Games for Semantics
An Introduction
DRAFT

Andrea Schalk
Department of Computer Science
University of Manchester

March 12, 2001

Abstract

These are draft notes for an introductory (mini-)course on games in semantics which I gave to graduate students in Manchester in the spring of 2001. Any feedback is highly welcome, in particular regarding the exercises and/or ways of improving the exposition.

1 What is a game?

Games have become very successful in the semantics of both, logics and programming languages. There notion of 'game' as such is very wide-ranging even in the narrow context of semantics, reaching from dialogue games for classical logic (see [Fel86] for an overview) to ever more sophisticated versions of so-called 'games' for semantics, see for example [AJ94, AJM, AJM94, AM95, AM97, AM99, Abr97, BDER97, BDER98, Bla92, Bla95, HO, HO93, Hyl97, LS91, Lai97, McC96].

All the games used for semantics have a few properties in common. A game is understood to consist of rules determining how various entities, the *players*, may interact. At any point in time it can only be one player's turn to take an action, or *make a move*.

All the games mentioned above have two players. In the case of games for logic we may think of one of them as a 'prover' and the other as a 'skeptic'. If the formula to be proven is, say, a conjunction then the skeptic may insist on having either of the two conjuncts proven. If, on the other hand, it is a disjunction then the prover may choose to just prove one of the sub-formulae. A similar case can be made for universal versus existential quantification. We will not, however, cover approaches using games commonly used in model theory such as Ehrenfeucht games or evaluation games. In the context of a semantics for a programming language the two players can be thought of as 'the program' and 'the environment'. No program runs without having been started, and it may require the environment to deliver, for example, values for its arguments. We usually refer to the two players as *Player* and *Opponent*—the choice of terms indicates who we see ourselves sympathize with.

We will give a detailed description of a category of games that can be understood as the basis for the games used in the literature cited above. We will concentrate on its structure as a model of (fragments of) linear logic, and the various notions of such models are developed in the Appendix. Using standard machinery that will give us a cartesian closed category which allows interpretation of versions of the simply typed λ-calculus. On the way we stress the flexibility regarding the presentation and give some hints regarding how one might arrive at a more abstract notion of 'game'. One side-route exhibits why the category with the 'obvious' dualization cannot be made $*$-autonomous, following some work by A. Blass [Bla92]. This is meant to be a gentle introduction and as such proceeds at a pace that is probably too slow for somebody who has had exposure to these sorts of games before. A number of exercises are included that are designed to help explore the various ideas introduced. Not all exercises will be suitable for all readers since some assume background knowledge in areas such as category theory or domain theory.

1.1 Game trees

Games are usually thought of as being given by their so-called *game trees*. These are intuitively understandable. Usually there is a unique starting position, in which it is somebody's turn. We say that that player *starts the game*. The branches of the tree from that initial position show the choice of moves the starting player has, and the nodes they lead to signify the position the game is in once the corresponding move has been made. In games for logic it is not necessarily the case that the two players move strictly alternately, the way we are used to it from board games such as chess and Go. For our purposes, however, we find it useful to have a more structured environment where this condition is imposed.

In the example below, the starting player has only one available move. Once that move has been carried out it will be the other player's turn, and in the example there are three choices for this second move. This naturally gives us a notion of *position* and *move*. A *play* consists of a (non-empty) path starting at the root. Hence such plays are partial—the game might be continued where the play ends.

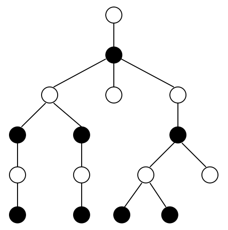

Figure 1: A game

In the example we have coloured the nodes (or positions) black or white. This shows us whose turn it is in that position. We adopt the convention that positions where it is Player's turn will be coloured black, and those where it is Opponent's turn will be coloured white. Because the two players move strictly alternately that means that a white Position is one reached by a Player move, and we call those Player or P-positions. A black position is one reached by an Opponent move, and we call those Opponent or O-positions. (It is worth noticing that different authors have different conventions.) We use A^P to denote the set of P-positions in a given game A, and A^O to refer to the set of O-positions in the same game. We use $*_A$ to refer to the initial position in A

For all our games we demand that Opponent always starts. The reason for this will become apparent in Section 4. Hence it has to be the case that $*_A \in A^P$. Thus we think of the game tree as in Figure 2.

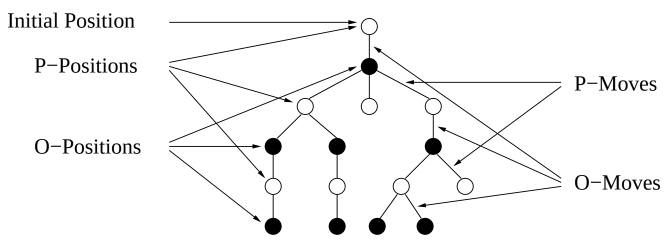

Figure 2: A game

Note that game trees are traditionally drawn as growing *downwards* from the root, no matter what biologists think of that. Because the colour coding gives us all the information on the labels we thus attach to positions (nodes) and moves (edges) we do not have to make them explicit when we draw a game tree. The conventions regarding Player and Opponent are summarized in the following table:

	○	●
colour	white	black
alternative 'colour'	unfilled	filled
kind of position	Player position	Opponent position
belongs to which set	element of A^P	element of A^O
what has happened last	Player has just moved	Opponent has just moved
what will happen next	Opponent to move next	Player to move next

We assume that every node in the game tree has a finite height, but we do allow infinite trees in two ways: We do not assume that our trees are finite branching, nor do we assume that all paths through the tree are finite. Hence our only finiteness requirement is a well-foundedness condition.

1.2 Representations

When it comes to a mathematical formalization of game trees two choice can be made. A popular one is to assume the existence of a set of *moves* M_A. The game tree can then be described as a prefix-closed subset of the set of words over the alphabet M_A. A position is then naturally labelled by a finite sequence of moves, namely the *play* that led to this particular position. The initial position will be labelled by the empty sequence. All the plays of even length will end in a position of the same colour (black or white, that is, Player or Opponent), and the same is true for plays of odd length—they will end in positions of the remaining colour. It has the nice property that there is a play leading from one given position to another if and only if the former is a prefix of the latter. Similarly, there is a move from one position to another if and only if the latter extends the former by precisely one move ('letter' in our alphabet M_A).

Definition 1 Alternative 1. *Let M be a set; we call the elements of M* moves*. A* game *is a prefixed-closed set of strings over M.*

This is a simple setting and quite sufficient for many purposes. It is how people often describe games such as chess or Go: A move is coded as a description of what occurs without any reference to the current position on the board, and the game tree can be created from rules describing whether a given move is legal at any given time. It is quite easy to describe rules in this way even for games whose game tree is huge: they concentrate on the change that occurs when going from one (legal) position to another. Additionally this easily allows restrictions to the notion of 'strategy' (see Exercise 1.(6)). One may want to be slightly more discriminating and decree that some moves are only for Player and others only for Opponent to play. This gives rise to our first alternative definition.

Definition 1 Alternative 2. *Let M^P and M^O be two sets. A* game *is a prefix-closed set of strings $m_1 \cdots m_n$ over $M^P + M^O$ such that*

- *$m_1 \in M^O$;*
- *for $1 \leq i < n$ $m_i \in M^P$ implies $m_{i+1} \in M^O$;*
- *for $1 \leq i < n$ $m_i \in M^O$ implies $m_{i+1} \in M^P$.*

We call the elements of M^P moves for Player *and those of M^O* moves for Opponent.

For games given in this way, a *play* is given by an element of the game, that is a valid sequence of moves.

Starting with just a game tree, however, it is not obvious how to come up with a natural set of moves so that the game tree can be described through them. Obviously, different choices for labels have to be made for arcs starting from the same node, but other than that, repetition is certainly optional. In fact, for a given game tree there may be many ways of representing it in this way; examples can be found in Figure 3.

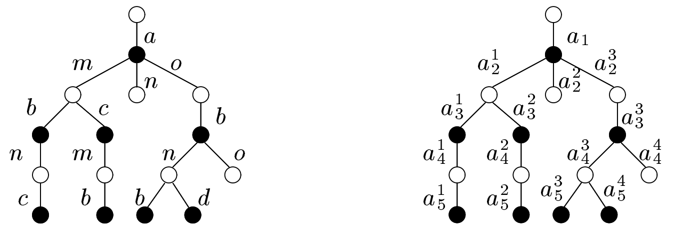

Figure 3: A game in terms of moves

Obviously for every game that is given via its set of moves we can easily transform this description into one via positions—we can just use the sequence of moves leading there to (uniquely) label any position.

While we have not yet said what a morphism of games should be it is worth pointing out here that different choices of labels (for the same game tree) will in general lead to non-isomorphic games in the category. There is also the question of whether a move can be legal for both, Player and Opponent. A more natural representation of a game tree might then be obtained by labelling the positions and coding when it is legal to move from one position to another. By necessity this labelling has to be unique, that is a given label cannot be used for more than one position. If we wish to label moves as well, this will typically be done by the two positions they connect. Clearly the set of positions then is partitioned naturally in a set of Player positions and a set of Opponent positions.

Definition 1 Alternative 3. *Let A^P and A^O be sets of so-called* positions *and let $*_A \in A^P$. A* game *is an acyclic (A^P, A^O)-bipartite graph such that every node has a finite path to $*_A$.*

Note that such a graph is indeed a tree with root $*_A$, and therefore it is legitimate to think of its edges as being directed. We will tacitly assume that they are directed *away* from the root. In this setting, a *play* is given by a sequence of positions, starting with the initial position $*_A$, such that there is an edge between any position and its successor in the sequence, should one such exist. We will use $a \longrightarrow a'$ to indicate that there is a move from position a to position a', and $a \longrightarrow^* a'$ to denote that there is a play to position a that can be continued to reach position a'. This representation makes it easier to find abstract versions of categories of games. (This might be the topic of a follow-up course.)

We will use these two representations interchangeably—for the simple games we are about to introduce, it does not matter which one we choose. We often will give a construction in terms of just one and it can be a useful exercise to work out the details for the other. It should be pointed out, however, that once the notions of game and strategy become more sophisticated, the two representations diverge.

1.3 Strategies

The main concept that brings games to life is that of a *strategy*. In fact in game theory games (at least simple ones) are often reduced to the sets of strategies for Player and Opponent and some codification of the *outcome* when playing those against each other.

Informally a *strategy for Player*, or *P-strategy*, consists of a collection of instructions telling Player what to do in every position when it is his turn. Since we see the world from Player's point of view, when we say 'strategy' we mean a P-strategy. Quite obviously there is a dual concept of *Opponent strategy*. We do include the possibility of refusing to move although there are still options available. Hence the concept of strategy is a *partial* one. Such a strategy for, say, chess would be huge, and for big games this is not something that is easily handled in practice. For our purposes, however, it serves to encapsulate the idea of 'proof' or 'program'. We write $\rho\colon A$ if ρ is a P-strategy on A.

Not surprisingly there are various ways in which the notion of a strategy can be described. If we assume that our game is given by its game tree then it makes sense to view a strategy ρ as a *subtree* of the game tree, subject to the following condition: Whenever it is Player's turn, the strategy picks at most one of the possible moves. In other words, if an Opponent position a is in ρ and $a \longrightarrow a'$ as well as $a \longrightarrow a''$ then a' and a'' in ρ implies that $a' = a''$. Such a subtree is given for our example game in Figure 4.

This gives us a notion of a *reachable position* when playing according to some strategy. The set of reachable positions $R(\rho)$ for a strategy ρ is the set of all positions contained in the corresponding subtree and all (Opponent) positions which can be reached from such a position with one (Opponent) move. We refer to the set of reachable P-positions as $R^P(\rho)$.

This one condition, while sufficient to tie down the notion of strategy, still allows several subtrees to encode the same strategy. It is useful to have just one representative for each

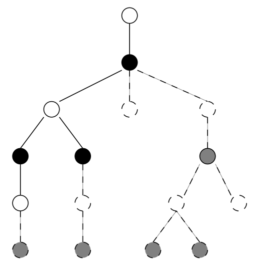

Figure 4: A strategy as a subtree

strategy, which means we have to impose a second condition. As it is, there is a minimal and a maximal representative for a given strategy: One that includes all the reachable Opponent positions to which the strategy has no reply, and one where all these are left out. It is natural to view a subtree as a restriction of the original game; we would like it to contain all the plays that are possible when Player plays according to this strategy. This reading means that the maximal subtree representing a strategy is the appropriate one. As we shall see in a moment, this has a useful consequence. Hence we add the second condition: Whenever a Player position a is an element of ρ and $a \longrightarrow a'$ then a' is in ρ.

Here is a formal definition that is based on the setting of Definition 1, Alternative 3.

Definition 2 Alternative 1. *A strategy ρ for a game A is a connected subgraph containing $*_A$ such that if $a, a' \in R(\rho)$*

- *if $a \in A^O$, $a \longrightarrow a'$, $a \longrightarrow a''$, $a'' \in R(\rho)$ then $a' = a''$;*
- *if $a \in A^P$ and $a \longrightarrow a'$ then $a' \in R(\rho)$.*

The canonical representative for the strategy given in Figure 4 can be seen in Figure 5. The obvious dualization leads to the notion of Opponent strategy. An example is given in Figure 6.

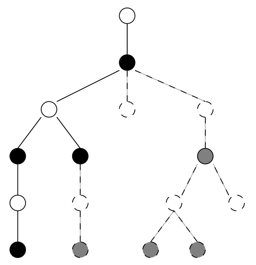

Figure 5: A strategy as a canonical subtree

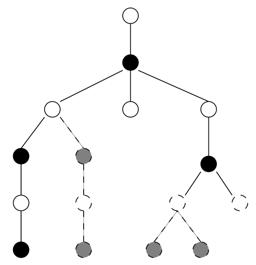

Figure 6: An O-strategy

An immediate result is the following:

Proposition 1.1 *A Player and an Opponent strategy intersect in precisely one play. If this play is finite then the final position in this play is the position where the game ends when playing the one strategy against the other.*

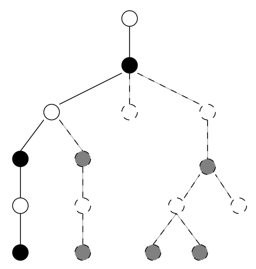

Figure 7: A play resulting from Figures 5 and 6

The intersection of the strategies from Figures 5 and 6 is given in Figure 7. This provides us with a notion of a 'result' of playing two strategies against each other, namely the actual play or, in the case this is finite, the final position in same. (Note that this only works because we decided to aim for the maximal representative of a strategy.) Again this is a notion that is useful when trying to abstract away from concrete games.

Quite obviously the subtree representation of a strategy is not the only one one might think of. An alternative is to use a partial map instead. Depending on which presentation is used, the co-domain of this partial map may vary. In the case of a presentation via moves a strategy will be given by a partial function from the set of O-positions (when it is Player's turn) to moves $A^O \longrightarrow M_A$. In the case of a presentation via positions, this turns into a partial function from O-positions to P-positions $A^O \longrightarrow A^P$. In either case a condition has to be imposed to ensure that this yields a valid set of instructions: It has to be the case that the strategy's suggested move is valid in the given position. Let ρ be such a strategy and a an O-position in the domain of ρ. In the first presentation, $\rho(a)$ has to be either a move such that the concatenation of a and $\rho(a)$ is a valid P-position, that is an element of A^P. In the second case, $\rho(a)$ will automatically be a P-position, but it also has to be the case that $a \longrightarrow \rho(a)$ is a valid move.

While any partial function satisfying this kind of position can be read as a strategy it is once again the case that this is not sufficient to ensure that there is one unique representative for each strategy: Nothing prevents the strategy to give instructions for positions which can never be reached due to earlier decisions made by the strategy. In the example of Figure 3, if Player chooses to play the move a_2^1 then the positions with prefix $a_1a_2^3$ will never be reached and hence it does not make sense to include instructions for what to do in position $a_1a_2^3a_3^3$. In the subtree representation, this was not a problem since the property of being a subtree precludes this. We therefore impose a second condition, giving us the *minimal* partial function that represents a given strategy by imposing the following condition. For any a in the domain of ρ, the unique predecessor of a is in the image of ρ. In other words ρ does not contain any redundant information. Equivalently, $\mathsf{dom}(\rho) \subseteq R^O(\rho)$ (where R^O is adapted to this presentation). It is clear that $\mathsf{im}(\rho) = R^P(\rho)$. Once again here is a formal definition.

Definition 2 Alternative 2. *A* strategy *on a game A is a partial function $A^O \rightharpoonup A^P$*

such that

- *for all* $a \in \mathsf{dom}(\rho)$, $a \longrightarrow \rho(a)$;
- $\mathsf{dom}(\rho) \subseteq R^O(\rho)$.

Exercise 1.2 *(1) Write out the definition of 'strategy' for Definition 1, Alternative 2 for both, the subtree and the partial function interpretation.*

(2) Given any partial function $A^O \longrightarrow M$*, is there a way of interpreting it as a strategy? If yes, how would you obtain the partial function corresponding to that strategy?*

(3) There are more ways of representing a strategy. Show that any strategy ρ *is uniquely determined by its set of reachable* P*-positions* $R^P(\rho)$*. (Choose your favourite representation for this.)*

(4) Yet another way of representing a strategy ρ *works as follows: Take all* P*-positions reachable by play according to* ρ *and add those reachable* O*-positions to which* ρ *has no reply. If you choose the dual representation for* O*-strategies, what can you say about their interaction, provided the game in question is finite?*

(5) Assume that the game A *is given by a prefix-closed subset of the set of strings over some alphabet* M*. Let* a *and* a' *be positions in a game* A *that are reachable for some strategy* ρ *on* A*. Show that one of the following is true:*

- a *is a prefix of* a';
- a' *is a prefix of* a;
- *the largest joint prefix of* a *and* a' *is of even length.*

(6) The set of P*-strategies on a game* A *is obviously pre-ordered since the strategies considered are partial. What can you say about the structure of this set? You might want to calculate the various operations for the two different presentations.*

(7) What would a notion of total *strategy be? How about non-determinism? Can these two notions be combined?*

(8) Assume that games are given as pre-fixed closed sets over sets of moves. One condition we might want to impose on a strategy is that it should be dependent only on the previous (Opponent-)move, no matter from which position it was made. Such strategies are called 'history-free'. Formulate a suitable condition for history-freeness.

