_A| = W$*). If* p_B *is a finite play is it odd or even?*
   (iii) Suppose that p is a maximal position in $T_{A\multimap B}$, *but that* $p_{A^\perp}$ *is not maximal in* $T_{A^\perp}$. *Show that* p_B *is maximal in* T_B, *and that* $p_{A^\perp}$ *is odd (so that necessarily* $|p_{A^\perp}| = L$*). If* p_B *is a finite play is it odd or even? Repeat the other way round.*
   (iv) Deduce that the definition of winning conditions given above accords with the normal play convention.

3. *Show that one can reformulate the condition for P winning in* $A \otimes B$ *purely in terms of the notion of winning on maximal plays in A and B as follows: P wins a maximal play p in* $A \otimes B$ *if and only if P wins each* p_A, p_B *which is maximal.*

[21] It is worth noting that in a maximal position p in $T_{A\otimes B}$ (respectively in $T_{A\multimap B}$), at least one of p_A and p_B is maximal.

4. *Let S be the game with just one initial move, after which the game ends as a loss for Player. Give concrete descriptions of the games* $S \otimes S$ *and* $S \multimap S$. *Set* $\Sigma = (S \multimap S)$. *Give concrete descriptions of the games* $\Sigma \otimes \Sigma$ *and* $\Sigma \multimap \Sigma$.

5. *Let R be the game in which there is just one infinite maximal play which (regrettably) is a loss for P. Describe the maximal plays in* $R \otimes R$ *and* $R \multimap R$.

3.1.3 The category $\mathcal{LWG}$ of games: elementary structure

We adopt the definition of identity strategy and composition of strategies from 2.4; but since we now require winning strategies, there is something to prove.

Proposition 3.1 *For any game A, the strategy* ι_A *is a winning strategy. And if* $\sigma : A \multimap B$ *and* $\tau : B \multimap C$ *are winning strategies, then so is their composite* $\sigma;\tau$.

We indicate briefly how insisting on winning strategies ensures that composed strategies stay total.[22] The problem when composing $\sigma : A \multimap B$ and $\tau : B \multimap C$ is the possibility of making an infinite play on the scratchpad B with $B^\perp$, out of which we do not emerge to play a further move in the game $A \multimap C$. Let us see roughly why this cannot occur. Suppose for simplicity that neither of the resulting plays in $A^\perp$ or C is maximal. Then we know that the values of the positions in $A^\perp$ and in C are both L. But the value on the scratchpad will be

either L for $B^\perp$ and W for B, so the value of the play in $B \multimap C$ is L,

or W for $B^\perp$ and L for B, so the value of the play in $A \multimap B$ is L.

In the first case we have a play according to τ which is not winning, and in the second case we have a play according to σ which is not winning.

It is now straightforward to check that we do as before have the structure of a category; winning works out well.

Proposition 3.2 *Suppose that A and B are games with* $\sigma : A \multimap B$; *then* $\iota_A;\sigma = \sigma = \sigma;\iota_B$.
Suppose that A, B, C, and D are games with $\rho : A \multimap B$, $\sigma : B \multimap C$ *and* $\tau : C \multimap D$; *then* $\rho;(\sigma;\tau) = (\rho;\sigma);\tau$.

Hence we can define a category.

Definition 3.4 *The category* $\mathcal{LWG}$ *of* linear win-games *has as objects, win-games as defined in Definition 3.1; as maps from A to B, winning strategies in* $A \multimap B$ *as defined in Definition 3.2; and identities and composition as in Definition 2.4.*

[22] This is a bit misleading, as in a suitable formulation Proposition 3.1 is constructive.

We let $\mathcal{LWG}(A)$ denote the set of (total winning) strategies in the game A and $\mathcal{LWG}(A, B)$ the set of maps from A to B in $\mathcal{LWG}$. Then we have $\mathcal{LWG}(A, B) = \mathcal{LWG}(A \multimap B)$.

Now consider the elementary structure on $\mathcal{LWG}$. Naturally this parallels the corresponding structure for $\mathcal{LFG}$ in Section 2 closely.

Symmetric monoidal closed structure on $\mathcal{LWG}$ This is exactly as in the case of $\mathcal{LFG}$ from Section 2, and presents no difficulties.

Categorical products in $\mathcal{LWG}$ We now have to deal with winning and losing, but otherwise the definition is as in Section 2. So we just display the clauses concerned with winning.

Winning positions *The terminal game* 1 *is again the unit game* I, *with no moves and with the empty position winning for Player.*

For $p \in \widehat{T_{A\times B}}$, $|p|_{A\times B} = W$ *if and only if* $|p|_A = W$ *or* $|p|_B = W$.

It is easy to see that all this works just as before.

Proposition 3.3 *The object* 1 *is a terminal object in* $\mathcal{LWG}$.
For games A *and* B, *there are maps* **fst** $: A \times B \longrightarrow A$, **snd** $: A \times B \longrightarrow B$ *exhibiting* $A \times B$ *as a product in* $\mathcal{LFG}$.

We sum up what we have so far.

Theorem 3.4 *The category* $\mathcal{LWG}$ *of games and strategies is a* SMCC *with products.*

Enrichment in spaces Recall that the collection of all P-strategies $\mathcal{LFG}(A)$ of a fun game A forms a dI-domain. For a game to win A, there is a corresponding fun game $F(A)$ in which we forget about winning. Each $\mathcal{LWG}(A)$ is a subset of (the maximal elements in) $\mathcal{LFG}(F(A))$ and so inherits the structure of a topological space. The spaces which arise are rather special, but it is clear that all the hom-sets $\mathcal{LWG}(B, C)$ are naturally spaces of some kind. So we should consider how the operations of $\mathcal{LWG}$ as a structured category are reflected in maps of spaces. This has been set as an exercise.

Exercises 13

1. *Any win-game* A *can be thought of as a fun-game,* $F(A)$ *say, simply by forgetting about winning.*
   (i) Show that F *gives rise to a functor* $F : \mathcal{LWG} \longrightarrow \mathcal{LFG}$.
   (ii) Is the forgetful functor F *a monoidal functor?*
   (iii) Does the forgetful functor $F : \mathcal{LWG} \longrightarrow \mathcal{LFG}$ *preserve products?*
2. *Establish the following facts.*
   (i) The contravariant functor $(-) \multimap S$ *is self dual on* $\mathcal{G}$.
   (ii) The game $A \multimap S$ *can be identified with* $S(A)$, *say, the game which is obtained from* A *by adding a fresh initial move* • *say for* O, *and then letting play continue as in* $A^{\perp}$.
   (iii) The self duality is enriched in the closed structure:

$$A \multimap S(B) \cong S(A \otimes B) \cong B \multimap S(A)$$

3. *Does* $\mathcal{LWG}$ *have coproducts? Does it have weak coproducts?*
4. *Find an appealing monoidal closed category of spaces in which to enrich* $\mathcal{LWG}$.

3.2 A cartesian closed category of games

3.2.1 The exponential comonad and the CCC

Again we take the exponential $!A$ of a game A to be in effect an infinite ordered tensor product of (versions of) A. We now have to deal with winning, and for completeness give the full definition.

