

Linear Logic and Petri Nets:
Categories, Algebra and Proof

Carolyn T. Brown

Doctor of Philosophy
University of Edinburgh

1990



Abstract

This thesis explores three ways in which linear logic may be used to define a specification language for Petri nets, by giving precise correspondences, at different levels, between linear logic and Petri nets.

Firstly, we define categories **NC** by analogy with de Paiva's dialectica categories **GC**. The category **NSet** has objects the elementary Petri nets and morphisms refinement maps. We show that **GC** induces in **NC** sufficient structure for **NC** to be a sound model of linear logic. We demonstrate the computational significance of the net constructors induced by the interpretation in **NSet** of the linear connectives \otimes , \wedge , \multimap , \oplus and $(-)^{\perp}$. Our framework unifies several existing approaches to categories of nets, and gives a model of full linear logic based on nets.

Secondly, we show that the possible evolutions of a net generate a quantale. Quantales are algebraic models of linear logic. Further, we show that certain restrictions on nets, including being safe or bounded, arise as subquantales induced by suitable conuclei. This approach allows us to give a sound semantics for linear logic using sets of markings of a given net. Thus the provability of certain assertions in linear logic corresponds to properties of nets.

Thirdly, we define a semantics for a fragment of linear logic \mathcal{L}_0 in terms of nets, by giving a partial function from formulae of linear logic to nets. This semantics is complete and sound where defined. Further, we show that whenever a net N can evolve to a net N' , there is a canonical proof in \mathcal{L}_0 that the formula interpreted by N entails the formula interpreted by N' . A canonical proof expresses the causal dependencies of a net in a precise way, using the (Cut) rule. This approach allows us to use the techniques of proof theory to study the evolution of nets.

Acknowledgements

I would like to thank my first supervisor, Stuart Anderson, for being a unique and wonderful person, for giving me much support, countless lively discussions and a sane philosophy.

I also owe a great debt to my second supervisor, the inimitable John Power, for his expert guidance and assiduous attention. I greatly appreciate all that Stuart and John have done for me, and the commitment they have both shown to the welfare of post-graduate students.

I would also like to thank Martin Hyland for his inspiration and support over many years, and Valeria de Paiva, Glynn Winskel and Barry Jay for their encouragement and interesting conversations. Paul Taylor's diagram macros were used to produce many of the diagrams in this thesis.

Nothing would have been possible without my friends and colleagues. In particular, many thanks to Chris, Healf, James, Simon and, above all, Doug.

This work was partly supported by an SERC research studentship. I am also grateful to the SERC for its Logic for IT initiative, which has provided the opportunity to meet people and exchange ideas.

Declaration.

This thesis was composed by myself and the work presented in it is my own.

Carolyn T. Brown

Table of Contents

I	Introduction and Preliminaries	9
1.	Introduction	10
2.	Preliminary Definitions: Petri Nets	14
2.1	Multisets and Multirelations	14
2.2	Petri Nets	18
2.2.1	Representing Petri Nets Graphically	19
2.3	The Evolution of Petri Nets	20
2.3.1	Subnets, Augmentation and Restriction	21
2.4	Important note on the Definition of Nets	22
3.	An Introduction to Linear Logic	24
3.1	Salient Features of Linear Logic	24
3.2	The Sequent Calculi of Linear Logic	25
3.2.1	Sequents with Multiple Conclusions	26
3.2.2	The Fragment LL_* of Linear Logic	28
3.2.3	Sequents with Single Conclusions	28
3.3	Some Useful Derived Rules	30
3.4	Formulae of Linear Logic	32

<i>Table of Contents</i>	5
II A Categorical Linear Framework for Petri Nets	38
II Introduction to Part II	39
4. A New Category of Petri Nets	41
4.1 Summary of the Chapter	41
4.2 Existing Categories of Petri Nets	42
4.2.1 Winskel's Category, Net	42
4.2.2 Structure in Net	43
4.2.3 Meseguer and Montanari's Categories of Nets	47
4.3 The Dialectica Category GC	50
4.4 A Category of Elementary Petri Nets	58
4.4.1 Some Examples of Refinement Maps	63
5. Structure in NSet	68
5.1 Characterising NC as a limit in Cat	69
5.2 Restriction	73
5.3 The Product of Two Nets	74
5.3.1 Restriction of the Product Net	77
5.3.2 The Terminal Object in NC	78
5.3.3 Sequentialising the Behaviour of a Net using Products	79
5.4 The Coproduct of Two Nets	80
5.4.1 The Initial Object in NC	83
5.5 Tensor Product as Concurrent Composition	83
5.5.1 Other Tensor Products	87
5.5.2 The Unit of the Monoidal Structure \otimes	87

<i>Table of Contents</i>	6
5.6 Expressing Synchrony and Asynchrony	88
5.6.1 Restrictions of the Product of two Nets	90
5.6.2 Restrictions of the Tensor Product of two Nets	91
5.7 The Exponential of Two Nets	93
5.8 Negation in NSet	95
5.9 The Constants in NC	98
6. Exploiting the Generality of the Framework	100
6.1 A Different Base Category C	100
6.2 Modifying the Notion of Refinement	102
6.2.1 Interpreting the Morphisms in $\mathbf{NSet}_{\underline{C}}^{\underline{C}}$, $\mathbf{NSet}_{\underline{C}}^{\supseteq}$ and \mathbf{NSet}^{co}	111
6.3 The Category $\mathbf{NC}^=$	113
6.3.1 The Category $\mathbf{NSet}^=$ of Elementary Nets	124
6.4 The Kleisli Category \mathbf{NC}_T	125
6.4.1 Definitions	125
6.4.2 A Kleisli Category on NC	126
6.4.3 Kleisli Categories with objects the Elementary Nets	129
6.5 Comparison of the NC categories with SafeNet	130
6.5.1 Comparing $\mathbf{NSet}^=$ with SafeNet	131
III Petri Nets and Quantaes	133
III Introduction to Part III	134
7. Quantaes of Behaviours of a Net	135
7.1 Introduction	135

7.2	Constructing a Quantale from a Net	135
7.2.1	A Quantale of Markings of a Petri Net	137
7.2.2	Backwards closed sets of markings	141
7.3	Traps	142
7.4	Interpreting Linear Logic in a Net-Quantale	146
7.5	Linear Negation	149
7.6	Equivalences on Nets	150
7.6.1	Examples of Equivalent Nets	152
7.7	Restrictions on the top element of a net-quantale	153
7.7.1	Arbitrary Sets of Initial Markings	156
IV	Evolution as Linear Proof	158
IV	Introduction to Part IV	159
8.	Expressing Nets as Canonical Formulae	161
8.1	Chapter Summary	161
8.2	The Formula Representing a Net	161
8.3	Representing Arbitrary Nets as Formulae	163
8.4	A Reduction Relation S on F	167
8.4.1	S is Strongly Normalising	173
8.4.2	S is Church-Rosser	177
9.	Reachability and Provability	181
9.1	Introduction	181
9.2	A Preorder on Nets	181

9.2.1	Examples of the Preorder \sqsupseteq	182
9.3	Composing Nets	185
9.3.1	The Action of \otimes on Nets	185
9.3.2	The Action of \wedge on Nets	188
9.3.3	The Issue of Choice	189
9.4	Nets and Equivalent formulae	193
9.4.1	The Interpretation of $!(A)$	194
9.5	A Semantics for Canonical Formulae	196
9.5.1	The Semantics of Linear Implication	197
9.5.2	The Semantics of $F \wedge G$	198
9.6	Reachable markings as Provable Formulae	199
9.6.1	Soundness and Completeness for L_0	199
9.6.2	Soundness and Completeness for L_1	203
9.7	Canonical Derivations	205
9.7.1	Example of a Canonical Derivation	208
9.7.2	Canonical derivations and Evolution	209
9.7.3	A Remark about Categories	209
V	Conclusion and Further Work	211
	Bibliography	216
	Index	222
	Index of Notation	227

Part I

Introduction and Preliminaries

Chapter 1

Introduction

The theory of computable functions is relatively well-understood, and for many purposes this theory is adequate for reasoning about the meanings of sequential computer programs. Ultimately, such reasoning is essential to the practice of programming, since one can prove important properties of programs within the theory. Logics for computable functions also provide a means of formally specifying and verifying programs, and have formed the theoretical basis for machine assisted proof systems.

By contrast, the theory of the behaviour of processes and its application to the study of concurrent programs is less well developed. There is little consensus on what is an appropriate treatment, and a number of alternative approaches exist. Petri nets and event structures ([Win82] [Win80] [Rei85]) use causal independence to establish when two processes can occur concurrently, whereas other models, such as state-transition systems (which include CCS [Mil89] and CSP [Hoa85]) simulate concurrency by the non-deterministic interleaving of atomic actions. Petri nets, event structures and transition systems are all relatively concrete models. There are more abstract models, such as the powerdomain model [Plo76], and models based on observational equivalence or failure-sets. There is still more diversity if we consider also the various process logics which can be used to express properties and reason about any of these models.

Among these models, that of Petri Nets is appealing on intuitive grounds. However, it has proved difficult to provide tractable theories of the behaviours

described by Petri Nets (see for example [Win87], and the introduction to [Mil89]). The aim of this thesis is to explore the theory of Petri Nets from a novel point of view, in particular by applying ideas from linear logic and from category theory to address the problems of specification and compositionality.

In this thesis, we establish precise relationships between Petri nets, a concrete model of concurrency, and linear logic [Gir86]. Each of the Parts II, III and IV of the thesis describes a correspondence between linear logic and Petri nets: each approach could be developed further to give a linear specification language for Petri nets, thus addressing a central problem in the theory of concurrent programming languages. In the analysis of concurrent programs, simultaneous satisfaction of conditions (their “conjunction”) is a primary consideration, and so linear logic, which is meticulous in the “book-keeping” of resources, has a natural application. The results of Parts III and IV exploit this feature of linear logic to describe the behaviour of Petri nets.

The results of Part II suggest a deeper connection between linearity and concurrency. In Part II, we define a category \mathbf{NSet} , which is a sound model of linear logic, and has as objects the elementary Petri nets and morphisms which are refinements. This allows us to address a fundamental weakness in the theory of Petri nets, the lack of a compositional approach. Petri nets are an appealing model of concurrency as they are based on the idea of causal independence and thus on synchrony, rather than a non-deterministic, interleaving semantics. However, there is as yet no fully satisfactory description of how to combine nets as algebraic structures in such a way that the behaviour of a composite net can be expressed in terms of the behaviour of its component nets. This problem is most fully considered in [Win88]. Such a compositional description is essential to developing both a specification language for Petri nets, and a means of verification that a net satisfies its specification. If we have a compositional model, we can verify the behaviour of a complex concurrent process by verifying that the behaviours of its component parts satisfy sub-specifications, and then composing these verifications. Without compositionality we cannot establish a means of step-wise refinement such as is used to verify sequential programs [Wir71], [ST88]. The category \mathbf{NSet} provides

a compositional approach to Petri net structure. In particular, since \mathbf{NSet} is a sound model of linear logic, each of the connectives \otimes , \multimap , \wedge , $\#$ and $(-)^{\perp}$ has an interpretation as a combinator of nets. This gives us a rich language for describing composite nets.

The introduction to Part II indicates our reasons for choosing a categorical approach to models of concurrency. In Chapter 4, we describe some existing categories whose objects are Petri nets, and describe de Paiva's dialectica category \mathbf{GC} , which is a sound model of linear logic. We define a category \mathbf{NC} by analogy with \mathbf{GC} , and consider the category \mathbf{NSet} with objects the elementary Petri nets, obtained by putting $\mathbf{C} = \mathbf{Set}$. In Chapter 5, we characterise \mathbf{NC} as a limit in \mathbf{Cat} , showing that the structure of \mathbf{GC} induces in \mathbf{NC} sufficient structure for \mathbf{NC} to be a sound model of linear logic. We consider in detail the interpretation as net combinators of the linear connectives \otimes , \multimap , \wedge , \oplus and $(-)^{\perp}$. In Chapter 6, we illustrate the flexibility of our approach, giving several categories of nets based on the construction \mathbf{NC} . In particular, we are able to unify in our framework two of the existing categories of nets. Thus \mathbf{NPSet} is the category of elementary nets defined in [NRT90], while Winskel's category $\mathbf{SafeNet}$ [Win88] can be obtained as a Kleisli category on $\mathbf{NSet}^{\#}$, a subcategory of \mathbf{NSet} . The work presented here is the first model of full linear logic to have been based on Petri nets: all other approaches model only a fragment of the logic.

In Part III, we show that the possible evolutions of a Petri net generate a quantale. Quantales are algebraic models of linear logic. Further, we show that such restrictions on nets as being safe, or bounded, arise as the subquantales induced by suitable conuclei. This approach allows us to give a sound semantics for linear logic in terms of sets of markings of a particular Petri net. Thus the provability of certain assertions in linear logic indicates properties of the net. This approach could be expanded to give a specification language for net behaviour. A useful extension would be the addition of quantifiers to the logic. Several of the results presented here have also been shown independently by Engberg and Winskel [EW89].

In Part IV, we show that certain formulae of linear logic correspond precisely to

Petri nets with finite sets of events and conditions. We give a semantics for linear logic in terms of nets using a partial function from formulae of linear logic to Petri nets. This semantics is a complete model of the fragment of linear logic \mathcal{L}_1 , and is sound where it is defined. Further, we show that whenever a net N can evolve to a net N' , there is a canonical proof in \mathcal{L}_0 that the formula interpreted by N entails the formula interpreted by N' . A canonical proof expresses causal dependencies of the net precisely in its restricted use of the (Cut) rule. This approach allows us to apply techniques of proof theory to the study of the evolution of nets, as is done by Gehlot and Gunter [GG90]. The results presented here are independent of those of Gehlot and Gunter, and use a considerably larger fragment of linear logic.

Each of the three parts has an introduction describing its aims and indicating related work. The necessary definitions of Petri net theory, together with some references, are given in Chapter 2. Chapter 3 offers a basic introduction to linear logic, with references and essential definitions.

There are several possibilities for further development of the ideas presented in this thesis. Work in progress with de Paiva shows how to generalise the dialectica categories to obtain categories with arbitrary nets as object set. The quantales based on net behaviour seem limited in scope: work in progress with Tofts will describe quantales generated by CCS processes. The results of Part IV could be extended to give a sound and complete model of (second-order) linear logic in terms of evolution of nets. Further application of proof theory to this work is also appropriate. Finally, we should consider the relationship between the results of the three parts of this thesis. The natural approach to this is to regard the models as categories and consider the functors between them.

Chapter 2

Preliminary Definitions: Petri Nets

An introduction to Petri nets is given in [Rei85]. Interesting papers can also be found in [ACP86b], [ACP86a], and elsewhere.

2.1 Multisets and Multirelations

We give the elementary definitions concerned with relations and multirelations which are necessary to define Petri nets.

Notation 2.1.1 *We write $\mathbf{1}$ for the one-element set, $\{\star\}$,
 $\mathbf{2}$ for the two-element set and
 \mathbb{N} for the natural numbers, including 0.*

Remark 2.1.2

We shall also use $\mathbf{1}$ to denote the terminal object of categories other than \mathbf{Set} .

Definition 2.1.3 *A relation from a set \mathcal{E} to a set \mathcal{B} is a function $R: \mathcal{E} \times \mathcal{B} \rightarrow \mathbf{2}$.*

Notation 2.1.4 *We write \mathbf{Set} for the category with sets as objects and functions as morphisms, with functional composition.*

Remark 2.1.5

A relation $R: \mathcal{E} \times \mathcal{B} \rightarrow \mathbf{2}$ can be regarded as a subobject in \mathbf{Set} of $\mathcal{E} \times \mathcal{B}$.

Definition 2.1.6 Let $R: \mathcal{E} \times \mathcal{B} \rightarrow \mathbf{2}$ be a relation. The opposite relation to R is the function R^{op} from $\mathcal{B} \times \mathcal{E}$ to $\mathbf{2}$ given by

$$R^{op}(\langle b, e \rangle) = R(\langle e, b \rangle) \quad \text{for all } \langle e, b \rangle \in \mathcal{E} \times \mathcal{B}.$$

Definition 2.1.7 A multirelation from a set \mathcal{E} to a set \mathcal{B} is a function $\alpha: \mathcal{E} \times \mathcal{B} \rightarrow \mathbf{N}$, which we shall denote $\alpha: \mathcal{E} \rightarrow_m \mathcal{B}$.

Notation 2.1.8 Let $\alpha: \mathcal{E} \rightarrow_m \mathcal{B}$ be a multirelation. We write $\alpha_{\langle e, b \rangle}$ for $\alpha(\langle e, b \rangle)$.

Definition 2.1.9 Let $\alpha: \mathcal{E} \rightarrow_m \mathcal{B}$ and $\beta: \mathcal{B} \rightarrow_m \mathcal{B}'$ be multirelations. Their composition is the multirelation $(\alpha; \beta): \mathcal{E} \rightarrow_m \mathcal{B}'$ is the formal sum (not necessarily convergent) given by

$$(\alpha; \beta)_{\langle e, b' \rangle} = \sum_{b \in \mathcal{B}} \alpha_{\langle e, b \rangle} \beta_{\langle b, b' \rangle}$$

Definition 2.1.10 Let $\alpha: \mathcal{E} \rightarrow_m \mathcal{B}$ be a multirelation. Let $\mathcal{E}' \subseteq \mathcal{E}$. Then the restriction of α to \mathcal{E}' is the multirelation $\alpha|_{\mathcal{E}'}: \mathcal{E}' \rightarrow_m \mathcal{B}$ given by

$$\alpha|_{\mathcal{E}'}(\langle e, b \rangle) = \alpha(\langle e, b \rangle) \quad \text{for } \langle e, b \rangle \in \mathcal{E}' \times \mathcal{B}.$$

Definition 2.1.11 Let \mathcal{E} and \mathcal{E}' be disjoint sets. Let $\alpha: \mathcal{E} \rightarrow_m \mathcal{B}$ and $\alpha': \mathcal{E}' \rightarrow_m \mathcal{B}'$ be multirelations. The union of α and α' is the multirelation $(\alpha \cup \alpha'): \mathcal{E} \cup \mathcal{E}' \rightarrow_m \mathcal{B} \cup \mathcal{B}'$ given by

$$(\alpha \cup \alpha')(\langle e, b \rangle) = \begin{cases} \alpha(\langle e, b \rangle) & e \in \mathcal{E} \text{ and } b \in \mathcal{B} \\ \alpha'(\langle e, b \rangle) & e \in \mathcal{E}' \text{ and } b \in \mathcal{B}' \\ 0 & e \in \mathcal{E}' \text{ and } b \in (\mathcal{B} \setminus \mathcal{B}') \\ 0 & e \in \mathcal{E} \text{ and } b \in (\mathcal{B}' \setminus \mathcal{B}). \end{cases}$$

Definition 2.1.12 A multiset over a set A is a multirelation $\alpha: \mathbf{1} \rightarrow_m A$, $\mathbf{1} = \{\star\}$ is a distinguished one-element set.

Notation 2.1.13 Let α be a multiset over A . We write $\alpha(a)$ for $\alpha(\langle \star, a \rangle)$.

Definition 2.1.14

A multirelation $\alpha: A' \rightarrow A$ is an inclusion if $A' \subseteq A$ and α is given by

$$\alpha_{a,a'} = \begin{cases} 1 & \text{if } a' = a \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.1.15 Let $\alpha: \mathbf{1} \rightarrow_m A$ and $\beta: \mathbf{1} \rightarrow_m B$ be multisets. We say β is a multi-subset of α , written $\beta \subseteq_m \alpha$, if $B \subseteq A$ and for every $b \in B$ $\beta(b) \leq \alpha(b)$.

Let $a \in A$. If $\alpha(a) > 0$ then a is a member of the multiset α over A , written $a \in_m \alpha$.

Definition 2.1.16 Let α be a multiset over A and β a multiset over B . If $\beta \subseteq_m \alpha$ then we define the subtraction of β from α to be the multiset $(\alpha - \beta): \mathbf{1} \rightarrow_m A$ given by

$$(\alpha - \beta)(a) = \begin{cases} \alpha(a) - \beta(a) & a \in B \\ \alpha(a) & a \in (A \setminus B). \end{cases}$$

Definition 2.1.17 Let $\alpha: \mathbf{1} \rightarrow_m A$ and $\beta: \mathbf{1} \rightarrow_m B$ be multisets. We define the sum of α and β to be the multiset $(\alpha + \beta): \mathbf{1} \rightarrow_m (A \cup B)$ given by

$$(\alpha + \beta)(a) = \begin{cases} \alpha(a) + \beta(a) & a \in (A \cap B) \\ \alpha(a) & a \in (A \setminus B) \\ \beta(a) & a \in (B \setminus A). \end{cases}$$

We define the intersection of α and β to be the multiset $\alpha \cap \beta: \mathbf{1} \rightarrow_m (A \cap B)$ given by

$$(\alpha \cap \beta)(a) = \min\{\alpha(a), \beta(a)\}.$$

Definition 2.1.18 A multiset α over a set A is finite if

$$\{a \in A \mid \alpha(a) \geq 1\}$$

is a finite set.

Definition 2.1.19

A multiset α over a set A is non-empty if there exists $a \in A$ such that $\alpha(a) \geq 1$.

Definition 2.1.20

The empty multiset over a set A is the multiset \emptyset given by $\emptyset(a) = 0$ for all $a \in A$.

Definition 2.1.21 Let α be a multiset over A and let $B \subseteq A$. The restriction of α to B , written $\alpha|_B$, is the multiset over B given by

$$\alpha|_B(b) = \alpha(b) \quad \text{for all } b \in B.$$

Remark 2.1.22 Let $F: \mathcal{B}' \rightarrow \mathcal{B}$ be a function. We regard F^{-1} as a multirelation $F^{-1}: \mathcal{B} \rightarrow_m \mathcal{B}'$ by putting

$$F^{-1}(b, b') = \begin{cases} 1 & \text{if } F(b') = b \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.1.23 Let $F: A \rightarrow B$ be a function. The linear extension of F to multisets over A is given by

$$F(\alpha) = \sum_{a \in A} \alpha(a) F(a), \quad \text{for any multiset } \alpha \text{ over } A.$$

Remark 2.1.24 We often regard multisets as sums. Thus the sum $A + 2B$ represents the multiset α over $\{A, B\}$ given by $\alpha(A) = 1, \alpha(B) = 2$.

If A and B are linear logic atoms, we shall often regard α as a tensor sum of atoms, written $A \otimes 2B$.

2.2 Petri Nets

Definition 2.2.1 A Petri Net is a 4-tuple $\langle \mathcal{E}, \mathcal{B}, pre, post \rangle$, where \mathcal{E} and \mathcal{B} are sets, and pre and $post$ are multirelations from \mathcal{E} to \mathcal{B} .

Notation 2.2.2 We write **Petri** for the set of Petri nets.

Remark 2.2.3 A note on size considerations is given in Section 2.4.

We shall call elements of \mathcal{E} *events* and elements of \mathcal{B} *conditions*. We shall call pre and $post$ the *pre-* and *post-condition relations* of N respectively.

Notation 2.2.4 We write N for the Petri net $\langle \mathcal{E}, \mathcal{B}, pre, post \rangle$, N_0 for the net $\langle \mathcal{E}_0, \mathcal{B}_0, pre_0, post_0 \rangle$, and so on.

Notation 2.2.5 With each of the multirelations pre and $post: \mathcal{E} \rightarrow_m \mathcal{B}$, we associate a function with the same name, from \mathcal{E} to multisets over \mathcal{B} , defined by

$$pre(e) = \sum_{b \in \mathcal{B}} pre(e, b)b \quad \text{and} \quad post(e) = \sum_{b \in \mathcal{B}} post(e, b)b.$$

We call $pre(e)$ the *pre-condition set* of e , and $post(e)$ the *post-condition set* of e .

We extend the functions pre and $post$ to multisets of events as follows:

$$pre(A) = \sum_{e \in \mathcal{E}} A(e)pre(e) \quad \text{for any multiset } A \text{ over } \mathcal{E}, \text{ and}$$

$$post(A) = \sum_{e \in \mathcal{E}} A(e)post(e) \quad \text{for any multiset } A \text{ over } \mathcal{E}.$$

Definition 2.2.6 Let N be a Petri net. A *marking* of N is a multiset M over \mathcal{B} .

Remark 2.2.7 Various authors add further conditions to the definition of a net, or of its markings, to ensure convergence of the formal sums involved in the composition of multirelations. We do not make such restrictions here. In Part II

we consider structure rather than behaviour, and so considerations of convergence are inappropriate. In Part III, the definitions of evolution are sufficient to ensure convergence, while in Part IV, the finiteness conditions on elements of MPetri are sufficient to ensure convergence. We shall therefore always identify the formal sum of Definition 2.1.9 with its evaluation.

Notation 2.2.8 *We write $\text{Mark}(N)$ for the set of all markings of N .*

Definition 2.2.9 *A marked Petri net is a 5-tuple $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M \rangle$ such that $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post} \rangle$ is a Petri net N , and M is a marking of N . We call M the initial marking of the marked net $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M \rangle$.*

Notation 2.2.10 *We write N for the marked Petri net $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M \rangle$, N_0 for the marked net $\langle \mathcal{E}_0, \mathcal{B}_0, \text{pre}_0, \text{post}_0, M_0 \rangle$, and so on. To avoid ambiguity we shall state whether a net is marked or not.*

Definition 2.2.11 *Let N be a Petri net. An event $e \in \mathcal{E}$ is a multiple event if there exists an $e' \in \mathcal{E}$ such that $e \neq e'$ and both*

$$\text{pre}(e) = \text{pre}(e') \text{ and } \text{post}(e) = \text{post}(e').$$

Definition 2.2.12 *We write MPetri for the set of marked Petri nets N such that \mathcal{E} and \mathcal{B} are finite and \mathcal{E} contains no multiple events.*

The set of nets MPetri is of particular concern to us in Part IV.

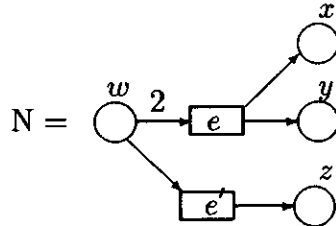
2.2.1 Representing Petri Nets Graphically

There is a graphical representation of Petri nets in which events are represented by labelled boxes, conditions by labelled circles, and the pre- and post-condition relations by weighted, directed arcs.