2 A category of games

2.1 Morphisms and linear function spaces

As is often the case with models for linear logic (and symmetric monoidal closed categories) it is, in fact, easier to describe the (linear) function space first and derive the morphisms from that (see Appendix A for a brief account of this connection).

As we have stressed before, the notion of strategy is the central one, and arguably a game is just useful just in so far as it is a vehicle for describing strategies. A morphism $A \longrightarrow B$

will be a strategy on the linear function space $A \multimap B$. We also expect such a strategy to map strategies on A to strategies on B. Compare this to the old proof theoretic idea that a proof of $A \multimap B$ should transfer proofs of A to proofs of B, or the idea that a term of a function type $A \to B$ should take terms of type A as its arguments. Hence $A \multimap B$ has to be such that the game A 'occurs' within it; and as the result of this kind of application should be in B, the same is true for B.

Note, however, that a role reversal takes place here as far as the game A is concerned: There are numerous ways of seeing this. We say, for example, that A occurs 'contravariantly' in $A \multimap B$ which means that $- \multimap B$ is a functor $\mathbf{C} \longrightarrow \mathbf{C}^{\mathsf{op}}$ rather than $\mathbf{C} \longrightarrow \mathbf{C}$. If we think in terms of programs then Player (who we still identify with) takes over the role of the *environment* in the game A: he will 'start' the program and provide any arguments which are required—this is the role played by Opponent. Or, if we view strategies in terms of proofs then one way of proving an implication is to prove that the premise is false—hence Player has an interest in 'disproving' (rather than proving) A.

The basic idea is that in the linear function space $A \multimap B$ can be viewed as playing A and B 'in parallel' (but still sequentially, that is one move (from either game) at a time), but where Player takes on the role of Opponent in A. Note that strictly speaking, switching roles in a game A does not give another game since part of the definition of a game is that Opponent always starts. Sometimes the result referred to as a 'cogame', and written as $A^\perp$. This can be misleading (see Section 4) and we will not make much use of it. Because of sequentiality it might be more accurate to say that we *interleave* the playing of A and B. This is subject to some conditions. First of all note that if Player takes on the role of Opponent in A (and thus is to start A), while Opponent is to start the game $A \multimap B$, then Opponent will have to start play by choosing an opening move in B. This creates some awkwardness regarding the initial position. One could leave out the initial position and start with a game forest (rather than a game tree) but that creates similar problems elsewhere. We give a formal definition of this game based on Definition 1, Alternative 3.

Definition 3 *Positions of the linear function space $A \multimap B$ of two games A and B are sequences $t_1 \cdots t_n$ over $A^P \setminus \{*_A\} + A^O + B^P \setminus \{*_B\} + B^O$ such that*

- *the restriction to positions from A, $t_1 \dots t_n|_A$, gives a valid play of A when prefixed by the initial positions $*_A$;*
- *the restriction to positions from B, $t_1 \dots t_n|_B$, gives a valid play of B when prefixed by the initial positions $*_B$;*
- *for $1 \leq i \leq n$ even, $t_i \in A^O + B^P$;*
- *for $1 \leq i \leq n$ odd, $t_i \in A^P + B^O$.*

The set of P-position $(A \multimap B)^P$ is given by strings of even length and the set of O-positions $(A \multimap B)^O$ is given by strings of odd length. The initial position is the empty string. Edges exist precisely between positions $t_1 \cdots t_n \longrightarrow t_1 \cdots t_n t_{n+1}$.

An example for a linear function space is given in Figure 10. We adopt the convention that edges pointing towards the right originate from moves of the game B, whereas edges pointing towards the left arise from the game A. This makes it more obvious that such a game can be viewed as interleaving the two games A and B. The labels attached to the positions in $A \multimap B$ are not the sequences we obtain from Definition 3. Rather than putting the full sequence at each node we only give the last element—the remainder is determined by the labels along the path from the initial position.

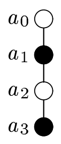

Figure 8: The game A

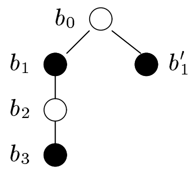

Figure 9: The game B

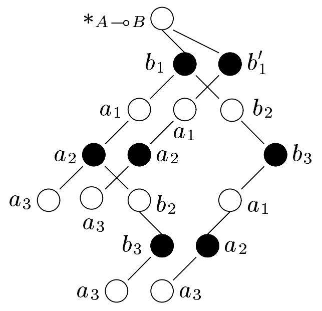

Figure 10: The game $A \multimap B$

Exercise 2.1 *(1) Draw a (small) game tree, say A, and then draw the game tree for $A \multimap A$.*

(2) Write out a definition of $A \multimap B$ for A and B being games according to Definition 1, Alternative 2.[1]

The conditions for the linear function space may look weak at first sight but they impose a certain discipline upon the players (mostly Opponent).

Proposition 2.2 *For the linear function space $A \multimap B$ of two games it is the case that*

- *Opponent must start with a move in B;*
- *Opponent can never 'switch' between the games.*

Exercise 2.3 *Prove Proposition 2.2.*

Note that every position in $A \multimap B$ can be projected onto a pair of positions, one in A and one in B, as follows: In the sequence presenting the position, say t, in $A \multimap B$ take the last occurring positions from A and B respectively. If no such occurs for A then choose the initial position. This gives us a function from the positions of $A \multimap B$ to the cartesian product of the set of positions of A and the set of positions of B which we refer to as $q_{A,B}$. Intuitively, if we view the game $A \multimap B$ consisting of playing $A \multimap B$ in parallel, these projections give the positions in the games A and B reached so far; they totally ignore how play interleaved between the two component games.

Exercise 2.4 *(1) Show that if t is a P-position in $A \multimap B$ then it projects under $q_{A,B}$ onto a pair in $A^P \times B^P$ or onto one in $A^O \times B^O$. What do O-positions in $A \multimap B$ project onto? What about the remaining combination? What connection is there between this discipline and the one imposed on Opponent when playing a linear function space game?*

(2) If we view a strategy ρ as being represented by its set of reachable P-positions $R^P(\rho)$ then such a strategy on $A \multimap B$ can be mapped to a pair of relations, one $A^P \longrightarrow\!\!\!\!\!\!\!+\;\; B^P$ and one $A^O \longrightarrow\!\!\!\!\!\!\!+\;\; B^O$. Show that no information is lost by this process, that is the strategy can be recovered from the relations.

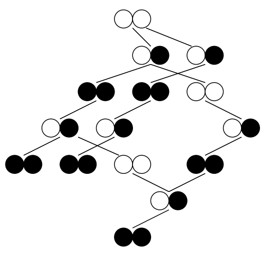

Figure 11: The game $A \multimap B$, projected onto pairs of positions from A and B

Figure 11 shows the image of $A \multimap B$ from Figure 10 under $q_{A,B}$ in the cartesian product of positions of A and positions of B where

- we have left out the combination $A^O \times B^P$ since it does not occur in this image;
- we have retained the edges of the game tree for purposes of orientation.

Note how some positions from $A \multimap B$ become identified under this mapping.

As we have said before, a morphism $A \longrightarrow B$ will be given by a strategy on $A \multimap B$. Hence to define a category we still have to say how to compose these strategies, and what the identity morphism should be. This undertaking is non-trivial, although the algorithm for the composite can be explained informally. We will do this first, using an example. Let $\rho\colon A \multimap B$ and $\sigma\colon B \multimap C$ be two morphism, namely $\rho\colon A \longrightarrow B$ and $\sigma\colon B \longrightarrow C$. An example is given in Figures 12 and 13.

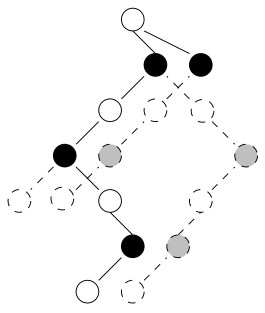

Figure 12: $\rho\colon A \multimap B$

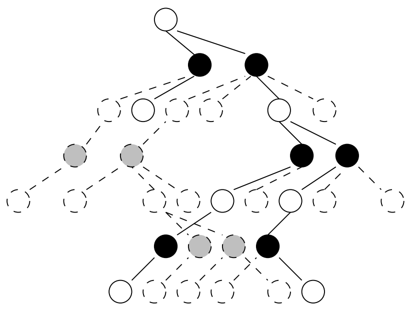

Figure 13: $\sigma\colon B \multimap C$

Exercise 2.5 *From $B \multimap C$ in Figure 13 (and B in Figure 9) find the game C.*

[1]This definition appears to be more natural than that in terms of positions given above and could be seen as an argument that Alternative 2 is more natural.

So how do we compose these strategies? To make it easier to follow the argument all the games in question are pictured in Figure 14.

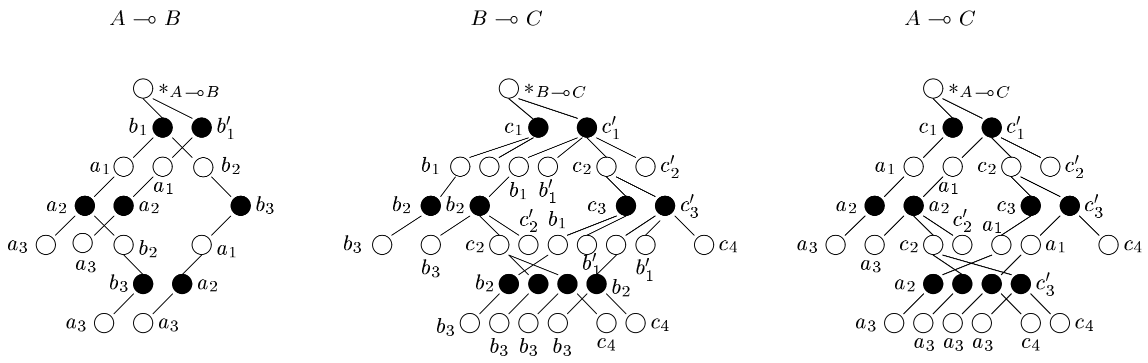

Figure 14: The games $A \multimap B$, $B \multimap C$ and $A \multimap C$

In order to find $(\sigma \circ \rho)$'s answer to Opponent's 'right' opening move to c'_1 in C (and therefore in $A \multimap C$) we look at σ's reply to that move (in $B \multimap C$) and find that it takes us to position $c'_1c'_2$. Hence this should be the reply of $\sigma \circ \rho$ to that opening move.

For the 'left' opening move to c_1 in C we consider σ's reply to that move, which is the right of the two opening moves in B, leading us to $c'_1b'_1$. We now look up ρ's reply to b'_1 viewed as a position in $A \multimap B$ and find that there is none. Hence $\sigma \circ \rho$ is undefined for c_1.

Next we want to find $(\sigma \circ \rho)$'s response to the two O-positions in $A \multimap C$ which can be reached so far, $c'_1c'_2c_3$ and $c'_1c'_2c'_3$.

We discuss the latter alternative first. We once again mimic the corresponding moves in $B \multimap C$ and find that σ's reply to $c'_1c'_2c'_3$ (now viewed in $B \multimap C$) is the left opening move leading to $c'_1c'_2c'_3b_1$ in $B \multimap C$. Looking at ρ's reply to b_1 (now viewed as a position in $A \multimap B$) we find b_1a_1. Hence we use this last move as $(\sigma \circ \rho)$'s reply to $c'_1c'_2c_3$, leading us to $c'_1c'_2c_3a_1$.

Opponent can now move to $c'_1c'_2c_3a_1a_2$, and to obtain $(\sigma \circ \rho)$'s reply to that move we copy the last move to the position we had reached in $A \multimap B$ in play so far to get $b_1a_1a_2$. ρ's reply to that is $b_1a_1a_2b_2$, and once again we copy that last move over to play in $B \multimap C$ so far to give us $c'_1c'_2c_3b_1b_2$. We look up σ's reply to that which takes us to $c'_1c'_2c_3b_1b_2b_3$, and copy the last move to the position reached in $A \multimap B$ so far, which gets us to $b_1a_1a_2b_2b_3$. Now we once again look up ρ's answer to that and get $b_1a_1a_2b_2b_3a_3$. Hence (after much work!) we have once again reached one of the two 'outer' games A and C and can record that $(\sigma \circ \rho)$'s reply to $c'_1c'_2c_3a_1a_2$ is $c'_1c'_2c_3a_1a_2a_3$.

It remains to determine $(\sigma \circ \rho)$'s reply to $c'_1c'_2c'_3$. Again we look up σ's reply to this position, now viewed in $B \multimap C$. The interaction between ρ and σ leads us to $c'_1c'_2c'_3a_1$.

From there Opponent may continue to $c'_1c'_2c'_3a_1a_2$. We look up ρ's reply to the position $b_1a_1a_2$ and find $b_1a_1a_2b_2$, so control passes once again to σ which answers $c'_1c'_2c'_3b_1b_2$ with $c'_1c'_2c'_3b_1b_2c_4$. Hence $(\sigma \circ \rho)$'s reply to $c'_1c'_2c'_3a_1a_2$ is $c'_1c'_2c'_3a_1a_2c_4$. The result is the strategy in Figure 15.[2]

[2]Unfortunately we cannot produce dashed circles while at the same time having indexed labels. Hence this presentation differs slightly from the one we have had so far. Again we are using line thickness to emphasize the positions of interest. The labels should be read to refer to precisely those positions.

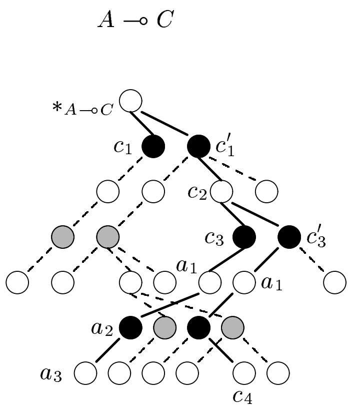

Figure 15: The composite $\sigma \circ \rho$: $A \multimap A$

Note that any play in B is uniquely determined by ρ and σ feeding their respective replies to each other—this is a consequence of Opponent being unable to switch between games. This means that control is carefully passed between the strategies ρ and σ in the sense that at any point when it is Player's turn in the game $A \multimap C$, precisely one of ρ and σ can be used at each step to determine $(\sigma \circ \rho)$'s reply. If we did not have the condition that 'Opponent may not switch between games' then this would not work.

What we do in general is to use a 'hidden' copy of the game B to work out the reply of $\sigma \circ \rho$ in the 'outer' games A and C as appropriate. Sometimes people refer to the action on B as being calculated using a 'scratch-pad'. That means we keep playing in the hidden game until play finally 'emerges' once more in one of the outer games, A or C. If play fails to emerge then the composite is not defined at the position in question. Figure 16 tries to give some idea of what is going on. The dashed lines 'copy' the move to make it appear in its correct colour in the game B and A (rather than in the co-games $B^\perp$ and $A^\perp$).

That this works is more remarkable than it might seem at first. It is easy enough to find the answers $\sigma \circ \rho$ gives to an opening move in $A \multimap C$. But what about positions further down the line? If they *are* reachable in play according to $\sigma \circ \rho$ then any play in B that occurs 'on the way there' is uniquely determined, and so this information is provided by the given position. In other words, in order to find $(\sigma \circ \rho)$'s answer to a given reachable O-position in $A \multimap C$, consider the play (in A and C) leading to it. Then there is precisely one play in the 'hidden' game B that goes along with this which is defined via the interaction of ρ and σ.

Making this precise (using whichever representation of games and strategies) is surprisingly tedious. We will work out the definition according to the above idea for games which are given via their sets of moves, Definition 1, Alternatives 1 and 2. The same method also works for Alternative 3 but is more tedious to formalize. We will treat this case when we introduce a useful shortcut for calculating the composite.

So for the time being let A, B, and C be given by a prefix-closed set of strings over their sets of moves M_A, M_B and M_C respectively. Then we can label the *positions* of a function space game, say $A \multimap B$, with moves from the direct sum $M_A + M_B$ in the obvious way. A play in $A \multimap B$ is given by a sequence of such positions, but we do not lose any information if we merely state the sequence of moves that are 'added on' at each step. Hence we can describe a play in $A \multimap B$ by a sequence of moves from $M_A + M_B$.

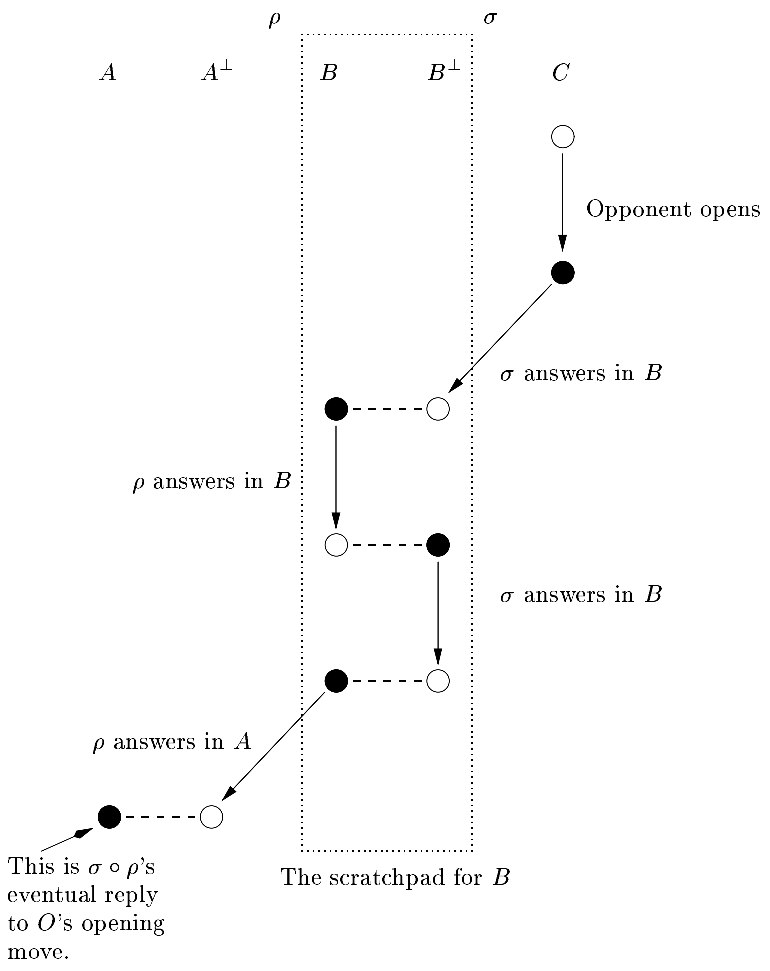

Figure 16: Play using a scratch-pad for B

For such a play $r_1 \cdots r_l$ we define

$$r_1 \cdots r_l|_A$$

to be the sequence of moves of A which we obtain by striking all moves from B, and we similarly define $r_1 \cdots r_l|_B$.