Definition 3.5 *Suppose that* A *is a game to win as above. Then the game* $!A$ *is defined as follows:*
Moves *The moves are* $M_{!A} = M_A \times \mathbf{N}$ *the countable copower of* M_A.
Game tree $T_{!A}$ *is the subtree of* $(M_{!A})^*$ *consisting of those sequences p such that (i) all the projections* p_k *are in* T_A, *and (ii) the first move in the* $k+1$*st copy is made after the first move in the* k*th.*
Winning plays *For a maximal play* $p \in \widehat{T_{!A}}$, *we set* $|p| = W$ *if and only if* $p_k = W$ *for all* k.

The monoidal structure $m_I : I \longrightarrow !I$ and $m_{AB} : !A \otimes !B \longrightarrow !(A \otimes B)$, the comonad structure $\varepsilon_A : !A \longrightarrow A$ and $\delta_A : !A \longrightarrow !!A$, and the operations $e_A : !A \longrightarrow I$ and $d_A : !A \longrightarrow !A \otimes !A$ of discard and duplication, can all be defined as in Section 2. One can check that all the strategies given there are in fact total winning ones. Thus the structure satisfies the same crucial properties as the corresponding structure on $\mathcal{LFG}$.

Proposition 3.5 *The components of the natural transformations* $e_A : !A \to I$ *and* $d_A : !A \to !A \otimes !A$ *give each free coalgebra* $\delta_A : !A \to !!A$ *the structure of a commutative comonoid. And this structure is preserved by coalgebra maps: whenever* $f : (!A, \delta_A) \to (!B, \delta_B)$ *is a coalgebra, then* f *is also a comonoid morphism.*

As before this means that there must be a cartesian closed category of games, and (as before) the existence of products means that we can concentrate on the Kleisli category.

Definition 3.6 *The category $\mathcal{WG}$ of win-games (without qualification) is the Kleisli category for the comonad ! on $\mathcal{LWG}$*

Again general considerations show what we want.

Proposition 3.6 *The category $\mathcal{WG}$ is cartesian closed. The categorical product is the product $A \times B$ and the function space $B \Rightarrow C$ is defined by*

$$B \Rightarrow C = !B \multimap C .$$

The standard adjunction induced by the comonad ! is a monoidal adjunction.

A concrete description of the CCC is as in the case of partial strategies, and now the hom-sets in $\mathcal{WG}$ are naturally spaces. We do not go into details.

Exercises 14

1. *(i) Show that a maximal play in $!A$ can involve starting only the first version A_0 of A and then playing a maximal play.*
(ii) Show that a maximal play in $!A$ can involve starting all of the ordered sequence of versions of A in term, but completing none of them.
(iii) Suppose that there is an infinite play in A. Show that there are infinite plays in $!A$ which involve finishing all the versions A_k with the chosen infinite play. (How many of them are there?)
2. *(i) Suppose that we have a play in $A \Rightarrow B$ where P infinitely often starts a new version of A, none of which are finished. Who wins?*
(ii) Suppose that we have a play in a game of the form

$$((A \Rightarrow B) \Rightarrow C) \Rightarrow D$$

in which (i) O opens in D, (ii) P replies by starting $C^\perp$, (iii) O responds by starting B, (iv) P responds by starting A and (v) thereafter P responds to every move of O by starting a fresh version of A. Who wins?
3. *Recall the game S with just an initial O-move. Show that the general and linear function spaces coincide: $S \Rightarrow S \cong S \multimap S$.*
4. *Show that the category $\mathcal{WG}$ is not extensional.*

3.2.2 $\mathcal{WG}$ as a model for constructive proofs

The category $\mathcal{WG}$ is a CCC in which not all objects have points, so prima facie it is a reasonable model of proof theory, and we ask how good it is.

Write $\Phi(\vec{p})$ and $\Psi(\vec{p})$ for propositional formulae in the $(\wedge, \Rightarrow)$-fragment of propositional logic with free propositional variables among the list $\vec{p} = p_1, \dots p_n$. Every interpretation in $\mathcal{WG}$ of $\vec{p} = p_1, \dots p_n$ as games $\vec{A} = A_1, \dots A_n$ gives rise to interpretations written $\Phi(\vec{A})$ and $\Psi(\vec{A})$ of Φ and Ψ as games. It is this interpretation which we wish to assess.

Ideally we would look for a 'full completeness' theorem in the sense of Abramsky and Jagadeesan (1994). This would take something like the following form.

Strong completeness for $\mathcal{C}$ Every sequence $\Phi(\vec{A}) \longrightarrow \Psi(\vec{A})$ of maps uniform in $\vec{A}$, is the interpretation of a unique (up to $\beta\eta$-equality) proof of $\Phi \vdash \Psi$. (Equivalently, it is the interpretation of a unique (up to $\beta\eta$-equality) λ-term.)

To make this precise in the case of $\mathcal{WG}$ would require us to elucidate a suitable sense of 'uniform' in terms of some form of polymorphism. Let us concern ourselves only with a weaker question.

Weak completeness for $\mathcal{C}$ If there is a map $\Phi(\vec{A}) \longrightarrow \Psi(\vec{A})$ in $\mathcal{C}$ for every $\vec{A}$, then $\Phi \vdash \Psi$ is provable in intuitionistic logic. (Equivalently, there is a λ-term of type $\Phi(\vec{p}) \longrightarrow \Psi(\vec{p})$.)

Let us consider the question of whether $\mathcal{WG}$ is weakly complete for the $(\wedge, \Rightarrow)$-fragment of intuitionistic logic. The answer seems a little delicate. We start by considering determined games, that is games in which one or other of the players has a winning strategy. Suppose that A and B are determined games. Then so is $A \Rightarrow B$, indeed the conditions are obvious.

P wins $A \Rightarrow B$ if and only if P wins B or P wins $A^\perp$.
O wins $A \Rightarrow B$ if and only if O wins B and O wins $A^\perp$.

It follows that if $\Phi \vdash \Psi$ in classical logic, then for all determined games $\vec{A}$ there is a map $\Phi(\vec{A}) \longrightarrow \Psi(\vec{A})$ in $\mathcal{WG}$. But the converse is equally trivial. Suppose that $\Phi \not\vdash \Psi$ in classical logic. Take an interpretation of $\vec{p}$ making Φ true and Ψ false; set

$$A_i = \begin{cases} I & \text{if } p_i \text{ is set true;} \\ S & \text{if } p_i \text{ is set false.} \end{cases}$$

I and S are determined and are wins for P and O respectively. Arguing inductively, we see that $\Phi(\vec{A})$ is a win for P, and $\Psi(\vec{A})$ is a win for O. Hence the game $\Phi(\vec{A}) \Rightarrow \Psi(\vec{A})$ is a win for O, and there is no map $\Phi(\vec{A}) \longrightarrow \Psi(\vec{A})$ in $\mathcal{WG}$.

If we restricted to games which are determined, then $\mathcal{WG}$ would be an excessively complicated model, weakly complete for classical logic. (The two-valued semantics seems a bit simpler.) But there are non-determined games (Mycielski 1964), and the general situation is not clear. Under the assumption of the Axiom of Choice, I believe that one can adapt and extend arguments from Blass (1972) to show the following conjecture.