Example 2.2.13 The net N with $\mathcal{E} = \{e, e'\}$, $\mathcal{B} = \{w, x, y, z\}$ and pre- and post-condition relations given by

$$\text{pre}(e) = 2w \quad \text{pre}(e') = w \quad \text{post}(e) = x + y \quad \text{post}(e') = z$$

is represented graphically by



The marking of a net is represented by small, shaded circles within the circle representing the marked condition. Thus

\odot^a represents the marking $2a$, and

\odot^n represents the marking na .

We shall call these shaded circles *tokens*.

2.3 The Evolution of Petri Nets

Definition 2.3.1 Let N be a marked Petri net, and let A be a multiset over \mathcal{E} . The multiset A is enabled if $\text{pre}(A) \subseteq_m M$, where M is the initial marking of N .

Definition 2.3.2 Let N be a marked Petri net. Let A be a multiset over \mathcal{E} enabled in N . We say that N one-step evolves under A from the marking M to the marking M' if

$$M' = (M - \text{pre}(A)) + \text{post}(A).$$

In this case, the events of A are said to occur concurrently.

Definition 2.3.3 Let N be a Petri net and let A be a multiset over \mathcal{E} . A firing of A is the process of one-step evolution under A .

Definition 2.3.4 Let N be a marked net. Let A_0 and A_1 be multisets over \mathcal{E} enabled by marking M . If $A_0 \neq A_1$ and $\text{pre}(A_0) \not\subseteq_m \text{pre}(A_1) \not\subseteq_m M$, then A_0 and A_1 are in conflict.

Notation 2.3.5

If N one-step evolves under A from M to M' then we write $M' \leq_1 M$.

Definition 2.3.6

The derivability relation of a net N , written \leq , is the transitive closure of \leq_1 .

Remark 2.3.7 By definition, \leq_1 is reflexive, since any net can evolve under the empty multiset of events, leaving its marking unchanged.

Definition 2.3.8 A net N evolves from a marking M to a marking M' if $M' \leq M$.

Definition 2.3.9 Let k be a positive integer. A k -bounded Petri net is a marked Petri net $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M \rangle$ such that

$$M(b) \leq k \text{ for all } b \in \mathcal{B}, \text{ and}$$

$$\text{whenever } M' \leq M, \text{ we have } M'(b) \leq k \text{ for all } b \in \mathcal{B}.$$

Definition 2.3.10 A safe Petri net is a 1-bounded Petri net.

2.3.1 Subnets, Augmentation and Restriction

Definition 2.3.11 Let N and N' be Petri nets. N' is a subnet of N if there is an inclusion $\eta: \mathcal{E}' \rightarrow \mathcal{E}$ and an inclusion $\beta: \mathcal{B}' \rightarrow_m \mathcal{B}$ such that for every multiset A over \mathcal{E}' ,

$$\text{pre}'(\eta A) = \beta(\text{pre}(A)) \quad \text{and} \quad \text{post}'(\eta A) = \beta(\text{post}(A)).$$

Definition 2.3.12 Let $N = \langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M \rangle$ be a Petri net. Let $\mathcal{E}' \subseteq \mathcal{E}$. Let \mathcal{B}' be the set $\{b \in \mathcal{B} \mid \exists e \in \mathcal{E}'. (b \in_m (\text{pre}(e) + \text{post}(e)))\}$. Let pre' and

$post'$ be the restrictions of pre and $post$ respectively to multirelations from \mathcal{E}' to \mathcal{B}' . Then the restriction of N to event set \mathcal{E}' , written $N[\mathcal{E}'$ is the net

$$\langle \mathcal{E}', \mathcal{B}', pre|_{\mathcal{E}'}, post|_{\mathcal{E}'} \rangle.$$

Remark 2.3.13 $N[\mathcal{E}'$ is the subnet of N with event set \mathcal{E}' and pre and $post$ relations given by

$$\text{for each } e \in \mathcal{E}', \quad pre'(e) = pre(e)$$

and

$$post'(e) = post(e)$$

Definition 2.3.14 Let N be a Petri net. The removal of an event e from N results in the net $N[(\mathcal{E} \setminus \{e\})$.

Definition 2.3.15 Let N be a Petri net. Let e be an event not in \mathcal{E} with pre-condition set $pre'(e)$ and post-condition set $post'(e)$ (the multisets $pre'(e)$ and $post'(e)$ need not be disjoint from \mathcal{B}). Let \mathcal{B}' be the set $\mathcal{B} \cup \{b \mid b \in_m pre'(e) + post'(e)\}$.

The augmentation of N by e results in the net

$$\langle \mathcal{E} \cup \{e\}, \mathcal{B}', pre \cup pre', post \cup post' \rangle.$$

2.4 Important note on the Definition of Nets

We have tried throughout to avoid unnecessarily strong restrictions on our definition of a Petri net. Thus our basic definition allows a net to have event and condition sets of arbitrary size. In particular, they may be uncountable. Also, our basic definition allows a net to have multiple events.

Certain parts of the theory presented here require a more restricted notion of net: where such restrictions are necessary, they are stated.

In Part II, we consider primarily elementary nets, all of whose pre- and post-condition sets and markings are multisets with no multiplicities exceeding 1.

In Part III, we need not make any restrictions. However, the use of quantales renders equivalent two nets which differ only with respect to multiple events, or events which are never enabled. In Section 7.7 of Part III, we show how more restricted notions of net (for example, nets which are bounded or safe) can be expressed using quantales.

In Part IV we consider marked nets with finite event sets and condition sets, and no multiple events.

Chapter 3

An Introduction to Linear Logic

Linear Logic was introduced by Girard in [Gir86], and this remains the most complete reference. A useful introduction is given by Lafont in the Appendix to [GLT89]. In this chapter, we review the salient features which distinguish linear logic from other logics, give sequent calculi for the various versions of the logic which concern us, define some derived rules and give some definitions pertaining to linear logic formulae.

3.1 Salient Features of Linear Logic

Linear logic differs from intuitionistic logic primarily in the absence of the structural rules of *weakening* and *contraction*. Weakening allows us to prove a proposition in the context of irrelevant (unused) assumptions, while contraction allows us to use a premise an arbitrary number of times. Linear logic has been called a “resource-conscious logic”, since the premises of a sequent must appear exactly as many times as they are used. If the rules for weakening and contraction were added to the logic, then the rules for \otimes and \wedge would be inter-derivable, and we would lose the distinction which linear logic makes between these two “flavours” of *and*. $A \otimes B$ is to be regarded as a resource consisting of exactly one resource A and one resource B : by contrast, $A \wedge B$ has the potential to be either a resource A or a resource B , but cannot be both. Dropping weakening and contraction decreases

the expressibility of the logic in some ways, although we can regain their power in a controlled way by using the “of course” operator, written $!$. This is a modal operator with the following proof rules:

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} (\text{Cont}) \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} (\text{Derel})$$

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B} (\text{Weak}) \qquad \frac{! \Gamma \vdash A}{! \Gamma \vdash !A} (!R)$$

These rules are together equivalent to the single rule

$$\frac{}{!A \vdash I \wedge A \wedge (!A \otimes !A)} (!)$$

From this rule we see that if we can assert $!A$, then we can make arbitrarily many (that is, zero or more) assertions of A .

3.2 The Sequent Calculi of Linear Logic

There are two basic presentations of linear logic, one using sequents with single conclusions, and one allowing multiple conclusions. The use of multiple conclusions makes clearer the symmetries between connectives, and is used in Part II, where we consider categorical models of linear logic. In Parts III and IV we are not concerned with the connective \sharp (which Girard calls *par*, and others have called a *tensor sum*), and therefore use calculi with single conclusions.

Remark 3.2.1 *There are several approaches to naming the constants of linear logic. We here follow a category-theoretic tradition, using I as the unit of the symmetric monoidal structure \otimes , $\mathbf{1}$ for the terminal object and unit of \wedge , $\mathbf{0}$ for the initial object and unit of \oplus , and \perp for absurdity, the unit of the symmetric monoidal structure \sharp .*

Elsewhere, the reader will find \mathbf{F} and \mathbf{T} used as the units of \wedge and \oplus respectively, in the tradition of lattice-based models. Also, $\mathbf{1}$ is sometimes used for the

unit of \otimes , since it is essentially a unit with respect to multiplication, by contrast with $\mathbf{0}$, the unit with respect to addition (\oplus).

We now present the various calculi which will be used in this thesis.

3.2.1 Sequents with Multiple Conclusions

Our sequents have the form

$$G_0, G_1, \dots, G_n \vdash D_0, D_1, \dots, D_m,$$

where the commas on the left are interpreted as conjunction and correspond to the connective \otimes , while the commas on the right are interpreted as disjunction and correspond to the connective $\#$.

We assume a countably infinite set of linear atoms, and a set comprising the constants $\mathbf{1}$, \perp , $\mathbf{0}$ and $\mathbf{1}$. The sequents of **Classical Linear Logic** are generated using the following rules, in which roman capitals represent formulae of **Lin** as defined in Section 3.4.1, and Greek capitals represent sequences of such formulae:

Axioms:

$$\begin{array}{ccc} \frac{}{A \vdash A} (\text{Identity}) & \frac{}{\vdash \mathbf{1}} (\text{IR}) & \frac{}{\perp \vdash} (\perp\text{L}) \\ \frac{}{\Gamma \vdash \mathbf{1}, \Delta} (\mathbf{1}\text{R}) & & \frac{}{\Gamma, \mathbf{0} \vdash \Delta} (\mathbf{0}\text{L}) \\ \frac{}{A \vdash A^{\perp\perp}} (\text{negR}) & & \frac{}{A^{\perp\perp} \vdash A} (\text{negL}) \end{array}$$

Structural Rules¹:

$$\frac{\Gamma \vdash \Delta}{\sigma\Gamma \vdash \tau\Delta} (\text{Exch}) \quad \frac{\Gamma \vdash A, \Delta \quad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta', \Delta} (\text{Cut})$$

Logical Rules:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta} (\text{varR}) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta} (\text{varL})$$

¹In the rule (Exch), σ and τ are permutations of the sequences Γ and Δ respectively.

Rules for the Multiplicatives:

$$\frac{\Gamma \vdash \Delta}{\Gamma, I \vdash \Delta} (\text{IL})$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta} (\perp\text{R})$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta} (\otimes\text{L})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} (\otimes\text{R})$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \# B \vdash \Delta, \Delta'} (\#\text{L})$$

$$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \# B, \Delta} (\#\text{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'} (\multimap\text{L})$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta} (\multimap\text{R})$$

Rules for the Additives:

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} (\wedge\text{R}) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} (\wedge\text{L1}) \quad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} (\wedge\text{L2})$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta} (\oplus\text{L}) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \oplus B, \Delta} (\oplus\text{R1}) \quad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \oplus B, \Delta} (\oplus\text{R2})$$

Rules for the Modalities²:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} (\text{Derel}) \quad \frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} (\text{Weak})$$

$$\frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} (\text{Cont}) \quad \frac{! \Gamma \vdash \Delta}{! \Gamma \vdash ! \Delta} (!\text{R})$$

²If Γ is the sequence G_0, \dots, G_n , we understand $! \Gamma$ to stand for the sequence obtained by applying $!$ to each of the members of Γ , thus $! \Gamma = !G_0, !G_1, \dots, !G_n$

3.2.2 The Fragment LL_* of Linear Logic

The fragment LL_* of linear logic, which we shall be using in Part II of this thesis, is intuitionistic rather than classical in flavour.

It consists of all the rules for classical linear logic given above, other than the rules (negL) and (varR). Thus there are only two rules for negation:

$$\frac{}{A \vdash A^{\perp\perp}}(\text{negR}) \quad \text{and} \quad \frac{\Gamma \vdash B, \Delta}{\Gamma, B^{\perp} \vdash \Delta}(\text{varL}).$$

3.2.3 Sequents with Single Conclusions

We now present the single conclusioned sequent calculus for linear intuitionistic logic without negation.

Axioms:

$$\frac{}{A \vdash A}(\text{Identity}) \quad \frac{}{\vdash \mathbf{1}}(\text{IR})$$

$$\frac{}{\Gamma \vdash \mathbf{1}}(\mathbf{1}) \quad \frac{}{\Gamma, \mathbf{0} \vdash A}(\mathbf{0})$$

Structural Rules:

$$\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}(\text{Cut}) \quad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}(\text{Exch})$$

Rules for the Multiplicatives:

$$\frac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A}(\text{IL}) \quad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}(\otimes\text{L}) \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}(\otimes\text{R})$$

$$\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}(\multimap\text{L}) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}(\multimap\text{R})$$

Rules for the Additives:

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} (\wedge R) \quad \frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C} (\wedge L1) \quad \frac{\Gamma, B \vdash C}{\Gamma, A \wedge B \vdash C} (\wedge L2)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} (\oplus R1) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} (\oplus R2) \quad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C} (\oplus L)$$

Rules for the Modalities:

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} (\text{Derel}) \quad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} (\text{Weak})$$

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} (\text{Cont}) \quad \frac{! \Gamma \vdash A}{! \Gamma \vdash !A} (!R)$$

The fragment of this logic which we shall use most often is \mathcal{L}_1 , which consists of all the above rules other than the three rules for \oplus , the rule $(-\circ R)$ and the rules **(0)** and **(1)**.

The fragment \mathcal{L} is \mathcal{L}_1 together with the rule $(-\circ R)$.

The fragment \mathcal{L}_0 is \mathcal{L}_1 without the three rules for \wedge . Thus \mathcal{L}_0 is essentially the multiplicative fragment of linear logic, together with the modalities. These three fragments of linear logic are the calculi used in Parts III and IV of this thesis.

3.3 Some Useful Derived Rules

There are three derived rules which we shall find useful. The first of these gives a useful insight into the meaning of the ! operator:

$$\frac{}{!A \vdash (A \wedge I) \otimes !A} \text{(Copy)}.$$

Using this rule it is easy to see that from a resource of type !A we can obtain as many copies as we like of the resource A. We derive this axiom in \mathcal{L}_1 as follows:

$$\frac{\frac{\frac{}{!A \vdash I} \text{(Identity)} \quad \frac{}{!A \vdash A} \text{(\wedge R)}}{!A \vdash A \wedge I} \text{(\otimes R)} \quad \frac{}{!A, !A \vdash (A \wedge I) \otimes !A} \text{(Cont)}}{!A \vdash (A \wedge I) \otimes !A} \text{(Id)}$$

Another useful derived rule is the rule (Imp),

$$\frac{A \vdash A \quad \Gamma, B \vdash C}{\Gamma, A, (A \multimap B) \wedge I \vdash C} \text{(Imp)}$$

This rule is derived in \mathcal{L}_1 using the rules ($\multimap L$) and ($\wedge L1$).

We use the derived rule (Id), which is in fact an axiom scheme, to abbreviate a sequence of ($\otimes R$) and (Identity) rules. (Id) is given by the following scheme, where n is a non-zero integer:

$$\frac{}{C_1, C_2, \dots, C_n \vdash C_1 \otimes C_2 \otimes \dots \otimes C_n} \text{(Id)}$$

(Id) is derived using the rules (Identity) and ($\otimes R$). For instance, for $n = 3$, the derivation is:

$$\begin{array}{c}
 \frac{}{A \vdash A} \text{ (Identity)} \quad \frac{}{B \vdash B} \text{ (Identity)} \\
 \hline
 A, B \vdash A \otimes B \quad \frac{}{C \vdash C} \text{ (Identity)} \\
 \hline
 A, B, C \vdash A \otimes B \otimes C \quad (\otimes R)
 \end{array}$$

Girard's rule for dereliction, which we have called (Derel), is as follows:

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \text{ (Derel)}$$

In place of this rule we shall often use the rule (Der), which is as follows:

$$\frac{\Gamma, A \wedge I, !A \vdash B}{\Gamma, !A \vdash B} \text{ (Der)}$$

It is easy to show that (Der) and (Derel) are interderivable in \mathcal{L}_1 and \mathcal{L} :

(Der) \Rightarrow (Derel)

$$\begin{array}{c}
 \frac{}{\Gamma, A \vdash B} \text{ (}\wedge\text{ L1)} \\
 \hline
 \Gamma, (A \wedge I) \vdash B \text{ (Weak)} \\
 \hline
 \Gamma, (A \wedge I), !A \vdash B \text{ (Der)} \\
 \hline
 \Gamma, !A \vdash B
 \end{array}$$

(Derel) \Rightarrow (Der)

$$\begin{array}{c}
 \frac{}{\Gamma, (A \wedge I), !A \vdash B} \text{ (}\otimes\text{L)} \quad \frac{}{} \text{ (Copy)} \\
 \hline
 \Gamma, (A \wedge I) \otimes !A \vdash B \quad !A \vdash (A \wedge I) \otimes !A \text{ (Cut)} \\
 \hline
 \Gamma, !A \vdash B
 \end{array}$$

3.4 Formulae of Linear Logic

Definition 3.4.1

Linear logic formulae are words over the following alphabet:

parentheses “(” and “)” ,

atoms a_0, a_1, \dots , which are assumed to constitute a countably infinite set,

constants \mathbb{I} , $\mathbf{1}$, $\mathbf{0}$, and \perp , and

binary operators \otimes , \wedge , \multimap , $\#$, \oplus , and

a unary operator $!$.

The set of linear logic formulae \mathbf{Lin} is defined inductively to be the least set satisfying the rules:

$$\frac{}{a \in \mathbf{Lin}} \text{ } a \text{ an atom} \qquad \frac{}{c \in \mathbf{Lin}} \text{ } c \text{ a constant}$$

$$\frac{M \in \mathbf{Lin} \quad N \in \mathbf{Lin}}{(M \circ N) \in \mathbf{Lin}} \text{ } \circ \text{ a binary operator}$$

$$\frac{M \in \mathbf{Lin}}{!M \in \mathbf{Lin}}$$

Convention 3.4.2

1. Outermost parentheses are not written.
2. \otimes is associative.
3. \otimes binds more strongly than any other binary operator, and so whenever $(M \otimes N)$ is not bound by a $!$ operator, we shall omit the parentheses and write $M \otimes N$.

4. The binary connectives all associate to the right.

5. We shall write A^\perp as an abbreviation for $A \multimap \perp$.

Remark 3.4.3

The above conventions allow us to omit parentheses from formulae without ambiguity.

Definition 3.4.4 Let A and F be linear logic formulae. We define a relation “is a subformula of” on **Lin**, written \leq , as follows:

$$\frac{}{A \leq A} \qquad \frac{A \leq M}{A \leq !M}$$

$$\frac{A \leq M}{A \leq (M \circ N)} \circ \text{ a binary operator}$$

$$\frac{A \leq N}{A \leq (M \circ N)} \circ \text{ a binary operator}$$

Definition 3.4.5

Let F be a linear logic formula. We define the set At of atoms of F by

$$At(F) = \{a \mid a \leq F, a \text{ an atom.}\}$$

Definition 3.4.6 Let A and F be linear logic formulae. We define a relation “is a factor of” on **Lin**, written $|$, as follows:

$$\frac{}{A|A} \qquad \frac{A|M}{A|!M}$$

$$\frac{A|M}{A|(M \otimes N)} \qquad \frac{A|N}{A|(M \otimes N)}$$

Remark 3.4.7 We use the intuitive definition of an occurrence of a subformula in a formula. Formal details of the definition may be found in [Hue85].

Definition 3.4.8 Let F be a linear logic formula, and let A be a set of atoms distinct from $At(F)$ such that there is a bijective function $r: At(F) \rightarrow A$. Then r gives rise to a formula F' with $At(F') = A$, where F' is obtained by replacing each occurrence of each atom a in F by $r(a)$.

We say F' is obtained from F by replacement using r .

Example 3.4.9 Let $F = A \otimes ((B \multimap C) \wedge D)$, and let $r: \{A, B, C, D\} \rightarrow \{W, X, Y, Z\}$ be given by $r(A) = W$, $r(B) = X$, $r(C) = Y$ and $r(D) = Z$. Then the formula $W \otimes ((X \multimap Y) \wedge Z)$ is the formula obtained from F by replacement using r .

Definition 3.4.10 Let F and F' be linear logic formulae. We say there is a formula isomorphism from F to F' if either

1. we can find a bijection $r: At(F) \rightarrow At(F')$ such that F' is obtained from F by replacement using r , or
2. we can find a formula F'' such that $At(F'')$ is distinct from $At(F)$ and from $At(F')$ and bijections $r_1: At(F) \rightarrow At(F'')$ and $r_2: At(F'') \rightarrow At(F')$ such that

F'' is obtained from F by replacement using r_1 , and

F' is obtained from F'' by replacement using r_2 .

By abuse of notation, we shall say that F' is obtained from F by replacement using r , where $r = r_1; r_2$.

Definition 3.4.11 A transposition of a formula M is either

1. the substitution of a subformula $A \otimes B$ of M by $B \otimes A$, or
2. the substitution of a subformula $A \wedge B$ of M by $B \wedge A$.

Definition 3.4.12

M is a permutation of N if M results from N by a sequence of transpositions.

Notation 3.4.13 We shall be concerned with the set of equivalence classes of \mathbf{Lin} under permutation, which we shall denote \mathbf{Lin}_{\cong} .

Convention 3.4.14

We shall use formulae in the same equivalence class of \mathbf{Lin}_{\cong} interchangeably.

Remark 3.4.15 The motivation for this convention is that formulae which are permutations of one another are interderivable in the fragment \mathcal{L}_1 of linear logic. Interderivability of $A \otimes B$ and $B \otimes A$ follows immediately from the (Exch) rule. It is also easy to show that $A \wedge B$ and $B \wedge A$ are interderivable:

$$\begin{array}{c}
 \frac{}{A \vdash A} \text{ (Id)} \quad \frac{}{B \vdash B} \text{ (Id)} \\
 \frac{A \vdash A}{A \wedge B \vdash A} (\wedge L1) \quad \frac{B \vdash B}{A \wedge B \vdash B} (\wedge L2) \\
 \hline
 \frac{A \wedge B \vdash A \quad A \wedge B \vdash B}{A \wedge B \vdash B \wedge A} (\wedge R)
 \end{array}$$

We shall call the derived axiom $\overline{A \wedge B \vdash B \wedge A}$ the rule (Com).

It follows that formulae which are permutations of one another are always interderivable using only the rules (Exch) and (Com), and so we identify them when doing proofs. This convention prevents proofs from becoming unnecessarily long and cumbersome, as it allows us to omit applications of the rules (Com) and (Exch). For example, we write

$$\frac{\frac{}{A \vdash A} \text{ (Id)} \quad \frac{}{B, C, D \vdash B \otimes C \otimes D} \text{ (Id)}}{B, A, A \multimap C, D \vdash B \otimes C \otimes D} (\multimap L)$$

to stand for the derivation

$$\begin{array}{c}
\frac{}{A \vdash A} \text{ (Id)} \quad \frac{}{B, C, D \vdash B \otimes C \otimes D} \text{ (Id)} \\
\frac{}{A \vdash A} \text{ (Id)} \quad \frac{B, C, D \vdash B \otimes C \otimes D}{B, D, C \vdash B \otimes C \otimes D} \text{ (Exch)} \\
\frac{}{A \vdash A} \quad \frac{B, D, C \vdash B \otimes C \otimes D}{B, D, A, A \multimap C \vdash B \otimes C \otimes D} \text{ (}\multimap\text{L)} \\
\frac{B, D, A, A \multimap C \vdash B \otimes C \otimes D}{B, A, D, A \multimap C \vdash B \otimes C \otimes D} \text{ (Exch)} \\
\frac{B, A, D, A \multimap C \vdash B \otimes C \otimes D}{B, A, A \multimap C, D \vdash B \otimes C \otimes D,} \text{ (Exch)}
\end{array}$$

and

$$\frac{\Gamma, A \wedge B \vdash C \quad \Delta \vdash B \wedge A}{\Gamma, \Delta \vdash C} \text{ (Cut)}$$

to stand for the derivation

$$\begin{array}{c}
\frac{}{\Delta \vdash B \wedge A} \text{ (Com)} \quad \frac{}{B \wedge A \vdash A \wedge B} \text{ (Com)} \\
\frac{\Delta \vdash B \wedge A \quad B \wedge A \vdash A \wedge B}{\Delta \vdash A \wedge B} \text{ (Cut)} \\
\frac{\Gamma, A \wedge B \vdash C \quad \Delta \vdash A \wedge B}{\Gamma, \Delta \vdash C} \text{ (Cut)}
\end{array}$$

The connectives of most interest to us in Part IV are \otimes and \multimap . If we think in terms of resources, then whenever A and B represent resources, $A \otimes B$ represents a resource comprising both A and B . The rule (\otimes R) expresses the idea that if resources Γ can be used to prove A , and also resources Δ can be used to prove B , then the resources $\Gamma \otimes \Delta$ can be used to prove $A \otimes B$. We can think of the two proofs going on in parallel, since there is no communication between them and each is valid independent of the other.

Thus \otimes joins two resources which have no causal interdependence.

In contrast, because of the absence of weakening in linear logic (other than in the special case of formulae of the form $!F$), whenever it is the case that $A \multimap B$ and there is no formula F such that $A = !F$, then it must be the case that B has a real

causal dependence on A , since A is used up whenever B is produced. Therefore, linear implication can be used to show when a necessary causal dependence exists between two formulae or resources.

In order to produce an algorithm which does a computation in parallel, it is necessary to establish exactly which parts of the algorithm can proceed independently of one another, and which are necessarily dependent. Since (as we saw above) linear logic explicitly expresses both these relationships between portions of a computation, it is natural that linear logic should lend itself to describing parallel computation. In what follows, we shall see that there is a close connection between linear logic and Petri nets.

Part II

A Categorical Linear Framework for Petri Nets

Introduction to Part II

In this part we shall define several closely related categories, each of which has as its object set either the set **Petri** of Petri nets or a subset of **Petri**. We now outline briefly three major benefits of putting Petri nets (and also other models of concurrency) in a categorical framework.

Firstly, viewing processes as the objects of a category gives us a compositional treatment of processes. In particular, if the category has sufficient structure, it will give us various ways of building up a large process from smaller ones in such a way that the behaviour of the whole can be expressed in terms of the behaviour of its parts. This modular approach facilitates both specification and verification of complex processes.

Secondly, given a category whose objects are processes, we aim to use categorical logic to develop proof systems and specification languages which describe parallel processes.

Finally, the use of categories allows us to relate different models of concurrency. Glynn Winskel [Win84a] has shown that if we view each class of models for processes as a category, by providing it with a suitable notion of morphism, then in many cases the relationship between the models arises as a co-reflection of categories. Since right adjoints preserve limits and left adjoints colimits, this allows us to pass smoothly from the semantics of one model to those of another.