Given a play in $A \multimap B$, say $r_1 \cdots r_l$ and another play in $B \multimap C$, say $s_1 \cdots s_m$ such that

$$r_1 \cdots r_l|_B = s_1 \cdots s_m|_B$$

we can define a unique *interleaving* of the two plays, that is a sequence

$$t_1 \cdots t_{l+m}$$

of moves from A, B and C such that

$$t_1 \cdots t_{l+m}|_{A \multimap B} = r_1 \cdots r_l \qquad \text{and} \qquad t_1 \cdots t_{l+m}|_{B \multimap C} = s_1 \cdots s_m.$$

Since $t_1 \cdots t_{l+m}|_A = r_1 \cdots r_l|_A$ is a valid play in A, and $t_1 \cdots t_{l+m}|_C = s_1 \cdots s_m|_C$ is a valid play in C, $t_1 \cdots t_{l+m}|_{A \multimap C}$ is a valid play in $A \multimap C$.

Hence we can define the composite of ρ and σ to be the subtree

$$\sigma \circ \rho = \{t_1 \cdots t_n|_{A \multimap C} \mid t_1 \cdots t_n \text{ is a sequence of moves from } A,\, B \text{ and } C \text{ with } t_1 \cdots t_n|_{A \multimap B} \in \rho, t_1 \cdots t_n|_{B \multimap C} \in \sigma\}.$$

Exercise 2.6 *(1) Assume that $r_0 \cdots r_l$ and $s_0 \cdots s_m$ are plays on $A \multimap B$ and $B \multimap C$ respectively such that $r_0 \cdots r_l|_B = s_0 \cdots s_m|_B$. For the sake of simplicity assume that the sets of moves are split up into Player and Opponent moves for each game. Show that precisely one of the following is the case:*

- *the last moves in A, B and C are Player moves;*
- *the last moves in A, B and C are Opponent moves;*
- *the last moves in A and B are Player moves and the last move in C is an Opponent move;*
- *the last move in A is a Player move and the last moves in B and C is an Opponent move.*

Conclude that the interleaving above is indeed unique.

(2) Show that if $t_1 \cdots t_n$ is a play in accord with $\sigma \circ \rho$ then there is precisely one sequence of moves from A, B, and C whose restriction to moves from $A \multimap C$ is $t_1 \cdots t_n$. In other words, the play in the 'hidden' game B is uniquely determined by the visible play.

(3) If you really want to make sure that composition can be defined in the same way in the positional setting, you might want to do that.

This makes it rather complicated to calculate $(\sigma \circ \rho)$'s reply to a given O-position in $A \multimap C$: We first have to reconstruct the play in B that took us there (which is uniquely determined by Exercise 2.6(2)), and then we can think about constructing the next move in the sequence! Fortunately there is a useful shortcut. We will now switch back to viewing games as being given via their sets of positions, which is more general.

Consider the projection $q_{A,B}$ for positions from $A \multimap B$ defined above. We can split this up (cf. Exercise 2.4 (2)) into maps

$$q^P_{A,B} \colon (A \multimap B)^P \longrightarrow (A^P \times B^P) + (A^O \times B^O) \qquad \text{and} \qquad q^O_{A,B} \colon (A \multimap B)^O \longrightarrow A^P \times B^O.$$

These projections are not injective (even jointly), but they become so when restricted to the positions reachable according to some strategy on $A \multimap B$.

Proposition 2.7 *Let t, t' be positions on $A \multimap B$ which are reachable when playing according to some strategy ρ. If $q_{A,B}(t) = q_{A,B}(t')$ then $t = t'$.*

Proof. We use Exercise 1.2 (5). Clearly if t is a prefix of t' (or *vice versa*) and $q_{A,B}(t) = q_{A,B}(t')$ then they must be equal. Alternatively their largest joint prefix t'' is of odd length, and the corresponding position is a P-position in $A \multimap B$. Since $q_{A,B}(t) = q_{A,B}(t')$ they can only differ if the same moves from A are played in the same order, and the same is true for B. Therefore the only difference that can occur is a different interleaving of these sequences of moves. That would mean that one of t and t' would play in A while the other does so in B.

This, however, can not occur following from a Player position since Opponent is not allowed to switch between games. In other worse

$$q_{A,B}(t'') = q^P_{A,B}(t'') \in (A^P \times B^P) + (A^O \times B^O).$$

If $q_{A,B}(t'') = q^P_{A,B}(t'') \in A^P \times B^P$ then the next move (which is Opponent's) has to be in B for both t and t', but since $q_{A,B}(t) = q_{A,B}(t')$ they both have the same move here. Similarly if $q_{A,B}(t'') \in A^O \times B^O$, which requires the next move to be made in A. Thus $t = t'' = t'$. □

From Exercise 1.2 (3) we know that we may represent a strategy by its set of reachable P-positions. Lemma 2.7 now tells us that we may replace these P-positions by their projections without losing information. For the strategies ρ and σ these sets can be seen in Figures 17 and 18. We have emphasized the positions of interest by using a thicker line.

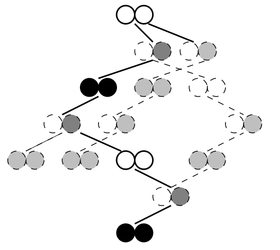
Figure 17: ρ, projected onto pairs of P-positions

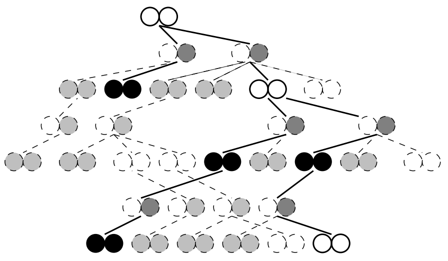
Figure 18: σ, projected onto pairs of P-positions

Lemma 2.8 *Let t be a P-position in $A \multimap C$ that is reachable by play according to $\sigma \circ \rho$. Then there exists precisely one position b in B such that the following hold:*

- *there is a position r in $A \multimap B$ with $\pi_1(q_{A,B}(r)) = \pi_1(q_{A,C}(t))$ and $\pi_2(q_{A,B}(r)) = b$;*
- *there is a position s in $B \multimap C$ with $\pi_1(q_{B,C}(S)) = b$ and $\pi_2(q_{B,C}(s)) = \pi_2(q_{A,C}(t))$.*

Proof. Since we have not given a formal definition for the composite in the positional setting we will only prove this for the case that games are given via their sets of moves. So a position in $A \multimap C$ is given by a sequence of moves from $M_A + M_C$. By definition of $\sigma \circ \rho$ we have a sequence of moves $t_1 \cdots t_n$ from $M_A + M_B + M_C$ such that

- $t_1 \cdots t_n|_{A \multimap C} = t$
- $t_1 \cdots t_n|_{A \multimap B}$ is a valid play on $A \multimap B$ and
- $t_1 \cdots t_n|_{B \multimap C}$ is a valid play of $B \multimap C$.

Note that in this setting

$$\pi_1(q_{A \multimap C}(t_1 \cdots t_n)) = t_1 \cdots t_n|_A \qquad \text{and} \qquad \pi_2(q_{A \multimap C}(t_1 \cdots t_n)) = t_1 \cdots t_n|_C.$$

Hence $r = t_1 \cdots t_n|_{A \multimap B}$ and $s = t_1 \cdots t_n|_{B \multimap C}$. Clearly this satisfies the condition if we set $b = t_1 \cdots t_n|_B$.

It remains to show the uniqueness of b. We know from Exercise 2.6 (2) that play in B is determined uniquely. It remains to show that we cannot have plays that are continuations of each other leading to different positions satisfying the conditions. So let $t_1 \cdots t_n$ be the least sequence of moves from $M_A + M_B + M_C$ such that $t_1 \cdots t_n|_{A \multimap C} = t$. Since $t_1 \cdots t_n|_{A \multimap C}$ is a P-position, by Exercise 2.4 (1) it must be the case that

$$(t_1 \cdots t_n|_A \in A^P \text{ and } t_1 \cdots t_n|_C \in C^P) \qquad \text{or} \qquad (t_1 \cdots t_n|_A \in A^O \text{ and } t_1 \cdots t_n|_C \in C^O).$$

Similarly for the projections onto the components of $A \multimap B$, that is A and B, and for the projections on the components of $B \multimap C$. Taking all these together, only two cases remain, namely

$$t_1 \cdots t_n|_A \in A^P \text{ and } t_1 \cdots t_n|_B \in B^P \text{ and } t_1 \cdots t_n|_C \in C^P \qquad \text{or}$$
$$t_1 \cdots t_n|_A \in A^O \text{ and } t_1 \cdots t_n|_B \in B^O \text{ and } t_1 \cdots t_n|_C \in C^O.$$

But if we were to extend this sequence by a move in B then in the first case the projection onto $B \multimap C$ would not be a valid position, whereas in the second case, the projection onto $A \multimap B$ would not be a valid position. Thus any continuation of $t_1 \cdots t_n$ will have to be a move in A or C, and will not project onto the given position t. □

Exercise 2.9 *Let $t_1 \cdots t_n$ be a sequence of moves from A, B and C such that $t_1 \cdots t_n|_{A \multimap B}$ is a play according to ρ and such that $t_1 \cdots t_n|_{B \multimap C}$ is a play according to σ. Show that if it is Player's turn in $t_1 \cdots t_n|A \multimap C$ then precisely one of ρ and σ can be used to determine the next move. Conclude that at any point in determining $\sigma \circ \rho$'s reply to some position in $A \multimap C$, precisely one of ρ and σ has control over what happens next.*

Corollary 2.10 *A P-position t in $A \multimap C$ is reachable by play according to $\sigma \circ \rho$ if and only if the following conditions are satisfied:*

- *$q_{A,C}(t)$ is an element of the relational composite of $q_{A,B}[R^P(\rho)] = \{q_{A,B}(r) \mid r \in R^P(\rho)\}$ and $q_{B,C}[R^P(\sigma)]$;*
- *all prefixes of t which are P-positions are reachable play according to $\sigma \circ \rho$.*

Proof. Lemma 2.8 establishes that a P-position that is reachable by play according to $\sigma \circ \rho$ does satisfy the conditions. Hence it remains to show the converse. But the relational composite means that we have plays in $A \multimap B$ and $B \multimap C$ whose restrictions to B coincide, so by definition of $\sigma \circ \rho$, they give rise to a play in accord with this composite strategy. □

We take up the example again to demonstrate what happens. Then

$$q_{A,B}(R^P(\rho)) = \{(a_0, b_0), (a_1, b_1), (a_2, b_2), (a_3, b_3)\}$$

and

$$q_{B,C}(R^P(\sigma)) = \{(b_0, c_0), (b_1', c_1), (b_0, c_2), (b_1, c_3), (b_1, c_3'), (b_3, c_3), (b_2, c_4)\}$$

as can be seen from Figure 19.

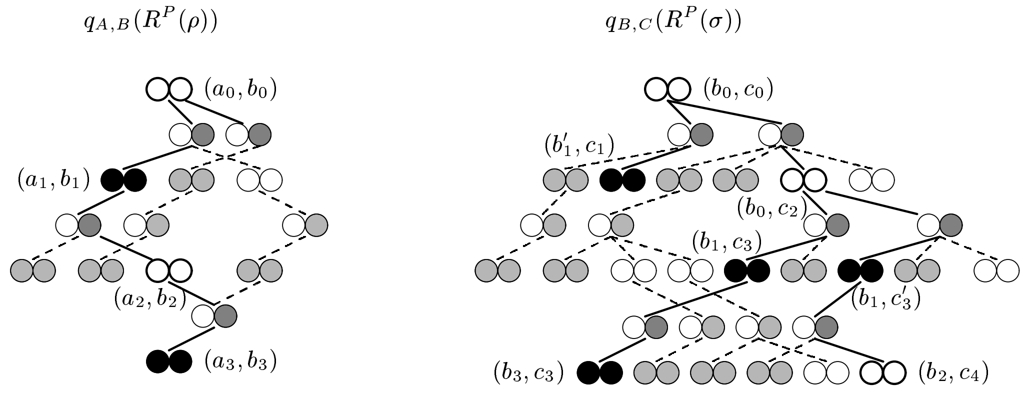

Figure 19: ρ and σ as relations

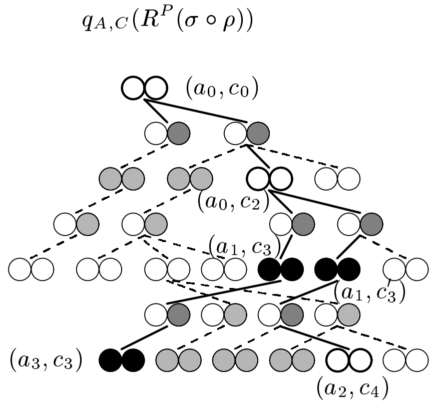

Figure 20: $\sigma \circ \rho$ as a relation

The relational composite is

$$\{(a_0, c_0), (a_0, c_2)(a_1, c_3), (a_1, c'_3), (a_2, c_4), (a_3, c_3)\}$$

which is equal to $q_{A,C}(R^P(\sigma \circ \rho))$ as can be seen from Figure 20.

Exercise 2.11 *Show that composition is associative. Even though this is not the shortest way of doing this it is worthwhile to examine how control is passed between the strategies ρ, σ and τ in each case.*

To establish that we have a category it remains to establish identity morphisms for all games. If A is any game then $A \multimap A$ has one canonical strategy which can intuitively be described as follows. Opponent starts play in the 'covariant' copy of A, and Player can 'copy' this move by playing that same opening move in the other, the 'contravariant' copy of A. Opponent has to reply in that copy, and Player can copy the answer back to the first game. They can continue in this way until they run out of valid moves. We call this strategy the *copy-cat strategy* on $A \multimap A$. It serves as the identity morphism $A \longrightarrow A$.

Remark. This copy-cat strategy has some amusing qualities. First of all note that Player can never run out of moves to make—Opponent would have to do so first. Secondly, this allows pitching a Player strategy on A against an Opponent strategy on the same game, by letting them make up an Opponent strategy on $A \multimap A$ in the obvious way and playing the copy-cat strategy against the result. In particular playing, say, chess against Kasparov and Deep Blue we can pitch them against each other provided that

- one of them takes white, and the other black and
- the game where one of them has white is started first.

We just copy, say, Kasparov's white move and make it our opening move against Deep Blue. Unless both games end in a draw this guarantees that we will win at least one of them!

The relational presentation of id_A is

$$q_{A,A}(R_P(\mathsf{id}_A)) = \{(a,a) \mid a \in A^P + A^O\}.$$

Not surprisingly this is the identity relation on $A^P + A^O$, and therefore the relational composite with the relational presentation of any strategy on A is equal to that strategy. Therefore the proposed identity on A behaves indeed as desired.

Definition 4 *Let* **Gam** *be the category whose objects are games; morphisms $A \longrightarrow B$ are strategies on the linear function space $A \multimap B$.*

In fact by using the relational presentation of strategies we have shown the following theorem. Let **Rel** be the category of sets and relations.

Theorem 2.12 *There is a faithful functor* **Gam** $\longrightarrow$ **Rel**.

Proof. On objects, take $F(A) = A^P + A^O$. On morphisms, map $\rho\colon A \longrightarrow B$ to the relation $R^P(\rho)\colon A^P + A^O \longrightarrow B^P + B^O$. Faithfulness was established in Proposition 2.7. □

This functor can be used as the basis of establishing an 'abstract' category of games [HS99]. This works by building up from the category of sets and relations **Rel**, creating ever more sophisticated categories. At each step there is a functor 'translating' from games to these resulting categories which carries across more and more of the properties of games. The first step on this way is to observe for a morphism $\rho\colon A \longrightarrow B$ the image $F(\rho)$ only intersects two of the sectors of $(A^P + A^O) \times (B^P + B^O)$, namely $A^P \times B^P$ and $A^O \times B^O$. Hence we can view F as a functor **Gam** $\longrightarrow$ **Rel** $\times$ **Rel**$^{\mathsf{op}}$. The latter is a $*$-autonomous category (and, in fact, a model of classical linear logic), but we will not describe its structure here. The second step refines the presentation of strategies, using that developed in Exercise 1.2 (4) and noticing that then $F(\rho)$ intersects only three of these four quadrants ($A^P \times B^O$ being added; compare Exercise 2.4(1)).

The third step uses the following observation. Let ρ be a P-strategy on a game A and σ a morphism $A \longrightarrow B$. Now consider what happens if we play ρ off against σ: Opponent opens in B and we look up σ's reply to that move. If this reply is in B we can use it as an answer. Otherwise we can apply ρ to the reply and chase the play between ρ and σ until play once more emerges in B. It should be clear that in this manner we obtain a P-strategy on B. In other words: A morphism $A \longrightarrow B$ transforms P-strategies on A into P-strategies on B. Compare that with the idea that a proof of $A \multimap B$ should transfer proofs of A to proofs of B.