Plausible Conjecture Assuming the Axiom of Choice, the category $\mathcal{WG}$ is weakly complete for intuitionistic conjunctive-implicational logic.

The dependence on set-theoretic combinatorics here is quite unsatisfactory, and it seems better to take uniformity seriously. That is not a topic for these notes, but see Abramsky (1997).

Exercises 15

1. *Check explicitly that some propositional formulae of your choice are not validated in the model $\mathcal{WG}$.*

2. *Show directly that the strategy Υ_A is not winning for some specific game A. Is there a game A for which Υ_A is a winning strategy?*

3. *Check the claim made in the discussion about winning $A \Rightarrow B$ when A and B are deterministic. (There are a couple of trivial points to make.)*

4. *Take the games $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ corresponding to the Pierce formula. Find winning strategies in the case of some specific games A and B. Can you detect a sense in which your strategies are not produced by a uniform algorithm?*

5. *(Blass 1972) Fix $\mathcal{U}$ a non-principal ultrafilter on $\mathbb{N}$. Consider the game A with $M_A = \mathbb{N}$ and $T_A = \{p \in \mathbb{N}^* | p_0 < p_1 < p_2 < \cdots\}$. For $p \in$ ... set $E(p) = \{n | p_{2k} \leq n, p_{2k+1}$ for some $k\}$.*
(i) Show that Opponent has a winning strategy for the game $A \otimes A$.
(ii) Recall $S(A) \cong (A \multimap S)$ from Exercises 13. Show that Opponent has a winning strategy for the game $S(A) \otimes S(A)$.
(iii) Deduce that the game A is not determined.

4 Dialogue Games

A feature of the main strategies considered in Section 2 is that a discipline of questions and answers ran through the resulting plays. This is obvious in the case of the games $\mathbb{N}$ and $\mathbb{B}$ in which an initial question • is answered by an appropriate value; but that is a trivial instance of a more general feature of all strategies denoting PCF-terms. In this part we explain in rough outline how making this intuition precise leads to a richer notion of game (a dialogue game) and to a restricted notion of strategy (an innocent strategy); and we indicate applications both to computation and to logic.

4.1 Categories of Dialogue Games

4.1.1 Moves in dialogue games

Questions and answers In a dialogue game, the moves are of four distinct kinds: Player's question which we represent generically as "(", Opponent's answer ")", Opponent's question"[" and Player's answer "]". The representation of questions and answers as left and right matching parentheses respectively reflects the following convention: Player's question can only be answered by Opponent, and *vice versa*. In addition every answer will be associated with a unique question. Questions need not be immediately answered; the immediate response to a question may either be an answer, or some further question.

A play in a dialogue game is required to satisfy the following basic condition.[23]

> **Principle of Pertinence** Whenever an answer occurs in a play, it answers the latest unanswered question.

Another way of putting this discipline of questions and answers is that questions pending in a dialogue are answered on a "last-asked-first-answered" basis. This has the following global consequence. If one looks at the pattern of brackets in any play of a game, it will be *potentially well-bracketed*, that is, the sequence can be extended to one in which the brackets match up in the standard way. Indeed, we may as well restrict ourselves to games in which the finite total plays are well bracketed, and in which any finite play extends to a finite total one.

Justification Some discipline is also maintained on the subsidiary questions which may be asked in response to a question (and answered before that question is answered). This is done by the use of a notion of explicit justification which can be thought of as providing a pointer from the given move (or the resulting position) to an earlier position. Restrictions on these pointers are given by the following convention.

> **Justification Convention** The justification for a P-question (must be an instance of some earlier O-question [which is not yet answered. The justification for an O-question [must be either the initial position of the game (so the question does not require justification), or else an instance of some earlier P-question (which is not yet answered. Answers are taken to be justified by the unique instance of the question which they answer.

Let us call pointers in accord with this convention *Justification Pointers*. It is natural for what follows to think of them as pointing from instances of moves to instances of moves. (When the pointer is to the initial position, think of it as pointing to the First Cause.)

Moves In the earlier parts of these notes it made sense to regard moves as tokens carrying no specific information. Where there is a structure of justification this is less satisfactory, and it is better to think of moves as carrying with them their justification history and possible futures[24]. However this effects nothing that we need worry about here and we omit the details.

[23] The Principle of Pertinence can be found in the established tradition in game semantics of intuitionistic logic, see e.g. Felscher's survey paper (Felscher 1986). I learnt of its importance for the theory of algorithms from Robin Gandy who invented it independently and called it the 'no-dangling-question-mark condition'.

[24] This is in line with a reading of dialogue games as involving 'menu-driven' computation.

4.1.2 Dialogue games and innocent strategies

Views Suppose now that we are given a play in some game which is a sequence of (explicitly justified) questions and answers satisfying the Principle of Pertinence and the Justification Convention. Call such a sequence well-formed. Suppose that P is to reply. We are going to require that P's strategy makes use only of some relevant part of this history: this is the *P-view* of the play.

Definition 4.1 *The* Player's view, *or* P-view, $\ulcorner p \urcorner$ *of a well-formed sequence p of moves is defined recursively. Let q range over well-formed sequences of moves, and r over segments of well-formed sequences.*

$$\begin{array}{lcll}
\ulcorner [\urcorner & \stackrel{\text{def}}{\equiv} & [& \textit{if "[" is initial,} \\
\ulcorner q \cdot (\cdot r \cdot [\urcorner & \stackrel{\text{def}}{\equiv} & \ulcorner q \urcorner \cdot (\cdot [& \textit{if "(" explicitly justifies "[",} \\
\ulcorner q \cdot) \urcorner & \stackrel{\text{def}}{\equiv} & \ulcorner q \urcorner \cdot) & \\
\ulcorner q \cdot [\cdot r \cdot] \urcorner & \stackrel{\text{def}}{\equiv} & \ulcorner q \urcorner & \textit{if "]" explicitly answers "[",} \\
\ulcorner q \cdot (\urcorner & \stackrel{\text{def}}{\equiv} & \ulcorner q \urcorner \cdot (. &
\end{array}$$

The P-view of a well-formed sequence of moves has the typical shape

$$[\cdot(\cdot)\cdot(\cdot)\cdot(\cdot[\cdot(\cdot)\cdot(\cdot)\cdot(\cdot[\cdot(\cdot)\cdots.$$

By construction, whenever there is a pattern "$(\cdot[$" in a P-view, the O-question "[" is explicitly justified by the P-question "(". Also there can be no segments of the form "$[\cdots]$" in a P-view. This may be read in the following two ways:

(i) Player simply ignores answers to questions posed by Opponent;

(ii) Player imagines that Opponent always answers questions directly.