This part consists of three chapters. Some of the work is to be found in [BG90] and [BG].

In Chapter 4 we give an overview of existing approaches to categories of Petri nets and describe Valeria de Paiva's dialectica category **GC**, which is a sound

model of linear logic. We modify her construction to define a category **NC** whose objects are elementary Petri nets, and discuss the interpretation of morphisms in **NC** as refinements of Petri nets.

In Chapter 5 we characterise **NC** as the kernel pair of the forgetful functor from **GC** into $\mathbf{C} \times \mathbf{C}^{op}$. This enables us to show that the relevant structure of **GC** lifts to **NC**, which is therefore also a sound model of linear logic. We describe the structure of **NC** in detail, and interpret it in terms of Petri nets.

In Chapter 6 we show that the category **NSet** is closely related to Winskel's category **SafeNet** and that **NPSet** is precisely the category described by Nielsen, Rozenberg and Thiagarajan in [NRT90], with object set the set of elementary nets. We discuss three modifications to our notion of refinement, and the extent to which the modified categories remain sound models of linear logic. We show that Winskel's category **SafeNet** is isomorphic to the Kleisli category on **NSet** for a particular choice of monad.

In ~~our conclusion~~ we describe briefly a generalisation of de Paiva's work which allows us to define a category with object set the set **Petri** of all Petri nets. This category has sufficient structure to be a sound model of linear logic.

Chapter 4

A New Category of Petri Nets

4.1 Summary of the Chapter

In this chapter we aim to establish the connection between linear logic and Petri nets by relating categories of Petri nets to the *Dialectica category* models of linear logic constructed by de Paiva [deP89a]. For a suitable choice of base category \mathbf{C} , the Dialectica category \mathbf{GC} is a model of a slight variant of intuitionistic linear logic based on that presented by Girard [Gir86]. Using the constructions described in [deP89a], we define a category whose objects are Petri nets and whose morphisms, constructed from those in \mathbf{GC} , are refinement maps of Petri nets. All the connectives of linear logic can be interpreted by constructions in \mathbf{GC} , and these give us constructions in our category of nets which have a computational interpretation in terms of Petri nets. This approach generates both existing constructions on nets, and also some novel constructions which have a useful computational interpretation.

The structure of this chapter is as follows. We first describe some existing work on categories with object set a subset of \mathbf{Petri} . We next define de Paiva's category \mathbf{GC} , and describe its structure. We define by analogy with \mathbf{GC} a category \mathbf{NC} and show that the category \mathbf{NSet} obtained by putting $\mathbf{C} = \mathbf{Set}$ has elementary Petri nets as objects. We discuss morphisms in \mathbf{NSet} in detail and with examples, interpreting them as refinement maps of nets.

In Chapter 5, we shall show that \mathbf{NC} is the kernel pair in \mathbf{Cat} of the forgetful functor from \mathbf{GC} into $\mathbf{C} \times \mathbf{C}^{op}$.

4.2 Existing Categories of Petri Nets

There are at least three important approaches to a categorical treatment of Petri nets in the literature. The first is that of Winskel, described in [Win87] and [Win88], which relates very closely to the categories presented here, as is shown in Section 6.5.1. The second is that of Meseguer and Montanari, described in [MM88a], which is also closely related to our approach. In [DMM89] a different approach is taken, in which an individual Petri net is seen as generating a category. This approach will not concern us here.

4.2.1 Winskel's Category, Net

Winskel defines a category of Petri nets as follows:

- objects are marked Petri nets,
- a morphism from $N = \langle \mathcal{E}, \mathcal{B}, pre, post, M \rangle$ to $N' = \langle \mathcal{E}', \mathcal{B}', pre', post', M' \rangle$ is a pair (η, β) , where η is a partial function from \mathcal{E} to \mathcal{E}' , and β is a multirelation from \mathcal{B} to \mathcal{B}' , such that

$$\beta M = M' \text{ and}$$

for every multiset A over \mathcal{E} ,

$$pre'(\eta A) = \beta(pre(A)) \quad \text{and} \quad post'(\eta A) = \beta(post(A)),$$

- and composition is componentwise.

We shall call this category \mathbf{Net} . A morphism $(\eta, \beta): N \rightarrow N'$ in \mathbf{Net} is interpreted as associating with every possible behaviour of N an induced behaviour in N' . The maps η and β describe the components in N' of events and conditions in N . The

components in N' of the event $e \in \mathcal{E}$ are local in the sense that $\eta(e)$, where defined, is a single event. Because η is partial, not every event of N has a component in N' , and so we think of the map (η, β) as giving a “partial simulation” of N in N' . The following theorem shows how morphisms in **Net** preserve behaviour:

Theorem 4.2.1 Let $(\eta, \beta): N \rightarrow N'$ be a morphism in **Net**. Whenever a marking M of N can evolve under a multiset of events A to marking M' , the marking βM can evolve in N' under ηA to the marking $\beta M'$.

Definition 4.2.2 A morphism (η, β) in **Net** is synchronous if η is total.

Notation 4.2.3 We write **Net**_{syn} for the subcategory of **Net** with object set Petri and morphisms the synchronous morphisms.

4.2.2 Structure in Net

We now describe briefly some of the categorical structure of **Net**.

Net has an *initial object*, which is the net $nil = \langle \phi, \{*\}, \phi, \phi, * \rangle$, consisting of a single condition, marked with multiplicity one, and no events.

Net has a *terminal object*, which is the net $null = \langle \phi, \phi, \phi, \phi, \emptyset \rangle$, with no conditions, events or marking.

The set of safe nets is the object set of a full subcategory of **Net**, which we shall call **SafeNet**. **SafeNet** has a subcategory **SafeNet**_{syn} of safe nets and synchronous morphisms. **SafeNet** has the same initial object and terminal object as **Net**.

Winskel states that **Net** has all finite products, but not all finite coproducts, while **SafeNet** has all finite products and coproducts.

We now give the binary product in **Net** explicitly.

Let $N_k = \langle \mathcal{E}_k, \mathcal{B}_k, pre_k, post_k, M_k \rangle$ be Petri nets for $k = 0, 1$, and let 0 be distinct from each \mathcal{E}_k . The product net $N_0 \times N_1$ is the net $\langle \mathcal{E}, \mathcal{B}, pre, post, M \rangle$, where $\mathcal{E}, \mathcal{B}, pre, post$ and M are defined as follows:

$$\mathcal{E} = \{(e_0, 0) \mid e_0 \in \mathcal{E}_0\} \cup \{(0, e_1) \mid e_1 \in \mathcal{E}_1\} \cup \{(e_0, e_1) \mid e_0 \in \mathcal{E}_0, e_1 \in \mathcal{E}_1\}$$

and

$$\mathcal{B} = \mathcal{B}_0 \uplus \mathcal{B}_1, \text{ the disjoint union of } \mathcal{B}_0 \text{ and } \mathcal{B}_1.$$

For $k = 0, 1$ we define a partial function π_k from \mathcal{E} to \mathcal{E}_k by

$$\pi_k(e_0, e_1) = \begin{cases} e_k & \text{if } e_k \in \mathcal{E}_k \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For $k = 0, 1$ we define ρ_k^{op} to be the opposite relation to the injection $\rho_k: \mathcal{B}_k \rightarrow \mathcal{B}$.

Finally,

$$M = \rho_0^{op} M_0 + \rho_1^{op} M_1,$$

$$pre(e) = \rho_0^{op}[pre_0(\pi_0 e)] + \rho_1^{op}[pre_1(\pi_1 e)] \quad \text{for all } e \in \mathcal{E}, \quad \text{and}$$

$$post(e) = \rho_0^{op}[post_0(\pi_0 e)] + \rho_1^{op}[post_1(\pi_1 e)] \quad \text{for all } e \in \mathcal{E}.$$

The projections from $N_0 \times N_1$ to N_k are the morphisms (π_k, ρ_k^{op}) .

The following theorem shows how the behaviour of the net $N_0 \times N_1$ is related to the behaviour of its component nets.

Theorem 4.2.4

Let N_0 and N_1 be Petri nets.

A multiset M is a marking of $N_0 \times N_1$ if and only if $\rho_0 M$ is a marking of N_0 and $\rho_1 M$ is a marking of N_1 .

Moreover, the marking M of $N_0 \times N_1$ can evolve by a multiset of events A over \mathcal{E} to a marking M' if and only if for $i = 0$ and $i = 1$, the marking $\rho_i M$ can evolve in N_i by the multiset of events $\pi_i A$ to the marking $\rho_i M'$.

It can be shown that **Net** does not have all finite coproducts (see [Win87]). However, Winskel states that **SafeNet** does have all finite coproducts. We now describe explicitly the coproduct in **SafeNet** of two nets with non-empty initial

markings.

Let $N_k = \langle \mathcal{B}_k, \mathcal{E}_k, pre_k, post_k, M_k \rangle$ for $k = 0, 1$ be safe nets with non-empty initial markings. Their coproduct $N_0 + N_1$ is the net $N = \langle \mathcal{E}, \mathcal{B}, pre, post, M \rangle$, where $\mathcal{E}, \mathcal{B}, pre, post$ and M are defined as follows:

$\mathcal{E} = \mathcal{E}_0 \uplus \mathcal{E}_1$, the disjoint union of the event sets of the components. For $k = 0, 1$ we write in_k for the injection from \mathcal{E}_k into \mathcal{E} .

$M = M_0 \times M_1$, the cartesian product of sets.

$$\mathcal{B} = \{(b_0, 0) \mid b_0 \in \mathcal{B}_0 \setminus M_0\} \cup \{(0, b_1) \mid b_1 \in \mathcal{B}_1 \setminus M_1\} \cup M.$$

The injection relations ρ_0 and ρ_1 on the condition sets are opposite to the evident partial functions taking an element of \mathcal{B} to its first or second component, and they are given by:

$$b_0 \rho_0 b \Leftrightarrow \exists b_1 \in \mathcal{B}_1 \cup \{0\}. b = (b_0, b_1), \quad \text{and}$$

$$b_1 \rho_1 b \Leftrightarrow \exists b_0 \in \mathcal{B}_0 \cup \{0\}. b = (b_0, b_1).$$

We define pre as follows: for $e_0 \in \mathcal{E}_0$,

$$pre(in_0 e_0) = \rho_0[pre_0(e_0)] = \{b \mid b \in \mathcal{B}, b = (b_0, b_1) \text{ with } b_0 \in pre_0(e_0), b_1 \in \mathcal{B}_1 \cup \{0\}\}$$

and similarly $pre(in_1 e_1)$ for $e_1 \in \mathcal{E}_1$.

We define $post$ analogously.

The behaviour of the coproduct net can be described in terms of the behaviour of its components using the following theorem:

Theorem 4.2.5

Let N_0 and N_1 be safe Petri nets with non-empty initial markings.

A marking M can evolve in $N_0 + N_1$ by a multiset of events A over \mathcal{E} to the marking M' if and only if *either* for $i = 0$ *or* for $i = 1$, there exist multisets M_i, A_i and M'_i with $M = \rho_i M_i$, $A = in_i A_i$ and $M' = \rho_i M'_i$, such that the marking M_i can evolve in N_i by the multiset of events A_i to the marking M'_i .

Winskel states that the definition and theorem extend to sums of safe nets with empty initial markings.

Synchronisation Algebras

A variety of parallel compositions on nets can be obtained from the product. Winskel shows how restrictions of the event set of the product net can be specified using a synchronisation algebra [Win84b], in a way which we describe here briefly.

Definition 4.2.6

A synchronisation algebra is a tuple $\langle L, \bullet, \mathbf{X}, \mathbf{0} \rangle$ where L is a set containing the distinct, distinguished elements \mathbf{X} and $\mathbf{0}$, the set $L \setminus \{\mathbf{X}, \mathbf{0}\}$ has at least one element, and \bullet is a commutative, associative, binary operation on L such that for all $\alpha, \alpha_0, \alpha_1 \in L$,

$$\alpha \bullet \mathbf{0} = \mathbf{0} \text{ and}$$

$$\alpha_0 \bullet \alpha_1 = \mathbf{X} \text{ if and only if } \alpha_0 = \alpha_1 = \mathbf{X}.$$

We call elements of L labels.

We interpret the synchronisation algebra $\langle L, \bullet, \mathbf{X}, \mathbf{0} \rangle$ in the following way. We associate with a set of events \mathcal{E} a function $l: \mathcal{E} \rightarrow (L \setminus \{\mathbf{X}, \mathbf{0}\})$. An event $e \in \mathcal{E}$ can occur asynchronously if and only if $l(e) \bullet \mathbf{X} \neq \mathbf{0}$. Two events e_0 and e_1 can never synchronise if $l(e_0) \bullet l(e_1) = \mathbf{0}$.

Definition 4.2.7

Let N_0 and N_1 be Petri nets. Let $\langle L, \bullet, \mathbf{X}, \mathbf{0} \rangle$ be a synchronisation algebra.

For $k = 0, 1$ let l_k be a function from \mathcal{E}_k to $(L \setminus \{\mathbf{X}, \mathbf{0}\})$, called a labelling function.

We define the parallel composition of N_0 and N_1 with respect to L to be the net obtained by restricting $N_0 \times N_1$ to events

$$\mathcal{E}' = \{e \in \mathcal{E}_0 \times \mathcal{E}_1 \mid l_0\pi_0(e) \bullet l_1\pi_1(e) \neq \mathbf{0}\}.$$

There is an induced labelling function $l: \mathcal{E}' \rightarrow L$ given by

$$l(e) = l_0\pi_0(e) \bullet l_1\pi_1(e).$$

Synchronisation algebras can be used to obtain uniformly all binary, commutative, associative parallel compositions which are based on synchronisation. Unfortunately, Winskel can define restriction and parallel composition as categorical operations only in the more complex framework of indexed categories, see [Win88]. The difficulties involved are a special case of the problem of presenting subobjects in a category constructively.

4.2.3 Meseguer and Montanari's Categories of Nets

In [DMM89], Degano, Meseguer and Montanari define four categories to correspond to each Petri net N . These categories are all aimed at axiomatising the *behaviour* of N as a category, and thus work at a different level from the constructions which we shall be considering in this chapter.

In their earlier paper, Meseguer and Montanari [MM88a] introduce more than 25 categories whose objects are either Petri nets or behaviours of Petri nets. We summarise their constructions briefly. Their approach is to regard a Petri net as algebraic structure over a directed graph.

Definition 4.2.8 *A (directed) graph is a 4-tuple $\langle E, V, \delta_0, \delta_1 \rangle$ where E and V are sets, called edges and nodes respectively, and δ_0 and δ_1 are functions from E to V called respectively Source and Target.*

Definition 4.2.9

A (directed) graph morphism from $\langle E, V, \delta_0, \delta_1 \rangle$ to $\langle E', V', \delta'_0, \delta'_1 \rangle$ is a pair of functions (f, g) with $f: E \rightarrow E'$ and $g: V \rightarrow V'$ such that

$$g\delta_0 = \delta'_0 f \quad \text{and} \quad g\delta_1 = \delta'_1 f.$$

In a natural way, we can regard a Petri net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ as a directed graph whose set of nodes is the free commutative monoid \mathcal{B}^\oplus over the set of conditions \mathcal{B} , the source and target maps δ_0 and δ_1 from \mathcal{E} to \mathcal{B}^\oplus being determined in the evident way by *pre* and *post* respectively. Nets regarded in this way, together with

(directed) graph morphisms which respect the monoid operation \oplus and leave the neutral element, $\mathbf{0}$, of \mathcal{B}^\oplus fixed, form a category Petri with finite products and coproducts. The product of two nets in Petri is a *synchronous* product in much the same way as Winskel's product, pairing events and taking the union of the condition sets. The coproduct represents juxtaposition without interaction.

To allow events to be "erased" by a mapping (equivalently, to allow the map on events to be partial), we build a new category Petri₀ by adding to the event set of each net a special element 0, and restricting morphisms to morphisms of Petri which leave 0 fixed. Meseguer and Montanari state that product and coproduct are as before. Further, if we give the event set of each net a commutative monoid structure $(\mathcal{E}, +, 0)$ (thought of as representing parallel composition) and restrict morphisms to those morphisms of Petri₀ whose first component is a monoid homomorphism, we have a category CMonPetri. CMonPetri has products and coproducts, and these coincide. Meseguer and Montanari suggest that the monoid on an event set need not be free, as it is intended to reflect the synchronisation structure of the net. However, there is no apparent way to prevent its being free, unless we consider some further structure to be associated with the net, such as a synchronisation algebra.

If we add to every net an identity event corresponding to each node, we obtain from Petri and CMonPetri two further categories, RPetri and CMonRPetri, with forgetful functors as follows:

$$\begin{array}{ccc}
 \underline{CMonRPetri} & \longrightarrow & \underline{RPetri} \\
 \downarrow & & \downarrow \\
 \underline{CMonPetri} & \longrightarrow & \underline{Petri}
 \end{array}$$

Finally, given a Petri net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$, we can define a partial function \cdot ; from $\mathcal{E} \times \mathcal{E}$ to \mathcal{E} which is defined for exactly those pairs (e, e') for which $\delta_1(e) = \delta_0(e')$. \cdot ; is intended to represent sequential composition of events. Several axioms are satisfied by \cdot ; , the most important of which are associativity and a form of distributivity of $+$ over \cdot ; . This states that for all events e, e', x, x' for

which the compositions are defined, we have

$$(e + e'); (x + x') = (e; x) + (e'; x').$$

Morphisms are morphisms of *CMonRPetri* with the additional constraint that whenever $e; e'$ is defined for events e, e' in a net N , and (f, g) is a morphism from N to N' , $f(e; e') = f(e); f(e')$. This defines a category which we shall call *CatPetri*.

Between these categories there are evident forgetful functors as follows:

$$\underline{\text{CatPetri}} \longrightarrow \underline{\text{CMonRPetri}} \longrightarrow \underline{\text{CMonPetri}} \longrightarrow \underline{\text{Petri}_0} \longrightarrow \underline{\text{Petri}}.$$

There exists a left adjoint T to the composite forgetful functor from *CatPetri* to *Petri* which assigns to each net N a category $T(N)$ which has an arrow for every possible computation of the net N , including all sequential and parallel compositions of events in N or identity events. Further details are given in [DMM89].

The morphisms in the categories defined are classified as follows, according to their action on events:

- in *Petri*, an event e in N maps to an event e' in N' ,
- in *Petri*₀, an event e in N maps to an event e' in N' , or is erased,
- in *CMonPetri*, an event e in N maps to a parallel composition of events in N' , or is erased,
- in *CMonRPetri*, an event e in N maps to a parallel composition of events in N' and idle (identity) events, or is erased
- in *CatPetri*, an event e in N maps to a computation in N' with possibly many sequential and parallel steps, or is erased.

In the construction of the above categories, the objects are given increasingly more structure, and morphisms are chosen to respect that structure. It is of course possible to form categories with object set **Petri**, but with increasingly general morphisms to correspond to those in the categories defined above. In particular, using

morphisms from the categories \underline{Petri}_0 , $\underline{CMonPetri}$, $\underline{CMonRPetri}$ and $\underline{CatPetri}$, we obtain the categories $\underline{AsynchPetri}$, $\underline{LinPetri}$, \underline{CPetri} and $\underline{ImplPetri}$ respectively, which relate to one another as follows:

$$\underline{Petri} \subseteq \underline{AsynchPetri} \subseteq \underline{LinPetri} \subseteq \underline{CPetri} \subseteq \underline{ImplPetri}.$$

For any of these categories we could prove a theorem analogous to Theorem 4.2.1, showing that morphisms preserve dynamic behaviour.

Each of the categories discussed in this section has an analogue for marked nets whose initial marking is a set.

Meseguer and Montanari do not offer any judgement as to which of these is the most useful category for increasing our understanding of Petri nets. It seems that some of the categories relate better to existing concepts than others. The choice of strict monoidal functors as morphisms between nets is a strong restriction on morphisms, and perhaps does not give rise to categories with interesting, useful structure. Moreover, it is debatable whether parallel composition should be assumed to distribute over sequential. For instance, if we wish to consider the *time* taken by processes, or their *causal dependencies*, we certainly do not want such a distributivity law (see [Gur90]).

4.3 The Dialectica Category GC

The work of de Paiva on dialectica categories, described in [deP89b], [deP89a] and [deP87], was begun with the aim of providing a categorical treatment of Gödel's "Dialectica Interpretation" of higher order arithmetic [Göd58] [Göd80]. De Paiva defines a category whose objects represent essentially the Φ^D , where Φ is a formula in higher-order arithmetic and $(\)^D$ is the Dialectica translation, see [Tro73]. Morphisms then correspond to Dialectica interpretations of implication. In the usual manner of categorical logic, a map from Φ^D to Ψ^D can be taken to be some kind of realisation of the formula " $\Phi^D \rightarrow \Psi^D$ ". In the case of the

Dialectica Interpretation this realisation can be given abstractly, leading to the notion of a Dialectica category \mathbf{DC} for an arbitrary category \mathbf{C} with finite limits.

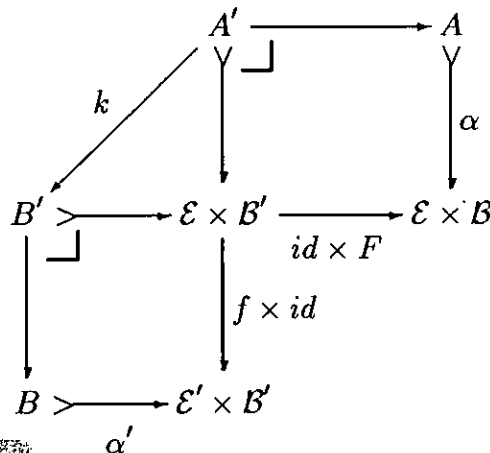
The dialectica category \mathbf{DC} is not cartesian closed, but has a product and a symmetric monoidal structure, each of which corresponds to a notion of conjunction. These observations led to the realisation that \mathbf{DC} is a model of the intuitionistic fragment of linear logic [GL87]. Following a suggestion of Girard, de Paiva constructed a category \mathbf{GC} which is a model of the version LL_* of linear logic. It is the category \mathbf{GC} which we shall use and extend in this part. \mathbf{GC} is an interesting model of linear logic in the sense that it has distinct interpretations of all five binary connectives (\otimes , \wedge , \oplus , $\#$ and \multimap), and distinct objects interpreting the units \top , $\mathbf{1}$, $\mathbf{0}$ and \perp . Many models identify one or more of these connectives, and similarly one or more of the constants.

We now define the category \mathbf{GC} and describe its structure.

Definition 4.3.1

Let \mathbf{C} be a category with finite limits. The Dialectica category \mathbf{GC} is given as follows:

- an object of \mathbf{GC} is a pair $\langle \mathcal{E}, B \rangle$ of objects of \mathbf{C} , together with a subobject $A \hookrightarrow \mathcal{E} \times B$ in \mathbf{C} . We shall denote such an object $(\mathcal{E} \xleftarrow{\alpha} B)$.
- A morphism in \mathbf{GC} from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$, is a pair of morphisms $f : \mathcal{E} \rightarrow \mathcal{E}'$ and $F : B' \rightarrow B$ in \mathbf{C} such that there exists a morphism k (necessarily monic and unique up to isomorphism) such that the triangle in :



commutes, and

- composition is given by composition in \mathbf{C} in each component.

Notation 4.3.2 We shall depict a morphism from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$ in \mathbf{GC} by:

$$\begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha} & B \\
 f \downarrow & & \Downarrow \\
 \mathcal{E}' & \xleftarrow{\alpha'} & B'
 \end{array}$$

In the case where \mathbf{C} is \mathbf{Set} and α and α' are set-theoretic relations, the pair (f, F) is a morphism from α to α' if and only if, whenever $e \alpha F(b')$ then $f(e) \alpha' b'$. This is the intended interpretation of \Downarrow . Composition respects \Downarrow .

In [deP89a], de Paiva considers the system LL_* of intuitionistic linear logic. The rules of LL_* are given in Section 3.2.2 of Chapter 3. They include all the usual rules for the connectives \otimes , \sharp , \wedge , \oplus , and \multimap , together with the structural rules (Cut) and (Exch) and the axiom (Id). The rules for negation are

$$\frac{\Gamma \vdash A, C}{\Gamma, A^\perp \vdash C}(\text{varL}) \quad \text{and} \quad \frac{}{A \vdash A^{\perp\perp}}(\text{negR}).$$

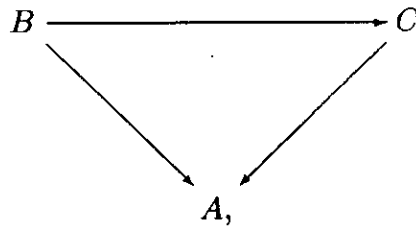
In this part, unless otherwise stated, the phrase “is a sound model of linear logic” is to be understood to mean “is a sound model of the fragment LL_* of linear logic”.

If we choose a base category \mathbf{C} with certain additional structure, to be described below, then we find that \mathbf{GC} has induced additional structure.

Notation 4.3.3 We shall write A to stand for the object $(\mathcal{E} \xleftarrow{\alpha} B)$ and A' to stand for the object $(\mathcal{E}' \xleftarrow{\alpha'} B')$.

Definition 4.3.4 Let A be an object of a category \mathbf{C} . The slice category of \mathbf{C} over A , written \mathbf{C}/A , has as objects the morphisms of \mathbf{C} with codomain A and as

morphisms commuting triangles in \mathbf{C} of the following form:



together with the evident composition.