Clearly the method we have just described is very similar to our intuitive description of composition, and this is something we can make precise. Let $\mathbf{I}$ be the game that consists of no moves at all, that is just the initial position. It is easy to see that $A \cong \mathbf{I} \multimap A$. Hence we can view every strategy on A as a morphism $\mathbf{I} \longrightarrow A$. The strategy on B we have described above is nothing but the composite of this strategy $\mathbf{I} \longrightarrow A$ with $\rho\colon A \longrightarrow B$, taking into account that $\mathbf{I} \multimap B \cong B$.

If, on the other hand, an O-strategy β on B is given then we can employ a similar method to combine it with a strategy $\sigma\colon A \longrightarrow B$. Indeed we can consider σ's reply to β's opening move in B and chase play between σ and β until we reach an opening move in A (or decide that play will never leave B). We thus obtain an O-strategy on A. It is not as straightforward to describe this as a composite of two strategies since O-strategies do not correspond to morphisms.

Exercise 2.13 *Consider the game* S *which consists of just one move. Now consider the game* $B \multimap \mathsf{S}$ *where* B *is arbitrary. Show that Opponent strategies on* B *are in bijective correspondence with Player strategies on* $B \multimap \mathsf{S}$*. The composite of some* $\sigma\colon A \longrightarrow B$ *with a strategy on* $B \multimap \mathsf{S}$ *gives a strategy on* $A \multimap \mathsf{S}$*.*

Hence a morphism $A \longrightarrow B$ gives rise to two maps.

$$\begin{tikzcd}
P\text{-strategies on } A \arrow[d] & O\text{-strategies on } B \\
P\text{-strategies on } B & O\text{-strategies on } B \arrow[u]
\end{tikzcd}$$

In [HS99] this idea is used to introduce a so-called 'double-glueing' construction which gives rise to a category built over **Rel** so that the functor from **Gam** to this category is full and faithful.

Exercise 2.14 *We know that we can (faithfully) view a morphism* $A \longrightarrow B$ *via two relations, one* $A^P \relbar\joinrel\mapstochar\joinrel\rightarrow B^P$ *and one* $B^O \relbar\joinrel\mapstochar\joinrel\rightarrow A^O$*. Can you come up with a condition on such relations that all strategies satisfy? Does that give you an idea for refining the category* $\mathbf{Rel} \times \mathbf{Rel}^{\mathsf{op}}$ *so that it mimics* **Gam** *more faithfully?*

Exercise 2.15 *Consider mapping a game* A *to its (pre-ordered) set of player strategies. Make this a functor from* **Gam** *to the category of posets. If you have done Exercise 1.2 (6) then use a suitable category of domains as the target category. Find a suitable enrichment for the resulting functor.*

2.2 The multiplicative structure

While establishing the category required some effort, the symmetric monoidal closed structure on this category falls into place quite readily. We start by describing the tensor product.

Definition 5 *The tensor product of two games* A *and* B *is defined as follows. Positions are strings* $t_1 \cdots t_n$ *over* $A^P \setminus \{*_A\} + A^O + B^P \setminus \{*_B\} + B^O$ *such that*

- *the restriction to positions from A, $t_1 \dots t_n|_A$, gives a valid play of A when prefixed by the initial positions $*_A$;*
- *the restriction to positions from B, $t_1 \dots t_n|_B$, gives a valid play of B when prefixed by the initial positions $*_B$;*
- *for $1 \leq i \leq n$ even, $t_i \in A^P + B^P$;*
- *for $1 \leq i \leq n$ odd, $t_i \in A^O + B^O$.*

The set of P-position $(A \otimes B)^P$ is given by strings of even length while the set of O-positions $(A \otimes B)^O$ is given by strings of odd length. Edges exist precisely in the following situation $t_1 \cdots t_n \longrightarrow t_1 \cdots t_{n+1}$.

Exercise 2.16 *Choose two (small) games and draw their tensor product.*

Intuitively, this game is quite easy to understand. The games A and B are played 'in parallel' in in a way similar to the linear function space (without any role reversals this time), that is, the playing of the two games is interleaved. Again it is the case that this definition imposes a certain discipline upon the players, only this time, it is mostly player who is restricted. Obviously we can define restrictions/projections of positions to the constituent games as we did for the linear function space.

Proposition 2.17 *For the tensor product $A \otimes B$ of two games it is the case that*

- *Opponent may start with a move in A or B;*
- *Player can never 'switch' between the games and must answer in whichever game Opponent last moved.*

Proof. The proof is analogous to that of Proposition 2.2. □

Let I be the game that consists of no moves at all, so that its only position is the initial position $*_{\mathbf{I}}$. Then $A \otimes \mathbf{I} \cong A \cong \mathbf{I} \otimes A$.

Exercise 2.18 *Write out definitions for these isomorphisms (as strategies).*

Given $\rho\colon A \longrightarrow A'$ and $\sigma\colon B \longrightarrow B'$ we can define $\rho \otimes \sigma\colon A \otimes B \longrightarrow A' \otimes B'$ as follows. Given an O-position r in $A \otimes B$ that is reachable then if the last move was made in A then play ρ's reply to the restriction of r to A, if the last move was made in B then play σ's reply to the restriction of r to B.

Exercise 2.19 *(1) Write out a formal definition of $\rho \otimes \sigma$ in the setting of your choice.*

(2) Show that $- \otimes -$ is a functor $\mathbf{Gam} \times \mathbf{Gam} \longrightarrow \mathbf{Gam}$.

It should be intuitively clear that $A \otimes B \cong B \otimes A$ and that $(A \otimes B) \otimes C \cong A \otimes (B \otimes C)$.

Proposition 2.20 *The category* **Gam** *is symmetric monoidal.*

Exercise 2.21 *Prove Proposition 2.20.*

Moreover, tensor product and linear function space are related by an adjunction.

Proposition 2.22 *The category* **Gam** *is symmetric monoidal closed.*

Proof. Since morphisms $A \otimes B \longrightarrow C$ correspond to strategies on $(A \otimes B) \multimap C$ and morphisms $A \longrightarrow B \multimap C$ correspond to strategies on $A \multimap (B \multimap C)$ it is sufficient to establish that

$$(A \otimes B) \multimap C \cong A \multimap (B \multimap C),$$

provided the isomorphism is natural. A position in $(A \otimes B) \multimap C$ is given by a sequence of positions from $(A \otimes B)$ (excluding the initial position $*_{A \otimes B}$) and positions from C such that

- the restriction to positions from $A \otimes B$, prefixed with the initial position, gives a valid play of $A \otimes B$ and
- the restriction to positions from C gives a valid play of C;
- positions with odd indices are elements of $(A \otimes B)^P + C^O$;
- positions with even indices are elements of $(A \otimes B)^O + C^P$.

This can be further simplified by using sequences over positions from A, B and C, coding the same information. A position in $(A \otimes B) \multimap C$ is given by a sequence of positions from A (excluding the initial position), B (excluding the empty position) and C such that

- the restriction to positions from A, prefixed with the initial position, gives a valid play of A;
- the restriction to positions from B, prefixed with the initial position, gives a valid play of B and
- the restriction to positions from C gives a valid play of C;
- positions with odd indices are elements of $A^P + B^P + C^O$;
- positions with even indices are elements of $A^O + B^O + C^P$.

It is easy to see that positions of $A \multimap (B \multimap C)$ are given in precisely the same way. $\square$

Proposition 2.23 *The functor* **Gam** $\longrightarrow$ **Rel** *from Theorem 2.12 is monoidal.*

Exercise 2.24 *Prove Proposition 2.23.*

Exercise 2.25 *For games A and B define a directed graph as follows. Nodes are pairs of positions (a, b) from A and B respectively, and there is an edge $(a, b) \longrightarrow (a', b')$ if and only if one of the following is the case*

- *$a = a'$ and $b \longrightarrow b'$ in B or*
- *$a \longrightarrow a'$ in A and $b = b'$.*

Convince yourself that this new graph is acyclic and has one node from which each other node can be reached in at most finitely many steps. In this graph identify two subgraphs which are reminiscent of $A \multimap B$ and $A \otimes B$ in the following sense: Any directed graph connected graph with a unique 'source node' can be made into a tree by choosing paths in the graphs as nodes with connections between such graphs if and only if the second extends the first by precisely one node. Applying this process to the subgraphs identified yields $A \multimap B$ and $A \otimes B$ respectively.

This idea suggests that it might be possible to define a category played on graphs rather than on trees. In [HS99], a sketch of such a category is given.

Exercise 2.26 Speculative. *The tree constructed in Exercise 2.25 is reminiscent of 'unfolding' a labelled transition system* lts *into a tree. Can you define a notion of 'game' arising from ltss? Can you make this a functor with, for example, simulations as morphisms? Does this functor preserve any of the structure?*

3 Additive structure

The product of two games A and B is easy to describe. Opponent chooses between A and B by playing one of the opening moves of these games, and thereafter play continues in that game.

Definition 6 *The product $A \times B$ of two games A and B has*

- *P-positions $A^P \setminus \{*_A\} + B^P \setminus \{*_B\} + \{*_{A\times B}\}$;*
- *O-positions $A^O + B^O$.*

There is a move $c \longrightarrow c'$ if an only if both c and c' are from A (B) and there is such a move in A (B) or if $c = \{*_{A\times B}\}$ and there is a move from the initial position in A or B to c'.

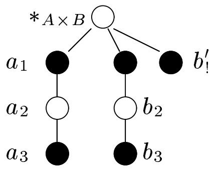

Figure 21: The product $A \times B$; compare Figures 8, 9

The projection $\pi_1\colon A \times B \longrightarrow A$ is similarly easy to describe. Opponent opens play in A and Player copies that move across to $A \times B$. Thereafter he plays a copy-cat strategy on A.

Definition 7 *Describe $A \times B$ for Alternative 2 of Definition 1. Further find the relational description of the projections. Remind yourself of how the categorical product on* **Rel** *works. What do you conclude about the functor* **Gam** $\longrightarrow$ **Rel**?

We have to show that the universal property for products is satisfied. So let us assume we have morphisms $\rho\colon C \longrightarrow A$ and $\sigma\colon C \longrightarrow B$. Define $\langle\rho,\sigma\rangle\colon C \longrightarrow A \times B$ as follows: If Opponent opens $A \times B$ by playing in A then play ρ, if Opponent opens in B play σ.

Exercise 3.1 *Show that $A \times B$ is indeed the product of A and B in* **Gam**.

Note that infinite products can be built in just the same way. This leaves us to ensure that there is a terminal object or 'empty product' in **Gam**. The game **I** that consists of just the initial position is also the terminal object. Indeed, $A \multimap \mathbf{I}$ is isomorphic to **I** again, since there is no opening move in **I** which would get the game going. Hence there is precisely one strategy on $A \multimap \mathbf{I}$, namely the one that does nothing, and therefore there is precisely one morphism $A \longrightarrow \mathbf{I}$ for every game A.

Proposition 3.2 *The category* **Gam** *has all products.*

Exercise 3.3 *Products in* **Rel** *are given by disjoint sums; the projections by the dual of the relations* inl *and* inr. *The functor* $\mathbf{Gam} \longrightarrow \mathbf{Rel}$ *does not preserve products—why? Show that it is at least monoidal with respect to the product structure. Can you come up with a variation of a definition of a game such that the corresponding functor preserves products? What happens to the monoidal structure in that case?*

4 The problem of dualization

This category of games seems to come with a natural idea of what the dual of a game might be—just changing the roles of Player and Opponent seems to do just the kind of thing that is envisioned. Certainly it would be the case that taking the dual twice is the same as doing nothing at all. Of course, in our setup, this candidate for a dual is not actually a game, but something we called a 'cogame'. There is a good reason for this, namely that the 'obvious' idea of getting this to work will not result in a category. This section is dedicated to showing why this is the case.

We would like to allow games where Player starts as well as those where Opponent does. In order to make these the objects of a category we have to think about the new definition of a linear function space (so that we can define morphisms between these new 'games'). We call such 'games' *generalized games.*

Definition 8 Alternative 3. *Let A^P and A^O be sets of so-called* positions *and let $*_A$ be one such position. A* generalized game *is an acyclic (A^P, A^O)-bipartite graph such that every node has a finite path to $*_A$.*

If the linear function space is to be a generalized game then it must be the case that at any position in the generalized game $A \multimap B$ it is precisely one player's turn, including right at the beginning. Hence we have to decide who should start which game. We use the idea once again that we wish to play the two games 'in parallel' (interleaving) and that Player should take the role of Opponent in the game A. In other words, as before, positions in $A \multimap B$ should be sequences of positions ensuring that Player has Opponent's moves in A and Player moves in B, and such that the projections onto the constituent games gives valid plays in A and B respectively. The only point where problems arise are at the start of the game. The reason for that is that we would like to keep the 'switching conditions' intact. This is desirable because it is the reason that our composition for (ordinary) games actually works—it induces a protocol that carefully transfers control between the two strategies to be composed in such a way that precisely one of these determines the next move in A, B, or C.

The following table gives an overview whose turn it is in which game at the start of the proposed $A \multimap B$ for all four combinations.

	P starts in B	O starts in B
P starts in A	O's turn in A P's turn in B	O's turn in A O's turn in B
O starts in A	P's turn in A P's turn in B	P's turn in A O's turn in B

We use (O,O) to indicate a case where it is O's turn to start in both, A and B. Clearly it must be O to start in the (P,O) case and P to start in the (O,P) case. That leaves us with a decision to make in the cases (P,P) and (O,O). If we wish to stick with the idea that Player is the only one who is allowed to switch between games then clearly we must have Opponent start in both these cases. This will ensure that after Opponent has made an opening move it will be Player's turn in both games. But looking back at the first two cases, we find a problem here: If it is Opponent's turn in both games as is the case for (P,O) then he has to start only one of them, and as a consequence Player will be required to answer in whichever game Opponent played last—he cannot switch, but after his move, Opponent can. This can be fixed quite easily: We just demand that Opponent start in *both* games A and B, so that thereafter Player is in control of the switching. We summarize below who starts where in the four cases.

	P starts in B	O starts in B
P starts in A	O starts in A	O starts in A and B
O starts in A	P starts in A or B	O starts in B

Clearly this is not quite the same thing as copying Definition 3 and merely replacing 'game' by 'generalized game', but it should be clear how that definition can be adjusted in the appropriate way. The resulting generalized game $A \multimap B$ does satisfy the switching condition.

We would like to define a category of generalized games where morphisms are strategies on this linear function space. We have laid the foundations for this by ensuring that the 'switching condition' holds on our proposed linear function space. So assume that we have $\rho\colon A \multimap B$ and $\sigma\colon B \multimap C$ for generalized games A, B and C. There are eight cases to be considered here, which we will abbreviate by triples indicating which player starts in which of the constituent games. The underlying idea for the composite is precisely the same as before—we use plays in all three games A, B and C to determine $\sigma \circ \rho$. Rather than giving a formal definition we will just convince ourselves that control is passed between ρ and σ as before. Once the game has been started the transfer of control obviously can be handled just as before (due to the switching condition holding), and we only have to make sure that at the start, precisely one ρ and σ can take control, that is can suggest a valid move.

Here are the eight cases.

(P,P,P): O starts in A and σ is waiting for the opening move in B, so ρ has control.

(P,P,O): O starts in both, A and C. σ expects an opening move in both, B and C, so ρ has control.

(P,O,P): O starts in A, but ρ expects an opening move in both, A and B. P starts in $B \multimap C$, so σ has control.

(P,O,O): O starts in both, A and C. ρ expects an opening move in both, A and B, so σ has control.

(O,P,P): P can choose whether to start in A or in C, but σ is waiting for O to start in B, so ρ has control and P's opening move will be in A.

(O,P,O): O starts in C. σ expects O to start in both, B and C, whereas ρ is prepared to start in one of A and B. If ρ's first move is in A then so will be $\sigma \circ \rho$'s; if ρ's first move is in B then where $\sigma \circ \rho$'s first play takes place will depend on σ's reply to the combined opening moves for B and C played so far. In any case, ρ has control at the start.

(O,O,P): P may start in A or C, but ρ is waiting for an opening move in B. Hence σ has control.

(O,O,O): This is the original case. O starts in C, and σ has control.

So this looks good. In order for this to be a category, we have to show that there is an identity for each object and that composition is associative. The identity will have to live on a game of pattern (P,P) or (O,O), and since in either case Opponent starts in precisely one copy A (but not the other), we can play a copy-cat strategy. This will behave as desired. It remains to consider associativity of composition. And that is the point where it goes wrong.

Consider games A, B, C and D of pattern (P,P,O,O). Let $\rho\colon A \multimap B$, $\sigma\colon B \multimap C$ and $\tau\colon C \multimap D$. Since the pattern determining how $\sigma \circ \rho$ starts is (P,P,O), this strategy starts with ρ in control. The pattern for $\tau \circ (\sigma \circ \rho)$ is (P,O,O), so τ has control. In particular, if τ opens with a move in C, so does $\tau \circ (\sigma \circ \rho)$. On the other hand, for $\tau \circ \sigma$ we have the pattern (P,O,O), so τ has control. For $(\tau \circ \sigma) \circ \rho$, the pattern switches to (P,P,O), which means that control lies with ρ. In particular, if ρ opens with a move in A, then so does $(\tau \circ \sigma) \circ \rho$. Clearly that means that $(\tau \circ \sigma) \circ \rho$ and $\tau \circ (\sigma \circ \rho)$ are different strategies, one opening with a move in A, the other with a move in C.

Hence we cannot define a category of generalized games in this way. In other words, we cannot make the category of games **Gam** $*$-autonomous by adding in the 'obvious' dual objects for games and keep the linear function space much as it was.

It is difficult to say just where it goes wrong in terms of the definitions that have been made. Clearly there is some hope that this might be fixable if Player is allowed to open both games simultaneously, in which case Opponent would have to reply in both games in order to maintain the idea of control.