There is a dual definition of *Opponent's view*, or *O-view*, $\llcorner p \lrcorner$ of a well-formed sequence p of moves:

$$\begin{array}{lcll}
\llcorner q \cdot [\cdot r \cdot (\lrcorner & \stackrel{\text{def}}{\equiv} & \llcorner q \lrcorner \cdot [\cdot (& \text{if "[" explicitly justifies "(",} \\
\llcorner q \cdot] \lrcorner & \stackrel{\text{def}}{\equiv} & \llcorner q \lrcorner \cdot] & \\
\llcorner q \cdot (\cdot r \cdot) \lrcorner & \stackrel{\text{def}}{\equiv} & \llcorner q \lrcorner & \text{if ")" explicitly answers "(",} \\
\llcorner q \cdot [\lrcorner & \stackrel{\text{def}}{\equiv} & \llcorner q \lrcorner \cdot [. &
\end{array}$$

The O-view of an empty sequence is the empty sequence. Since a well-formed sequence never begins with a P-question, we omit the case of $\llcorner (\lrcorner$. An O-view can never have a segment of the form $(\cdots)$: Opponent ignores answers to questions posed by Player. An O-view may, for example, have the shape

$$[\cdot(\cdot[\cdot]\cdot[\cdot]\cdot[\cdot(\cdot[\cdot]\cdot[\cdot(\cdots.$$

The following properties of P-view and O-view are easy to verify:

- If $q \cdot (\cdot r \cdot)$ is a well-formed sequence and if "(" explicitly justifies ")", then
$$\ulcorner q \cdot (\cdot r \cdot) \urcorner \quad = \quad \ulcorner q \urcorner \cdot (\cdot).$$
Dually if $q \cdot [\cdot r \cdot]$ is a well-formed sequence and if "[" explicitly justifies "]", then
$$\llcorner q \cdot [\cdot r \cdot] \lrcorner \quad = \quad \llcorner q \lrcorner \cdot [\cdot].$$
- If p is a well-formed sequence ending with an O-move (respectively P-move), then the last move of p is preserved by P-view (respectively O-view); that is to say, the last move of $\ulcorner p \urcorner$ (respectively $\llcorner p \lrcorner$) comes from the same last move of p.

Legal positions The idea is that moves in a P-strategy should be determined, not by the position itself, but by the P-view of the position. However there is a problem. In the above definitions we tacitly assume that the justification pointers take care of themselves. In fact they need not do so; the justifying move may disappear in the P-view. So we need a further condition on justification pointers.

Visibility Condition The visibility condition for a sequence q is as follows. For any initial subsequence $p \cdot ($ of the sequence q, the O-question "[" explicitly justifying the P-question "(" occurs in the P-view of p. Similarly for any initial subsequence $p \cdot [$ of the sequence q, the P-question explicitly justifying "[" occurs in the O-view of p.

Definition 4.2 *A play in a game of questions and answers is a* legal position *if it satisfies the Principle of Pertinence, the Justification Convention and the Visibility Condition.*

Clearly the set of legal positions is prefix closed. Furthermore for legal positions, the justification pointers do take care of themselves.

Proposition 4.1 *If p is a legal position, then both the O-view and the P-view of p are legal positions.*

Definition 4.3 *A* dialogue game for fun *or* dialogue fun game *consists of a non-empty tree (that is, prefix closed collection) of legal positions.*

Innocent strategies We are now in a position to describe a restricted notion of strategy. It is simplest to give this in function mode.

Definition 4.4 *Suppose that A is a simple dialogue game.*
A P-strategy (as in Section 2 but presented in function mode) $\sigma : O_A \longrightarrow M_A$ is an innocent strategy *if and only if whenever p and q are odd and $\ulcorner p \urcorner = \ulcorner q \urcorner$ then $\sigma(p) = \sigma(q)$ (in the usual sense that if one is defined then so is the other).*
The force of the definition is that an innocent strategy only makes use of the P-view. Hence it makes sense to regard the innocent strategy as being given by its restriction to P-views. Then a finite innocent strategy *is one whose restriction to P-views is a finite function.*

4.1.3 Categories of dialogue games

We now describe some categories of dialogue games. These come in two flavours: games for fun (with partial strategies) and games to win (with total strategies). We can follow the pattern already established in Sections 2 and 3. So first, there is a tensor product and linear function space for dialogue fun-games and for dialogue win-games. We can adopt the definition of identity strategy and composition of strategies from Definition 2.4; but, since we are now dealing with innocent strategies, there is something to prove. The basic (non-trivial) combinatorial fact is that innocent strategies compose.

Proposition 4.2 *For any dialogue game A, the strategy ι_A is innocent. And if $\sigma : A \multimap B$ and $\tau : B \multimap C$ are innocent, then so is their composite $\sigma;\tau$.*

As an immediate consequence, we have SMCCs of games.

Theorem 4.3 *There is a* SMCC *$\mathcal{LDFG}$ of linear dialogue fun-games, and a* SMCC *$\mathcal{LDWG}$ of linear dialogue win-games.*

In each case, there is a comonad $(!, \varepsilon, \delta)$ carrying a comonoid structure. Hence we get CCCs of games, and since we have products in the linear categories, these may be taken to be the Kleisli categories.

Theorem 4.4 *The Kleisli category of the comonad $(!, \varepsilon, \delta)$ on the* SMCC *$\mathcal{LDFG}$ of linear dialogue fun-games is a* CCC *$\mathcal{DFG}$ of dialogue fun-games. The Kleisli category of the comonad $(!, \varepsilon, \delta)$ on the* SMCC *$\mathcal{LDWG}$ of linear dialogue fun-games is a* CCC *$\mathcal{DWG}$ of dialogue win-games.*

A subcategory of $\mathcal{DFG}$ is described in detail in Hyland and Ong (1995) where it is used as the basis for the construction of an intensionally fully abstract model for PCF.[25]

Exercises 16

1. *Show that in any play satisfying the Principle of Pertinence the number of questions is always greater than or equal to the number of answers. (Recall that this is the simple algorithm for checking correct bracketing of expressions.)*
2. *Show that the operation of taking the P-view is idempotent. Similarly for taking the O-view.*
3. *Describe the sequences of the form $\ulcorner \llcorner p \lrcorner \urcorner$, and of the form $\llcorner \ulcorner p \urcorner \lrcorner$. What do you notice?*
4. *Show that what goes for questions goes for answers, that is, that the explicitly justifying question of every P-answer (respectively O-answer) in a legal position appears in the P-view (respectively O-view) of the legal position up to that point.*

[25] A model is intensionally fully abstract just when its observational or contextual quotient is fully abstract.

4.2 Dialogue Games and Logic

4.2.1 A Compactness Theorem for Strategies

In this section we identify a subcategory of the category $\mathcal{DWG}$ of dialogue win-games which is suited to modelling the proof theory of finitary logic.

Let us say that a game is *acyclic* just when there are no justification cycles, and *finitary* just when in addition, the number of questions and answers is finite. These are simple and plausible requirements on a game for finitary logic. We need an additional more subtle property.

Justice Suppose that p is an infinite play in a (finitary) game. We wish to catch the intuition that it is either P's fault or O's fault that the play has gone on so long; and that it is the one at fault who should lose. We do not go into the formal details, but give the basic idea.