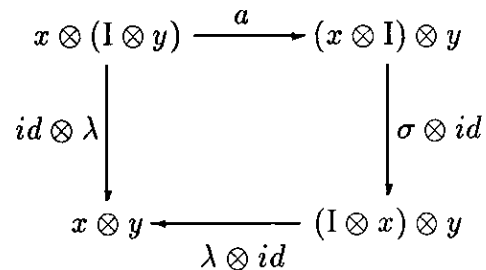
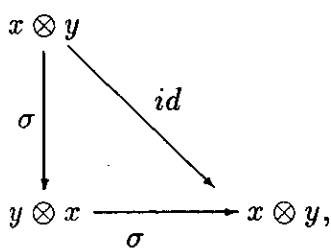
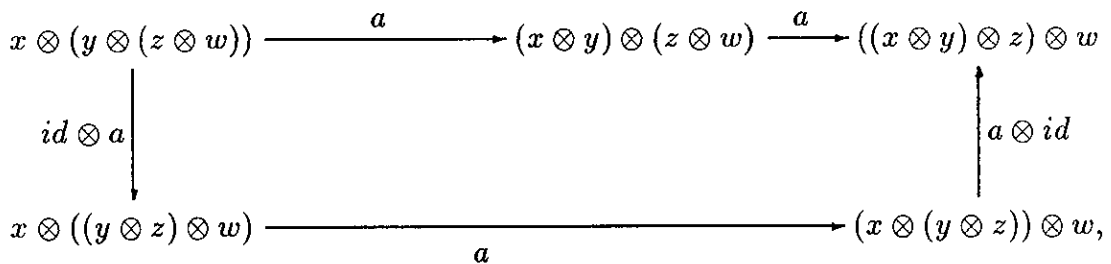
Definition 4.3.5 A category \mathbf{C} is locally cartesian closed if \mathbf{C} has a terminal object, and for every object A of \mathbf{C} , the slice category \mathbf{C}/A is cartesian closed.

Remark 4.3.6 From this definition we see that if \mathbf{C} is locally cartesian closed then \mathbf{C} is also cartesian closed.

Definition 4.3.7 A symmetric monoidal category is a tuple $\langle \mathbf{C}, \otimes, I, a, \lambda, \sigma \rangle$ where \mathbf{C} is a category, \otimes is a functor from $\mathbf{C} \times \mathbf{C}$ to \mathbf{C} , I is an object of \mathbf{C} , and a, λ and σ are natural isomorphisms such that:

$$\begin{aligned}
 a &= \alpha_{x,y,z}: x \otimes (y \otimes z) \cong (x \otimes y) \otimes z, \\
 \lambda &= \lambda_x: I \otimes x \cong x, \\
 \sigma &= \sigma_{x,y}: x \otimes y \cong y \otimes x,
 \end{aligned}$$

and the following diagrams commute:



and

$$\begin{array}{ccccc}
 x \otimes (y \otimes z) & \xrightarrow{a} & (x \otimes y) \otimes z & \xrightarrow{\sigma} & z \otimes (x \otimes y) \\
 \downarrow id \otimes \sigma & & & & \downarrow a \\
 x \otimes (z \otimes y) & \xrightarrow{a} & (x \otimes z) \otimes y & \xrightarrow{\sigma \otimes id} & (z \otimes x) \otimes y.
 \end{array}$$

Definition 4.3.8 A symmetric monoidal closed category is a symmetric monoidal category $\langle \mathbf{C}, \otimes, I, a, \lambda, \sigma \rangle$ in which each functor $(-)\otimes A: \mathbf{C} \rightarrow \mathbf{C}$ has a specified right adjoint $[A, -]$.

Notation 4.3.9 If $\langle \mathbf{C}, \otimes, I, a, \lambda, \sigma \rangle$ is a symmetric monoidal (closed) category, we say that \mathbf{C} has a symmetric monoidal (closed) structure \otimes .

Notation 4.3.10 Let $\langle \mathbf{C}, \otimes, I, a, \lambda, \sigma \rangle$ is a symmetric monoidal category. We call the functor \otimes a tensor product.

We now give some of the results of [deP89b], omitting some details.

Lemma 4.3.11 If \mathbf{C} is locally cartesian closed then \mathbf{GC} has a symmetric monoidal structure given by

$$A \otimes A' = (\mathcal{E}'' \xleftarrow{\alpha \otimes \alpha'} \mathcal{B}''),$$

where $\mathcal{E}'' = \mathcal{E} \times \mathcal{E}'$ and $\mathcal{B}'' = \mathcal{B}^{\mathcal{E}'} \times \mathcal{B}'^{\mathcal{E}}$, and $\alpha \otimes \alpha'$ is defined by taking pullbacks.

We shall describe $\alpha \otimes \alpha'$ explicitly in Section 5.5.

The unit of \otimes is $I = (1 \xleftarrow{id} 1)$.

The proof appears in [deP89b]. When $\mathcal{E}, \mathcal{E}', \mathcal{B}$ and \mathcal{B}' are sets, the relation $\alpha \otimes \alpha'$ can be expressed by

$$(e, e') \alpha \otimes \alpha' (b, b') \quad \text{if and only if} \quad e \alpha f(e') \text{ and } b \alpha' g(e).$$

Lemma 4.3.12

There exists an internal hom $[-, -]_{\mathbf{GC}}: \mathbf{GC}^{op} \times \mathbf{GC} \rightarrow \mathbf{GC}$ given by

$$[A, A']_{\mathbf{GC}} = (\mathcal{E}'' \xleftarrow{(\alpha')^\alpha} \mathcal{B}''),$$

where $\mathcal{E}'' = \mathcal{E}^{\mathcal{E}'} \times \mathcal{B}^{\mathcal{B}'}$, $\mathcal{B}'' = \mathcal{E} \times \mathcal{B}'$. Again, $(\alpha')^\alpha$ is defined by taking pullbacks.

When $\mathcal{E}, \mathcal{E}', \mathcal{B}$ and \mathcal{B}' are sets, the relation $(\alpha')^\alpha$ can be expressed by

$$(f, F) (\alpha')^\alpha (e, b') \quad \text{if and only if} \quad (e \alpha F(b') \Rightarrow f(e) \alpha' b').$$

The adjunction $-\otimes A \dashv [A, -]_{\mathbf{GC}}$ gives \mathbf{GC} a symmetric monoidal closed structure.

Definition 4.3.13 A category \mathbf{C} has finite (small) disjoint coproducts if for any finite (small) family $(A_\lambda)_{\lambda \in \Lambda}$ of objects in \mathbf{C} , the coproduct $A = \coprod_{\lambda \in \Lambda} A_\lambda$ exists, each of the canonical injections $in_\lambda: A_\lambda \rightarrow A$ is a monomorphism, and for each pair of distinct indices λ, λ' the pullback of in_λ along $in_{\lambda'}$ is the initial object.

Definition 4.3.14 We say that A as above is stable under pullbacks if given any map $f: B \rightarrow A$, if we take the pullbacks of each of the canonical injections in_λ along $f: B \rightarrow A$ and call them $f^* A_\lambda$, then the induced map from $\coprod_{\lambda \in \Lambda} f^* A_\lambda$ to B is an isomorphism.

Definition 4.3.15 \mathbf{C} has disjoint, stable coproducts if \mathbf{C} has disjoint coproducts and every coproduct in \mathbf{C} is stable.

Lemma 4.3.16 Let \mathbf{C} be a finitely complete category with disjoint, stable, finite coproducts. There exists a second symmetric monoidal structure on \mathbf{GC} , denoted \sharp , given by

$$A \sharp A' = (\mathcal{E}'' \xleftarrow{\alpha \sharp \alpha'} B''),$$

where $\mathcal{E}'' = \mathcal{E}^{\mathcal{B}'} \times \mathcal{E}'^{\mathcal{B}}$ and $\mathcal{B}'' = \mathcal{B} \times \mathcal{B}'$. The unit of \sharp is given by $\perp = (1 \xleftarrow{\phi} 1)$.

When $\mathcal{E}, \mathcal{E}', \mathcal{B}$ and \mathcal{B}' are sets, the relation $\alpha \sharp \alpha'$ can be expressed by

$$(f, g) \alpha \sharp \alpha' (b, b') \quad \text{if and only if} \quad f(b') \alpha b \quad \text{or} \quad g(b) \alpha' b'.$$

Lemma 4.3.17 Let \mathbf{C} be a finitely complete category with disjoint, stable finite coproducts. Then \mathbf{GC} has finite products and finite coproducts.

Remark 4.3.18 If \mathbf{C} has small products and disjoint, stable, small coproducts, then \mathbf{GC} has small products and small coproducts.

Binary products in \mathbf{GC} are given by

$$A \wedge A' = (\mathcal{E}'' \xleftarrow{\alpha \wedge \alpha'} B''),$$

where $\mathcal{E}'' = \mathcal{E} \times \mathcal{E}'$ and $B'' = B + B'$.

When $\mathcal{E}, \mathcal{E}', B$ and B' are sets, the relation $\alpha \wedge \alpha'$ can be expressed by

$$(e, e') \alpha \wedge \alpha' \begin{cases} (b, 0) & \text{if } e \alpha b \\ (b', 1) & \text{if } e' \alpha' b'. \end{cases}$$

The terminal object of \mathbf{GC} is $\mathbf{1} = (1 \xleftarrow{\phi} 0)$, where 1 and 0 are the terminal and initial objects of \mathbf{C} respectively.

Binary coproducts in \mathbf{GC} are given

$$A \oplus A' = (\mathcal{E}'' \xleftarrow{\alpha \oplus \alpha'} B''),$$

where $\mathcal{E}'' = \mathcal{E} + \mathcal{E}'$ and $B'' = B \times B'$.

When $\mathcal{E}, \mathcal{E}', B$ and B' are sets, the relation $\alpha \oplus \alpha'$ can be expressed by

$$(e, 0) \alpha \oplus \alpha' (b, b') \text{ if and only if } e \alpha b, \text{ and}$$

$$(e', 1) \alpha \oplus \alpha' (b, b') \text{ if and only if } e' \alpha' b'.$$

The initial object of \mathbf{GC} is $\mathbf{0} = (0 \xleftarrow{\phi} 1)$.

Notation 4.3.19

We write $(-)^{\perp}$ for the functor $[-, \perp]_{\mathbf{GC}}$ from \mathbf{GC}^{op} to \mathbf{GC} .

To an object $A = (\mathcal{E} \xleftarrow{\alpha} B)$ of \mathbf{GC} , the functor $(-)^{\perp}$ assigns the object $(B \xleftarrow{\perp^{\alpha}} \mathcal{E})$. When \mathcal{E} and B are sets, for all $e \in \mathcal{E}$ and $b \in B$,

$$b \perp^{\alpha} e \text{ if and only if } (e \alpha b \Rightarrow \perp).$$

Since \perp is the empty relation, if our objects are decidable relations in \mathbf{Set} , then

$$b \perp^{\alpha} e \text{ if and only if it is not the case that } e \alpha b.$$

We therefore call the functor $(-)^{\perp}$ *linear negation*.

Formulae of linear logic can now be interpreted as objects of the category \mathbf{GC} . We assume an interpretation of the atoms of the logic as objects of \mathbf{GC} . We then interpret the logical connectives as follows:

the connective \otimes is interpreted by the monoidal structure \otimes ,

linear implication \multimap by internal hom $[-, -]_{\mathbf{GC}}$,

\wedge by binary product, \wedge ,

\oplus by binary coproduct, $+$,

linear negation by the functor $(-)^{\perp}$,

$\mathbf{1}$ and \perp , the units of \otimes and $\#$ by the units $\mathbf{1}$ and \perp of the monoidal structures \otimes and $\#$,

and

$\mathbf{1}$ and $\mathbf{0}$, the units of \wedge and \oplus by the terminal object $\mathbf{1}$ and initial object $\mathbf{0}$ respectively.

Notation 4.3.20 We write $|A|$ for the interpretation in \mathbf{GC} of the formula A of linear logic. It is evident that $| - |$ is a function from formulae of linear logic to objects of \mathbf{GC} .

Theorem 4.3.21 Let \mathbf{C} be a locally cartesian closed category with disjoint, stable, finite coproducts. Then the symmetric monoidal closed category \mathbf{GC} is a model of LL_* : that is, whenever there is an entailment $\Gamma \vdash_{LL_*} A$, there exists a morphism (f, F) in \mathbf{GC} from the interpretation of Γ to the interpretation of A .

Remark 4.3.22 Let \mathbf{T} be a topos. Then \mathbf{T} is locally cartesian closed and has disjoint, stable coproducts. Hence \mathbf{GT} is a sound model of intuitionistic linear logic.

De Paiva shows how the modal operators of linear logic, “!” and “?” , can be modelled in **GC** by a comonad and a monad respectively. We shall not go into the details of her constructions here, as we shall not be discussing the action of the modalities on nets. It is certainly possible to interpret the action of “!” and “?” on Petri nets, and if that interpretation proves useful, it will be considered in future work.

4.4 A Category of Elementary Petri Nets

We shall now show how a simple extension of de Paiva’s work gives rise to a category which is also a model of linear logic, but whose objects are Petri nets. In general in this part we concentrate on the structural properties of a net rather than its dynamic behaviour, and so we discuss properties of unmarked nets.

Definition 4.4.1 *We say a Petri net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ is elementary if for all events $e \in \mathcal{E}$, $pre(e)$ and $post(e)$ are both multisets in which no element occurs with multiplicity greater than one.*

Remark 4.4.2 *Elementary nets are studied in [NRT90], [Roz87] and [Thi87], where they are called elementary net systems.*

We shall first consider elementary nets, and illustrate the application to Petri nets of constructions arising from **GC**. Analogous constructions exist for general nets, as we mention in *our Conclusion* and in [BG90]. A full treatment of general nets requires an extension of de Paiva’s work, and is to be the subject of a joint paper with de Paiva.

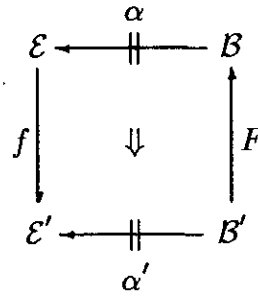
Let **C** be a category with finite limits.

We define a category **NC** by analogy with **GC** as follows:

- objects of **NC** are pairs of subobjects $A_0 \xrightarrow{\alpha^0} \mathcal{E} \times \mathcal{B}$, and $A_1 \xrightarrow{\alpha^1} \mathcal{E} \times \mathcal{B}$, which we shall write as $(\mathcal{E} \xleftarrow{\alpha} \text{---} \mathcal{B})$,

- a morphism from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$ is a pair (f, F) of morphisms in \mathbf{C} such that for $i = 0$ and $i = 1$, (f, F) is a morphism in \mathbf{GC} from $(\mathcal{E} \xleftarrow{\alpha^i} B)$ to $(\mathcal{E}' \xleftarrow{(\alpha')^i} B')$,
- and composition is componentwise.

Notation 4.4.3 A morphism from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$ in \mathbf{NC} will be depicted thus:



Notation 4.4.4 Unless otherwise stated, any diagram \mathcal{D} in \mathbf{NC} with arrows labelled α, α' etc. stands for two diagrams in \mathbf{GC} , \mathcal{D}_i , in which for $i = 0, 1$ the labels α and α' are replaced by α^i and $(\alpha')^i$ respectively.

We are primarily interested in the particular case where \mathbf{C} is \mathbf{Set} , the category of sets and functions. Since \mathbf{Set} is a topos, it has all the properties required for its dialectica category \mathbf{GSet} to be a sound model of intuitionistic linear logic.

Remark 4.4.5 For any monic α in \mathbf{Set} into an object $\mathcal{E} \times B$, we have an evident canonical choice for the subobject A of $\mathcal{E} \times B$. Consequently, pullbacks are defined up to equality, rather than up to isomorphism, and so the conditions (4.1) hold up to equality, rather than isomorphism.

Observe that objects of \mathbf{NSet} are precisely elementary nets. We shall identify the net $\langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ with the object $(\mathcal{E} \xleftarrow{\alpha} B)$ of \mathbf{NSet} by putting

$$\alpha^0 = pre \quad \text{and} \quad \alpha^1 = post.$$

Suppose that (f, F) is a morphism in \mathbf{NSet} from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$.

The pullback condition for morphisms in \mathbf{NSet} can be expressed as

$$\text{for all } e \in \mathcal{E} \text{ and } b' \in B', \quad e \alpha F(b') \Rightarrow f(e) \alpha' b'.$$

Expressing this as a condition on each relation, we have

$$F^{-1}(\text{pre}(e)) \subseteq \text{pre}'(f(e)) \quad \text{and} \quad F^{-1}(\text{post}(e)) \subseteq \text{post}'(f(e)). \quad (4.1)$$

Conversely, if $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B} \rightarrow \mathcal{B}'$ are morphisms in **Set**, (f, F) is a morphism in **NSet** whenever the conditions (4.1) are satisfied.

The map (f, F) shows how N' is a refinement of N , in a way we now make precise.

Definition 4.4.6 Let $N = \langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post} \rangle$ be a Petri net, and let $e, e' \in \mathcal{E}$. Then e is a refinement of e' if

$$\text{pre}(e) \subseteq_m \text{pre}(e') \quad \text{and} \quad \text{post}(e) \subseteq_m \text{post}(e').$$

Thus the event e' consumes at least as many resources as e , and produces at least as many: we can think of this as meaning that the pre- and post-condition sets of e' are more specified than those of e .

Definition 4.4.7 Let $N = \langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post} \rangle$ and $N' = \langle \mathcal{E}', \mathcal{B}', \text{pre}', \text{post}' \rangle$ be Petri nets, and let F be a function from \mathcal{B}' to \mathcal{B} . Then e' refines e relative to F if for each $e \in \mathcal{E}$,

$$F^{-1}(\text{pre}(e)) \subseteq_m \text{pre}'(e') \quad \text{and} \quad F^{-1}(\text{post}(e)) \subseteq_m \text{post}'(e').$$

Definition 4.4.8 A refinement map from a Petri net $N = \langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post} \rangle$ to a Petri net $N' = \langle \mathcal{E}', \mathcal{B}', \text{pre}, \text{post} \rangle$ is a pair of functions (f, F) with $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B}' \rightarrow \mathcal{B}$ such that for every $e \in \mathcal{E}$, $f(e)$ refines e relative to F .

N' is a refinement of N if there exists a refinement map $(f, F): N \rightarrow N'$.

Proposition 4.4.9

Refinement maps between elementary nets are precisely the morphisms of **NSet**.

Proof: Immediate. □

Proposition 4.4.10 Let (f, F) be a morphism in \mathbf{NSet} from the safe net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ to the safe net $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$. Whenever a marking M_0 can evolve in N by a multiset of events A to a marking M_1 , there exist markings S_0 and S_1 of N' with

$$F^{-1}(M_0) \subseteq S_0 \quad \text{and} \quad F^{-1}(M_1) \subseteq S_1$$

such that S_0 can evolve by the multiset of events $f(A)$ to S_1 .

Proof: Put $S_0 = F^{-1}(M_0) + pre'(fA) - F^{-1}(pre(A))$ and

$$S_1 = F^{-1}(M_1) + post'(fA) - F^{-1}(post(A)).$$

Since $M_1 = M_0 - pre(A) + post(A)$, we have

$$F^{-1}(M_1) = F^{-1}(M_0) - F^{-1}(pre(A)) + F^{-1}(post(A)).$$

By linearity, the conditions (4.1) give

$$F^{-1}(pre(A)) \subseteq pre'(fA) \quad \text{and} \quad F^{-1}(post(A)) \subseteq post'(fA).$$

Now by hypothesis $pre(A) \subseteq M_0$, and so $F^{-1}(pre(A)) \subseteq F^{-1}(M_0)$.

Hence $pre'(fA) \subseteq F^{-1}(M_0) + pre'(fA) - F^{-1}(pre(A)) = S_0$, and the multiset of events fA is enabled in N' with the marking S_0 .

After a firing of fA , N' has marking

$$\begin{aligned} S_1 &= F^{-1}(M_0) + post'(fA) - F^{-1}(pre(A)) \\ &\supseteq F^{-1}(M_0) + F^{-1}(post(A)) - F^{-1}(pre(A)) \\ &= F^{-1}(M_1). \end{aligned}$$

This completes the proof. \square

Remark 4.4.11 *This proposition is exactly analogous to the result which Winskel presents in Theorem 4.2.1. It illustrates what we mean by refinement, since it shows that given any evolution of a net N , a refinement of N is able to simulate that evolution.*

Remark 4.4.12 *It should be possible to extend this proposition to the behaviour of all elementary nets, rather than just safe nets. The delicacy lies in the fact that if M_0 is a multiset containing some element of \mathcal{B} in multiplicity greater than one,*

$F^{-1}(M_0)$ may not be well-defined. Where the linear extension of F to multisets over \mathcal{B}' is an invertible multirelation, the extension of the proposition is straightforward.

The following proposition characterises isomorphism in \mathbf{NSet} , and is of considerable importance in later chapters.

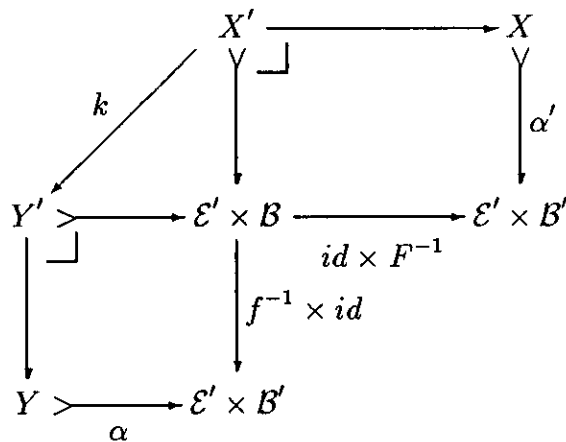
Proposition 4.4.13 A morphism (f, F) in \mathbf{NSet} from $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ to $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$ is an isomorphism if and only if

- $f: \mathcal{E} \rightarrow \mathcal{E}'$ is a bijection in \mathbf{Set} ,
- $F: \mathcal{B}' \rightarrow \mathcal{B}$ is a bijection in \mathbf{Set} , and
- for each $e \in \mathcal{E}$,

$$F(pre'(fe)) = pre(e) \quad \text{and} \quad F(post'(fe)) = post(e).$$

Proof: The inverse of a morphism (f, F) in \mathbf{NC} , where it exists, is (f^{-1}, F^{-1}) , where f^{-1} is the inverse in \mathbf{C} of f , and F^{-1} the inverse in \mathbf{C} of F . Thus (f, F) is an isomorphism in \mathbf{NC} if and only if f and F are isomorphisms in \mathbf{C} , and (f^{-1}, F^{-1}) is a morphism in \mathbf{NC} .

Thus a morphism (f, F) in \mathbf{NSet} is an isomorphism if and only if f and F are isomorphisms in \mathbf{Set} and the triangle in the diagram



commutes.

Since isomorphisms in **Set** are preserved under pullback, recalling Remark 4.4.5 we have

$$X' = X \quad \text{and} \quad Y' = Y.$$

Since (f, F) is a morphism in **NSet**, the diagram

$$\begin{array}{ccccc}
 & & X'' & \xrightarrow{\quad} & X \\
 & & \downarrow & \lrcorner & \downarrow \alpha \\
 & k' \nearrow & & & \\
 Y'' & \xrightarrow{\quad} & \mathcal{E} \times B' & \xrightarrow{id \times F} & \mathcal{E} \times B \\
 \downarrow & \lrcorner & \downarrow f \times id & & \\
 Y & \xrightarrow{\quad} & \mathcal{E}' \times B' & & \\
 & \alpha' \searrow & & &
 \end{array}$$

commutes in **Set**, and so we have

$$X'' = X \quad \text{and} \quad Y'' = Y.$$

k is a unique map from X' to Y' , and k' is a unique map from X'' to Y'' . Since $X' = X''$ and $Y' = Y''$, we have $X'' = Y''$.

Now

$$X'' = \{\langle e, b' \rangle \mid F(b') \alpha e\} \quad \text{and} \quad Y'' = \{\langle e, b' \rangle \mid b' \alpha' fe\}.$$

Since $X'' = Y''$, $F(b') \in pre(e)$ if and only if $b' \in pre'(fe)$ and similarly for the post-condition relations, that is,

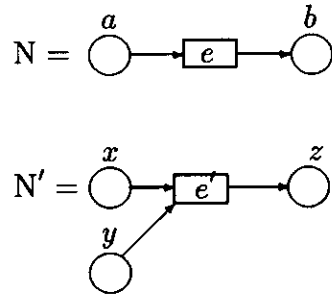
$$F(pre(fe)) = pre(e) \quad \text{and} \quad F(post(fe)) = post(e).$$

□

4.4.1 Some Examples of Refinement Maps

We now give some examples of nets N and N' with refinement maps (f, F) from N to N' .

Example 4.4.14



Put $f(e) = e'$, $F(x) = F(y) = a$ and $F(z) = b$.

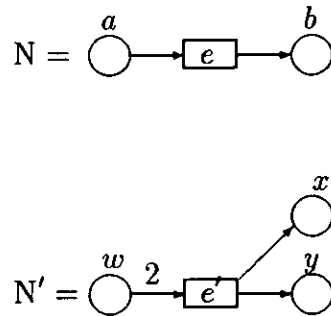
Observe that the conditions (4.1) are satisfied, since

$$F^{-1}(\text{pre}(e)) = F^{-1}(\{a\}) = \{x, y\} = \text{pre}(fe) \quad \text{and}$$

$$F^{-1}(\text{post}(e)) = F^{-1}(\{b\}) = \{z\} = \text{post}(fe).$$

Thus (f, F) is a refinement map from N to N' .

Example 4.4.15



Put $f(e) = e'$, $F(x) = F(y) = b$ and $F(w) = a$.

Observe that the conditions (4.1) are satisfied, since

$$F^{-1}(\text{pre}(e)) = F^{-1}(\{a\}) = \{w\} \subseteq_m 2w = \text{pre}(fe) \quad \text{and}$$

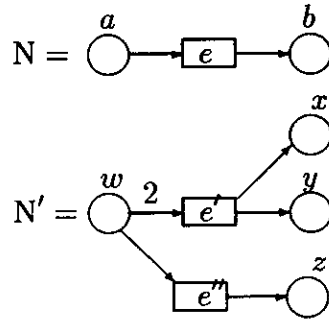
$$F^{-1}(\text{post}(e)) = F^{-1}(\{b\}) = \{x, y\} = \text{post}(fe).$$

Hence (f, F) is a refinement map from N to N' .

Notice that we can refine an elementary net into a net which is not elementary.

To illustrate this notion of refinement further, we now give an example of two nets between which there cannot exist a refinement in the sense of Definition 4.4.8.

Example 4.4.16



We now consider possible functions $f: \{e\} \rightarrow \{e', e''\}$ and $F: \{w, x, y, z\} \rightarrow \{a, b\}$. We shall consider the case where $f(e) = e'$, the argument being similar when $f(e) = e''$.