Exercise 4.1 *What happens if you try to take generalized games and map them to the category of sets and relations? What happens to morphisms? Can you see more easily why composition fails to be associative?*

Exercise 4.2 *(1) Examine the category* **Gam** $\times$ **Gam**$^{\mathsf{op}}$ *as another attempt of adding a dualization operation. If you are not sure how to define the $*$-autonomous structure on this category consult [HS01].* **Speculative.** *Is it possible to view this category with a formal dual as a category of games?*

(2) **Speculative.** *Is it possible to define a linear function space where both games are played truly in parallel, that is moves are pairs of moves, one in each game? What kind of structure is achieved in this way?*

(3) **Speculative.** *Is it possible to define a function space for generalized games in which moves are moves from either game or pairs of moves? Is the result a category?*

(4) **Speculative.** *Define a category of generalized games where morphisms are strategies on the linear function space only for pattern (P, P) and (O, O), and where all other hom-sets are empty. What is the structure of the resulting category?*

5 Linear exponential structure

Finding a linear exponential structure for any category is usually quite a bit harder than coming up with, say, a symmetric monoidal closed on. The existence of finite products, of course, is a simple matter of 'yes' or 'no'—either they exist or they do not.

In our search for a suitable such structure we let ourselves be guided by the idea that $!A \multimap B$ should give us some kind of 'intuitionistic function space'.

We will contrast this idea with that of the linear function space $A \multimap B$. If we think of a Player strategy τ on $A \multimap B$ taking a Player strategy ρ on A as an argument then it is clear that in any one play of $A \multimap B$, Player only is allowed to 'run through' no more than one play of ρ. This is an extremely limited way of being allowed to examine the argument which is more in the spirit of a linear world. (Note, however, that nothing forces τ to look at ρ at all, if play on $A \multimap B$ according to τ occurs exclusively in B.)

So for $!A \multimap B$ to overcome this difficulty there must be numerous copies of A available, so that any argument living on A can be examined from all angles. The simplest way of providing this is to make $!A$ something akin to an infinite tensor product: Then in $!A \multimap B$ Player is allowed to start as many new copies of A as he likes. However, it is not quite an infinite tensor product that we are after here—we merely want Opponent to be able to start multiple copies of A when playing $!A$. This gives us an order on these copies ('the first copy started', 'the second copy started', and so on), and therefore we use an infinite *ordered* tensor product.

Definition 9 *The game $!A$ has as positions sequences $r_1 \cdots r_n$ of positions (a, i) where a is a position of A excluding the initial position and $i \in \mathbb{N}$ such that*

- *for every $i \in \mathbb{N}$ the restrictions to positions with second component i, $r_1 \cdots r_n|_i$, gives a valid play of A when prefixed by the initial position $*_A$;*
- *for every $i \in \mathbb{N}$, if a position with second component $i+1$ occurs then there is an earlier position with second component i in the sequence;*
- *for $m \in \mathbb{N}$ even, $t_m \in A_P$;*
- *for $m \in \mathbb{N}$ odd, $t_m \in A_O$.*

The set of P-position $(!A)^P$ is given by strings of even length and the set of O-positions $(!A)^O$ is given by strings of odd length. The initial position is the empty string. Edges exist precisely between positions $r_1 \cdots r_n \longrightarrow r_1 \cdots r_n r_{n+1}$.

An example for such a game is given in Figure 22. The labels marking the positions are not the entire strings, but only the last element of the corresponding string. To obtain the proper label for a position just take all consecutive labels from the root (excluding $*_{!B}$).

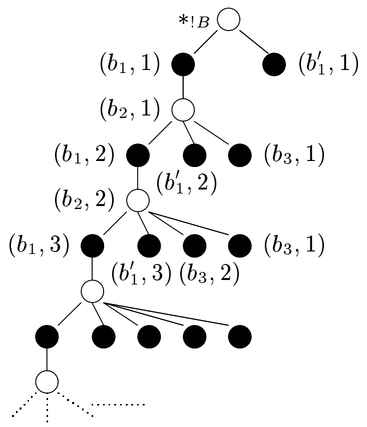

Figure 22: The linear exponential $!B$; Figure 9

Exercise 5.1 *Write a definition of* $!A$ *based on Alternative 2 for the definition of a game.*

Proposition 5.2 *For the linear exponential* $!A$ *is is the case that*

- *only Opponent is allowed to switch between different copies of* A*, hence Player is required to respond in whichever copy Opponent played last;*
- *Opponent may only start play in the* i*th copy of* A *if all copies with a smaller index have been started already. Hence Opponent starts with an opening move in the first copy of* A*.*

We have to establish the existence of the various structure maps forming a linear exponential comonad as defined in Definition 19. We will only argue this informally since proving all the equations formally would make this section a lot longer. We start with the definition making ! a functor. Let $\tau\colon A \longrightarrow B$, that is $\tau\colon A \multimap B$. The game $!A \multimap !B$ works as follows. Opponent opens with an opening move in (the first copy of) B. Player is now allowed to start the first copy of A or to continue playing in B as he chooses—he will take action in accordance with τ. If he does the former then Opponent is obliged to answer in the first copy of A, otherwise he may play in the first copy of B or open a second copy of B. Opponent has control over opening copies of B while Player does so for A. Player plays the strategy $!\tau$ on $!A \multimap !B$ as follows. For every copy of B that Opponent opens, Player starts a copy of τ. Whenever τ requires him to open (a copy of) A he does so, keeping track of which copy of A corresponds to which copy of B and treating them as one copy of $A \multimap B$. As Opponent switches between copies of B, Player switches between the corresponding copies of τ, which in turn lead to corresponding copies of A. In other words, Player secretly treats the game $!A \multimap !B$ like $!(A \multimap B)$ by keeping a list of indices that tell him which copies of A correspond to which copies of B. On $!(A \multimap B)$, he just plays τ on each copy of $A \multimap B$. Note that this correspondence may differ from play to play.

Exercise 5.3 *Define games A and B and a strategy $\tau\colon A \multimap B$ such that*

- *there are plays of $!\tau$ in which the first and second copy of B correspond to the first and second copy of A respectively;*
- *there are plays of $!\tau$ in which the first copy of B corresponds to the second copy of A, and the second copy of B corresponds to the first copy of A.*

Formally, the reply of $!\tau$ to some O-position $t_1 \cdots t_n$ of $!A \multimap !B$ is found as follows. If t_n is a position in $!A$, say in the ith copy of A, then find the corresponding copy of B as follows: Find the opening move of the ith copy of A. The immediately preceding position occurs in, say, the jth copy of B. Restrict $t_1 \cdots t_n$ to those positions which are either of the form (a, i) or of the form (b, j). Ignoring i and j, this is a valid play of $A \multimap B$. Find τ's reply to this and play accordingly. If t_n is a position in $!B$, say in the jth copy of B, then proceed as follows. If there is an O-position in this jth copy of B that is followed by a position in a copy of A, say the ith copy, then restrict play to these two copies as before, determine τ's answer, and make the appropriate move. Otherwise find τ's reply to the restriction to positions which are in the jth copy of B. If the reply is in B, continue accordingly. Otherwise the reply is an opening move of A which we play in a newly created copy of A.

Exercise 5.4 *Show that once i and j have been determined as above, restricting a sequence to positions from the ith copy of A and the jth copy of B gives a play of $A \multimap B$ in accord with τ. In other words, the correspondence between copies of A and copies of B can be derived as described.*

We next turn to the counit of the comonad we would like to build around the functor $!$. Consider the game $!A \multimap A$. Opponent opens by making a move in A. Now Player can copy this move by opening a first copy of A in $!A$. Opponent has to reply to that move in the same game, and this answer can be copied back to A. In other words, Player employs a copy-cat strategy between A and the first copy of A in $!A$, and no further copies of A are ever opened. We define ϵ_A to be this strategy.

We have to show that ϵ is natural, that is that for all $\tau\colon A \longrightarrow B$, $\tau \circ \epsilon_A = \epsilon_B \circ !\tau$. Now it pays off that we have a faithful functor F to the category of sets and relations: To show that this equation holds we merely have to show that $F(\tau) \circ F(\epsilon_A) = F(\epsilon_B) \circ F(!\tau)$. Clearly $F(\epsilon_A) = \{((a_1, 1) \cdots (a_n, 1), a_n) \mid *_A a_1 \cdots a_n \text{ is a play in } A\}$. Therefore

$$F(\tau) \circ F(\epsilon_A) = \{((a_1, 1) \cdots (a_n, 1), b) \mid (a_n, b) \in F(\tau),\ *_A a_1 \cdots a_n \text{ is a play in } A\}.$$

When calculating the composite $\epsilon_B \circ !\tau$, only one copy of B is ever started in the 'hidden' game $!B$. Hence at most one copy of A is started in the game $!A$, namely the first one, and the second composite, $\epsilon_B \circ !\tau$, evaluates as the same set.

This brings us to the comultiplication of our proposed comonad. For that we have to define a morphism $\delta_A\colon !A \longrightarrow !!A$ for each game A. At first sight, $!!A$ may look as if it allowed more options than $!A$, but this is not really true: In the end, a countable number of copies of A are available in each one. The reason for this is that Opponent opens $!!A$ by opening a first copy of $!A$, which is done by opening a first copy of A. While on paper Opponent has the choice between opening a new copy of $!A$ or a copy of A in an existing copy of $!A$, it all comes down to opening new copies of A in the end. Hence Player can play a copy-cat strategy $!A \longrightarrow !!A$, opening a new copy of A whenever Opponent does the same.

All Player has to do is to keep track of which copies of A in $!A$ (these will be indexed by some $i \in \mathbb{N}$) corresponds to which copy of A in $!!A$ (these will be indexed by some (j,k)). This correspondence can at any time be derived from a given position. We take δ_A to be this copy-cat strategy. Notice once again that the correspondence between the various copies of A may vary in different plays.

Exercise 5.5 *Write out a more detailed definition of δ_A and show that δ is natural as desired. (Doing this very rigorously may be quite arduous!)*

The comonad equations are not difficult to establish informally. For example the composite $!\delta_A \circ \delta_A \colon !A \longrightarrow !!!A$ establishes a correspondence between copies of A in the two games; it is the same as that established by $\delta_{!A} \circ \delta_A \colon !A \longrightarrow !!!A$. We next consider $\epsilon_{!A} \circ \delta_A \colon !A \longrightarrow !A$. Opponent opens play in the first copy of A in $!A$ and $\epsilon_{!A}$ copies this move to the first copy of A in the first copy of $!A$ in $!!A$. δ_A copies this move across to the first copy of A in $!A$. Ultimately this corresponds to the copy-cat strategy on $!A$. Note that because of the way $\epsilon_{!A}$ operates, only the first copy of $!A$ in $!!A$ is ever opened. Thus play is copied between $!A$, the first copy of $!A$ in $!!A$, and $!A$ again. Hence the nth copy of A in the covariant copy of $!A$ will clearly correspond to the nth copy of A in the contravariant one. It is not much more difficult to make an argument that $!\epsilon_A \circ \delta_A$ also is the identity on $!A$.

This brings us to the monoidal structure. First for the morphism $m_{\mathbf{I}} \colon \mathbf{I} \longrightarrow !\mathbf{I}$. Since $!I \cong \mathbf{I}$, and $\mathbf{I}$ is the terminal object, there is only one such morphism which we take. More interestingly, consider $(!A \otimes !B) \multimap !(A \otimes B)$. Opponent opens play by making an opening move from A or from B. This move is copied across, and the answer is copied back. Again it is a question of keeping up a correspondence between copies of A and B on either side of $\multimap$. Note that the copy of A corresponding to the A-component of the ith copy of $A \otimes B$ will not necessarily have the same index as the copy of B corresponding to the B-component of the same copy of $A \otimes B$.

It is worthwhile to note that $!A \otimes !B$ and $!(A \otimes B)$ are in general not isomorphic. Figure 23 shows an example for this.

Exercise 5.6 *Convince yourself that keeping up these correspondences as described leads to a valid play of $(!A \otimes !B) \multimap !(A \otimes B)$. In particular it is necessary to show that any move can be copied across to the 'right' copy of A or B. You might also want to convince yourself that $!A \otimes !B$ and $!(A \otimes B)$ are in general not isomorphic.*

Exercise 5.7 *Can you show that one cannot define a natural transformation with components $n_{A,B} \colon !(A \otimes B) \longrightarrow !A \otimes !B$?*

We will not try to prove here that $(!, \epsilon, \delta, m_{\mathbf{I}}, m)$ is a monoidal comonad, but the reader might well want to go through at least some of the equations involved.

It thus remains to introduce the comonoid structure on free coalgebras for the comonad. This is very easy in the case of the counit, since $!A \multimap \mathbf{I} \cong \mathbf{I}$ means that all $e_A \colon !A \longrightarrow \mathbf{I}$ will be the only strategy on this game. The comultiplication, on the other hand, is more interesting.

Consider $!A \multimap (!A \otimes !A)$. Again we use something akin to a copy-cat strategy, which should be no surprise by now. All Player does is to match copies of A in $!A \otimes !A$ with copies of

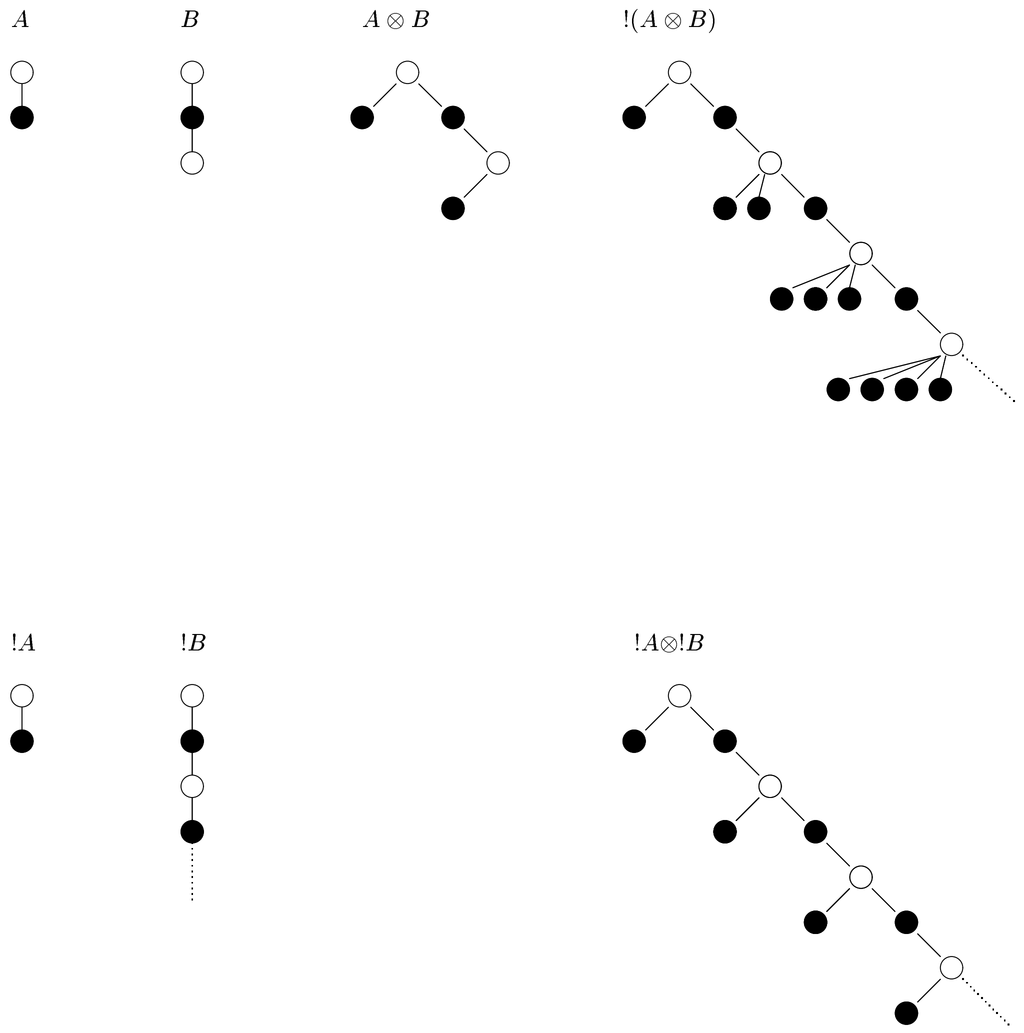

Figure 23: $!A \otimes !B$ versus $!(A \otimes B)$

A in $!A$, much as before. Establishing symmetry and associativity of the resulting operation should not pose any difficulties.

But the unit laws are no more difficult than that. Consider the game $!A \multimap (!A \otimes \mathbf{I})$ and the strategy given on it by $(\mathsf{id}_{!A} \otimes e_A) \circ d_A$. Opponent opens play in the first copy of A in $!A \otimes \mathbf{I} \cong !A$. This is copied across to the first copy of A in the left copy of $!A$ in $!A \otimes !A$ by $\mathsf{id}_A \otimes e_A$. Now d_A chooses this move to open the first copy of A in $!A$. Opponent can only switch between copies of A in $(!A \otimes \mathbf{I})$, but not in $!A$. Whenever he opens a new copy of A there, this results in a new copy of A being opened by Player in $!A$, with the same move. All in all this amounts to the isomorphism ρ_A as desired.

It is instructive to check that there is no natural transformation going the opposite way, that is with components $!A \otimes !A \longrightarrow !A$. The reason for this is that Player is forced to make a choice here which copies of A in $!A$ to pair with which in $!A \otimes !A$, and this time, there is no canonical way of doing so. Consider games A and B where A is the one-move game S from above, and B has precisely two opening moves for Opponent (and no replies). Note that

$!A \cong A$ and $!B \cong B$. Then defining a non-trival morphism $!A \otimes !A \longrightarrow !A$ is the same as defining one $A \otimes A \longrightarrow A$. This amounts to deciding in which copy of A (in $A \otimes A$) to copy this move. A similar choice has to be made for B, only now there are two options (all others can be reduced to those). It is not difficult to come up with morphisms $A \longrightarrow B$ that show that neither of these choices results in a natural transformation.