> **Principle of Justice** Suppose that p is an infinite play in a game A, that Player asks questions justified by a specific instance of [infinitely often in such a way Player can see (from his view) that he asks infinitely often; then we may say that Player is *time-wasting*. Similarly Opponent may be time-wasting. However in a given play only one can be time wasting. We say that A *satisfies the principle of justice* if and only if for every infinite play p, if Player is time-wasting then Player loses, and similarly for Opponent.

Definition 4.5 *A dialogue win game is* just *if and only if it is finitary and satisfies the principle of justice.*

Proposition 4.5 *The collection of just games forms a full sub*CCC *$\mathcal{JG}$ of the* CCC *$\mathcal{DWG}$.*

For just games there is a finiteness or compactness theorem, which seems fundamental for a good theory of proofs.

Theorem 4.6 *(Compactness Theorem) All winning strategies in a just game are finite.*[26]

4.2.2 Categories of Games of Argument

Our intention now is to obtain an appealing model for proofs, while avoiding questions of polymorphism or uniformity.[27] The approach is quite intuitive in as much as it relies on an idea which is fundamental to the Lorenzen tradition.

[26] Recall that an innocent strategy is finite just when the partial function (on views) giving the strategy is finite.

[27] We focus here on games with total strategies, though a version involving partial strategies is possible.

Start by fixing a set C of claims or confessions. All definitions and results are parameterised over this fixed set (which is to be regarded as a set of propositional constants).

Definition 4.6 *A* game of argument *is a dialogue win-game whose questions are indexed by elements of C; and where every question has as unique answer its index from C.*
In a play in a game of argument, both players may answer or admit $c \in C$ (possibly many times). A play is good *if and only if Player never admits $c \in C$ before Opponent does.*

The notions of tensor product and of linear function space carry over to this variation on the notion of a dialogue game. While the general notion of strategy (see Definitions 2.2 and 3.2) should by now be sufficiently clear, we are interested in rather special total strategies.

Definition 4.7 *A total P-strategy in an argument is*

- innocent *so long as it only makes use of the P-view of a position;*
- good *so long as all plays in accord with it are good;*
- winning *so long as all plays in accord with it are wins for Player.*

We again adopt the definition of identity strategy and composition of strategies from Definition 2.4; but now we need to show that good winning innocent strategies compose. We have commented on innocence in Section 4.1.2, dealt with winning in Section 3.1.1 and fortunately goodness takes care of itself.

Proposition 4.7 *For any argument A, the strategy $\iota_A : A \multimap B$ is a good, winning, innocent strategy. And if $\sigma : A \multimap B$ and $\tau : B \multimap C$ are good, winning innocent strategies, then so is their composite $\sigma;\tau : A \multimap C$.*

As an immediate consequence, we have SMCCs of games.

Theorem 4.8 *There is a* SMCC *$\mathcal{LAG}$ of linear games of argument.*

Again there is a comonad $(!, \varepsilon, \delta)$ carrying a comonoid structure. Hence we get a CCC of games, and since we have products in the linear categories, this may be taken to be the Kleisli categories.

Theorem 4.9 *The Kleisli category of the comonad $(!, \varepsilon, \delta)$ on the* SMCC *$\mathcal{LAG}$ of linear linear games of argument is a* CCC *$\mathcal{AG}$ of games of argument.*

4.2.3 $\mathcal{AG}$ as a model for constructive proofs

Regard the collection C of answers in the arguments as a set of propositional constants (or type constants). An interpretation of the proof theory of the $(\wedge, \Rightarrow)$-fragment of intuitionistic logic (or equivalently of the corresponding typed λ-calculus) is given by an interpretation of the elements of C as arguments.

Definition 4.8 *The* canonical interpretation *is given by interpreting each $c \in C$ as the argument in which O has just one opening question which can (only) be immediately answered by c.*

Inspection of this interpretation motivates the definition of 'good play'. In such a complex play, P acts in a cautious fashion and maintains no proposition which O has not already conceded.
The canonical interpretation is weakly complete for intuitionistic logic.

Theorem 4.10 *Let Φ and Ψ be propositional formulae in $(\wedge, \Rightarrow)$-logic with constants from C. Suppose that in the canonical interpretation there is a map $\Phi \longrightarrow \Psi$. Then $\Phi \vdash \Psi$ is provable in intuitionistic logic, and (equivalently) there is a λ-term of type $\Phi \Rightarrow \Psi$.*

The canonical interpretation in $\mathcal{AG}$ is far from being strongly complete; but this could reasonably be regarded as a positive feature. We get a new CCC and with it a more generous notion of proof. Let us close by considering briefly how the category $\mathcal{AG}$ (based on C) is related to the free CCC (on objects from C), or equivalently to the simply typed lambda calculus under $\beta\eta$-equality (with base types from C). Suppose we vary the notion of a game of argument so that once an answer has been given, all the outstanding questions must be answered in order. Then as the answers are determined by the questions, they are effectively redundant, and there is simply an option to call a halt. This gives us yet another CCC $\mathcal{RAG}$ of restricted games of argument: the objects are those of $\mathcal{AG}$, but the strategies are restricted. A restricted strategy can be read straightforwardly as a strategy, and so the CCC $\mathcal{RAG}$ embeds in the CCC $\mathcal{AG}$. This is hardly surprising in view of the following result.

Theorem 4.11 *The* CCC *$\mathcal{RAG}$ based on C is a free* CCC *generated by the set of objects C.*

This is a form of strong completeness: the proof is a simplified version of the definability result in Hyland and Ong (1995). Closely related ideas are in Felscher (1986), and Herbelin has independently made essentially the same observation from a somewhat different point of view.

Exercises 17

1. *Show that a play p in $A \otimes B$ can be good without its projection p_A being good.*

2. *Any argument A can be thought of as a win-game $W(A)$. Show that W extends to a functor $\mathcal{LA} \longrightarrow \mathcal{LWG}$. Is W a monoidal functor?*

3. *Test Theorem 4.10, by checking that, in the canonical interpretation, the Pierce formula does not hold.*

4. *How many maps $A \times A \longrightarrow A$ can you find in $\mathcal{AG}$?*

5. *Investigate further the relation between $\mathcal{AG}$ and $\mathcal{RAG}$.*

A Appendix: Monoidal Categories

The standard definitions of symmetric monoidal closed category, of monoidal functor and of monoidal natural transformation are as follows.

A *monoidal category* is a category $\mathcal{V}$ equipped with a functor $\otimes : \mathcal{V} \times \mathcal{V} \longrightarrow \mathcal{V}$, an object I of $\mathcal{V}$, and natural isomorphisms

$$a_{UVW} : (U \otimes V) \otimes W \longrightarrow U \otimes (V \otimes W),$$

$$l_U : I \otimes U \longrightarrow U \text{ and } r_U : U \otimes I \longrightarrow U,$$

such that the *coherence diagrams*

$$\begin{tikzcd}
((U \otimes V) \otimes W) \otimes X \arrow[r, "a"] \arrow[d, "a \otimes 1"'] & (U \otimes V) \otimes (W \otimes X) \arrow[r, "a"] & U \otimes (V \otimes (W \otimes X)) \\
(U \otimes (V \otimes W)) \otimes X \arrow[rr, "a"'] & & U \otimes ((V \otimes W) \otimes X) \arrow[u, "1 \otimes a"']
\end{tikzcd}$$

$$\begin{tikzcd}
(U \otimes I) \otimes V \arrow[rr, "a"] \arrow[dr, "r \otimes 1"'] & & U \otimes (I \otimes V) \arrow[dl, "1 \otimes l"] \\
& U \otimes V &
\end{tikzcd}$$

commute.