Suppose F is given by

$$F(x) = F(y) = b \text{ and } F(z) = F(w) = a.$$

Then we have

$$F^{-1}(pre(e)) = F^{-1}(\{a\}) = \{z, w\} \not\subseteq_m pre(fe).$$

The problem is caused by the extraneous z . Our only alternative is to map z to b by F , putting

$$F(x) = F(y) = F(z) = b \text{ and } F(w) = a.$$

Then we have

$$F^{-1}(pre(e)) = F^{-1}(\{a\}) = \{w\} \subseteq_m pre(fe) = 2w.$$

The condition on the *post*-condition relation is not satisfied, however as

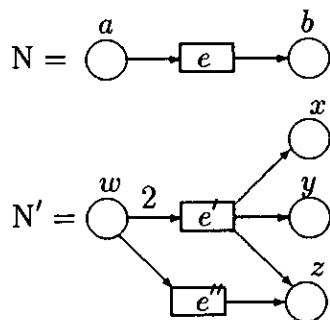
$$F^{-1}(post(e)) = F^{-1}(\{b\}) = \{x, y, z\} \not\subseteq_m post(fe).$$

Thus it is plain that we cannot have a refinement map from N to N' , as there is no condition in N to which we can map z without violating the conditions (4.1).

This example illustrates a strength of our approach. The morphisms in \mathbf{NSet} are contravariant in the second argument, and so any refinement map from net N to N' must map every condition in N' to some condition in N . While a refinement map allows the image net to contain a greater number of conditions than the net N being refined, the additional conditions must be intimately connected with their images in N . In Example 4.4.16, it is exactly because z is the post-condition of an event not in the image of f that there is no refinement map from N to N' . This conforms to the slogan of algebraic specification, “no junk!” [BG81].

It is not the case, however, that every event in the refinement N' must correspond to an event in N . We illustrate this point by the example below.

Example 4.4.17



Put $f(e) = e'$, $F(x) = F(y) = F(z)$ and $F(w) = a$.

Observe that the conditions (4.1) are satisfied, since

$$F^{-1}(\text{pre}(e)) = F^{-1}(\{a\}) = \{w\} \subseteq_m 2w = \text{pre}(fe) \quad \text{and}$$

$$F^{-1}(\text{post}(e)) = F^{-1}(\{b\}) = \{x, y, z\} = \text{post}(fe).$$

Thus there is a refinement map from N to N' , even though f is not surjective.

Thus our definition allows the refinement of a net N to have more events than N itself. This is also true of the maps in Winskel's category \mathbf{Net} , since the maps on events need not be surjective functions. Consider a morphism $(\eta, \beta): N \rightarrow N'$ in \mathbf{Net} . Since the map β on conditions is an arbitrary multirelation, there may be conditions in N' which have no relationship to any condition in N . By contrast, in the case of a morphism $(f, F): N \rightarrow N'$ in \mathbf{NSet} , the map on conditions requires

every condition in N' to relate to at least one in N : pre- and post-conditions of any events not in the image of f are heavily restricted in that they must be part of the refinement relative to F of at least one event of the original net.

Chapter 5

Structure in NSet

In this chapter we give an elegant characterisation of \mathbf{NC} as a limit in \mathbf{Cat} , where \mathbf{Cat} is the category whose objects are small categories and whose morphisms are functors. This allows us to prove that all the relevant structure of \mathbf{GC} lifts to \mathbf{NC} , proving it to be a sound model of linear logic. We then interpret the connectives of linear logic as constructions on Petri nets. The expressiveness of linear logic stems from the variety of its connectives: we exploit that expressiveness to compose nets in various ways. We now describe the interpretation as net constructors of the product, coproduct, tensor product, internal hom and negation of linear logic. We also show how restriction is expressed in the category \mathbf{NC} . This, combined with the constructions of product and tensor on nets, allows us to construct in \mathbf{NC} the net corresponding to the parallel composition of two nets with respect to any chosen synchronisation algebra. We also discuss the interpretation of the constants \perp , \top , $\mathbf{0}$ and $\mathbf{1}$ as nets. Essentially, we interpret product and tensor product as parallel composition of nets, coproduct as a choice between nets, and internal hom as refinement.

We can also interpret $\#$, the comonad $!$ and the monad $?$ as net constructors. These connectives are not discussed here.

Throughout this chapter we shall assume that \mathbf{C} has sufficient structure for \mathbf{GC} to be a sound model of linear logic. Further, we shall assume that all relevant

structure of \mathbf{C} is assigned. This ensures that the relevant structure of \mathbf{GC} and of \mathbf{NC} is also assigned.

5.1 Characterising \mathbf{NC} as a limit in \mathbf{Cat}

We now show that whenever \mathbf{C} is a locally cartesian closed category with disjoint, stable, finite coproducts, \mathbf{NC} inherits from \mathbf{GC} sufficient structure to be itself a model of linear logic. To do this we appeal to the following theorem:

Theorem 5.1.1

\mathbf{NC} is the pullback in \mathbf{Cat} of the forgetful functor $\mathcal{U}: \mathbf{GC} \rightarrow \mathbf{C} \times \mathbf{C}^{op}$ along itself.

Proof: Let the following diagram:

$$\begin{array}{ccc}
 \mathbf{X} & \longrightarrow & \mathbf{GC} \\
 \downarrow & \lrcorner & \downarrow \mathcal{U} \\
 \mathbf{GC} & \xrightarrow{\quad \mathcal{U} \quad} & \mathbf{C} \times \mathbf{C}^{op}
 \end{array}$$

be a pullback in \mathbf{Cat} .

Then an object of \mathbf{X} is a map from the terminal object 1 of \mathbf{Cat} into \mathbf{X} , equivalently, a pair of maps $(\tilde{A}_0, \tilde{A}_1)$ from 1 into \mathbf{GC} such that $\mathcal{U}\tilde{A}_0 = \mathcal{U}\tilde{A}_1$, equivalently a pair of objects (A_0, A_1) of \mathbf{GC} such that $\mathcal{U}A_0 = \mathcal{U}A_1$.

Similarly, an arrow in \mathbf{X} from $\langle A_0, A_1 \rangle$ to $\langle A'_0, A'_1 \rangle$ is a pair $\langle (f, F), (g, G) \rangle$ of arrows in \mathbf{GC} with $(f, F): A_0 \rightarrow A'_0$ and $(g, G): A_1 \rightarrow A'_1$ such that $\mathcal{U}(f, F) = \mathcal{U}(g, G)$.

Thus objects of \mathbf{X} are pairs of the form $\langle (\mathcal{E}_0 \xleftarrow{\alpha_0} B_0), (\mathcal{E}_1 \xleftarrow{\alpha_1} B_1) \rangle$ such that

$$\mathcal{U}(\mathcal{E}_0 \xleftarrow{\alpha_0} B_0) = \mathcal{U}(\mathcal{E}_1 \xleftarrow{\alpha_1} B_1),$$

that is, such that $\mathcal{E}_0 = \mathcal{E}_1$ and $\mathcal{B}_0 = \mathcal{B}_1$. Thus an object of \mathbf{X} is a 4-tuple $\langle \mathcal{E}, \mathcal{B}, \alpha^0, \alpha^1 \rangle$ in which \mathcal{E} and \mathcal{B} are objects of \mathbf{C} and α^0 and α^1 are subobjects of $\mathcal{E} \times \mathcal{B}$.

Similarly, morphisms in \mathbf{X} are pairs $\langle (f, F), (g, G) \rangle$ of morphisms in \mathbf{GC} for which for which $f = g$ and $F = G$.

Thus the object set of \mathbf{X} is the set of pairs of subobjects $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$, and a morphism from $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')$ is a pair (f, F) of morphisms in \mathbf{C} for which the following diagrams

$$\begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^0} & \mathcal{B} \\
 f \downarrow & & \downarrow \\
 \mathcal{E}' & \xleftarrow{(\alpha')^0} & \mathcal{B}' \\
 & & \uparrow F
 \end{array}
 \quad \text{and} \quad
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^1} & \mathcal{B} \\
 f \downarrow & & \downarrow \\
 \mathcal{E}' & \xleftarrow{(\alpha')^1} & \mathcal{B}' \\
 & & \uparrow F
 \end{array}$$

commute. Thus $\mathbf{X} = \mathbf{NC}$. □

Remark 5.1.2 \mathbf{NC} is the kernel pair of \mathcal{U} .

Corollary 5.1.3 \mathbf{NC} is a subcategory of $\mathbf{GC} \times \mathbf{GC}$.

Lemma 5.1.4 \mathcal{U} strictly preserves the product and coproduct of \mathbf{GC} .

Proof: We suppose that \mathbf{C} has assigned products, coproducts and internal homs. Then $\mathbf{C} \times \mathbf{C}^{op}$ has assigned products *defined* by

$$\langle \mathcal{E}, \mathcal{B} \rangle \tilde{\times} \langle \mathcal{E}', \mathcal{B}' \rangle = \langle \mathcal{E} \times \mathcal{E}', \mathcal{B} + \mathcal{B}' \rangle.$$

Now

$$\begin{aligned}
 \mathcal{U}((\mathcal{E} \xleftarrow{\alpha} \mathcal{B}) \times (\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')) &= \mathcal{U}(\mathcal{E} \times \mathcal{E}' \xleftarrow{\alpha \wedge \alpha'} \mathcal{B} + \mathcal{B}') \\
 &= \langle \mathcal{E} \times \mathcal{E}', \mathcal{B} + \mathcal{B}' \rangle \\
 &= \langle \mathcal{E}, \mathcal{B} \rangle \tilde{\times} \langle \mathcal{E}', \mathcal{B}' \rangle \\
 &= (\mathcal{U}(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})) \tilde{\times} (\mathcal{U}(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')).
 \end{aligned}$$

The empty case is trivial.

Thus \mathcal{U} strictly preserves the product of \mathbf{GC} .

The proof that \mathcal{U} strictly preserves the coproduct is similar. □

Lemma 5.1.5 \mathcal{U} strictly preserves the monoidal structures \sharp and \otimes of \mathbf{GC} , and the closed structure $[-, -]$ corresponding to \otimes .

Proof:

Consider the binary operation \circ on $\mathbf{C} \times \mathbf{C}^{op}$ defined by

$$(\mathcal{E}, \mathcal{B}) \circ (\mathcal{E}', \mathcal{B}') = (\mathcal{E} \times \mathcal{E}', \mathcal{B}^{\mathcal{E}'} \times \mathcal{B}'^{\mathcal{E}}).$$

It is readily seen that \circ induces a symmetric monoidal structure on $\mathbf{C} \times \mathbf{C}^{op}$. Then for objects A and A' of \mathbf{GC} ,

$$\mathcal{U}(A) \circ \mathcal{U}(A') = \mathcal{U}(A \otimes A').$$

Further, if a , λ and σ are the natural isomorphisms expressing respectively the associativity, left identity and symmetry of \otimes , then $\mathcal{U}a$, $\mathcal{U}\lambda$ and $\mathcal{U}\sigma$ are natural isomorphisms expressing the associativity, left identity and symmetry of \circ , and satisfy the appropriate coherence conditions. Thus \mathcal{U} strictly preserves the symmetric monoidal structure \otimes .

A similar argument shows that the binary operation \flat on $\mathbf{C} \times \mathbf{C}^{op}$ defined by

$$\langle \mathcal{E}, \mathcal{B} \rangle \flat \langle \mathcal{E}', \mathcal{B}' \rangle = \langle \mathcal{E}^{\mathcal{E}'} \times \mathcal{B}'^{\mathcal{B}}, \mathcal{B} \times \mathcal{B}' \rangle$$

gives $\mathbf{C} \times \mathbf{C}^{op}$ a symmetric monoidal structure. For objects A and A' of \mathbf{GC} ,

$$\mathcal{U}(A) \flat \mathcal{U}(A') = \mathcal{U}(A \sharp A').$$

Again, it is evident that \mathcal{U} strictly preserves the symmetric monoidal structure \sharp . Finally, \circ is a symmetric monoidal closed structure on $\mathbf{C} \times \mathbf{C}^{op}$, with internal hom given by

$$[(\mathcal{E}, \mathcal{B}), (\mathcal{E}', \mathcal{B}')] = (\mathcal{E}'^{\mathcal{E}} \times \mathcal{B}^{\mathcal{B}'}),$$

as shown by the following sequence of natural isomorphisms:

$$\begin{aligned}
& \mathbf{C} \times \mathbf{C}^{op}((\mathcal{E}, \mathcal{B}) \circ (\mathcal{E}', \mathcal{B}'), (\mathcal{E}'', \mathcal{B}'')) \\
&= \mathbf{C} \times \mathbf{C}^{op}((\mathcal{E} \times \mathcal{E}', \mathcal{B}^{\mathcal{E}'} \times \mathcal{B}^{\mathcal{E}}), (\mathcal{E}'', \mathcal{B}'')) \\
&\cong \mathbf{C}(\mathcal{E}, \mathcal{E}''^{\mathcal{E}'}) \times \mathbf{C}(\mathcal{B}'', \mathcal{B}^{\mathcal{E}'}) \times \mathbf{C}(\mathcal{B}'', \mathcal{B}^{\mathcal{E}}) \\
&\cong \mathbf{C}(\mathcal{E}, \mathcal{E}''^{\mathcal{E}'} \times \mathcal{B}'^{\mathcal{B}''}) \times \mathbf{C}(\mathcal{E}' \times \mathcal{B}'', \mathcal{B}) \\
&\cong \mathbf{C} \times \mathbf{C}^{op}((\mathcal{E}, \mathcal{B}), (\mathcal{E}''^{\mathcal{E}'} \times \mathcal{B}'^{\mathcal{B}''}, \mathcal{E}' \times \mathcal{B}'')).
\end{aligned}$$

Thus \mathcal{U} strictly preserves the closed structure of \mathbf{GC} as given in Lemma 4.3.12.

□

Theorem 5.1.1 and Lemmata 5.1.4 and 5.1.5 suffice to prove that the product, coproduct, symmetric monoidal structures and internal hom of \mathbf{GC} lift to \mathbf{NC} , as we now show.

Consider the following categories of (small) categories with structure:

- (symmetric) monoidal categories and strict (symmetric) monoidal functors,
- closed categories and functors strictly preserving the closed structure,
- categories with assigned finite products, and functors strictly preserving these,
- categories with assigned finite coproducts, and functors strictly preserving these.

These are all monadic over \mathbf{Cat} . The evident forgetful functor from any of them into \mathbf{Cat} creates all limits (see [ML71]). Since the product (\times), coproduct ($+$), symmetric monoidal structures (\otimes and \sharp) and internal hom of \mathbf{GC} are strictly preserved by \mathcal{U} , and since \mathbf{NC} is the kernel pair of \mathcal{U} , \mathbf{NC} has the products, coproduct, symmetric monoidal structures and internal hom induced by those in \mathbf{GC} .

Corollary 5.1.6 \mathbf{NC} is a sound model of linear logic, that is, whenever there is an entailment $\Gamma \vdash A$ in LL_* , there exists a morphism (f, F) in \mathbf{NC} from the interpretation of Γ as a net to the interpretation of A as a net.

Proof: Follows from Theorem 5.1.1 and Lemmata 5.1.5 and 5.1.4.

□

Thus we can interpret all the connectives of linear logic as constructors on Petri nets. For example, we can give a meaning as a net to “ $N_0 \otimes N_1$ ” or “ $N_0 \wedge N_1$ ” or “ $N_0 \multimap N_1$ ” for arbitrary nets N_0 and N_1 . Applying these constructors to Petri nets allows us to build up composite nets whose behaviour can be expressed in terms of the behaviour of their components.

We now examine in detail the structure of NSet, the category of elementary Petri nets.

5.2 Restriction

We now illustrate the application of the category NSet to obtain the restriction of a net elegantly, something which is not always possible in an algebraic model. Let $N = (\mathcal{E} \xrightarrow{\alpha_0} \mathcal{B})$ be a Petri net and let $\mathcal{E}' \subseteq \mathcal{E}$. Then $N[\mathcal{E}'$, the restriction of N to the event set \mathcal{E}' is readily constructed in NSet. We take pullbacks in Set as follows:

$$\begin{array}{ccc}
 X' & \xrightarrow{\quad} & X \\
 \downarrow \text{pre}' & \lrcorner & \downarrow \text{pre} \\
 \mathcal{E}' \times \mathcal{B} & \xrightarrow{\quad} & \mathcal{E} \times \mathcal{B}
 \end{array}
 \qquad
 \begin{array}{ccc}
 Y' & \xrightarrow{\quad} & Y \\
 \downarrow \text{post}' & \lrcorner & \downarrow \text{post} \\
 \mathcal{E}' \times \mathcal{B} & \xrightarrow{\quad} & \mathcal{E} \times \mathcal{B}
 \end{array}$$

Now $X' = \{\langle e, b \rangle \in \mathcal{E}' \times \mathcal{B} \mid e \in \mathcal{E}', e \text{ pre } b\}$ and

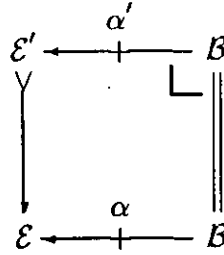
$Y' = \{\langle e, b \rangle \in \mathcal{E}' \times \mathcal{B} \mid e \in \mathcal{E}', e \text{ post } b\}$.

Thus the relations pre' and post' are such that for each $e \in \mathcal{E}'$,

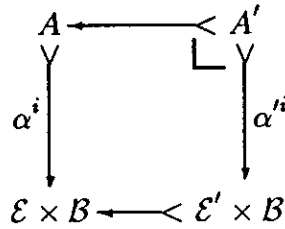
$$\text{pre}'(e) = \{b \mid e \text{ pre } b\} \quad \text{and} \quad \text{post}'(e) = \{b \mid e \text{ post } b\}.$$

Thus $N' = \langle \mathcal{E}', \mathcal{B}, \text{pre}', \text{post}' \rangle$ is precisely $N[\mathcal{E}'$, the restriction of N to event set \mathcal{E}' .

Notation 5.2.1 We shall write



to stand for the pair of diagrams



This simple construction is available precisely because we can work at two levels (that of \mathbf{C} and that of \mathbf{GC}), and the relations *pre* and *post* of a Petri net are morphisms in our base category \mathbf{C} , but are composed by morphisms in our higher level category \mathbf{GC} . By contrast, the *pre* and *post* condition relations in Winskel's category \mathbf{Net} appear only as structure on the objects of the category, and so cannot be manipulated directly.

We shall give examples and applications of restriction in the category \mathbf{NSet} in Sections 5.3.1 and 5.6.2.

5.3 The Product of Two Nets

By Theorem 5.1.1, the product in \mathbf{NSet} of nets $N_0 = (\mathcal{E}_0 \xleftarrow{\alpha_0} B_0)$ and $N_1 = (\mathcal{E}_1 \xleftarrow{\alpha_1} B_1)$ is that induced by the product in \mathbf{GSet} . We shall now describe the product in \mathbf{NSet} explicitly.

The product of two objects $(\mathcal{E}_0 \xleftarrow{\alpha_0} B_0)$ and $(\mathcal{E}_1 \xleftarrow{\alpha_1} B_1)$ in \mathbf{GSet} is $(\mathcal{E}_0 \times \mathcal{E}_1 \xleftarrow{\alpha_0 \wedge \alpha_1} B_0 + B_1)$, where the relation $\alpha_0 \wedge \alpha_1$ is given by

$$(e_0, e_1) \alpha_0 \wedge \alpha_1 \begin{cases} (b_0, 0) & \text{if } e_0 \alpha_0 b_0 \\ (b_1, 1) & \text{if } e_1 \alpha_1 b_1. \end{cases}$$

Hence the product in **NSet** of objects $(\mathcal{E}_0 \xleftarrow{\alpha_0} \mathcal{B}_0)$ and $(\mathcal{E}_1 \xleftarrow{\alpha_1} \mathcal{B}_1)$ is

$$(\mathcal{E}_0 \times \mathcal{E}_1 \xleftarrow{\alpha_0 \wedge \alpha_1} \mathcal{B}_0 + \mathcal{B}_1),$$

where the two relations $(\alpha_0 \wedge \alpha_1)^0$ and $(\alpha_0 \wedge \alpha_1)^1$ are given by

$$\langle e_0, e_1 \rangle (\alpha_0 \wedge \alpha_1)^0 \begin{cases} (b_0, 0) & \text{if } e_0 \alpha_0^0 b_0 \\ (b_1, 1) & \text{if } e_1 \alpha_1^0 b_1 \end{cases} \quad \text{and}$$

$$\langle e_0, e_1 \rangle (\alpha_0 \wedge \alpha_1)^1 \begin{cases} (b_0, 0) & \text{if } e_0 \alpha_0^1 b_0 \\ (b_1, 1) & \text{if } e_1 \alpha_1^1 b_1. \end{cases}$$

For $k = 0, 1$ let π_k be the projection from $\mathcal{E}_0 \times \mathcal{E}_1$ to \mathcal{E}_k , and let in_k be the injection of \mathcal{B}_k into $\mathcal{B}_0 + \mathcal{B}_1$. Then the projection in **NSet** from $N = N_0 \times N_1$ to N_k is the map (π_k, in_k) .

The pre- and post-condition relations in the product net $N_0 \times N_1$ are given as follows:

$$pre(\langle e_0, e_1 \rangle) = \{(b_0, 0) \mid e_0 pre_0 b_0\} \cup \{(b_1, 1) \mid e_1 pre_1 b_1\}, \quad \text{and}$$

$$post(\langle e_0, e_1 \rangle) = \{(b_0, 0) \mid e_0 post_0 b_0\} \cup \{(b_1, 1) \mid e_1 post_1 b_1\}.$$

An event in the product net is the synchronisation of two events, one in each of the component nets: that is, a firing in the product net is the concurrent firing of an event in the first component and an event in the second.

Evidently the product in **NSet** is closely related to the product of two nets N_0 and N_1 in Winskel's category **Net**, as described in Section 5.3. The event set of Winskel's product net $N = N_0 \times N_1$ is the set $((\mathcal{E}_0)_\perp \times (\mathcal{E}_1)_\perp) \setminus \langle *, * \rangle$, formed by lifting the two event sets, taking their cartesian product and removing the pair $\langle *, * \rangle$. The set of conditions is $\mathcal{B}_0 \uplus \mathcal{B}_1$, with the evident relations. In the product net N , events are either of the form $\langle e_0, * \rangle$ or $\langle *, e_1 \rangle$, in which case they correspond to a single event in the appropriate component net, or of the form $\langle e_0, e_1 \rangle$, in which case they correspond to a synchronisation of two events, one in each component

net. In Section 5.6, we show that the net $(N_0 + \perp) \times (N_1 + \perp)$ has events which correspond exactly to the events of Winskel's product net $N_0 \times N_1$, together with the additional isolated event \perp .

We relate the behaviour of the net $N_0 \times N_1$ to that of its component nets as follows:

Proposition 5.3.1 Let $N_0 = \langle \mathcal{E}_0, \mathcal{B}_0, pre_0, post_0 \rangle$ and $N_1 = \langle \mathcal{E}_1, \mathcal{B}_1, pre_1, post_1 \rangle$ be Petri nets, and let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ be the product net $N_0 \times N_1$.

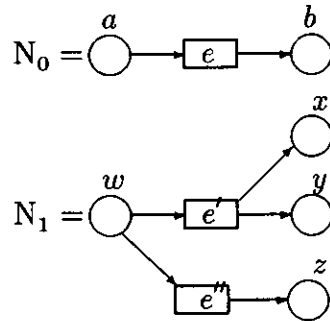
For $k = 0, 1$ let π_k be the projection from $\mathcal{E}_0 \times \mathcal{E}_1$ to \mathcal{E}_k , and let in_k be the injection from \mathcal{B}_k to $\mathcal{B}_0 + \mathcal{B}_1$.

- A multiset M is a marking of N if and only if $in_k^{op}(M)$ is a marking of N_k for each k , and
- the marking M of $N_0 \times N_1$ can evolve by a multiset of events A over \mathcal{E} to a marking M' if and only if for $k = 0$ and $k = 1$, the marking $in_k^{op}M$ can evolve in N_k by the multiset of events $\pi_k A$ to the marking $in_k^{op}M'$.

The proof is straightforward. The above proposition is the analogue for NSet of Winskel's Theorem 4.2.4.

The construction of a product is essential for modelling parallel compositions, since we can use a synchronisation algebra to determine exactly which synchronisations may occur.

Example 5.3.2 Consider the two nets N_0 and N_1 given below:



The product net $N_0 \times N_1$ has event set

$$\mathcal{E} = \{\langle e, e' \rangle, \langle e, e'' \rangle\}$$

and condition set

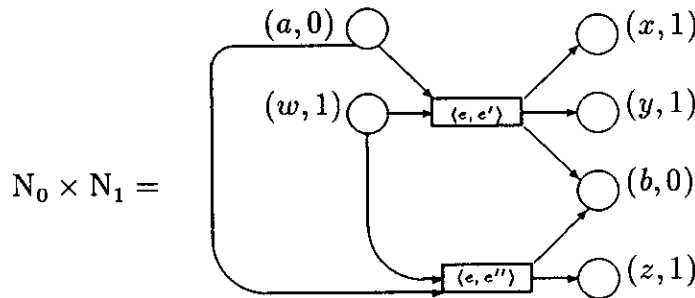
$$\mathcal{B} = \{(a, 0), (b, 0), (w, 1), (x, 1), (y, 1), (z, 1)\}.$$

The pre- and post-conditions of $N_0 \times N_1$ are given by:

$$pre(\langle e, e' \rangle) = \{(a, 0), (w, 1)\}, \quad post(\langle e, e' \rangle) = \{(b, 0), (x, 1), (y, 1)\},$$

$$pre(\langle e, e'' \rangle) = \{(a, 0), (w, 1)\} \quad \text{and} \quad post(\langle e, e'' \rangle) = \{(b, 0), (z, 1)\}.$$

Thus the product net $N_0 \times N_1$ is given by:



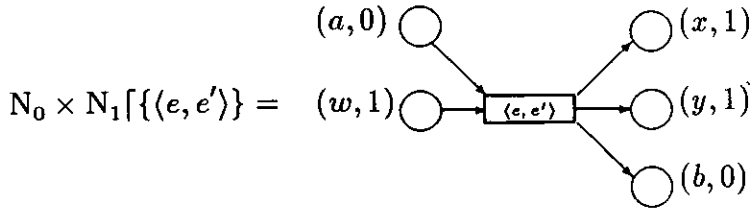
In [Win84a], Winskel defines the *synchronous product* of nets $N_0 = \langle \mathcal{E}_0, \mathcal{B}_0, pre_0, post_0 \rangle$ and $N_1 = \langle \mathcal{E}_1, \mathcal{B}_1, pre_1, post_1 \rangle$ to be $(N_0 \times_{\mathbf{Net}} N_1)[(\mathcal{E}_0 \times \mathcal{E}_1)]$, the restriction of their product in **Net** to the event set $\mathcal{E}_0 \times \mathcal{E}_1$. He states that the synchronous product of two nets is their product in the category \mathbf{Net}_{syn} , of Petri nets and synchronous morphisms (see Section 4.2.1).