Exercise 5.8 *Carry out the calculations that show that the above outline is correct, that is, there is no way of defining a copy-cat strategy* $!A \longrightarrow !A \otimes !A$ *which results in a natural transformation.*

We will not establish any more of the equations needed to prove Theorem 5.9, but the reader is invited to select some of them and show that they are valid.

Theorem 5.9 **Gam** *is a model of intuitionistic linear logic.*

Exercise 5.10 *Show that the functor* $F \colon \mathbf{Gam} \longrightarrow \mathbf{Rel}$ *is linearly distributive with respect to the linear exponential structure on* **Rel** *introduced in Exercise A.13.*

As we have seen in Theorem A.31 this is sufficient to define a cartesian closed category. We set $A \to B = !A \multimap B$. Let $\mathbf{Gam}_!$ be the category whose objects are games and where morphisms $A \longrightarrow B$ are given by Player strategies on $A \to B$. The identity is the copy-cat strategy on $!A \multimap A$ that copies between the first copy of A in $!A$ and A, and never opens any further copies (in other words, ϵ_A from above). The composite of $\rho \colon A \longrightarrow B$ and $\sigma \colon B \longrightarrow C$ in $\mathbf{Gam}_!$ is given by $\sigma \circ !\rho \circ \delta_A$. In a play according to this composite, Opponent opens play in C. Control then falls to σ. If σ answers in C then so does the composite. Otherwise a copy of B is opened, and control passes to $!\rho$. A copy of ρ is started and interacts with σ until an answer is reached. Whenever σ opens a new copy of B, a new copy of ρ is started, leading to new copies of A as required.

Exercise 5.11 *You might want to give an explicit definition of composition in* $\mathbf{Gam}_!$*, or at least try your hand at an example.*

The following is an immediate corollary of Theorem 5.9 and Theorem A.31.

Theorem 5.12 *The category* $\mathbf{Gam}_!$ *is cartesian closed.*

One criticism regarding this linear exponential is that it goes a step further than our original motivation indicated. For every game A which has at least one move each for Player and Opponent it is clearly the case that $!A$ is infinite. Hence we rather overshoot the original target of being able to examine a strategy on A viewed as the argument of a morphism $A \longrightarrow B$: If A is finite then every such strategy only contains finitely many decisions, and thus can be fully examined within finitely many copies of A. Recently there have been attempts at defining a linear exponential that is less crude.

The idea is as follows: Consider the set of strategies on A as the set of P-positions of $!'A$. O-positions are of the form (ρ, a) where ρ is a strategy on A and $a \in R_P(\rho)$ such that $\rho(a)$ is undefined. If ρ is such a strategy then it has moves to O-positions of the form (ρ, a). If (ρ, a) is an O-position of $!'A$ then it has moves precisely to those strategies ρ' which are minimal extensions of ρ such that $\rho'(a)$ is defined. (In other words the only difference between ρ and

ρ' is that ρ' has an answer to the position a. In general this will result in a directed acyclic bipartite graph rather than a tree. Such a graph can be treeified; compare Exercise 2.25. Clearly for a finite game A it will be the case that $!'A$ is finite. The structure maps can now be defined in a similar way to the ones for $!$.

Exercise 5.13 *Show that another linear exponential comonad on* **Gam** *can be defined in this way. (Showing naturality may be trickier than it seems!)*

A What is a categorical model for Linear Logic?

The aim of this section is to give an outline of the categorical structure required for a model of linear logic. That means that we only try to motivate the various conditions imposed on such models; this is by no means intended to be a full account. For intuitionistic linear logic a detailed description of the process of defining such a model can be found in Gavin Bierman's thesis [Bie94]. Such a semantics has to be subject to a valuation assigning objects to any variables any of these formulae contain. In this section we will take that for granted and not mention the valuation any longer—the reason for that is that any interpretation should be sufficiently independent from the actual valuation that no reference is made to any properties of the objects appearing in same.

A.1 Intuitionistic linear logic

When people talk about intuitionistic linear logic they mean the calculus with the connectives $\otimes$, $\multimap$, $\sqcap$ and $!$. Since there is no negation, the De Morgan duals of these can not be defined within this system. Further the system only allows *one-sided* judgements to be made. For a full discussion of the rules see Bierman's thesis [Bie94].

MILL—the multiplicative fragment

The rules for this fragment are given in Table 1, with the exception of the *exchange rule* which allows the exchange of any two neighbouring formulae in a sequent Γ.

Identity	Multiplicative connectives		Cut	Constants
	left	right		
$\dfrac{}{A \vdash A}$	$\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$ $\dfrac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}$	$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$ $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}$	$\dfrac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}$	$\dfrac{}{\vdash \mathbf{I}}$

Table 1: Rules for MILL.

Exercise A.1 *Compare this system to* $\mathsf{N}\{\to, \sqcap\}$, *a natural deduction system for implication and conjunction. What are the differences, what the similarities? Can you deduce some categorical structure suitable for modelling the above from this?*

We use the left rule for $\otimes$ is to justify modelling a judgement $\Gamma \vdash B$ as a morphism

$$A_1 \otimes \cdots \otimes A_m \longrightarrow B \qquad \text{in the category } \mathbf{C},$$

where Γ is the sequence $A_1, \ldots, A_m$. We are abusing notation here in that we expect the object denoting the formula A to be called A again. We will use $[\![\Gamma]\!] = A_1 \otimes \cdots \otimes A_m$. Obviously this is subject to an evaluation given to interpret any variables in the sequence.

It should we obvious that we need constructions on the objects of the category that mimic the formation of formulae, that is interpretations for the operations *tensor* and *linear implication*. Again we are stingy with notation and use the same symbols to denote the categorical operations.

The first thing to note is that these should not be constructions *just* on the objects, but that they should also work for morphisms—that is they should be functors of some description. With our chosen setting being categories this should come as no surprise, but we can, in fact make an argument for this.

For example, the rule

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$$

means that given morphisms

$$[\![\Gamma]\!] \longrightarrow A \qquad \text{and} \qquad [\![\Delta]\!] \longrightarrow B$$

we have to be able to define a morphism

$$[\![\Gamma, \Delta]\!] = [\![\Gamma]\!] \otimes [\![\Delta]\!] \longrightarrow A \otimes B$$

and if $\otimes$ is a bi-functor on $\mathbf{C}$, that is a functor $\mathbf{C} \times \mathbf{C} \longrightarrow \mathbf{C}$, then this is taken care of for us automatically. Functoriality of this assignment means the following: The leaf rule $A \vdash A$ is interpreted by the identity on (the interpretation of) A. So preservation of identities by $\otimes$ means that the identity on $A \otimes B$ is used to interpret

$$A, B \vdash A \otimes B,$$

which itself is a consequence of $A \vdash A$ and $B \vdash B$. Since from there we can, in fact, prove that $A \otimes B \vdash A \otimes B$ this seems to make sense.[3]

This leaves us with the question of why $\otimes$ should also preserve composition. Composition in the category is the interpretation of the Cut rule:

$$\frac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}.$$

From arrows

$$f\colon [\![\Gamma]\!] \longrightarrow A \qquad \text{and} \qquad g\colon A \otimes [\![\Delta]\!] \longrightarrow B$$

we have to define an arrow

$$[\![\Gamma]\!] \otimes [\![\Delta]\!] \longrightarrow B,$$

and fortunately there's something to fill the gap:

$$[\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{f \otimes \mathsf{id}_{[\![\Delta]\!]}} A \otimes [\![\Delta]\!] \xrightarrow{g} B.$$

[3]If we only allow axioms to introduce literals (say p, q) then tensor preserving the identity amounts to the identity morphism on $p \otimes q$ interpreting $p \otimes q \vdash p \otimes q$, which can be proven from $p \vdash p$ and $q \vdash q$.

Exercise A.2 *Find a justification for demanding that* $\otimes$ *preserve composition.*

Because lists of formulae are *associative* by definition, we expect $\otimes$ to be associative. Further, because of the exchange rule we expect it to be *symmetric*. And finally we need an 'empty' version of the tensor product in order to interpret judgements which have an empty context. We refer to this object as $\mathbf{I}$, and it is clear that it should behave like a units for $\otimes$.

Definition 10 *We say that a category* $\mathbf{C}$ *is (symmetric) monoidal if there is a bi-functor* $\otimes$ *on* $\mathbf{C}$ *that is associative, (symmetric) and has a unit* $\mathbf{I}$. *By that we mean that there exist natural isomorphisms* α, $(\sigma,)$ ρ *and* λ *such that the diagrams given below commute. These isomorphisms 'mediate' between objects providing us with associativity, symmetry and unit laws. Their naturality is expressed in the following set of diagrams.*

$$\begin{tikzcd}
A \otimes (B \otimes C) \arrow[r, "\alpha_{A,B,C}"] \arrow[d, "f\otimes(g\otimes h)"'] & (A \otimes B) \otimes C \arrow[d, "(f\otimes g)\otimes h"] \\
A' \otimes (B' \otimes C') \arrow[r, "\alpha_{A',B',C'}"'] & (A' \otimes B') \otimes C'
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes B \arrow[r, "\sigma_{A,B}"] \arrow[d, "f\otimes g"'] & B \otimes A \arrow[d, "g\otimes f"] \\
A' \otimes B' \arrow[r, "\sigma_{A',B'}"'] & B' \otimes A'
\end{tikzcd}
\quad
\begin{tikzcd}
\mathbf{I} \otimes A \arrow[r, "\lambda_A"] \arrow[d, "\mathsf{id}_{\mathbf{I}}\otimes f"'] & A \arrow[d, "f"] \\
\mathbf{I} \otimes A' \arrow[r, "\lambda_{A'}"'] & A'
\end{tikzcd}
\quad
\begin{tikzcd}
A \otimes \mathbf{I} \arrow[r, "\rho_A"] \arrow[d, "f\otimes\mathsf{id}_{\mathbf{I}}"'] & A \arrow[d, "f"] \\
A' \otimes \mathbf{I} \arrow[r, "\rho_{A'}"'] & A'
\end{tikzcd}$$

These natural isomorphisms interact with each other as given by the diagrams below.

$$\begin{tikzcd}
A \otimes (B \otimes (C \otimes D)) \arrow[rr, "\alpha_{A,B,C\otimes D}"] \arrow[d, "\mathsf{id}_A\otimes\alpha_{B,C,D}"'] & & (A \otimes B) \otimes (C \otimes D) \arrow[d, "\alpha_{A\otimes B,C,D}"] \\
A \otimes ((B \otimes C) \otimes D) \arrow[r, "\alpha_{A,B\otimes C,D}"] & (A \otimes (B \otimes C)) \otimes D \arrow[r, "\alpha_{A,B,C}\otimes\mathsf{id}_D"'] & ((A \otimes B) \otimes C) \otimes D
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes (B \otimes C) \arrow[r, "\alpha_{A,B,C}"] \arrow[d, "\mathsf{id}_A\otimes\sigma B{,}C"'] & (A \otimes B) \otimes C \arrow[r, "\sigma_{A\otimes B,C}"'] & C \otimes (A \otimes B) \arrow[d, "\alpha_{C,A,B}"] \\
A \otimes (C \otimes B) \arrow[r, "\alpha_{A,C,B}"] & (A \otimes C) \otimes B \arrow[r, "\sigma_{A,C}\otimes\mathsf{id}_B"'] & (C \otimes A) \otimes B
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes (\mathbf{I} \otimes C) \arrow[r, "\alpha_{A,\mathbf{I},C}"] \arrow[dr, "\mathsf{id}_A\oplus\lambda_C"'] & (A \otimes (\mathbf{I} \otimes C) \arrow[d, "\rho_A\otimes\mathsf{id}_A"] \\
 & A \otimes C
\end{tikzcd}
\quad
\begin{tikzcd}
A \otimes B \arrow[r, "\sigma_{A,B}"] \arrow[dr, "idA\oplus B"'] & B \otimes A \arrow[d, "\sigma_{B,A}"] \\
 & A \otimes B
\end{tikzcd}
\quad
\begin{tikzcd}
\mathbf{I} \otimes A \arrow[r, "\sigma_{\mathbf{I},A}"] \arrow[dr, "\lambda_A"'] & A \otimes \mathbf{I} \arrow[d, "\rho_A"] \\
 & A
\end{tikzcd}$$

Exercise A.3 *Show that the category of sets and relations* **Rel** *is symmetric monoidal. Hint: The tensor product is given by cartesian product. You might want to convince yourself that this is* not *the product in this category.*

Exercise A.4 *Convince yourself that a category with finite products is symmetric monoidal. Do the same for categories with finite coproducts.*

So in order to be able to interpret $\otimes$ we require our category $\mathbf{C}$ to be symmetric monoidal. The question of what a functor of symmetric monoidal categories should be can be answered in several ways. Maybe the most straightforward would be to demand that the symmetric monoidal structure is preserved (at least up to isomorphism), but that turns out to be too strong for many examples. Instead, we choose the following notion. Note that we do not distinguish between the tensor products in different categories in our notation, similarly for the unit $\mathbf{I}$. Also note that we are sloppy in our notation and write $- \otimes -$ for the functor $\mathbf{C} \times \mathbf{C} \longrightarrow \mathbf{C}$ although nothing in the notation suggests that we assume it to have two (possibly different) arguments rather than just one.

Definition 11 *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be symmetric monoidal categories.*

(1) A functor $F\colon \mathbf{C} \longrightarrow \mathbf{D}$ *is said to be* symmetric monoidal *if there exist a morphism* $m_{\mathbf{I}}\colon \mathbf{I} \longrightarrow F(\mathbf{I})$ *and a natural transformation* $m\colon F(-) \otimes F(-) \longrightarrow F(- \otimes -)$ *respecting the symmetric, associative, and unit structures.*

(2) Let F *and* G *be two symmetric monoidal functors* $\mathbf{C} \longrightarrow \mathbf{D}$ *with natural transformations* m *and* n *respectively. We say that a natural transformation* $\tau\colon F \longrightarrow G$ *is* symmetric monoidal *if the following diagrams commute.*

$$\begin{tikzcd}
\mathbf{I} \arrow[r, "m_{\mathbf{I}}"] \arrow[dr, "n_{\mathbf{I}}"'] & F(\mathbf{I}) \arrow[d, "\tau_{\mathbf{I}}"] \\
 & G(\mathbf{I})
\end{tikzcd}
\qquad
\begin{tikzcd}
F(A) \otimes F(B) \arrow[r, "m_{A,B}"] \arrow[d, "\tau_A \otimes \tau_B"'] & F(A \otimes B) \arrow[d, "\tau_{A \otimes B}"] \\
G(A) \otimes G(B) \arrow[r, "n_{A,B}"'] & G(A \otimes B)
\end{tikzcd}$$

Exercise A.5 *(1) If you are unsure about what it meant by 'respecting the ... structures' try drawing some diagrams.*

(2) Show that if F *is a symmetric monoidal endofunctor on a symmetric monoidal category* $\mathbf{C}$ *then so is* $F \circ F$*.*

This brings us to the next connective, *linear implication* $\multimap$. This connective is very tightly linked to the morphisms of our category $\mathbf{C}$ for the following reason. Any judgement $A \vdash B$ is interpreted by a morphism $A \longrightarrow B$. This judgement allows us to deduce $\vdash A \multimap B$, which is interpreted by a morphism $\mathbf{I} \longrightarrow A \multimap B$. Hence there is a connection between morphism $A \longrightarrow B$ and those $\mathbf{I} \longrightarrow A \multimap B$ which is tightened by the following derivation. Assume that we now have a derivation of $\vdash A \multimap B$.

$$
\dfrac{\dfrac{\vdots}{\vdash A \multimap B} \qquad \dfrac{\dfrac{}{A \vdash A} \quad \dfrac{}{B \vdash B}}{A, A \multimap B \vdash B}}{A \vdash B}\ Cut
$$

For our category that means that we must be able to go the other way, too. We expect the following one-to-one correspondence:

$$
\begin{array}{c}
A \longrightarrow B \\
\hline\hline
\mathbf{I} \longrightarrow A \multimap B
\end{array}
$$

or, more generally,

$$
\begin{array}{c}
C \otimes A \longrightarrow B \\
\hline\hline
C \longrightarrow A \multimap B
\end{array}
$$

Comparing this to the structure of a cartesian closed category we find a close resemblance: Assuming we make the usual naturality assumptions this means that the functor $- \otimes A$ is left-adjoint to $A \multimap -$. In a cartesain closed category the monoidal structure, that is the tensor product, is given by product, which makes it a stronger construct, but that is the only difference.

Definition 12 *A* symmetric monoidal closed category *is a symmetric monoidal category in which each functor $- \otimes A$ has a right adjoint.*

Note that the adjunction does impose more structure on the linear function space construct than one might think at first.

Exercise A.6 *(1) Show that in every symmetric monidal closed category $\mathbf{I} \multimap A \cong A$ naturally.*

(2) Show that in every symmetric monoidal closed category $(A \otimes B) \multimap C$ is naturally isomorphic to $A \multimap (B \multimap C)$.

Symmetric monoidal closed categories have all the categorical structure required to model the multiplicative fragment of intuitionistic linear logic.

Definition 13 *A* model for MILL *is any symmetric monoidal closed category.*

Exercise A.7 *Show that* **Rel** *is a symmetric monoidal closed category. Hint: the underlying set for $A \multimap B$ is again the cartesian product of A and B.*

Exercise A.8 *For every rule in Table 1 find the categorical counterpart, that is given the morphisms interpreting the hypothesis, find the morphism interpreting the conclusion. Do the same* vice versa*, that is for every equation imposed on a symmetric monoidal closed category, find a proof justifying it. (For this it is useful to recall how adjunctions can be described using equations only.)*

One might assume that something additional is required for a functor between symmetric monoidal closed (rather than merely symmetric monoidal) categories, but this is not the case.

Exercise A.9 *Assume that* **C** *and* **D** *are symmetric monoidal closed categories. Show that if $F\colon \mathbf{C} \longrightarrow \mathbf{D}$ is a symmetric monoidal functor then there exist natural transformations $F(A \multimap -) \longrightarrow F(A) \multimap F(-)$ and $F(- \multimap B) \longrightarrow F(-) \multimap F(B)$.*

MALL—the multiplicative additive fragment

We add one of the additve connectives to the multiplicative fragment MILL, namely the conjunction $\sqcap$. For that we add the following rules to those given in Table 1, and call the result MALL.