A *symmetry* for a monoidal category is a natural isomorphism

$$c_{UV} : U \otimes V \longrightarrow V \otimes U$$

with $c^2 = 1$, and such that the *coherence diagrams*

$$\begin{tikzcd}
(U \otimes V) \otimes W \arrow[r, "a"] \arrow[d, "c \otimes 1"'] & U \otimes (V \otimes W) \arrow[r, "c"] & (V \otimes W) \otimes U \arrow[d, "a"] \\
(V \otimes U) \otimes W \arrow[r, "a"'] & V \otimes (U \otimes W) \arrow[r, "1 \otimes c"'] & V \otimes (W \otimes U)
\end{tikzcd}$$

$$\begin{tikzcd}
U \otimes I \arrow[rr, "c"] \arrow[dr, "r"'] & & I \otimes U \arrow[dl, "l"] \\
& U &
\end{tikzcd}$$

commute.

A *closed structure* on a (symmetric) monoidal category is given by a bifunctor $[-,-] : \mathcal{V}^{op} \times \mathcal{V} \rightarrow \mathcal{V}$ together with an isomorphism

$$\mathcal{V}(U \otimes V, W) \cong \mathcal{V}(U, [V, W])$$

natural in U, V and W.

A *symmetric monoidal closed category* (SMCC) is a monoidal category equipped with a symmetry and a closed structure.

Suppose that $\mathcal{U}$ and $\mathcal{V}$ are SMCCs. (We shall not trouble to distinguish between the respective structures on the categories.) A *symmetric monoidal functor* is a functor $F : \mathcal{U} \longrightarrow \mathcal{V}$ equipped with *mediating natural transformations*

$$m_I : I \longrightarrow F(I)$$

$$m_{UV} : F(U) \otimes F(V) \longrightarrow F(U \otimes V)$$

such that the diagrams

$$\begin{tikzcd}
FI \otimes FU \arrow[r, "m_{IU}"] & F(I \otimes U) \arrow[d, "F(l_U)"] \\
I \otimes FU \arrow[u, "m_I \otimes 1_{FU}"] \arrow[r, "l_{FU}"'] & FU
\end{tikzcd}
\qquad
\begin{tikzcd}
FU \otimes FI \arrow[r, "m_{UI}"] & F(U \otimes I) \arrow[d, "F(\rho_A)"] \\
FU \otimes I \arrow[u, "1_{FU} \otimes m_I"] \arrow[r, "\rho_{FU}"'] & FU
\end{tikzcd}$$

$$\begin{tikzcd}
(FU \otimes FV) \otimes FW \arrow[r, "m_{UV} \otimes 1_{FW}"] & F(U \otimes V) \otimes FW \arrow[r, "m_{(U \otimes V)W}"] & F((U \otimes V) \otimes W) \\
FU \otimes (FV \otimes FW) \arrow[u, "a"] \arrow[r, "1_{FU} \otimes m_{VW}"'] & FU \otimes F(V \otimes W) \arrow[r, "m_{U(V \otimes W)}"'] & F(U \otimes (V \otimes W)) \arrow[u, "Fa"']
\end{tikzcd}$$

$$\begin{tikzcd} FU \otimes FV \arrow[r, "m_{UV}"] \arrow[d, "c_{UV}"'] & F(U \otimes V) \arrow[d, "F(c_{UV})"] \\ FV \otimes FU \arrow[r, "m_{VU}"'] & F(V \otimes U) \end{tikzcd}$$

commute.

Suppose that $F, G : \mathcal{U} \longrightarrow \mathcal{V}$ are symmetric monoidal functors. (Again we do not trouble to distinguish between the associated structures.) A *monoidal natural transformation* $\alpha : F \longrightarrow G$ is a natural transformation such that the diagrams

$$\begin{tikzcd} & I \arrow[dl, "m_I"'] \arrow[dr, "m_I"] & \\ FI \arrow[rr, "\alpha_I"'] & & GI \end{tikzcd} \qquad \begin{tikzcd} FU \otimes FV \arrow[r, "\alpha_U \otimes \alpha_V"] \arrow[d, "m_{UV}"'] & GU \otimes GV \arrow[d, "m_{UV}"] \\ F(U \otimes V) \arrow[r, "\alpha_{U \otimes V}"'] & G(U \otimes V) \end{tikzcd}$$

commute.

It is straightforward to compose symmetric monoidal functors and monoidal natural transformations.

Proposition SMCCs, *symmetric monoidal functors and monoidal natural transformations form a 2-category.*

A *monoidal comonad* on a SMCC $\mathcal{G}$ consists of a symmetric monoidal functor $! : \mathcal{G} \to \mathcal{G}$ (equipped with mediating natural transformations) together with monoidal natural transformations $\varepsilon : ! \longrightarrow 1_{\mathcal{G}}$ and $\delta : ! \longrightarrow !!$ which give a comonad on $\mathcal{G}$. (Thus a monoidal comonad is just a comonad in the 2-category of SMCCs.)

Definition A.1 *A* Linear category *is a* SMCC $\mathcal{C}$, *together with a monoidal comonad* $(!, \varepsilon, \delta, m_{AB}, m_I)$ *on* $\mathcal{C}$, *which is equipped with monoidal natural transformations with components* $e_A : !A \to I$ *and* $d_A : !A \to !A \otimes !A$ *which give each free coalgebra* $\delta_A : !A \to !!A$ *the structure of a commutative comonoid, this structure being preserved by coalgebra morphisms between free coalgebras.*

The condition that $(!A, d_A, e_A)$ forms a commutative comonoid means that the following three diagrams commute.

$$\begin{tikzcd} & !A \arrow[dl, "r^{-1}"'] \arrow[d, "d_A"] \arrow[dr, "l^{-1}"] & \\ !A \otimes I & !A \otimes !A \arrow[l, "1_{!A} \otimes e_A"] \arrow[r, "e_A \otimes 1_{!A}"'] & I \otimes !A \end{tikzcd} \qquad \begin{tikzcd} !A \arrow[r, "d_A"] \arrow[dr, "d_A"'] & !A \otimes !A \arrow[d, "c_{!A\,!A}"] \\ & !A \otimes !A \end{tikzcd}$$

$$\begin{tikzcd} !A \arrow[rr, "d_A"] \arrow[d, "d_A"'] & & !A \otimes !A \arrow[d, "1_{!A} \otimes d_A"] \\ !A \otimes !A \arrow[r, "d_A \otimes 1_{!A}"'] & (!A \otimes !A) \otimes !A \arrow[r, "\alpha_{!A\,!A\,!A}"'] & !A \otimes (!A \otimes !A) \end{tikzcd}$$

In addition we require that e_A and d_A be coalgebra morphisms, that is,that the following diagram commutes.