Remark 5.3.3

The product of two nets in NSet is precisely their synchronous product.

5.3.1 Restriction of the Product Net

The restriction of the net $N_0 \times N_1$ of Example 5.3.2 to the singleton event set $\{\langle e, e' \rangle\}$ is the net



The synchronisation algebra for the restriction given above is $\langle L, \bullet, \mathbf{X}, \mathbf{0} \rangle$ where $L = \{\lambda_0, \lambda_1, \lambda_2, \mathbf{X}, \mathbf{0}\}$ and the operation \bullet is given as follows:

$$\lambda_i \bullet \mathbf{0} = \mathbf{0} \quad \text{for } i = 0, 1, 2,$$

$$\lambda_i \bullet \mathbf{X} = \mathbf{0} \quad \text{for } i = 0, 1, 2,$$

$$\lambda_0 \bullet \lambda_1 = \lambda_3, \quad \lambda_1 \bullet \lambda_2 = \mathbf{0} \quad \text{and} \quad \lambda_2 \bullet \lambda_3 = \mathbf{0}.$$

The labelling function of the net N_0 is given by $l_0(e) = \lambda_0$.

The labelling function of the net N_1 is given by $l_1(e') = \lambda_1$ and $l_1(e'') = \lambda_2$.

The synchronisation algebra indicates that in the parallel composition of N_0 and N_1 with respect to \mathcal{L} , none of the events e, e' and e'' can occur asynchronously, e can synchronise with e' but not with e'' , and e' cannot synchronise with e'' . Thus the only event of the parallel composition of N_0 and N_1 with respect to \mathcal{L} is the synchronisation of e and e' , that is, the event $\langle e, e' \rangle$.

Remark 5.3.4 *Given nets N_0 and N_1 together with a synchronisation algebra L for them, we can construct a subset \mathcal{E} of $\mathcal{E}_0 \times \mathcal{E}_1$ such that the net $(N_0 \times N_1)[\mathcal{E}]$ has precisely the synchronisations specified by the algebra L .*

5.3.2 The Terminal Object in NC

The terminal object in **NC** is $\mathbf{1} = (1 \xleftarrow{\phi} \text{---} 0)$, the unit of product. In **Set**, the initial object 0 is the empty set ϕ , and the terminal object is the singleton set $\{\star\}$. Hence the constant $\mathbf{1}$ in **NSet** is the net consisting of a single event, no conditions and empty pre- and post-condition relations. By contrast, the terminal

object $\mathbf{1}_{\mathbf{Net}}$ in Winskel's category \mathbf{Net} [Win88] is the *null* net, with no events or conditions and empty initial marking. $\mathbf{1}$ is also the terminal object of Winskel's category \mathbf{Net}_{syn} .

Thus we have

$$\mathbf{1}_{\mathbf{NSet}} = \mathbf{1}_{\mathbf{Net}_{syn}} = \square \quad \text{and} \quad \mathbf{1}_{\mathbf{Net}} = \{ \} .$$

Example 5.3.5 We illustrate the action of \times on nets by proving that $\mathbf{1}$ is the unit of product in \mathbf{NSet} .

Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ be a net.

We shall show that $N \times \mathbf{1}$ is isomorphic in \mathbf{NSet} to N .

The event set of the net $N \times \mathbf{1}$ is $\mathcal{E} \times \mathbf{1} \cong \mathcal{E}$.

The condition set of the net $N \times \mathbf{1}$ is $\mathcal{B} \uplus \phi \cong \mathcal{B}$.

For each $e \in \mathcal{E}$,

$$pre(\langle e, \star \rangle) = pre(e) \uplus \phi = pre(e),$$

and similarly $post(\langle e, \star \rangle) = post(e)$.

Applying Lemma 4.4.13, we see that $N \times \mathbf{1}$ is isomorphic to N in \mathbf{NSet} .

Similarly, $\mathbf{1} \times N$ is isomorphic to N in \mathbf{NSet} .

5.3.3 Sequentialising the Behaviour of a Net using Products

We illustrate an application of the product in \mathbf{NSet} with an example taken from [Win84a].

Example 5.3.6 We can represent a ticking clock by the net I , given by

$$I = \boxed{t} \rightleftarrows \bigcirc p .$$

Remark 5.3.7 Notice that I is the unit of the tensor \otimes in \mathbf{NSet} , see Section 5.5.2.

For any safe net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$, we can interleave the occurrences of events in N by synchronising the occurrence of each event with a tick of the clock. We do this by forming the product in NSet of N with I .

Events in $N \times I$ are pairs $\langle e, t \rangle$, for $e \in \mathcal{E}$. The condition set of $N \times I$ is the set $\mathcal{B} \uplus \{p\}$. We shall assume that $p \notin \mathcal{B}$ and write b for $(b, 0)$ and p for $(p, 1)$.

For each $e \in \mathcal{E}$,

$$pre(\langle e, t \rangle) = pre(e) + p \quad \text{and} \quad post(\langle e, t \rangle) = post(e) + p.$$

Proposition 5.3.8

Let N have initial marking the set M . Let I have initial marking p .

M' is a reachable marking of $N \times I$ if and only if $(M' - p)$ is a reachable marking of N .

Moreover, the marking M can evolve in $N \times I$ under the multiset A of events to marking M' if and only if $A = \langle e, t \rangle$ for some e , and the marking $(M - p)$ can evolve in N under the event e to the marking $(M' - p)$.

Proof: Follows immediately from the definition of the product net.

□

5.4 The Coproduct of Two Nets

The coproduct in GC is dual to the product, interpreting the linear logic connective \oplus . The coproduct of two nets $N_0 = (\mathcal{E}_0 \xleftarrow{\alpha_0} \mathcal{B}_0)$ and $N_1 = (\mathcal{E}_1 \xleftarrow{\alpha_1} \mathcal{B}_1)$ is the net $N = (\mathcal{E}_0 + \mathcal{E}_1 \xleftarrow{\alpha_0 \oplus \alpha_1} \mathcal{B}_0 \times \mathcal{B}_1)$. The pre- and post-condition relations in N are again determined by the behaviour of the coproduct in GSet, applying Theorem 5.1.1 and arguing as in Section 5.3. Thus we have:

$$\langle e_0, 0 \rangle pre_0 \oplus pre_1 \langle b_0, b_1 \rangle \text{ if } e_0 pre_0 b_0 \quad \text{and}$$

$$\langle e_1, 1 \rangle pre_0 \oplus pre_1 \langle b_0, b_1 \rangle \text{ if } e_1 pre_1 b_1,$$

and similarly for the post-conditions.

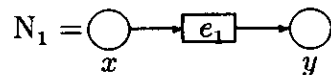
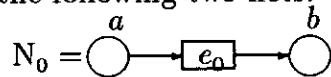
Consider the event $(e_0, 0)$ in N . We have

$$pre(e_0, 0) = \{\langle b_0, b_1 \rangle \mid b_0 \in pre_0(e), b_1 \in \mathcal{B}_1\} \quad \text{and}$$

$$post(e_0, 0) = \{\langle b_0, b_1 \rangle \mid b_0 \in post_0(e), b_1 \in \mathcal{B}_1\}.$$

We now give an example which illustrates the difficulties of expressing the behaviour of a coproduct net in terms of its components.

Example 5.4.1 Consider the following two nets:



The coproduct net $N_0 + N_1$ has event set $\{(e_0, 0), (e_1, 1)\}$, and condition set $\{\langle a, x \rangle, \langle a, y \rangle, \langle b, x \rangle, \langle b, y \rangle\}$. The pre- and post-condition relations of $N_0 + N_1$ are given as follows:

$$pre(e_0, 0) = \{\langle a, x \rangle, \langle a, y \rangle\}, \quad post(e_0, 0) = \{\langle b, x \rangle, \langle b, y \rangle\}$$

$$pre(e_1, 1) = \{\langle a, x \rangle, \langle b, x \rangle\} \quad \text{and} \quad post(e_1, 1) = \{\langle a, y \rangle, \langle b, y \rangle\}.$$

In order to establish the behaviour of the coproduct net, we must express its markings in terms of those of its component nets. The best way of doing this is not immediately obvious, and different choices imply different interpretations of the coproduct.

A natural choice is to define the pair $\langle c_0, c_1 \rangle$ to be marked in $N_0 + N_1$ if and only if c_0 is marked in N_0 and c_1 is marked in N_1 . Thus the event $(e_0, 0)$ can occur whenever a is marked in N_0 and both x and y are marked in N_1 . A firing of $(e_0, 0)$ marks b in N_0 and x and y in N_1 . This appears promising: an event derived from the first component has its usual properties in that component, and does not affect the marking of the second component.

The pre-conditions of the multiset of events $(e_0, 0) + (e_1, 1)$ are $\langle a, x \rangle + \langle a, y \rangle + \langle b, x \rangle + \langle a, x \rangle$, or $2\langle a, x \rangle + \langle a, y \rangle + \langle b, x \rangle$. It is not clear whether a is marked with multiplicity 1 or 2 in N_0 . This illustrates the fundamental difficulty in this choice of interpretation of a marking of the compound net.

The behaviour of the coproduct net, while easy to state, is difficult to interpret. There is no tidy theorem such as Theorem 4.2.5 or Theorem 5.3.1. Further work will examine the *behaviours* of nets in addition to their structural properties, and we hope this will allow us to give a computationally useful interpretation to the coproduct of two nets.

We now sketch a possible interpretation of the behaviour of the coproduct net. Our aim is to interpret the behaviour of the coproduct net in terms of its component nets by means of a function *Behav* from $\text{Mark}(N_0) \times \text{Mark}(N_1)$ to $\text{Mark}(N_0 + N_1)$. It is routine to construct a category from the transition graph $\mathcal{T}(N)$ of a net N : we shall denote this category by $\mathcal{T}(N)$. We expect our function *Behav* on markings to extend to a functor from $\mathcal{T}(N_0) \times \mathcal{T}(N_1)$ to $\mathcal{T}(N_0 + N_1)$. We extend our notion of marking to include functions from the condition set into the ordinal $\omega + 1$. A possible choice for *Behav* is the function *Behav*₀ which preserves the markings of N_0 and marks every condition of N_1 in multiplicity ω . This does give rise to a functor of transition graphs. Similarly, there is a function *Behav*₁ which preserves the markings of N_1 and marks every condition of N_0 in multiplicity ω . If we consider our map on markings to be a non-deterministic choice between *Behav*₀ and *Behav*₁, then the operator \oplus gives a non-deterministic choice between the two component nets (as does the coproduct in Meseguer and Montanari's category of *marked* Petri nets, *MPetri*). The net $N_0 + N_1$ behaves either as net N_0 or as net N_1 , but we cannot determine externally which net's behaviour it will adopt. The choice is made at the level of the function on markings.

This interpretation of the coproduct of two nets agrees with our intuition as to the meaning of linear logic \oplus . This interpretation appears to relate to Plotkin's powerdomain construction [Plo76].

5.4.1 The Initial Object in NC

The initial object in **NC** is $\mathbf{0} = (0 \xleftarrow{\phi} 1)$, the unit of the coproduct. Thus in **NSet**, $\mathbf{0}$ is the net with one condition and no events. It corresponds to Winskel's initial net *nil* which consists of one condition, marked in multiplicity one.

Thus we have

$$\mathbf{0} = \bigcirc \quad \text{and} \quad \mathbf{0}_{\mathbf{Net}} = \odot .$$

Example 5.4.2 We illustrate the action of $+$ on nets by proving that $\mathbf{0}$ is the unit of coproduct in **NC**.

Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ be a net.

We shall show that $N + \mathbf{0}$ is isomorphic in **NSet** to N .

The event set of the net $N + \mathbf{0}$ is $\mathcal{E} \uplus \phi \cong \mathcal{E}$.

The condition set of the net $N + \mathbf{0}$ is $\mathcal{B} \times 1 \cong \mathcal{B}$.

For each $e \in \mathcal{E}$,

$$pre(\langle e, 0 \rangle) = \{ \langle b, \star \rangle \mid e \text{ pre } b \} \cong pre(e).$$

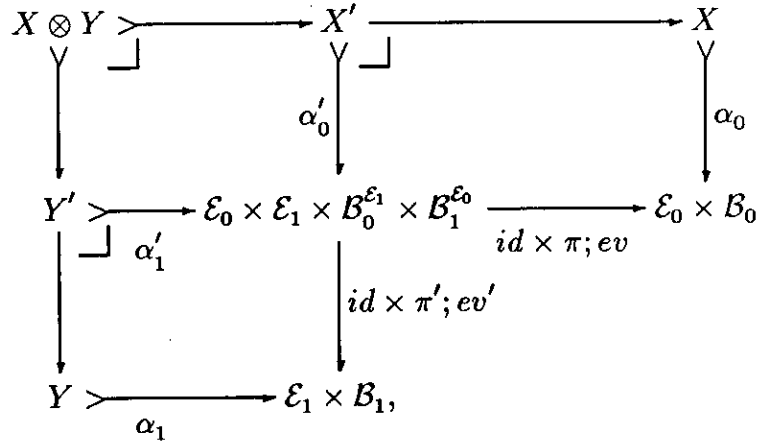
and similarly $post(\langle e, 0 \rangle) \cong post(e)$.

Applying Lemma 4.4.13, we see that $N + \mathbf{0}$ is isomorphic to N in **NSet**.

Similarly, $\mathbf{0} + N$ is isomorphic to N in **NSet**.

5.5 Tensor Product as Concurrent Composition

The tensor of a pair of objects $(\mathcal{E}_0 \xleftarrow{\alpha_0} \mathcal{B}_0)$ and $(\mathcal{E}_1 \xleftarrow{\alpha_1} \mathcal{B}_1)$ in **GC** is $(\mathcal{E}_0 \times \mathcal{E}_1 \xleftarrow{\alpha_0 \otimes \alpha_1} \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0})$. It is formed by pulling back α_0 to α'_0 along the map $id \times \pi; ev$ (where π is projection from $\mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0}$ to $\mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1}$ and ev is evaluation $ev : \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1} \rightarrow \mathcal{B}_0$), pulling back α_1 to α'_1 along the map $id \times ev'; \pi'$ (where π' is projection from $\mathcal{E}_0 \times \mathcal{B}_1^{\mathcal{E}_0} \times \mathcal{B}_0^{\mathcal{E}_1}$ to $\mathcal{E}_0 \times \mathcal{B}_1^{\mathcal{E}_0}$ and ev' is evaluation $ev' : \mathcal{E}_0 \times \mathcal{B}_1^{\mathcal{E}_0} \rightarrow \mathcal{B}_1$), and then pulling back α'_0 along α'_1 . Thus (suppressing the structural isomorphisms) we have:



and $\alpha_0 \otimes \alpha_1$ is the inclusion of $X \otimes Y$ in $\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0}$.

When $\mathcal{E}_0, \mathcal{E}_1, \mathcal{B}_0$ and \mathcal{B}_1 are sets, we have

$$\langle e_0, e_1 \rangle (\alpha \otimes \beta) \langle f, g \rangle \text{ if and only if } e_0 \alpha_0 f(e_1) \text{ and } e_1 \alpha_1 g(e_0).$$

Applying Theorem 5.1.1, this symmetric monoidal structure lifts to a symmetric monoidal structure on NC, which we shall also write as \otimes . For $e_0 \in \mathcal{E}_0$, and $e_1 \in \mathcal{E}_1$, the pre- and post-condition relations of $N_0 \otimes N_1$ in NSet is:

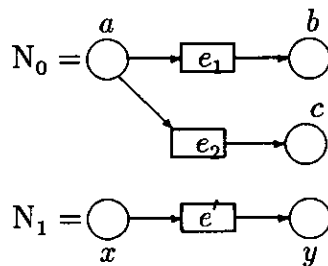
$$pre(\langle e_0, e_1 \rangle) = \{ \langle f, g \rangle \in \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0} \mid f(e_1) \in pre_0(e_0), g(e_0) \in pre_1(e_1) \}$$

and

$$post(\langle e_0, e_1 \rangle) = \{ \langle f, g \rangle \in \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0} \mid f(e_1) \in post_0(e_0), g(e_0) \in post_1(e_1) \}.$$

We now consider a simple example of a tensor product in NSet.

Example 5.5.1 Consider the nets N_0 and N_1 given below.



We put $N_0 = \langle \mathcal{E}_0, \mathcal{B}_0, pre_0, post_0 \rangle$ and $N_1 = \langle \mathcal{E}_1, \mathcal{B}_1, pre_1, post_1 \rangle$.

Thus $\mathcal{E}_0 = \{e_1, e_2\}$ and $\mathcal{E}_1 = \{e'\}$, and $\mathcal{B}_0 = \{a, b, c\}$ and $\mathcal{B}_1 = \{x, y\}$.

In our example, the possible functions $f: \mathcal{E}_1 \rightarrow \mathcal{B}_0$ and $g: \mathcal{E}_0 \rightarrow \mathcal{B}_1$ are given as follows:

$$\begin{array}{lll}
 f_a(e') = a & g_{xy}(e_1) = x & g_{xy}(e_2) = y \\
 f_b(e') = b & g_{yy}(e_1) = y & g_{yy}(e_2) = y \\
 f_c(e') = c & g_{xx}(e_1) = x & g_{xx}(e_2) = x \\
 & g_{yx}(e_1) = y & g_{yx}(e_2) = x.
 \end{array}$$

The event set of the tensor net $N_0 \otimes N_1$ is $\mathcal{E}_0 \times \mathcal{E}_1$.

The condition set of the tensor net $N_0 \otimes N_1$ is

$$\{ \langle f, g \rangle \mid f: \mathcal{E}_1 \rightarrow \mathcal{B}_0 \text{ and } g: \mathcal{E}_0 \rightarrow \mathcal{B}_1 \}.$$

The pre- and post-relations are as follows:

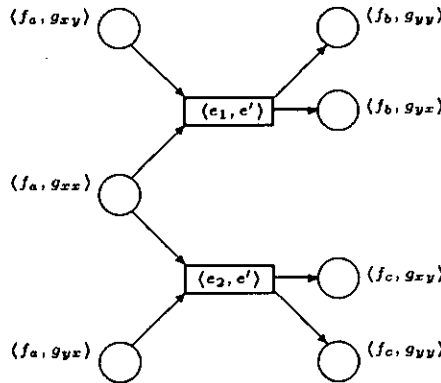
$$pre(\langle e_1, e' \rangle) = \{ \langle f_a, g_{xx} \rangle, \langle f_a, g_{xy} \rangle \},$$

$$post(\langle e_1, e' \rangle) = \{ \langle f_b, g_{yx} \rangle, \langle f_b, g_{yy} \rangle \},$$

$$pre(\langle e_2, e' \rangle) = \{ \langle f_a, g_{xx} \rangle, \langle f_a, g_{yx} \rangle \}$$

$$\text{and } post(\langle e_2, e' \rangle) = \{ \langle f_c, g_{xy} \rangle, \langle f_c, g_{yy} \rangle \}.$$

The tensor net $N_0 \otimes N_1$ is given by:



together with the isolated conditions

$$\{ \langle f_a, g_{yy} \rangle, \langle f_b, g_{xx} \rangle, \langle f_b, g_{xy} \rangle, \langle f_c, g_{xx} \rangle, \langle f_c, g_{yx} \rangle \}$$

none of which is connected to any part of the net $N_0 \otimes N_1$. Notice that the conflict in net N_0 is reflected by conflict in net $N_0 \otimes N_1$.

We now relate the behaviour of the composite net $N_0 \otimes N_1$ to synchronised behaviours of the component nets N_0 and N_1 .

We interpret the tensor product of arbitrary nets N_0 and N_1 as follows. Consider one of the pre-conditions $\langle f, g \rangle$ of an event $\langle e_0, e_1 \rangle$ in $N_0 \otimes N_1$. We understand the function f as representing a channel of communication between event e_0 in N_0 and $pre_1(e_1)$ in N_1 , and the function g as representing a channel of communication between e_1 in N_1 and $pre_0(e_0)$ in N_0 . This is justified by the fact that (interpreting $\pi_i(\langle f_0, f_1 \rangle)$ as f_i for $i = 0, 1$), we have $\pi_0(\langle f, g \rangle) (\pi_0\langle e_0, e_1 \rangle) \in pre_0(e_0)$ and $\pi_1(\langle f, g \rangle) (\pi_1\langle e_0, e_1 \rangle) \in pre_1(e_1)$:

We consider the channel f to be marked if a communication is possible along it, and that communication confirms that $f(e_0)$ is marked.

We consider the pair $\langle f, g \rangle$ to be marked exactly when the channels f and g are both marked.

The marking of $N_0 \otimes N_1$ at a certain moment then tells us which communications are possible and successful at that moment. Whenever the set

$$\{\langle f, g \rangle \mid f: \mathcal{E}_1 \rightarrow \mathcal{B}_0, g: \mathcal{E}_0 \rightarrow \mathcal{B}_1, f(e_1) \in pre_0(e_0), g(e_0) \in pre_1(e_1)\}$$

is marked in $N_0 \otimes N_1$, e_1 can “check” via the $\{f\}$ that all conditions in $pre_0(e_0)$ are marked, and similarly e_0 checks via the $\{g\}$ that $pre_1(e_1)$ is marked. The event $\langle e_0, e_1 \rangle$ can fire, with the effect of e_0 and e_1 firing in synchronisation. After this firing, a new set of communications $post(\langle e_0, e_1 \rangle)$ has become possible, indicating that $post_0(e_0)$ and $post_1(e_1)$ are marked in N_0 and N_1 respectively. Evolution of the net $N_0 \otimes N_1$ proceeds in this way.

Remark 5.5.2 *Returning to Example 5.5.1, the fact that the tensor net has a number of isolated conditions indicates that the communications channels which these conditions represent are not essential to any synchronisation of the two nets N and N' . This may prove to be an advantage of the tensor product over the cartesian product, (both in the category NSet and in Winskel’s category Net) in that the construction of the tensor product makes explicit the necessary communications between the component nets, while the cartesian product leaves such issues*

unresolved. Thus, forming the tensor product of two nets not only gives us information about what synchronisations are possible, but also about what physical links are required between the nets to achieve those synchronisations.

5.5.1 Other Tensor Products

The symmetric monoidal structure \otimes differs from the tensor in [MM88b]. It has a plausible computational interpretation. The tensor introduced by Brown in [Bro89b] (see Chapter 8.1), Section 9.3.1) which we shall here call \otimes' , is the second tensor product mentioned by de Paiva in [deP89a], and is not adjoint to the internal hom. It is the monic $(\mathcal{E}_0 \times \mathcal{E}_1 \xleftarrow{\alpha_0 \otimes' \alpha_1} \mathcal{B}_0 \times \mathcal{B}_1)$ given by

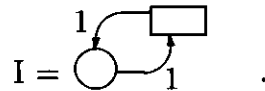
$$\langle e_0, e_1 \rangle (\alpha_0 \otimes' \alpha_1) \langle b_0, b_1 \rangle \text{ if and only if } e_0 \alpha_0 b_0 \text{ and } e_1 \alpha_1 b_1.$$

Let N_0 and N_1 be safe nets. As before, we shall consider the place $\langle b_0, b_1 \rangle$ in the composite net be marked exactly when b_0 is marked in N_0 and b_1 is marked in N_1 . Then the net $N_0 \otimes' N_1$ has precisely the same behaviour as the product net $N_0 \times N_1$.

5.5.2 The Unit of the Monoidal Structure \otimes

The unit of the tensor \otimes in **NC** is $I = 1 \xleftarrow{id} 1$. Thus the constant I in **NSet** is the net consisting of one condition \star and one event having pre-condition \star and post-condition \star .

Thus we have



Example 5.5.3

To illustrate the action of \otimes on nets, we prove I is the unit of \otimes in **NSet**.

Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$. We shall show that the tensor product $N \otimes I$ is isomorphic in **NSet** to N .

The event set of $N \otimes I$ is the set $\mathcal{E} \times 1 \cong \mathcal{E}$.

The condition set of $N \otimes I$ is the set $\mathcal{B}^1 \times 1^\mathcal{E} \cong \mathcal{B}$,

For each $e \in \mathcal{E}$ we have

$$pre(\langle e, \star \rangle) = \{ \langle f, g \rangle \mid f: 1 \rightarrow \mathcal{B}, g: \mathcal{E} \rightarrow 1, f(\star) \in pre(e) \text{ and } g(e) \in 1 \}.$$

The only function $g: \mathcal{E} \rightarrow 1$ is the constant map sending every element of \mathcal{E} to \star . Hence the condition on g is always satisfied, and we have

$$pre(\langle e, \star \rangle) \cong \{ f \mid f: 1 \rightarrow \mathcal{B} \text{ and } f(\star) \in pre(e) \}.$$

Now \mathcal{B} is isomorphic to \mathcal{B}^1 , via the map which sends $b \in \mathcal{B}$ to the function f_b which maps \star to b (that is, the map $\Lambda(\pi_1: 1 \times \mathcal{B} \xrightarrow{\cong} \mathcal{B}^1)$). Hence

$$pre(\langle e, \star \rangle) \cong \{ b \mid f_b \in pre(e) \} \cong pre(e).$$

Similarly, $post(\langle e, \star \rangle) \cong post(e)$ for each $e \in \mathcal{E}$.

Applying Lemma 4.4.13, we see that $N \otimes I$ is isomorphic in NSet to N .

A similar argument shows that $I \otimes N$ is isomorphic to N in NSet.

5.6 Expressing Synchrony and Asynchrony

We have seen that an event $\langle e_0, e_1 \rangle$ in the product net $N_0 \times N_1$ is enabled whenever the events e_0 and e_1 in the component nets are both enabled at the same time. Thus events of the product net correspond to *synchronisations* of events in the component nets. We use a new example to demonstrate how events may occur asynchronously in a product net.