Additive connectives			Constants
left		right	
$\dfrac{\Gamma, A \vdash C}{\Gamma, A \sqcap B \vdash C}$	$\dfrac{\Gamma, B \vdash C}{\Gamma, A \sqcap B \vdash C}$	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \sqcap B}$	$\dfrac{}{\Gamma \vdash \mathbf{1}}$

Table 2: The rules for $\sqcap$

The categorical interpretation for the new connective $\sqcap$ certainly should be something similar to a tensor product. But looking at the left introduction rules we find that we need something stronger here. In particular given $C \vdash A$ and $C \vdash B$ we obtain $C \vdash A \sqcap B$—so given morphisms $C \longrightarrow A$ and $C \longrightarrow B$ we have to be able to construct a morphism $C \longrightarrow A \sqcap B$. This looks just like the requirement we demand for products, at least the part about being able to construct such morphisms. So how about the second requirement, the projections? From $A \vdash C$ we have to be able to conclude that $A \sqcap B \vdash C$—that is, given a morphism $A \longrightarrow C$ we have to be able to construct one $A \sqcap B \longrightarrow C$. If we do have a projection $A \sqcap B \longrightarrow A$ then we can simply precompose that with the given morphism to obtain the desired result. Similar for the symmetric version. Since the usual categorical symbol for a product is $\times$ we often write that rather than $\sqcap$ for the interpretation of that connective in the category. The interpretation of $\mathbf{1}$ is similarly determined. Clearly there will have to be a morphism $A \longrightarrow \mathbf{1}$ for every object A. We demand that there be precisely one of those for each A, making $\mathbf{1}$ the terminal object or 'empty product'.

Definition 14 *A* model for MALL *is given by a symmetric monoidal closed category with finite products.*

Exercise A.10 *Show that* **Rel** *is a model for* MALL*. Hint: the categorical product is given by the disjoint union; projections are the 'reversed' injections.*

Exercise A.11 *Justify the equations defining a categorical product from the rules for* MALL*.*

ILL—the intuitionistic fragment

In order to obtain the intuitionistic fragment of linear logic we have to add one other connective, the unary !. This is our way of marking formulae which we allow to be subject to the weakening and contraction rules. This fragment is obtained by adding the rules in Table 3 to MALL.

We wish to interpret ! in the categorical model $\mathbf{C}$ by a functor $\mathbf{C} \longrightarrow \mathbf{C}$ which we also refer to as !. This functor will have to come with additional structure so that we can define morphisms corresponding to various rules. We postpone the argument for why ! should be defined on morphisms too.

The linear exponential !	
$\dfrac{\Gamma \vdash B}{\Gamma, !A \vdash B}$	$\dfrac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}$
$\dfrac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}$	$\dfrac{!\Gamma \vdash A}{!\Gamma \vdash !A}$

Table 3: The rules for the linear exponential !

The judgement $A \vdash B$ is interpreted by a morphism $A \longrightarrow B$, and from that we have to be able to obtain a morphism $!A \longrightarrow B$. This can be done most easily if we demand that there exist a morphism $\epsilon_A \colon !A \longrightarrow A$ for each object A, then the composite

$$!A \longrightarrow A \longrightarrow B$$

can be used to interpret $!A \vdash B$.

Further from $!A \vdash B$ we get $!A \vdash !B$ and thus for a morphism $!A \longrightarrow B$ we want to be able to define a morphism $!A \longrightarrow !B$, that is we want a map $(-)^* \colon \mathbf{C}(!A, B) \longrightarrow \mathbf{C}(!A, !B)$. If we want to enforce that certain proofs create the same morphisms in our model then we have to impose some conditions on these.

Consider the proof

$$\cfrac{\cfrac{\cfrac{}{A \vdash A}}{!A \vdash A}}{!A \vdash !A}$$

We might just as well have started with the axiom $!A \vdash !A$ and therefore we demand that $(\epsilon_A)^* = \mathsf{id}_{!A}$. Next consider the proof

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ !A \vdash B\end{array}}{!A \vdash !B} \quad \cfrac{\cfrac{}{B \vdash B}}{!B \vdash B}}{!A \vdash B}\,Cut$$

This amounts to cutting against the identity (well, almost), and should really be the same as the original proof of $!A \vdash B$. Hence we demand that the assignment $\epsilon_B \circ (-)^*$ be the identity on $\mathbf{C}(!A, B)$.

Finally consider the two proofs

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ !A \vdash B\end{array}}{!A \vdash !B} \quad \cfrac{!B \vdash C}{!B \vdash !C}}{!A \vdash !C}\,Cut \qquad\qquad \cfrac{\cfrac{\cfrac{\begin{array}{c}\vdots\\ !A \vdash B\end{array}}{!A \vdash !B} \quad !B \vdash C}{!A \vdash C}\,Cut}{!A \vdash !C}$$

Arguably we have performed 'the same' cut both times, and these proofs should be considered equal. Therefore we demand that for $f\colon !A \longrightarrow B$ and $g\colon !B \longrightarrow C$ we have that $g^* \circ f^* = (g \circ f^*)^*$.

Hence we have that $!$, ϵ and $(-)^*$ form the dual of a *Kleisli triple*. An alternative way of defining the same structure is the following.

Definition 15 *A* comonad *on a category* $\mathbf{C}$ *consists of a functor* $!$ *and natural transformations* $\epsilon\colon ! \longrightarrow \mathsf{id}_{\mathbf{C}}$ *and* $\delta\colon ! \longrightarrow !!$ *such that the following diagrams commute for all objects* A.

$$\begin{tikzcd}
!A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] \arrow[dr, "\mathsf{id}_{!A}"] & !!A \arrow[d, "\epsilon_{!A}"] \\
!!A \arrow[r, "!\epsilon_A"'] & !A
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & !!A \arrow[d, "\delta_{!A}"] \\
!!A \arrow[r, "!\delta_A"'] & !!!A
\end{tikzcd}$$

We often call ϵ *the* counit *of the comonad and* δ *its* comultiplication.

Exercise A.12 *Show that the dual of a Kleisli triple (that is, assignments* $(!, \epsilon, (-)^*)$ *as above satisfying the given equations) and a comonad are the same thing.*

Exercise A.13 *Show that the following is a comonad on the category of sets and relations. On objects let* $!A$ *be the set of multisets over* A. *For a morphism* $R\colon A \longrightarrow\!\!\!\!\!\!+\;\; B$ *set* $x \mathrel{!R} y$ *if and only if there exists* $c \subseteq R$ *such that* $\pi_1 c = x$ *and* $\pi_2 c = y$. *For the counit, take the reverse of the 'singleton' relation. For the comultiplication take the 'sum' of the multisets in question.*

This allows us to interpret the rules on the right hand side of Table 3. Before we turn to the rules on the left hand side of the same table we resolve another matter.

Consider the following proofs.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}}{!A \vdash A} \quad \dfrac{\dfrac{}{B \vdash B}}{!B \vdash B}}{!A, !B \vdash A \otimes B}}{!A, !B \vdash !(A \otimes B)}}{!A \otimes !B \vdash !(A \otimes B)} \qquad\qquad \dfrac{\dfrac{}{\vdash \mathbf{I}}}{\vdash !\mathbf{I}}$$

So there must be morphisms $m_{A,B}\colon !A \otimes !B \longrightarrow !(A \otimes B)$ which we expect to be natural in A and B, and there must be a morphism $\mathbf{I} \longrightarrow !\mathbf{I}$. In other words we expect that $!$ is a monoidal functor. We further demand that ϵ and δ are monoidal natural transformations, which is sometimes summarized by demanding that $(!, \epsilon, \delta, m, m_{\mathbf{I}})$ be a *monoidal comonad.*

Exercise A.14 *Show that the multiset comonad from Exercise A.13 is monoidal.*

Exercise A.15 *Can you find proof-theoretic reasons for demanding that* ϵ *and* δ *be monoidal? How about the naturality of* m*?*

We finally turn to the left rules for !. Given a morphism $\mathbf{I} \longrightarrow B$ which is the interpretation of $\vdash B$, we have to be able to define a morphism $!A \longrightarrow B$ which serves as the interpretation of $!A \vdash B$. Again we use a composite for this and demand that for every object A there be a morphism $e_A \colon !A \longrightarrow \mathbf{I}$. Similarly we demand the existence of a morphism $d_A \colon !A \longrightarrow !A \otimes !A$ to interpret the contraction rule. As is to be expected, the mere existence of these morphisms is not sufficient for our purposes.

First of all we expect symmetry, transitivity and unit laws to hold. Clearly when contracting three occurrences of, say, $!A$, it should not matter in which order this happens. Exchanging two such occurrences and then contracting them should be the same as just contracting them. And first introducing $!A$ by weakening and then contracting it with an existing copy should be the same as doing nothing at all.

Definition 16 *A* commutative comonoid *in a symmetric monoidal category* $\mathbf{C}$ *is an object* A *with two morphisms* $e \colon A \longrightarrow \mathbf{I}$ *and* $d \colon A \longrightarrow A \otimes A$ *such that the following diagrams commute.*

$$\begin{tikzcd}
A \arrow[r, "d"] \arrow[dr, "d"'] & A \otimes A \arrow[d, "\sigma_A"] \\
& A
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "d"] \arrow[dr, "\lambda_A^{-1}"'] & A \otimes A \arrow[d, "e \otimes \mathsf{id}_A"] \\
& \mathbf{I} \otimes A
\end{tikzcd}$$

$$\begin{tikzcd}
A \arrow[r, "d"] \arrow[d, "d"'] & A \otimes A \arrow[r, "d \otimes \mathsf{id}_A"'] & (A \otimes A) \otimes A \arrow[d, "\alpha_{A,A,A}"] \\
A \otimes A \arrow[rr, "\mathsf{id}_A \otimes d"'] & & A \otimes (A \otimes A)
\end{tikzcd}$$

A morphism of comonoids (A, e, d) *and* (A', e', d') *is a morphism* $f \colon A \longrightarrow A'$ *respecting the comonoid structure, that is the following diagrams commute.*

$$\begin{tikzcd}
A \arrow[r, "e"] \arrow[d, "f"'] & \mathbf{I} \\
A' \arrow[ur, "e'"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "d"] \arrow[d, "f"'] & A \otimes A \arrow[d, "f \otimes f"] \\
A' \arrow[r, "d'"'] & A' \otimes A'
\end{tikzcd}$$

Exercise A.16 *(1) Show that the multiset over a set* A *is a symmetric comonoid in the category* **Rel**.

(2) Show that if $\mathbf{C}$ *is a category with finite products then every object carries a comonoid structure, and every morphism preserves it.*

We want every triple $(!A, e_A, d_A)$ to be a commutative comonoid. Clearly objects of the form $!A$ play a particular role—they are the interpretations of formulae on which we may perform the structural rules such as weakening and contraction.

Definition 17 *The* category $\mathbf{C}_!$ of free coalgebras for a comonad $(!, \epsilon, \delta)$ *on a category* $\mathbf{C}$ *has objects* $!A$*, where* A *is an object of* $\mathbf{C}$*. Morphisms* $!A \longrightarrow !B$ *are morphisms* $f\colon !A \longrightarrow !B$ *in* $\mathbf{C}$ *such that the following diagram commutes.*

$$\begin{tikzcd}
!A \arrow[r, "\delta_A"] \arrow[d, "f"'] & !!A \arrow[d, "!f"] \\
!B \arrow[r, "\delta_B"'] & !!B
\end{tikzcd}$$

Exercise A.17 *Show that every morphism in* $\mathbf{C}_!$ *can be written as a composite of morphisms of the form* δ_A *and* $!f$*.*

There is a generalized form of 'coalgebra' for a comonad.

Definition 18 *The* category $\mathbf{C}^!$ of Eilenberg-Moore coalgebras for a comonad $(!, \epsilon, \delta)$ *on a category* $\mathbf{C}$ *has objects* $\xi\colon A \longrightarrow !A$ *which are arrows in* $\mathbf{C}$*. We also refer to these objects as* coalgebras (for the comonad)*. Morphisms between two such objects* $\xi\colon A \longrightarrow !A$ *and* $\zeta\colon B \longrightarrow !B$ *are given by morphisms* $f\colon A \longrightarrow B$ *such that the following diagram commutes.*

$$\begin{tikzcd}
A \arrow[r, "\xi"] \arrow[d, "f"'] & !A \arrow[d, "!f"] \\
B \arrow[r, "\zeta"'] & !B
\end{tikzcd}$$

Exercise A.18 *(1) Show that* $\mathbf{C}_!$ *embeds into* $\mathbf{C}^!$ *by mapping an object* $!A$ *to the object given by* $\delta_A\colon !A \longrightarrow !!A$*. Further show that this embedding is full.*

(2) Show that the category of free coalgebras is isomorphic to the co-Kleisli *category of a monad. This category has objects* A *where* A *is an object in* $\mathbf{C}$ *and morphisms* $A \longrightarrow B$ *are given by morphisms* $!A \longrightarrow B$ *in* $\mathbf{C}$*. The identity on an object* A *is* ϵ_A *and the composite of* $f\colon A \longrightarrow B$ *and* $g\colon B \longrightarrow C$ *is given by* $g \circ !f \circ \delta_A\colon A \longrightarrow C$*. If you haven't seen this construction before you might want to prove that it is indeed a category.*

Proposition A.19 *Let* $(!, \epsilon, \delta, m, m_{\mathbf{I}})$ *be a monoidal comonad on a symmetric monoidal category* $\mathbf{C}$*. Then* $\mathbf{C}^!$ *is symmetric monoidal. The tensor product of two objects* $\xi\colon A \longrightarrow !A$ *and* $\zeta\colon B \longrightarrow !B$ *is given by* $m_{A,B} \circ (\xi \otimes \zeta)\colon A \otimes B \longrightarrow !(A \otimes B)$*. The unit for the tensor product is given by* $m_{\mathbf{I}}\colon \mathbf{I} \longrightarrow !\mathbf{I}$*.*

Exercise A.20 *Prove Proposition A.19.*

Exercise A.21 *Show that there is an adjunction between* $\mathbf{C}$ *and* $\mathbf{C}_!$*. Hint:* $\mathbf{C} \longrightarrow \mathbf{C}_!$ *is the identity on objects.* $\mathbf{C}_! \longrightarrow \mathbf{C}$ *maps an object* A *of* $\mathbf{C}_!$ *to* $!A$*; a morphism* $A \longrightarrow B$ *in* $\mathbf{C}_!$ *given by* $f\colon !A \longrightarrow B$ *in* $\mathbf{C}$ *is mapped to* $!f \circ \delta_A\colon !A \longrightarrow !B$*.*

Do the same for $\mathbf{C}$ *and* $\mathbf{C}^!$. *Hint:* $\mathbf{C} \longrightarrow \mathbf{C}^!$ *maps an object* A *to* $\delta_A \colon !A \longrightarrow !!A$; $\mathbf{C}^! \longrightarrow \mathbf{C}$ *is the identity on objects. Show that this adjunction is monoidal, that is both functors are monoidal and so are the natural transformations connecting them.*

Proposition A.22 *Let* $(!, \epsilon, \delta, m, m_{\mathbf{I}})$ *be a monoidal comonad on a symmetric monoidal category* $\mathbf{C}$. *If every free coalgebra is a commutative comonoid in* $\mathbf{C}$ *then so is every coalgebra, that is every object of* $\mathbf{C}^!$, *(with respect to the tensor product from Proposition A.19). Further if every morphism of free coalgebras is a morphism of comonoids then every morphism of coalgebras is a morphism of comonoids with respect to the induced structure.*

Exercise A.23 *Prove Proposition A.22. Hint: Every coalgebra is a retract of a free one. Derive a comonoid structure using the existing one for free coalgebras.*

Hence demanding that every free coalgebra have a commutative comonoid structure is just the same as demanding that that is the case for all coalgebras. Viewing the structure as part of the bigger category allows us to state all that we require for our models.

Definition 19 *We say that a symmetric monoidal category has a* linear exponential comonad *if it has a monoidal comonad such that the category of Eilenberg-Moore coalgebras, with the induced tensor product, is a category of commutative comonoids. In other words every object of the Eilenberg-Moore category comes with a commutative comonoid structure, and every morphism preserves that structure.*

This is, in fact, equivalent to a weaker sounding structure. Note that in general the category $\mathbf{C}_!$ is not closed under the tensor product in $\mathbf{C}^!$. This makes talking about the structure slightly more awkward.

Proposition A.24 *A symmetric monoidal category* $\mathbf{C}$ *has a linear exponential comonad if and only if it has a monoidal comonad* $(!, \epsilon, \delta, m, m_{\mathbf{I}})$ *such that*

- *all objects of* $\mathbf{C}$ *which are of the form* $!A$ *are equipped with a commutative comonoid structure* (A, e_a, d_a) *which is natural in* A;
- *the natural transformations* e *and* d *are monoidal;*
- *every morphism of the form* $!f$ *and every morphism of the form* δ_A *is a morphism of comonoids.*

Exercise A.25 *Prove Proposition A.24. Hint: part of this was done with the proof of Proposition A.22. If you have a problem with showing that* e *and* d *are monoidal (in the 'only if' direction) then think about what it means for* $m_{\mathbf{I}}$ *and* $m_{A,B}$ *to be morphisms in* $\mathbf{C}^!$ *(between which objects?).*

For completeness' sake we also introduce a notion of morphism between categories with a linear exponential structure.