$$\begin{tikzcd} I \arrow[rr, "m_I"] & & !I \\ !A \arrow[u, "e_A"] \arrow[rr, "\delta_A"] \arrow[d, "d_A"'] & & !!A \arrow[u, "!e_A"'] \arrow[d, "!d_A"] \\ !A \otimes !A \arrow[r, "\delta_A \otimes \delta_A"'] & !!A \otimes !!A \arrow[r, "m_{!A\,!A}"'] & !(!A \otimes !A) \end{tikzcd}$$

Finally all coalgebra morphisms between (free) coalgebras are also comonoid morphisms: if $f : !A \longrightarrow !B$ is a coalgebra morphism, then it is also a comonoid morphism between the comonoids $(!A, e_A, d_A)$ and $(!B, e_B, d_B)$, i.e. it makes the following diagram commute.

$$\begin{tikzcd} & !A \arrow[r, "d_A"] \arrow[dl, "e_A"'] \arrow[dd, "f"] & !A \otimes !A \arrow[dd, "f \otimes f"] \\ I & & \\ & !B \arrow[ul, "e_B"] \arrow[r, "d_B"'] & !B \otimes !B \end{tikzcd}$$

These conditions were introduced in Benton, Bierman, de Paiva, and Hyland (1992), and their consequences were closely studied in Bierman (1993). The crucial point is the following.

Theorem *A Linear category, $\mathcal{C}$, is a categorical model for (intuitionistic) multiplicative exponential linear logic.*

Things simplify markedly in the presence of product, as first noticed by Seely, and further analysed by Bierman (1995).

Definition A.2 *A* new-Seely category *is a* SMCC *with finite products, $\mathcal{C}$, together with a comonad,* $(!, \varepsilon, \delta)$ *and natural isomorphisms,* $n_I : I \longrightarrow !1$ *and* $n_{AB} : !A \otimes !B \longrightarrow !(A \times B)$, *such that the adjunction,* $\langle F, G, \eta, \varepsilon \rangle$, *between $\mathcal{C}$ and the (co-)Kleisli category $\mathcal{C}_!$ is a* monoidal *adjunction.*

In the presence of products, a Linear Category will be a new-Seely category, and we shall have the following basic result.

Theorem (Seely, Bierman) *Given a Linear Category with products, $\mathcal{C}$, the co-Kleisli category $\mathcal{C}_!$ is cartesian closed and the adjunction between $\mathcal{C}$ and $\mathcal{C}_!$ is a* monoidal *adjunction.*

B Appendix: PCF

PCF is a typed programming language. Types of the language are Church's simple types (Church 1940) also referred to as PCF-types. They are defined as follows:

A	$::=$	ι	natural numbers
	$\mid$	o	booleans
	$\mid$	$A \Rightarrow A$	arrow or function type.

Let the meta-variable β range over ground types ι and o. As usual $\Rightarrow$ associates to the right: $A_1 \Rightarrow A_2 \Rightarrow A_3$ is read as $A_1 \Rightarrow (A_2 \Rightarrow A_3)$. Each simple type can be uniquely expressed as $A_1 \Rightarrow A_2 \cdots \Rightarrow A_n \Rightarrow \beta$ $(n \geq 0)$; in the traditional notation of type theory this is abbreviated as $(A_1, \cdots, A_n, \beta)$. For example the type $((\iota \Rightarrow \iota) \Rightarrow \iota \Rightarrow \iota) \Rightarrow \iota \Rightarrow \iota$ is abbreviated as $(((\iota, \iota), \iota, \iota), \iota, \iota)$.

For each type A, fix a denumerable set of variables. Raw PCF-terms are defined by the following grammar:

s	$::=$	Ω^A	undefined term
	$\mid$	c^A	constant
	$\mid$	x	variable
	$\mid$	$(s \cdot s)$	application
	$\mid$	$(\lambda x : A.s)$	abstraction
	$\mid$	$\mathbf{Y}^A(s)$	general recursive term, or Y-term;

where c^A ranges over the basic arithmetic constants. Type information is omitted where irrelevant. The application $(s \cdot t)$ is written st, and application associates to the left: $st_1 \cdots t_n$ abbreviates $(\cdots((st_1)t_2)\cdots t_n)$. The phrase $s : A$ means that the type of the term s is A, derived according to the following rules:

$$\Omega^A : A \qquad\qquad c^A : A$$

$$\frac{s : A \Rightarrow A}{\mathbf{Y}^A(s) : A} \qquad \frac{s : A_1 \Rightarrow A_2 \qquad t : A_1}{(s \cdot t) : A_2} \qquad \frac{s : A_2}{(\lambda x : A_1.s) : A_1 \Rightarrow A_2}$$

The basic arithmetic constants are as follows:

n	:	ι	numerals, for each natural number $n \geqslant 0$
t, f	:	o	booleans: truth and falsity
succ	:	$\iota \Rightarrow \iota$	successor
pred	:	$\iota \Rightarrow \iota$	predecessor
zero?	:	$\iota \Rightarrow o$	test for zero
cond^ι	:	$o \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota$	natural number conditional
cond^o	:	$o \Rightarrow o \Rightarrow o \Rightarrow o$	boolean conditional.

The notion of free and bound variables is completely standard; a closed term is a term without any free variables. Term substitution is written $s[t/x]$.

Operational Semantics *Programs* of PCF are closed terms of ground type. Values are λ-abstractions and constants of ground type; the meta-variable v ranges over values. Following the function paradigm, to compute a program in PCF is to evaluate it. The operational semantics of PCF is given in terms of a Martin-Löf style evaluation relation: $s \Downarrow v$ meaning "the closed term s evaluates to the value v".

$$v \Downarrow v \qquad \frac{u[t/x] \Downarrow v}{(\lambda x.u)t \Downarrow v} \qquad \frac{s \Downarrow v \quad vt \Downarrow v'}{st \Downarrow v'}$$

$$\frac{s \Downarrow \mathsf{t} \qquad u \Downarrow v}{\mathsf{cond}^\beta suu' \Downarrow v} \qquad \frac{s \Downarrow \mathsf{f} \qquad u' \Downarrow v}{\mathsf{cond}^\beta suu' \Downarrow v}$$

$$\frac{s\mathbf{Y}^A(s) \Downarrow v}{\mathbf{Y}^A(s) \Downarrow v}$$

$$\frac{s \Downarrow n}{\mathsf{succ}\, s \Downarrow n+1} \qquad \frac{s \Downarrow n+1}{\mathsf{pred}\, s \Downarrow n} \qquad \frac{s \Downarrow 0}{\mathsf{pred}\, s \Downarrow 0}$$

$$\frac{s \Downarrow 0}{\mathsf{zero?}\, s \Downarrow \mathsf{t}} \qquad \frac{s \Downarrow n+1}{\mathsf{zero?}\, s \Downarrow \mathsf{f}}$$