Definition 5.6.1

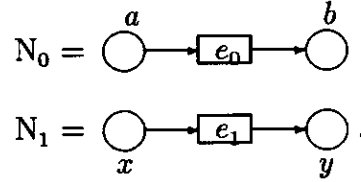
An event $\langle e_0, e_1 \rangle$ of the product net $N_0 \times N_1$ is asynchronous if either

$$pre(\langle e_0, e_1 \rangle) = in_0(pre(e_0)) \quad \text{and} \quad post(\langle e_0, e_1 \rangle) = in_0(post(e_0)), \quad \text{or}$$

$$pre(\langle e_0, e_1 \rangle) = in_1(pre(e_1)) \quad \text{and} \quad post(\langle e_0, e_1 \rangle) = in_1(post(e_1)).$$

Thus an asynchronous event is an event of a product net whose firing concerns only one of the component nets of the product.

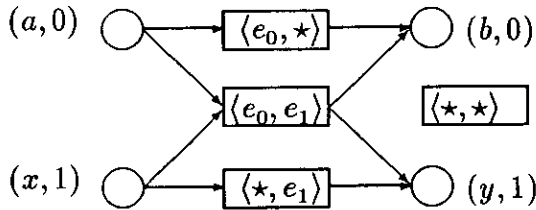
Example 5.6.2 Let N_0 and N_1 be the nets:



We shall form the product of the two nets $N_0 + \perp$ and $N_1 + \perp$ where \perp is the unit of \sharp , described in Section 5.8. The constant \perp is the net consisting of one event and one condition, with empty pre- and post-condition relations. Thus $\perp = (1 \xleftarrow{\phi} 1)$.

We put $\mathcal{E}_0 = \{e_0, \star\}$, $\mathcal{E}_1 = \{e_1, \star\}$, $\mathcal{B}_0 = \{\langle a, \star \rangle, \langle b, \star \rangle\}$ and $\mathcal{B}_1 = \{\langle x, \star \rangle, \langle y, \star \rangle\}$.

Then the product net $N_0 \times N_1$ is given by:



Since the event \star has empty pre-condition, the synchronisation of any event e with \star can occur whenever the pre-conditions of e are satisfied. Thus the event $\langle e_0, \star \rangle$ can fire whenever e_0 can fire in N_0 . Similarly, the event $\langle \star, e_1 \rangle$ can fire whenever e_1 can fire in N_1 . Thus the occurrence of an event synchronously with \star depends only on the marking of one of the component nets of the product $N_0 \times N_1$, and alters the marking of that component alone. Thus $\langle e_0, \star \rangle$ and $\langle \star, e_1 \rangle$ are asynchronous events in the product net $N_0 \times N_1$.

Thus the event set of net $(N_0 + \perp) \times (N_1 + \perp)$ contains all possible synchronisations of events in N_0 and N_1 , together with an asynchronous event corresponding to each event of either N_0 or N_1 . These constitute the events of Winskel's product net [Win88]. In addition, the product net in NSet includes the isolated event

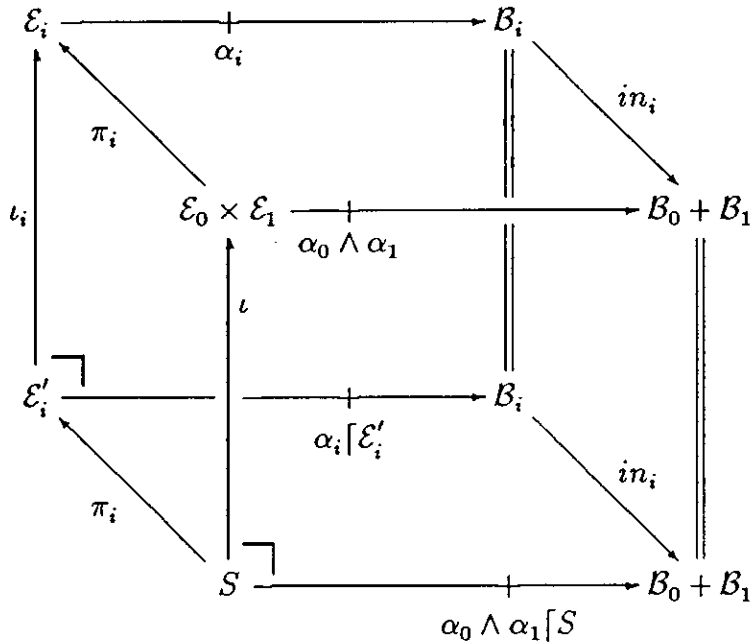
$\langle \star, \star \rangle$, which is precisely the event which Winskel must exclude in order that the universal property for the product is satisfied.

5.6.1 Restrictions of the Product of two Nets

Let $N_0 = (\mathcal{E}_0 \xleftarrow{\alpha_0} \mathcal{B}_0)$ and $N_1 = (\mathcal{E}_1 \xleftarrow{\alpha_1} \mathcal{B}_1)$ be objects of NSet. Restrictions of the product $N_0 \times N_1$ are given by a particular class of subobjects of the relation $\alpha_0 \wedge \alpha_1$. If we wish to restrict the product net to the subset of events $\mathcal{E}'_0 \times \mathcal{E}'_1 \subseteq \mathcal{E}_0 \times \mathcal{E}_1$, we simply use $\mathcal{E}'_0 \times \mathcal{E}'_1$ to construct a subobject S of $A \wedge B$. The monic from S into $(\mathcal{E}_0 \times \mathcal{E}_1) \times (\mathcal{B}_0 + \mathcal{B}_1)$ is the net $(N_0 \otimes N_1)[(\mathcal{E}'_0 \times \mathcal{E}'_1)]$, thus:

$$\begin{array}{ccc}
 \mathcal{E}'_0 \times \mathcal{E}'_1 & \xleftarrow{\alpha_0 \wedge \alpha_1|_S} & S' \\
 \downarrow \Upsilon & & \downarrow \Upsilon \\
 \mathcal{E}_0 \times \mathcal{E}_1 & \xleftarrow{\alpha_0 \wedge \alpha_1} & \mathcal{B}_0 + \mathcal{B}_1.
 \end{array}$$

In view of the following commuting diagrams (in which we write \mathcal{E} to stand for the event set $\mathcal{E}'_0 \times \mathcal{E}'_1$):



it is readily seen that

$$(\alpha_0 \wedge \alpha_1)|_{\mathcal{E}'_0 \times \mathcal{E}'_1} = \alpha_0|_{\mathcal{E}'_0} \wedge \alpha_1|_{\mathcal{E}'_1}.$$

5.6.2 Restrictions of the Tensor Product of two Nets

Let $N_0 = (\mathcal{E}_0 \xleftarrow{\alpha_0} \mathcal{B}_0)$ and $N_1 = (\mathcal{E}_1 \xleftarrow{\alpha_1} \mathcal{B}_1)$ be objects of NSet. Restrictions of the tensor product $N_0 \otimes N_1$ are given by a particular class of subobjects of the relation $\alpha_0 \otimes \alpha_1$. If we wish to restrict the tensor net to the subset of events $\mathcal{E}'_0 \times \mathcal{E}'_1 \subseteq \mathcal{E}_0 \times \mathcal{E}_1$, we simply use $\mathcal{E}'_0 \times \mathcal{E}'_1$ to construct a subobject S of $A \otimes B$. The monic from S into $\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_0} \times \mathcal{B}_1^{\mathcal{E}_0}$ is the net $(N_0 \otimes N_1)[(\mathcal{E}'_0 \times \mathcal{E}'_1)$.

Let ι be the inclusion of $\mathcal{E}'_0 \times \mathcal{E}'_1$ in $\mathcal{E}_0 \times \mathcal{E}_1$. Then S is simply the pullback of $\alpha_0 \otimes \alpha_1$ along $\iota \times id$, thus:

$$\begin{array}{ccc}
 S & \xrightarrow{\quad} & X \otimes Y \\
 \downarrow & \lrcorner & \downarrow \alpha_0 \otimes \alpha_1 \\
 \mathcal{E}'_0 \times \mathcal{E}'_1 \times \mathcal{B}_0^{\mathcal{E}'_0} \times \mathcal{B}_1^{\mathcal{E}'_0} & \xrightarrow{\iota \times id} & \mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_0} \times \mathcal{B}_1^{\mathcal{E}_0}
 \end{array}$$

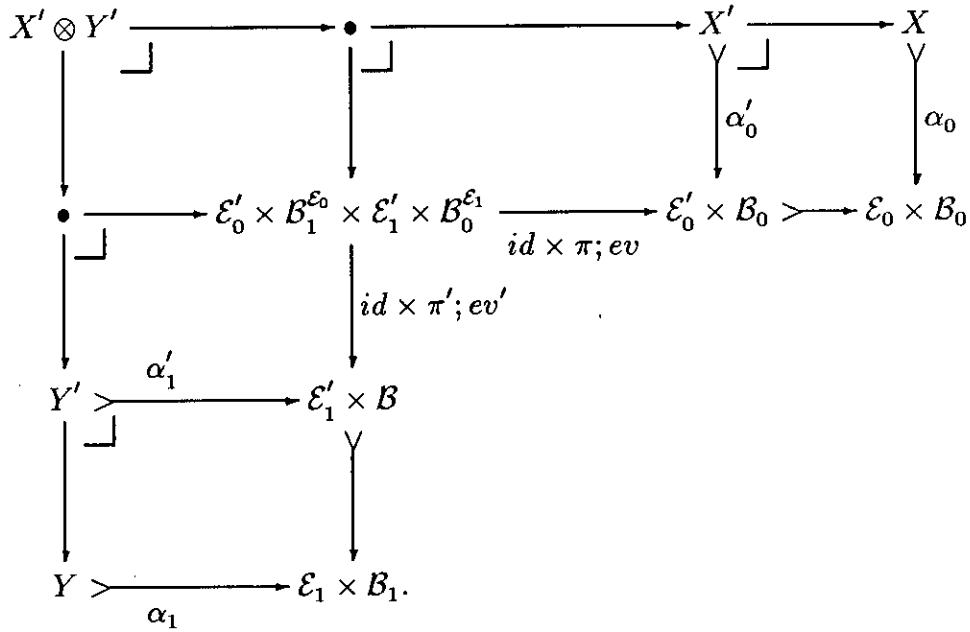
Then

$$S = \{(e_0, e_1, f, g) \mid (e_0, e_1) \in (\mathcal{E}'_0 \times \mathcal{E}'_1) \text{ and } e_1 \beta f(e_0) \text{ and } e_0 \alpha g(e_1)\}.$$

By constructing subobjects in NSet of product or tensor nets, we can restrict these nets on arbitrary sets of events. In particular, by restricting a product net, we can specify events which may only occur in synchronisations (as is done by restriction in Milner's CCS [Mil89]), or events which can only occur asynchronously. Thus subobjects give a way of constructing in this framework most existing notions of parallel composition (compare [Win87]). We have a straightforward way of handling restriction, which is often a difficulty in algebraic treatments of concurrency theory.

Remark 5.6.3 *The properties of pullbacks ensure that the two other candidates for the construction of the restriction $(N_0 \otimes N_1)[(\mathcal{E}'_0 \times \mathcal{E}'_1)$ yield the same net, S , as we now show.*

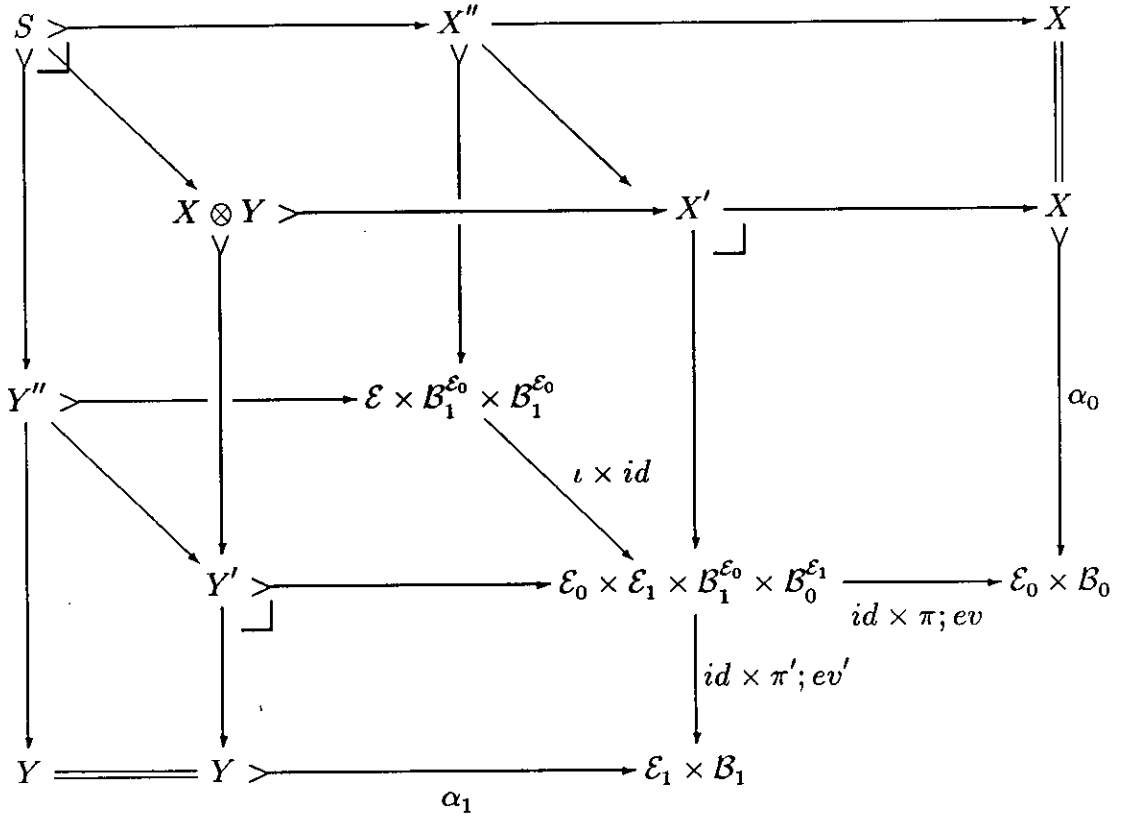
The first alternative is to take the tensor product of $N_0[\mathcal{E}'_0]$ and $N_1[\mathcal{E}'_1]$ as follows:



It is evident that $X' \otimes Y' = S$, with the same inclusion into $\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0}$.

The third possible approach is to pullback the relations α_0 and α_1 along the composition of the inclusion $\iota \times id$ of $\mathcal{E}'_0 \times \mathcal{E}'_1 \times \mathcal{B}_0^{\mathcal{E}'_1} \times \mathcal{B}_1^{\mathcal{E}'_0}$ into $\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0^{\mathcal{E}_1} \times \mathcal{B}_1^{\mathcal{E}_0}$ with the maps into $\mathcal{E}_0 \times \mathcal{B}_0$ and $\mathcal{E}_1 \times \mathcal{B}_1$ respectively, obtaining the objects X'' and Y'' respectively. Thus, X'' is the pullback of α_0 along $(\iota \times id); (id \times \pi; ev)$ and Y'' is the pullback of α_1 along $(\iota \times id); (id \times \pi'; ev')$. We then pullback again to S , as

shown below:



Again, it is readily seen that

$$S = \{ \langle e_0, e_1, f, g \rangle \mid \langle e_0, e_1 \rangle \in (\mathcal{E}'_0 \times \mathcal{E}'_1) \text{ and } e_1 \alpha_1 f(e_0) \text{ and } e_0 \alpha_0 g(e_1) \}.$$

5.7 The Exponential of Two Nets

We define an internal hom on **GC** as follows.

$$[(\mathcal{E}_0 \xleftarrow{\alpha_0} B_0), (\mathcal{E}_1 \xleftarrow{\alpha_1} B_1)]_{\mathbf{GC}} = (\mathcal{E}_1^{\mathcal{E}_0} \times B_0^{B_1} \xleftarrow{\alpha_1^{\alpha_0}} \mathcal{E}_0 \times B_1)$$

Again, $\alpha_1^{\alpha_0}$ is constructed using successive pullbacks.

The adjunction $- \otimes A \dashv [A, -]$ gives **GC** a symmetric monoidal closed structure.

Remark 5.7.1 *In order to guarantee the existence of a greatest subobject of $\mathcal{E}_1^{\mathcal{E}_0} \times B_0^{B_1} \times \mathcal{E}_0 \times B_1$, de Paiva requires that the underlying category **C** be locally cartesian closed: in fact, it is sufficient to ensure that the subobjects of any object*

of \mathbf{C} form a complete Heyting algebra, and this condition is easier to prove in some cases than de Paiva's.

When \mathcal{E}_0 , \mathcal{E}_1 , \mathcal{B}_0 and \mathcal{B}_1 are sets, we have

$$\langle f, F \rangle \alpha_1^{\alpha_0} \langle e_0, b_1 \rangle \text{ if and only if } e_0 \alpha_0 F(b_1) \Rightarrow f(e_0) \alpha_1 b_1.$$

In NSet, putting $N = [N_0, N_1]$ we have

$$\langle e_0, b_1 \rangle \in \text{pre}(\langle f, F \rangle) \text{ if and only if } F(b_1) \in \text{pre}_0(e_0) \Rightarrow b_1 \in \text{pre}_1(fe_0)$$

and

$$\langle e_0, b_1 \rangle \in \text{post}(\langle f, F \rangle) \text{ if and only if } F(b_1) \in \text{post}_0(e_0) \Rightarrow b_1 \in \text{post}_1(fe_0).$$

If for all $b_1 \in \mathcal{B}_1$ we have $\langle e_0, b_1 \rangle \in \text{pre}(\langle f, F \rangle)$ then

$$\text{for all } b_1, \quad F(b_1) \in \text{pre}_0(e_0) \Rightarrow b_1 \in \text{pre}_1(fe_0),$$

and hence

$$F^{-1}(\text{pre}_0(e_0)) \subseteq \text{pre}_1(fe_0),$$

and hence fe_0 refines e_0 relative to F .

Thus the pre-conditions of an event $\langle f, F \rangle$ in the exponential net $[N_0, N_1]$ indicate the extent to which N_1 is a refinement of N_0 .

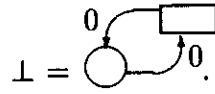
In particular, if $\mathcal{E}_0 \times \mathcal{B}_0 \subseteq \text{pre}(\langle f, F \rangle)$, then N_1 is a refinement of N_0 and (f, F) is a refinement map from N_0 to N_1 .

Conversely, if there is no event $\langle f, F \rangle$ of the exponential net $[N_0, N_1]$ for which $\mathcal{E}_0 \times \mathcal{B}_0 \subseteq \text{pre}(\langle f, F \rangle)$, then N_1 is not a refinement of N_0 .

5.8 Negation in NSet

We shall not be discussing the connective \sharp on Petri nets, although Theorem 5.1.1 shows that it is possible to interpret the logical connective \sharp as a constructor of nets. However, the unit \perp of \sharp is of interest to us in that it is used to define the linear negation of a net.

The unit of \sharp in NC is \perp , given by $\perp = 1 \xleftarrow{0} \text{---} 1$. Thus the constant net \perp in NSet is the net with one place and one event having empty pre- and post-conditions:



Recall that the linear negation A^\perp of an object A of GC, is the object $[A, \perp]$. In NSet, where relations are decidable, the negation of the object $(\mathcal{E} \xleftarrow{\alpha} \text{---} \mathcal{B})$ is the object $(\mathcal{B} \xleftarrow{\perp^\alpha} \text{---} \mathcal{E})$, where the relation \perp^α is given as follows, for $i = 0, 1$:

$$b \perp^{\alpha_i} e \quad \text{if and only if} \quad e \notin \phi_i b.$$

Thus we have:

$$pre(b) = \{e \in \mathcal{E} \mid b \notin pre(e)\} \quad \text{and}$$

$$post(b) = \{e \in \mathcal{E} \mid b \notin post(e)\}.$$

Remark 5.8.1 *Negating a net interchanges its condition and event sets.*

We now give some definitions from graph theory which characterise those nets whose negations are, in a precise way, simpler than the nets themselves.

Definition 5.8.2 *The order of a directed graph $G = \langle V, E, Source, Target \rangle$ is $|V|$, the number of vertices of G .*

Definition 5.8.3 *The size of a directed graph $G = \langle V, E, Source, Target \rangle$ is $|E|$, the number of edges of G .*

Definition 5.8.4 Let $G = \langle V, E, Source, Target \rangle$ be a directed graph. A loop in G is an edge $e \in E$ such that $Source(e) = Target(e)$.

Definition 5.8.5 Let G be a directed graph of order n and size m with no loops. The density of G , written $d(G)$, is $\frac{m}{n(n-1)}$.

Definition 5.8.6 Let G be a directed graph.

If $d(G) > \frac{3}{4}$ then G is dense.

If $d(G) < \frac{1}{4}$ then G is sparse.

The density of a directed graph is a measure of the number of pairs of its vertices which are joined by an edge. The specific choice of $\frac{1}{4}$ and $\frac{3}{4}$ in the definitions of sparse and dense graphs is unimportant, but these figures suffice for our purposes.

Definition 5.8.7 A directed graph $G = \langle V, E, Source, Target \rangle$ is a bipartite graph with vertex classes V_1 and V_2 if $V = V_1 \cup V_2$, $V_1 \cap V_2 = \phi$, and for each $e \in E$, V_1 contains either $Source(e)$ or $Target(e)$, but not both.

Remark 5.8.8 A directed bipartite graph has no loops.

Remark 5.8.9 A Petri net can be regarded as a directed bipartite graph by putting $V_1 = \mathcal{E}$, $V_2 = \mathcal{B}$ and

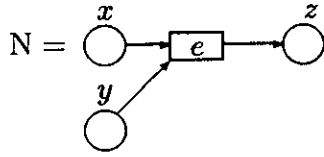
$$E = \{(v_1, v_2) \in V_1 \times V_2 \mid v_2 \in pre(v_1)\} \cup \{(v_2, v_1) \in V_2 \times V_1 \mid v_2 \in post(v_1)\}.$$

The negation of a net N is of particular interest where N is dense when viewed as a directed bipartite graph. In this case we can establish properties N by considering properties of its negation, which is sparse.

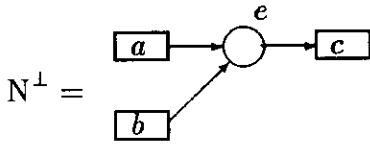
We now give some examples of nets and their negations.

Example 5.8.10

Consider the net N given below:

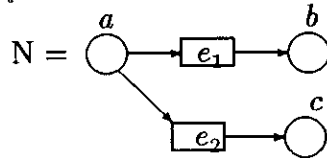


The net N^\perp is given by:

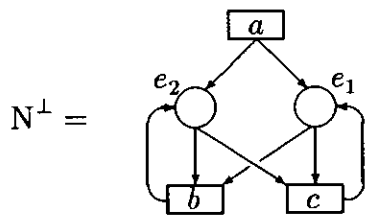


Example 5.8.11

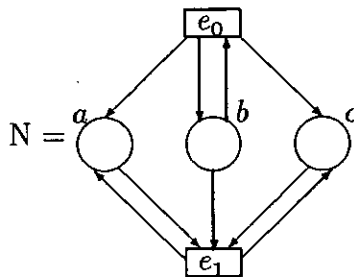
Consider the net N given below:



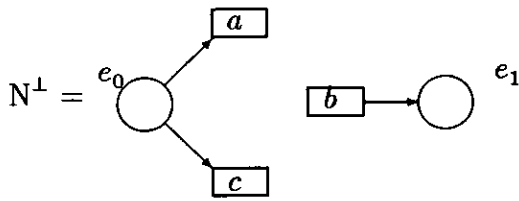
The net N^\perp is given by:



Example 5.8.12 The following net is a dense directed bipartite graph:



The negation of N is sparse:

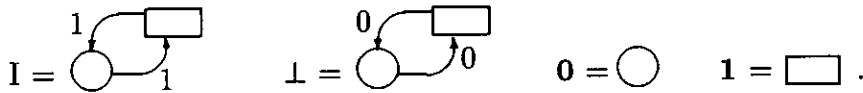


There are certain structural properties of a net which are easier to prove of a sparse net than of a dense net. Hence we can use negation to simplify the proof of certain properties of a dense graph.

5.9 The Constants in NC

NC has a distinct object interpreting each of the linear constants \perp , \top , $\mathbf{0}$ and $\mathbf{1}$. \perp and \top are interpreted by the units of $\#$ and \otimes respectively, while $\mathbf{0}$ and $\mathbf{1}$ are interpreted respectively by the initial and the terminal object.

The constants in NSet are as follows:



The constants in Winskel's category Net (and also in SafeNet) are as follows:

$$\mathbf{0} = \odot \quad \mathbf{1} = \{ \}$$

Each of the four constants has an important individual rôle. Although the constant nets $\mathbf{0}$ and \perp appear very similar, and one might argue that we cannot distinguish them by their behaviour, they have very different behaviour when placed in parallel with another net.

Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$. The net $N \times \mathbf{0}$ has event set $\mathcal{E} \times \phi \cong \phi$ and condition set $\mathcal{B} \uplus \mathbf{1} \cong \mathcal{B}_\perp$. The pre- and post-condition relations of $N \times \mathbf{0}$ are empty.

The net $N \times \perp$ has event set $\mathcal{E} \times 1 \cong \mathcal{E}$, and condition set $\mathcal{B} \uplus 1 \cong \mathcal{B}_\perp$. The pre- and post-condition relations are such that for each $e \in \mathcal{E}$,

$$pre(\langle e, \star \rangle) \cong pre(e) \quad \text{and} \quad post(\langle e, \star \rangle) \cong post(e).$$

Notice that $N \times \perp$ is not isomorphic to N , since there is no bijection between the condition sets of the two nets.

Let M_0 be an initial marking of $\mathbf{0}$ and of \perp . If M is reachable from M_0 in either $\mathbf{0}$ or \perp then $M = M_0$. In this sense $\mathbf{0}$ and \perp have the same behaviour.

We can associate with any Petri net a transition system by labelling vertices with markings and arrows with parallel and sequential compositions of events.

Then the transition system associated with $\mathbf{0}$ has no arrows, whereas the transition system associated with \perp has an identity arrow at each vertex, and no other arrows.

The transition systems of these two nets are neither bisimilar nor observationally equivalent, and so we should not be surprised by their different properties when placed in parallel with another net.

Chapter 6

Exploiting the Generality of the Framework

We can adapt the construction of **NC** in various ways, while retaining many of its properties. In this chapter we give several examples of modifications to the construction. Our aim is to define categories related to **NC** which are also sound models of linear logic. The generality of the framework given by the dialectica categories allows us to view as instances of the same fundamental construction a number of categories, some of which are new and some of which have been studied before.