Definition 20 *Let* **C** *and* **D** *be symmetric monoidal categories with linear exponential comonads. A functor* $F\colon \mathbf{C} \longrightarrow \mathbf{D}$ *is* linearly distributive *if and only if* F *is monoidal (with structure* $m_{\mathbf{I}}, m_{C,C'}$*) and is equipped with a distributive law* $\kappa\colon !F \longrightarrow F!$ *respecting the comonoid structure, in the sense that*

$$\begin{tikzcd}
\mathbf{I} \arrow[d, "m_{\mathbf{I}}"'] & !F(C) \arrow[l, "e_{F(C)}"'] \arrow[r, "d_{F(C)}"] \arrow[d, "\kappa_C"] & !F(C)\otimes !F(C) \arrow[d, "m_{!C,!C}\cdot(\kappa_C\otimes\kappa_C)"] \\
F(I) & F(!C) \arrow[l, "F(e_C)"] \arrow[r, "F(d_C)"'] & F(!C\otimes !C)
\end{tikzcd}$$

Definition 21 *We say that a category* **C** *is a* model for intuitionistic linear logic *if*

- *it is symmetric monoidal closed;*
- *it has finite products;*
- *it has a linear exponential comonad.*

Exercise A.26 *Show that* **Rel** *is a model for intuitionistic liner logic.*

Morphisms of models for intuitionistic linear logic are given by linearly distributive functors.

A.2 Cartesian closed categories

Models for intuitionistic linear logic are not just interesting for their own sake. They give rise to a cartesian closed category. Cartesian closed categories are the standard model for the $\{\sqcap, \rightarrow\}$ fragment of classical logic, and this is the right universe to give semantics for various programming languages.

Proposition A.27 *Let* **C** *be a symmetric monoidal category. If* **C** *has a linear exponential comonad* $(!, \epsilon, \delta, m, m_{\mathbf{I}})$ *then* $\mathbf{C}^!$ *has finite products which are given by the induced tensor product.*

Exercise A.28 *Prove Proposition A.27. Hint: The projections are given by the arrows* $\rho_A \circ e'\colon A \otimes A' \longrightarrow A$ *and* $\lambda_A \circ e\colon A \otimes A' \longrightarrow A'$*, where* e *and* e' *are the counits for the comonoid structure. Pairing of morphisms can be defined using the comultiplications.*

Exercise A.29 *Can you generalize the previous exercise? Where did you use the assumption that* $\mathbf{C}^!$ *is a category of coalgebras?*

Proposition A.30 *Let* **C** *be a model for intuitionistic linear logic. Then*

$$!A \otimes !B \cong !(A \times B).$$

In particular the co-Kleisli category $\mathbf{C}_!$ *for the linear exponential comonad is closed under the tensor product from* $\mathbf{C}^!$*. Hence the co-Kleisli category* $\mathbf{C}_!$ *has finite products.*

Proof. The isomorphisms are given by

$$!(A \times B) \xrightarrow{d_{A\times B}} !(A \times B)\otimes!(A \times B) \xrightarrow{!\pi_1\otimes!\pi_2} !A\otimes!B$$

and

$$!A\otimes!B \xrightarrow{\delta_A\otimes\delta_B} !!A\otimes!!B \xrightarrow{m_{!A,!B}} !(!A\otimes!B) \xrightarrow{!\langle\rho_A\circ(\epsilon_A\otimes e_B),\lambda_B\circ(e_A\otimes\epsilon_B)\rangle} !(A \times B).$$

We will show below that these maps are indeed mutually inverse. Therefore the tensor product of free coalgebras is (isomorphic to) a free coalgebra. By Exercise A.18, the embedding of $\mathbf{C}_!$ into $\mathbf{C}^!$ is full, and therefore a product in $\mathbf{C}^!$ which is a free coalgebra is also a product in $\mathbf{C}_!$. Therefore the co-Kleisli category has finite products. Consider the following diagram.

$$\begin{tikzcd}
!(A \times B) \arrow[r, "d_{A\times B}"] \arrow[dr, bend right=0, "\mathsf{id}_{!(A\times B)}"'] & !(A \times B)\otimes!(A \times B) \arrow[r, "!\pi_1\otimes!\pi_2"] \arrow[d, "e_{A\times B}"] & !A\otimes!B \arrow[d, "\mathsf{id}_{!A}\otimes e_B"'] \arrow[dr, "{\langle\rho_A\circ(\epsilon_A\oplus e_B), \lambda_B\circ(e_A\oplus\epsilon_B)\rangle}"] & \\
 & !(A \times B)\otimes \mathbf{I} \arrow[r, "\pi_1\otimes\mathsf{id}_{\mathbf{I}}"] \arrow[d, "\rho_{!(A\times B)}"] & !A \otimes \mathbf{I} \arrow[d, "\rho_{!A}"'] & A \times B \arrow[d, "\pi_1"] \\
 & !(A \times B) \arrow[r, "!\pi_1"'] & !A \arrow[r, "\epsilon_A"'] & A
\end{tikzcd}$$

It justifies the upper triangle in the following diagram which establishes one of the two equations we wish to show. For the sake of readability we leave out any associativity isomorphisms involved and therefore any bracketing of tensored expressions. Further we write f^2 for $f \otimes f$ at times.

$$\begin{tikzcd}
 & & !A\otimes!B \arrow[drr, "\delta_A\times\delta_B"] & & \\
 & !(A\times B)\otimes!(A\times B) \arrow[ur, "!\pi_1\otimes!\pi_2"] \arrow[r, "\delta_{A\times B}\otimes\delta_{A\times B}"] & !!(A\times B)\otimes!!(A\times B) \arrow[rr, "!!\pi_1\otimes!!\pi_2"] \arrow[d, "m_{!(A\times B),!(A\times B)}"] & & !!A\otimes!!B \arrow[d, "m_{!A,!B}"] \\
!(A\times B) \arrow[ur, "d_{A\times B}"] \arrow[r, "\delta_{A\otimes B}"] \arrow[drrrr, "\mathsf{id}_{!(A\times B)}"'] & !!(A\times B) \arrow[r, "!d_{A\times B}"] \arrow[drrr, "\epsilon_{A\times B}"] & !(!(A\times B)\otimes!(A\times B)) \arrow[rr, "!(!\pi_1\otimes!\pi_2)"] & & !(!A\otimes!B) \arrow[d, "!\langle\rho_A\circ(\epsilon_A\otimes e_B){,}\lambda_B\circ(e_A\otimes\epsilon_B)\rangle"] \\
 & & & & !(A\times B)
\end{tikzcd}$$

Now for the other direction.

$$\begin{tikzcd}
& !!A\otimes !!B \arrow[r, "m_{!A,!B}"] \arrow[d, "(\mathsf{id}\otimes\sigma_{!!A,!!B}\otimes\mathsf{id})\circ(d_{!A}\otimes d_{!B})"] & !(!A\otimes !B) \arrow[r, "!\langle\rho_A\circ(\epsilon_A\otimes e_B)\text{,}"', "\lambda_B\circ(e_A\otimes\epsilon_B)\rangle"'] \arrow[d, "d_{!A\otimes !B}"] & !(A\times B) \arrow[d, "d_{A\times B}"] \\
!A\otimes !B \arrow[ur, "\delta_A\otimes\delta_B"] \arrow[d, "(\mathsf{id}\otimes\sigma_{!A,!B}\otimes\mathsf{id})\circ(d_A\otimes d_B)"] & !!A\otimes !!B\otimes !!A\otimes !!B \arrow[r, "m_{!A,!B}", "\otimes m_{!A,!B}"'] \arrow[d, "!\epsilon_A\otimes\mathsf{id}\otimes\mathsf{id}\otimes !\epsilon_B"] & !(!A\otimes !B)\otimes !(!A\otimes !B) \arrow[r, "!\langle\rho_A\circ(\epsilon_A\otimes e_B)\text{,}", "\lambda_B\circ(e_A\otimes\epsilon_B)\rangle^2"'] \arrow[d, "!(\epsilon_A\otimes\mathsf{id})\otimes !(\mathsf{id}\otimes\epsilon_B)"] & !(A\times B)\otimes !(A\times B) \arrow[dddl, "!\pi_1\otimes !\pi_2"] \\
!A\otimes !B\otimes !A\otimes !B \arrow[ur, "\delta_A\otimes\delta_A\otimes\delta_B\otimes\delta_B"] \arrow[r, "\mathsf{id}\otimes\delta_A", "\otimes\delta_B\otimes\mathsf{id}"'] \arrow[d, "\mathsf{id}\otimes e_B\otimes e_A\otimes\mathsf{id}"] & !A\otimes !!B\otimes !!A\otimes !B \arrow[r, "m_{A,!B}", "\otimes m_{!A,B}"'] \arrow[d, "\mathsf{id}\otimes !e_B\otimes !e_A\otimes\mathsf{id}"] & !(A\otimes !B)\otimes !(!A\otimes B) \arrow[d, "!(\mathsf{id}\otimes e_B)\otimes !(e_A\otimes\mathsf{id})"] & \\
!A\otimes\mathbf{I}\otimes\mathbf{I}\otimes !B \arrow[r, "\mathsf{id}\otimes m_{\mathbf{I}}", "\otimes m_{\mathbf{I}}\otimes\mathsf{id}"'] \arrow[drr, "\rho_A\otimes\lambda_B"] & !A\otimes !\mathbf{I}\otimes !\mathbf{I}\otimes !B \arrow[r, "m_{A,\mathbf{I}}", "\otimes m_{\mathbf{I},B}"'] & !(A\otimes\mathbf{I})\otimes !(\mathbf{I}\otimes B) \arrow[d, "!\rho_A\otimes !\lambda_B"] & \\
& & !A\otimes !B &
\end{tikzcd}$$

The morphism along the bottom of the diagram composes to the identity on $!A\otimes !B$ by the counit rule for comonoids. $\square$

Theorem A.31 *Let* $\mathbf{C}$ *be a model for intuitionistic linear logic. Then its co-Kleisli category is cartesian closed.*

Proof. By the previous proposition the co-Kleisli category $\mathbf{C}_!$ has finite products, and the product of A and B is $A\times B$. It remains to find a right adjoint for this functor. This is given by $A\to B = !A \multimap B$ as is shown by the following series of equations.

$$\begin{aligned}
\mathbf{C}_!(A\times B, C) &\cong \mathbf{C}(!(A\times B), C)\\
&\cong \mathbf{C}(!A\otimes !B, C)\\
&\cong \mathbf{C}(!A, !B\multimap C)\\
&\cong \mathbf{C}_!(A, !B\multimap C)\\
&\cong \mathbf{C}_!(A, B\to C)
\end{aligned}$$

$\square$

A.3 Classical linear logic

Classical linear logic can be understood to be obtained by adding negation to the intuitionistic fragment—at least if one uses the De Morgan rules to define all the missing connectives (while this does not do the logic justice it is how the model is built). We will not go into details here regarding the full system—see for example [Gir87, Gir95]. Variables in this system come in dual pairs, and then dualization (or negation) can be defined for all formulae using De Morgan-style rules. The system is typically two-sided, that is its judgements are of the form

$$\Gamma \vdash \Delta,$$

where both, Γ and Δ are lists of formulae. Such judgements are interpreted by arrows

$$A_1 \otimes \cdots \otimes A_m \longrightarrow B_1 \oplus \cdots \oplus B_n,$$

where $\Gamma = A_1, \ldots, A_m$ and $\Delta = B_1, \ldots, B_n$.

Formulae can be moved to the other side of $\vdash$ by dualizing them. In particular, we can freely move between $A \vdash B$ and $B^\perp \vdash A^\perp$. Hence on our category we will have to be able to freely move from morphisms $A \longrightarrow B$ and morphisms $B^\perp \longrightarrow A^\perp$. This suggests having a contravariant functor $\mathbf{C} \longrightarrow \mathbf{C}$ as the interpretation of dualization, and one that is its own inverse.

Definition 22 *We say that a category* $\mathbf{C}$ *is self-dual if there is a functor* $(-)^\perp \colon \mathbf{C} \longrightarrow \mathbf{C}^{\mathsf{op}}$ *such that* $(-)^{\perp\perp}$ *is equivalent to the identity on* $\mathbf{C}$*.*

Exercise A.32 *Argue that dualization (as the interpretation of negation) should preserve composition and identities.*

Clearly we would like the categorical interpretation of dualization to satisfy the De Morgan equations connecting the various connectives. For $\oplus$, $\sqcup$ and ? we may indeed use the De Morgan equations to define interpretations, for example the semantics of $A \oplus B$ will be $(A^\perp \otimes B^\perp)^\perp$. But the connection we wish to achieve between $\otimes$ and $\multimap$ has to be built into the definition of what a model for (classical) linear logic should be.

Definition 23 *A symmetric monoidal category* $\mathbf{C}$ *is* $*$-autonomous *if has an involutary duality* $(-)^\perp$ *and if for every object* A*, the functor* $- \otimes A \colon \mathbf{C} \longrightarrow \mathbf{C}$ *is left adjoint to* $(A \otimes -^\perp)^\perp \colon \mathbf{C} \longrightarrow \mathbf{C}$*.*

Exercise A.33 *Show that* **Rel** *is* $*$*-autonomous. Hint: the dual of a set is the set itself; for morphisms the dual is defined by taking the reverse of the relation in question.*

In fact, **Rel** is more than just $*$-autonomous, it is what is called *compact closed.* A $*$-autonomous category is *compact closed* if for all objects A and B, $(A \otimes B)^\perp \cong A^\perp \otimes B^\perp$ or, equivalently, $A \multimap B \cong A^\perp \otimes B$. Compact closed categories are degenerate models of linear logic in the sense that the interpretation of $\oplus$, which is defined by the De Morgan laws applied to $\otimes$, is the same as that for $\otimes$.

Exercise A.34 *(1) Show that a symmetric monoidal closed category is* $*$*-autonomous if and only if there is an object* $\perp$ *such that the functor* $(- \multimap \perp) \multimap \perp \colon \mathbf{C} \longrightarrow \mathbf{C}^{\mathsf{op}}$ *is a self-involutary duality on* $\mathbf{C}$*.*

(2) Show that a self-dual category which has finite products also has finite coproducts.

Definition 24 *We say that a category* $\mathbf{C}$ *is a* model for (classical) linear logic *if and only if it is*

- *$*$-autonomous;*
- *has finite products (and thus finite coproducts);*
- *has a linear exponential comonad (and thus a linear exponential monad).*

In other words, all that is added to a model of intuitionistic linear logic is the self-dualization, which has to behave well when it comes to connecting the tensor and the linear function space. The rest of the structure comes for free.

References

[Abr97] S. Abramsky. Semantics of interaction: an introduction to game semantics. In Pitts and Dybjer [PD97], pages 1–31.

[AJ94] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *J. Symbolic Logic*, 59:543–574, 1994.

[AJM] S. Abramsky, R. Jagadeesan, and P. Malacaria. Games and full abstraction for PCF. *Information and Computation.* To appear.

[AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (extended abstract). In *Theoretical Aspects of Computer Software: TACS '94*, volume 789 of *Lecture Notes in Computer Science*, pages 1–15, 1994.

[AM95] S. Abramsky and G. McCusker. Games and full abstraction for the lazy λ-calculus. In *Proc. Logic in Computer Science 1995*, pages 234–243. IEEE Computer Society Press, 1995.

[AM97] S. Abramsky and G. McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions. In P.W. O'Hearn and R.D. Tennent, editors, *Algol-like languages*, pages 317–348. Birkhäuser, 1997.

[AM99] S. Abramsky and P.-A. Melliès. Concurrent games and full completeness. In *Proc. Logic in Computer Science 1999*, pages 431–442. IEEE Computer Society Press, 1999.

[BDER97] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Believe it or not, AJM's games model is a model of classical linear logic. In *Proc. Logic in Computer Science 1997*, pages 68–75. IEEE Computer Society Press, 1997.

[BDER98] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Timeless games. In *Proceedings of the tenth Workshop on Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science.* Springer-Verlag, 1998.

[Bie94] G.M. Bierman. On intuitionistic linear logic. Technical Report 346, University of Cambridge Computer Laboratory, August 1994.

[Bla92] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.

[Bla95] A. Blass. Questions and answers—a category arising in linear logic, complexity theory and set theory. In J.Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 61–81. CUP, 1995.

[Fel86] W. Felscher. *Handbook of Philosophical Logic*, chapter Dialogues as a foundation for intuitionisict logic, pages 341–372. Volume III of Gabbay and Guenther [GG86], 1986.

[GG86] D. Gabbay and F. Guenther, editors. *Handbook of Philosophical Logic*, volume III. D. Reidel Publishing Company, 1986.

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

[Gir95] J.-Y. Girard. Linear logic: its syntax and semantics. In Girard et al. [GLR95].

[GLR95] J.-Y. Girard, Y. Lafont, and L. Regnier, editors. *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1995.

[HO] J.M.E. Hyland and C.-H.L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation.* To appear.

[HO93] J.M.E. Hyland and C.-H.L. Ong. Fair games and full completeness for multiplicative linear logic without the mix-rule. ftp from ftp.comlab.ox.ac.uk as fcomplete.ps.gz in /pub/Documents/techpapers/Luke.Ong, 1993.

[HS99] M. Hyland and A. Schalk. Abstract games for linear logic. extended abstract. In Martin Hofmann, Giuseppe Rosolini, and Dusko Pavlovic, editors, *Proceedings of CTCS '99, Conference on Category Theory and Computer Science*, volume 29 of *Electronic Notes on Theoretical Computer Science*, 1999.

[HS01] M. Hyland and A. Schalk. Glueing and orthogonality in models of linear logic. To appear in Theoretical Computer Science, 2001.

[Hyl97] M. Hyland. Game semantics. In Pitts and Dybjer [PD97], pages 131–194.

[Lai97] J.G. Laird. Full abstraction for functional languages with control. In *Proc. Logic in Computer Science 1997*, pages 58–64. IEEE Computer Science Press, 1997.

[LS91] Y. Lafont and T. Streicher. Games semantics for linear logic. In *Proc. Logic in Computer Science 1991.* IEEE Computer Science Press, 1991.

[McC96] G. McCusker. Games and full abstraction for FPC. In *Proc. Logic in Computer Science 1996*, pages 174–183. IEEE Computer Society Press, 1996.

[PD97] A.M. Pitts and P. Dybjer, editors. *Semantics and Logics of Computation.* Cambridge University Press, 1997.