The type theory PCF The operational semantics for PCF reflects an intuitive understanding of the meaning of the terms. In this view the reductions are justified as the replacement of a term by an equal term. Thus the intuitive semantics can be given expression in an equational theory. In the case of PCF this amounts to a type theory related to Scott's original formulation. A (core) type theory for PCF $\mathbb{T}$ is given as follows. Take the typing rules already given, and define a relation $s = t$ on typed terms (in context) by taking, in addition to the usual rules for equality, the following:

$$\boxed{\begin{array}{c} (\lambda x : A.s)t = s[t/x] \qquad \lambda x : A.sx = s \quad (\text{if } x \text{ not free in } s) \\ \mathsf{cond}^{\beta}\mathsf{t}st = s \qquad \mathsf{cond}^{\beta}\mathsf{f}st = t \\ s(\mathrm{Y}^A(s)) = \mathrm{Y}^A(s) \\ \mathsf{succ}\ n = n+1 \qquad \mathsf{pred}\ n+1 = n \qquad \mathsf{pred}\ 0 = 0 \\ \mathsf{zero?}0 = \mathsf{t} \qquad \mathsf{zero?}n+1 = \mathsf{f} \end{array}}$$

It is important that there be a good relation between the reduction relation $\Downarrow$ of the operational semantics and the equality of the type theory. This is given by the following result.

Proposition *For any programs s and t, if s and t are equal in the type theory* $\mathbb{T}$ *then for any ground value* v $s \Downarrow v \iff t \Downarrow v$.

What is commonly called a denotational semantics for PCF is essentially some kind of interpretation of (model for) the type theory which we have just introduced. The usual form of a model for PCF is that the types are interpreted as domains and the terms as continuous (or stable continuous) maps between domains. The major concern in modelling programming languages is with issues of 'adequacy' and 'full abstraction'. For a survey of the famous full abstraction problem for PCF see Ong (1986). Game theoretic solutions to the problem are offered in Abramsky, Jagadeesan, and Malacaria (1994) Abramsky, Jagadeesan, and Malacaria (1995) and in Hyland and Ong (1995). For other recent approaches to sequentiality see Brookes and Geva (1992), Bucciarelli and Ehrhard (1994), Plotkin and Winskel (1994) and Winskel (1994).

References

Abramsky, S. (1997). Semantics of interaction. In this volume, pp 1–32.

Abramsky, S. and R. Jagadeesan (1994). Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic 59*, 543–574.

Abramsky, S., R. Jagadeesan, and P. Malacaria (1994). Full abstraction for PCF (extended abstract). In *Theoretical Aspects of Computer Software: TACS'94*, Volume 789 of *Lecture Notes in Computer Science*, pp. 1–15. Springer-Verlag.

Abramsky, S., R. Jagadeesan, and P. Malacaria (1995). Games and full abstraction of PCF. Submitted to Information and Computation.

Benton, P. N., G. M. Bierman, V. C. V. de Paiva, and J. M. E. Hyland (1992). Term assignment for intuitionistic linear logic. Technical Report 262, Computer Laboratory, University of Cambridge.

Berry, G. and P.-L. Curien (1982). Sequential algorithms on concrete data structures. *Theoretical Computer Science 20*, 265–321.

Bierman, G. M. (1993). *On Intuitionistic Linear Logic*. Ph. D. thesis, Computer Laboratory, Cambridge University.

Bierman, G. M. (1995). What is a categorical model of linear logic? In *Proceedings of Second International Conference on Typed Lambda Calculi and Applications*, Volume 902 of *Lecture Notes in Computer Science*, pp. 78–93. North-Holland.

Blass, A. (1972). Degrees of indeterminacy of games. *Fundamenta Mathematica 77*, 1972.

Blass, A. (1992). A game semantics for linear logic. *Annals of Pure and Applied Logic 56*, 183–220.

Brookes, S. and S. Geva (1992). Stable and sequential functions on scott domains. Technical Report CMU-CS-92-121, School of Computer Science, Carnegie-Mellon Univeristy.

Bucciarelli, A. and T. Ehrhard (1994). Sequentiality in an extensional framework. *Information and Computation 110*, 265–296.

Cartwright, R., P.-L. Curien, and M. Felleisen (1994). Fully abstract semantics for observably sequential languages. *Information and Computation 111*, 297–401.

Church, A. (1940). A formulation of a simple theory of types. *Journal of Symbolic Logic 5*, 56–68.

Conway, J. H. (1976). *On Numbers and Games*. Academic Press.

Coquand, T., C. A. Gunter, and G. Winskel (1989). Domain theoretic models of polymorphism. *Information and Computation 81*, 123–167.

Crole, R. L. (1993). *Categories for Types*. Cambridge University Press.

Curien, P.-L. (1993). *Categorical Combinators, Sequential Algorithms and Sequential Functions*. Birkhauser.

Felscher, W. (1986). Dialogues as a foundation for intuitionist logic. In D. Gabbay and F. Guenther (Eds.), *Handbook of Philosophical Logic*, Volume III, pp. 341–372. D. Reidel.

Freyd, P. J. and A. Scedrov (1990). *Categories and Allegories*. North-Holland.

Girard, J.-Y. (1986). The system **f** of variable types, fifteen years later. *Theoretical Computer Science 45*, 159–192.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science 50*, 1–102.

Girard, J.-Y. (1989a). Geometry of interaction i: interpretation of system f. In Ferro, Bonotto, Valantini, and Zanardo (Eds.), *Proceedings of Logic Colloquium '88*, pp. 221–260. North-Holland.

Girard, J.-Y. (1989b). *Proofs and Types*. Cambridge University Press.

Hyland, J. M. E. and C.-H. L. Ong (1995). On full abstraction for PCF: I, II and III. Submitted to Information and Computation.

Kahn, G. and G. D. Plotkin (1978). Domaines concrets. Technical Report 336, INRIA-LABORIA.

Kahn, G. and G. D. Plotkin (1993). Concrete domains. *Theoretical Computer Science 121*, 187–277.

Kelly, G. M. (1982). *Basic Concepts of Enriched Category Theory*. LMS Lecture Notes. Cambridge University Press.

Kelly, G. M. and M. L. Laplaza (1980). Coherence for compact closed categories. *Journal of Pure and Applied Algebra 19*, 193–213.

Lambek, J. and P. J. Scott (1986). *Introduction to Higher Order Categorical Logic*. Cambridge University Press.

Lorenzen, P. and K. Lorenz (1978). *Dialogische Logik*. Wissenschaft Buchgesellschaft.

Mycielski, J. (1964). On the axiom of determinateness. *Fundamenta Mathematica 53*, 205–224.

Ong, C.-H. L. (1986). Correspondence between operational and denotational semntics. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum (Eds.), *Handbook of Logic in Computer Science*, Volume 4, pp. 341–372. Oxford University Press.

Plotkin, G. D. (1977). LCF as a programming language. *Theoretical Computer Science 5*, 223–255.

Plotkin, G. D. and G. Winskel (1994). Bistructures, bidomains and linear logic. Technical Report RS-94-9, BRICS.

Taylor, P. (1986). *Recursive Domains, Indexed Category Theory and Polymorphism*. Ph. D. thesis, DPMMS, Cambridge University.

Winskel, G. (1994). Stable bistructure models of PCF. Technical Report RS-94-13, BRICS.