6.1 A Different Base Category **C**

The construction of **NC** is parametric in **C**, and so different models arise when we consider a different base category **C**.

For example, consider the category **PSet**, whose objects are pointed sets and whose morphisms are strict functions under functional composition, or the category **Rel**, whose objects are sets and whose morphisms are relations on sets, composed in the usual way. Taking **PSet** or **Rel** as our base category **C** gives rise to morphisms in **NC** whose first components are respectively partial functions or relations.

Recent work of Nielsen, Rozenberg and Thiagarajan [NRT90] uses a category of Petri nets in which morphisms from N to N' are pairs (η, β) where η is a partial

function from \mathcal{E} to \mathcal{E}' and β is the inverse of a partial function from \mathcal{B} to \mathcal{B}' . The objects of their category are elementary nets (which they call *elementary net systems*). It can readily be shown that this category is a subcategory of **NPSet**. In fact, their category is precisely the category **NPSet**⁼, which we shall define in Section 6.3.

Unfortunately, neither **PSet** nor **Rel** is cartesian closed, and so **NPSet** and **NRel** do not have all the structure we desire. Some at least of the structure required to model linear logic is found in **NPSet**, and investigation of this will be an interesting area for future work.

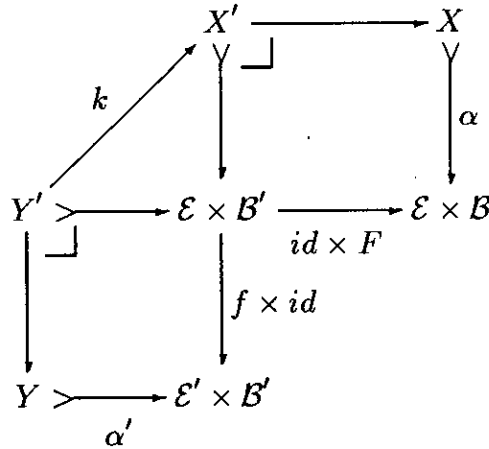
Another approach is to consider the category **Shv**(X) of sheaves on a topological space X [Joh77]. Since **Shv** is a topos, **NShv** is a sound model of linear logic. It is possible that using such a base category we could build a category with objects representing behaviours of Petri nets.

6.2 Modifying the Notion of Refinement

We can define different categories with Petri nets as objects by modifying our definition of morphism.

We now define a category \mathbf{GC}^{co} by analogy with \mathbf{GC} :

- objects of \mathbf{GC}^{co} are precisely the objects of \mathbf{GC} ,
- a morphism in \mathbf{GC}^{co} from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$ is a pair (f, F) of maps in \mathbf{C} such that there exists a morphism k making the triangle in the diagram :

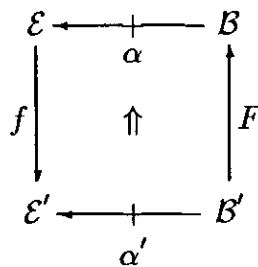


commute,

- and composition is componentwise.

Notation 6.2.1

We depict a morphism in \mathbf{GC}^{co} from $(\mathcal{E} \xleftarrow{\alpha} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} B')$ thus:



Remark 6.2.2 A morphism in \mathbf{GC}^{co} only differs from a morphism in \mathbf{GC} in the direction of the inclusion k , which is reflected in the use of \uparrow as opposed to \downarrow in the diagram of Notation 6.2.1.

Proposition 6.2.3 \mathbf{GC}^{co} is isomorphic to \mathbf{GC}^{op} .

Proof: Consider the assignment ι given as follows:

$$\iota: (\mathcal{E} \xleftarrow{\alpha} \mathcal{B}) \mapsto (\mathcal{B} \xleftarrow{\alpha^{\text{op}}} \mathcal{E}) \quad \text{and}$$

$$\iota: (f, F) \mapsto (F, f).$$

We shall show that ι is an isomorphism of categories.

Because we are introducing contravariance at two points, checking that ι is a functor is a little delicate. ι evidently maps identities to identities. To see that ι respects composition, observe that the composition in \mathbf{GC}^{op} of the morphisms (f, F) from $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')$ and (g, G) from $(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')$ to $(\mathcal{E}'' \xleftarrow{\alpha''} \mathcal{B}'')$ is the morphism $(gf, FG): (\mathcal{E} \xleftarrow{\alpha} \mathcal{B}) \rightarrow (\mathcal{E}'' \xleftarrow{\alpha''} \mathcal{B}'')$. Thus

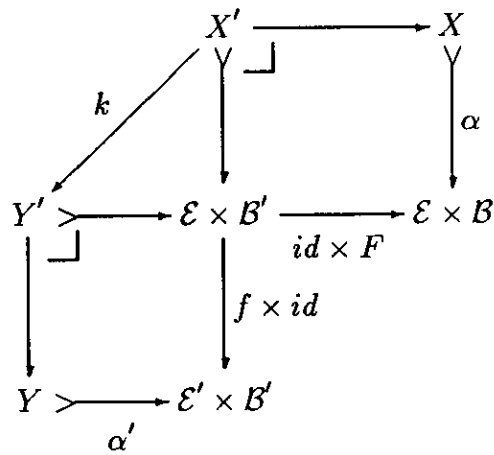
$$\iota(f, F); \iota(g, G) = (F, f); (G, g) = (FG, gf) = \iota(gf, FG),$$

and ι is a functor. It is evidently bijective on objects, and we now show that it is bijective on morphisms.

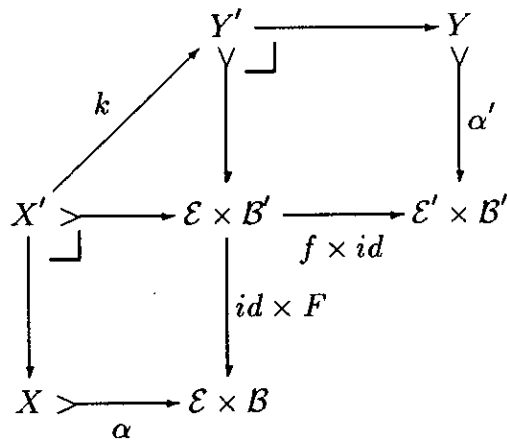
We shall show that (F, f) is a morphism in \mathbf{GC}^{op} from $(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')$ to $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ if and only if (f, F) is a morphism in \mathbf{GC}^{co} from $(\mathcal{B}' \xleftarrow{(\alpha')^{\text{op}}} \mathcal{E}')$ to $(\mathcal{B} \xleftarrow{\alpha^{\text{op}}} \mathcal{E})$.

Let (f, F) be a morphism in \mathbf{GC}^{op} from $(\mathcal{E}' \xleftarrow{\beta} \mathcal{B}')$ to $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$. Then from the definition of a morphism in \mathbf{GC}^{op} , there exists a k

making the triangle in the following diagram



commute in \mathbf{C} . Reflecting in a line in the page, perpendicular to k , we see that k makes the triangle in the following diagram:



commute in \mathbf{C} .

Let $\alpha: R \hookrightarrow X \times Y$ be a monic in \mathbf{C} . Then there is a monic $\alpha^{op}: R^{op} \hookrightarrow Y \times X$ induced by α in the evident way, and $R^{op} \cong R$ in \mathbf{C} . Hence the map k^{op} corresponding to k in the diagram above makes the triangle

in the following diagram:

$$\begin{array}{ccccc}
 & & Y'^{op} & \xrightarrow{\quad} & Y^{op} \\
 & & \downarrow & \lrcorner & \downarrow \\
 & & & & (\alpha')^{op} \\
 & k^{op} \nearrow & & & \\
 X'^{op} & \xrightarrow{\quad} & B' \times \mathcal{E} & \xrightarrow{id \times f} & B' \times \mathcal{E}' \\
 \downarrow & \lrcorner & \downarrow & & \\
 & & F \times id & & \\
 X^{op} & \xrightarrow{\alpha^{op}} & B \times \mathcal{E} & &
 \end{array}$$

commute.

The above diagram commutes in \mathbf{C} if and only if there is a morphism (F, f) in \mathbf{GC}^{co} from $(B' \xleftarrow{(\alpha')^{op}} \mathcal{E}')$ to $(B \xleftarrow{\alpha^{op}} \mathcal{E})$.

Thus (F, f) is a morphism in \mathbf{GC}^{co} from $(B' \xleftarrow{(\alpha')^{op}} \mathcal{E}')$ to $(B \xleftarrow{\alpha^{op}} \mathcal{E})$ if and only if (f, F) is a morphism in \mathbf{GC}^{op} from $(\mathcal{E}' \xleftarrow{\alpha'} B')$ to $(\mathcal{E} \xleftarrow{\alpha} B)$.

Hence ι is an isomorphism of categories. □

We now show that whenever \mathbf{GC} is a sound model of classical linear logic, both \mathbf{GC}^{op} and \mathbf{GC}^{co} are sound models of classical linear logic. This requires a simple lemma:

Lemma 6.2.4 Let \mathbf{GC} be a sound model of classical linear logic. Let A and B be objects of \mathbf{GC} . There is a morphism in \mathbf{GC} from $A^\perp \# B^\perp$ to $(A \otimes B)^\perp$. Further, there is a morphism in \mathbf{GC} from $(A \otimes B)^\perp$ to $A^\perp \# B^\perp$.

Further, there are morphisms in \mathbf{GC} from $A^\perp \otimes B^\perp$ to $(A \# B)^\perp$ and from $(A \# B)^\perp$ to $A^\perp \otimes B^\perp$.

Proof: Since \mathbf{GC} is a sound model of classical linear logic, it suffices to show that $A^\perp \# B^\perp$ and $(A \otimes B)^\perp$ are inter-derivable in classical linear logic, and secondly that $A^\perp \otimes B^\perp$ and $(A \# B)^\perp$ are inter-derivable in classical linear logic. We here give only one of the derivations: the others are similar.

$$\begin{array}{c}
\frac{}{A \vdash A^{\perp\perp}} \text{ (negR)} \quad \frac{}{B \vdash B^{\perp\perp}} \text{ (negR)} \quad \frac{}{A^{\perp} \vdash A^{\perp}} \text{ (Id)} \quad \frac{}{B^{\perp} \vdash B^{\perp}} \text{ (Id)} \\
\hline
\frac{}{A, B \vdash A^{\perp\perp} \otimes B^{\perp\perp}} \text{ (\otimes R)} \quad \frac{}{A^{\perp} \# B^{\perp} \vdash A^{\perp}, B^{\perp}} \text{ (\#L)} \\
\hline
\frac{}{A^{\perp\perp} \otimes B^{\perp\perp}, A^{\perp} \# B^{\perp} \vdash} \text{ (\otimes L)} \quad \frac{}{A^{\perp\perp} \otimes B^{\perp\perp}, A^{\perp} \# B^{\perp} \vdash} \text{ (varL)} \\
\hline
\frac{}{A, B, A^{\perp} \# B^{\perp} \vdash} \text{ (\otimes L)} \quad \frac{}{(A \otimes B), A^{\perp} \# B^{\perp} \vdash} \text{ (varR)} \\
\hline
\frac{}{A^{\perp} \# B^{\perp} \vdash (A \otimes B)^{\perp}} \text{ (Cut)}
\end{array}$$

□

Proposition 6.2.5 If \mathbf{GC} is a sound model of *classical* linear logic, then \mathbf{GC}^{op} is a sound model of classical linear logic.

Proof: We have a sound interpretation of classical linear logic in \mathbf{GC} given by the function $| - |$ from formulae of linear logic to objects of \mathbf{GC} .

We now show that the function $\| - \|$ from formulae of linear logic to objects of \mathbf{GC}^{op} given by

$$\|F\| = |F^{\perp}|$$

gives a sound interpretation of classical linear logic in \mathbf{GC}^{op} . Thus we show that whenever the sequent $\Gamma \vdash \Delta$ is provable in classical linear logic, there is a morphism in \mathbf{GC}^{op} from $\|\Gamma\|$ to $\|\Delta\|$.

Suppose $G_0, \dots, G_n \vdash D_0, \dots, D_m$ is a sequent of classical linear logic. Then there is a morphism in \mathbf{GC} from $|G_0 \otimes \dots \otimes G_n|$ to $|D_0 \# \dots \# D_m|$. Further, we can derive the sequent $D_0^{\perp}, \dots, D_m^{\perp} \vdash G_0^{\perp}, \dots, G_n^{\perp}$ in classical linear logic by repeated application of the rules (varL) and (varR).

Since \mathbf{GC} is a sound model of classical linear logic, there is a morphism in \mathbf{GC} from $|D_0^\perp \otimes \cdots \otimes D_m^\perp|$ to $|G_0^\perp \# \cdots \# G_n^\perp|$. Hence there is a morphism in \mathbf{GC}^{op} from $|G_0^\perp \# \cdots \# G_n^\perp|$ to $|D_0^\perp \otimes \cdots \otimes D_m^\perp|$.

Lemma 6.2.4 suffices to show that there is a morphism in \mathbf{GC} from $|G_0^\perp \# \cdots \# G_n^\perp|$ to $|(G_0 \otimes \cdots \otimes G_n)^\perp|$, and also that there is a morphism in \mathbf{GC} from $|(D_0 \# \cdots \# D_m)^\perp|$ to $|D_0^\perp \otimes \cdots \otimes D_m^\perp|$.

Thus we have the following composite morphism in \mathbf{GC}^{op} :

$$|(G_0 \otimes \cdots \otimes G_n)^\perp| \rightarrow |G_0^\perp \# \cdots \# G_n^\perp| \rightarrow |D_0^\perp \otimes \cdots \otimes D_m^\perp| \rightarrow |(D_0 \# \cdots \# D_m)^\perp|.$$

Thus whenever the sequent $G_0, \dots, G_n \vdash D_0, \dots, D_m$ is provable in classical linear logic, there is a morphism in \mathbf{GC}^{op} from $\|G_0 \otimes \cdots \otimes G_n\|$ to $\|D_0 \# \cdots \# D_m\|$.

Hence \mathbf{GC}^{op} is a sound model of classical linear logic. \square

Corollary 6.2.6 If \mathbf{GC} is a sound model of classical linear logic, then \mathbf{GC}^{co} is a sound model of classical linear logic.

Remark 6.2.7 *The proof of Proposition 6.2.5 depends on the fact that since \mathbf{GC} is a sound model of classical linear logic, \mathbf{GC} is a $*$ -autonomous category with the functor $*$ given by $(-)^\perp$ [deP89a].*

Remark 6.2.8

\mathbf{GSet} , \mathbf{GSet}^{op} and \mathbf{GSet}^{co} are all sound models of classical linear logic.

Proposition 6.2.9 $\mathbf{GC} \times \mathbf{GC}^{co}$ is a sound model of linear logic.

Proof: The category of small categories with assigned finite products and assigned finite coproducts, with two symmetric monoidal closed structures, one of which is closed, having as morphisms functors which preserve all this structure strictly, is monadic over \mathbf{Cat} . Hence the product in \mathbf{Cat} of any two small categories with such extra structure, coherently acquires that extra structure uniquely.

Let \mathbf{C} be a locally cartesian closed category with disjoint stable co-products. Let $|A|_{\mathbf{GC}}$ denote the interpretation of a linear logic formula A in the category \mathbf{GC} , which is a sound model linear logic.

We know that whenever $\Gamma \vdash_{LL} A$, there is a morphism (f, F) in \mathbf{GC} from $|\Gamma|_{\mathbf{GC}}$ to $|A|_{\mathbf{GC}}$, and also a morphism (F, f) in \mathbf{GC}^{co} from $|\Gamma|_{\mathbf{GC}^{\text{co}}}$ to $|A|_{\mathbf{GC}^{\text{co}}}$.

Since all the relevant structure of \mathbf{GC} and \mathbf{GC}^{co} lifts to $\mathbf{GC} \times \mathbf{GC}^{\text{co}}$, we see that whenever $\Gamma \vdash_{LL} A$, there is a morphism $((f, F), (F, f))$ in $\mathbf{GC} \times \mathbf{GC}^{\text{co}}$ from $|\Gamma|_{\mathbf{GC} \times \mathbf{GC}^{\text{co}}}$ to $|A|_{\mathbf{GC} \times \mathbf{GC}^{\text{co}}}$. \square

The category \mathbf{GC}^{co} was obtained by reversing the direction of the inclusion k in the definition of a morphism in \mathbf{GC} . Recall that objects of \mathbf{NC} are pairs of subobjects α^0 and α^1 of objects $\mathcal{E} \times \mathcal{B}$ in \mathbf{C} .

If we reverse the inclusion k for the subobject α^0 and leave it unchanged for the subobject α^1 , we obtain a category which we shall denote $\mathbf{NC}_{\frac{\subseteq}{\supseteq}}$. The object set of $\mathbf{NC}_{\frac{\subseteq}{\supseteq}}$ is the same as the object set of \mathbf{NC} . Morphisms in $\mathbf{NC}_{\frac{\subseteq}{\supseteq}}$ are pairs of maps (f, F) in \mathbf{C} such that

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^0} & \mathcal{B} \\
 f \downarrow & \uparrow & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^0} & \mathcal{B}'
 \end{array}
 & \text{and} &
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^1} & \mathcal{B} \\
 f \downarrow & \downarrow & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^1} & \mathcal{B}'
 \end{array}
 \end{array}$$

Thus a morphism in $\mathbf{NC}_{\frac{\subseteq}{\supseteq}}$ from $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{\alpha'} \mathcal{B}')$ is a pair (f, F) such that (f, F) is both a morphism in \mathbf{GSet} from $(\mathcal{E} \xleftarrow{\alpha^0} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{(\alpha')^0} \mathcal{B}')$, and a morphism in $\mathbf{GSet}^{\text{co}}$ from $(\mathcal{E} \xleftarrow{\alpha^1} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{(\alpha')^1} \mathcal{B}')$.

Composition is again componentwise. It is readily seen that $\mathbf{NC}_{\frac{\subseteq}{\supseteq}}$ is a category.

We can also define a category by reversing the inclusion k for the subobject α^1 and leaving it unchanged for the subobject α^0 . We shall denote this category

$\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$. It has the same object set as \mathbf{NC} , while its morphisms are pairs (f, F) of morphisms in \mathbf{C} such that

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^0} & \mathcal{B} \\
 f \downarrow & & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^0} & \mathcal{B}' \\
 & \Downarrow &
 \end{array}
 & \text{and} &
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^1} & \mathcal{B} \\
 f \downarrow & & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^1} & \mathcal{B}' \\
 & \Uparrow &
 \end{array}
 \end{array}$$

The category obtained by reversing the inclusion on both the pre- and post-conditions we shall call \mathbf{NC}^{∞} . It has the same object set as \mathbf{NC} , while its morphisms are pairs (f, F) of morphisms in \mathbf{C} such that

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^0} & \mathcal{B} \\
 f \downarrow & & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^0} & \mathcal{B}' \\
 & \Uparrow &
 \end{array}
 & \text{and} &
 \begin{array}{ccc}
 \mathcal{E} & \xleftarrow{\alpha^1} & \mathcal{B} \\
 f \downarrow & & \uparrow F \\
 \mathcal{E}' & \xleftarrow{(\alpha')^1} & \mathcal{B}' \\
 & \Uparrow &
 \end{array}
 \end{array}$$

Proposition 6.2.10 Let \mathbf{C} be a category with finite limits and disjoint, stable, finite coproducts. $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ has finite products and coproducts given as in $\mathbf{GC} \times \mathbf{GC}^{\infty}$.

Proof: We identify each object $(\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ of $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ with the object $\langle (\mathcal{E} \xleftarrow{\alpha^0} \mathcal{B}), (\mathcal{B} \xleftarrow{(\alpha^1)^{op}} \mathcal{E}) \rangle$ of $\mathbf{GC} \times \mathbf{GC}^{\infty}$. Then $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ is evidently a full subcategory of $\mathbf{GC} \times \mathbf{GC}^{\infty}$.

Inspection of the evident diagram shows that the product in $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ is the same as that in $\mathbf{GC} \times \mathbf{GC}^{\infty}$.

The terminal object $\mathbf{1} = \langle (1 \xleftarrow{\phi} 0), (0 \xleftarrow{\phi} 1) \rangle$ of $\mathbf{GC} \times \mathbf{GC}^{\infty}$ is an object of $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$, and (since $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ is a *full* subcategory of $\mathbf{GC} \times \mathbf{GC}^{\infty}$) is the terminal object of $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$.

Thus the product in $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ of the nets $N_0 = \langle \mathcal{E}_0, \mathcal{B}_0, \alpha_0, \alpha_1 \rangle$ and $N_1 = \langle \mathcal{E}_1, \mathcal{B}_1, \beta_0, \beta_1 \rangle$ is the net

$$N_0 \times N_1 = \langle \mathcal{E}_0 \times \mathcal{E}_1, \mathcal{B}_0 + \mathcal{B}_1, \alpha_0 \wedge \beta_0, (\alpha_1 \oplus \beta_1)^{op} \rangle.$$

We can show in a similar way that the coproduct in $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ coincides with the coproduct in $\mathbf{GC} \times \mathbf{GC}^{co}$, and so the coproduct of nets N_0 and N_1 in $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ is

$$N_0 + N_1 = \langle \mathcal{E}_0 + \mathcal{E}_1, \mathcal{B}_0 \times \mathcal{B}_1, \alpha_0 \oplus \beta_0, (\alpha_1 \wedge \beta_1)^{op} \rangle.$$

Hence $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ has finite products and coproducts, and they are given as in $\mathbf{GC} \times \mathbf{GC}^{co}$. \square

Remark 6.2.11 *We expect that the symmetric monoidal closed structure of $\mathbf{GC} \times \mathbf{GC}^{op}$ restricts to the subcategories $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ and $\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$, in which case $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ and $\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$ would be sound models of linear logic.*

Remark 6.2.12 *We do not as yet have any way of constructing either $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ or $\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$ as a limit in \mathbf{Cat} . This greatly increases the difficulty of studying their properties (as the above proposition and remark illustrate), and underlines the value of Theorem 5.1.1, which expresses \mathbf{NC} as a kernel pair in \mathbf{Cat} .*

Remark 6.2.13 *The isomorphism $\iota: \mathbf{GC} \rightarrow \mathbf{GC}^{co}$ induces an isomorphism $\tilde{\iota}$ from $\mathbf{NC}_{\underline{\mathbb{C}}}^{\subseteq}$ to $\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$, given by*

$$\tilde{\iota}(\langle (\mathcal{E} \xleftarrow{\alpha^0} \mathcal{B}), (\mathcal{B} \xleftarrow{(\alpha^1)^{op}} \mathcal{E}) \rangle) = \langle (\mathcal{B} \xleftarrow{(\alpha^0)^{op}} \mathcal{E}), (\mathcal{E} \xleftarrow{\alpha^1} \mathcal{B}) \rangle \quad \text{and}$$

$$\tilde{\iota}(f, F) = (F, f).$$

Hence $\mathbf{NC}_{\underline{\mathbb{C}}}^{\supseteq}$ also has finite products and coproducts whenever \mathbf{C} satisfies the conditions of Proposition 6.2.10, and they are given as in $\mathbf{GC}^{co} \times \mathbf{GC}$.

6.2.1 Interpreting the Morphisms in $\mathbf{NSet}_{\subseteq}^{\subseteq}$, $\mathbf{NSet}_{\subseteq}^{\supseteq}$ and \mathbf{NSet}^{co}

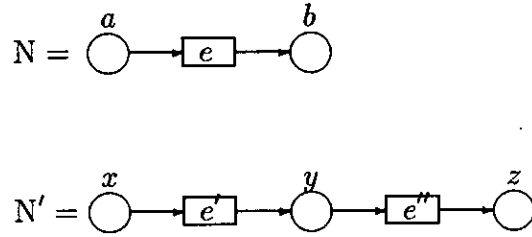
As before, we identify a Petri net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ with the object $\langle (\mathcal{E} \xleftarrow{\alpha^0} \mathcal{B}), (\mathcal{B} \xleftarrow{\alpha^1} \mathcal{E}) \rangle$ of $\mathbf{NSet}_{\subseteq}^{\subseteq}$ by putting $\alpha^0 = pre$ and $\alpha^1 = post^{op}$.

Let (f, F) be a morphism in $\mathbf{NSet}_{\subseteq}^{\subseteq}$ from $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ to $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$. Then for each multiset A over \mathcal{E} we have

$$F^{-1}(pre(A)) \subseteq_m pre'(fA) \quad \text{and} \quad F^{-1}(post(A)) \supseteq_m post'(fA).$$

Again, it is reasonable to interpret the morphisms between nets in $\mathbf{NSet}_{\subseteq}^{\subseteq}$ as expressing refinement.

To see why we might want to modify our notion of refinement map, consider the following two nets:



A little consideration shows that there is no morphism from $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ to $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$ in \mathbf{NSet} .

If we consider the pair of functions $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B}' \rightarrow \mathcal{B}$ given by

$$f(e) = e', \quad F(x) = a \quad \text{and} \quad F(y) = F(z) = b,$$

then we observe that

$$F^{-1}(pre(e)) = F^{-1}(\{a\}) = \{x\} = pre(fe) \quad \text{and}$$

$$F^{-1}(post(e)) = F^{-1}(\{b\}) = \{y, z\} \supseteq post(fe).$$

Thus (f, F) is a morphism of nets in the category $\mathbf{NSet}_{\subseteq}^{\subseteq}$. Notice that there is a morphism between N and N' in Winskel's category \mathbf{Net} , given by

$$\eta(e) = e' \quad \text{and} \quad \beta(c, c') = \begin{cases} 1 & \text{if } (c, c') = (a, x) \\ 1 & \text{if } (c, c') = (b, y) \\ 0 & \text{otherwise.} \end{cases}$$

Also, there is a morphism from N to N' in each of the categories discussed by Meseguer and Montanari which has nets as objects. Most existing concepts of refinement or specification would give rise to a map either from N to N' or from N' to N . This suggests that the category $\mathbf{NSet}_{\frac{\subseteq}{\supseteq}}$ may prove more appropriate to the study of Petri nets than \mathbf{NSet} . Here, however, we shall concentrate on the properties of \mathbf{NSet} .

Let (f, F) be a morphism in $\mathbf{NSet}_{\frac{\subseteq}{\supseteq}}$ from $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ to $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$. The pre-conditions of an event fe in the image net N' must include the pre-conditions of e , relative to F , while the post-conditions of fe are contained in the post-conditions of e , relative to F . Hence we summarise morphisms in $\mathbf{NSet}_{\frac{\subseteq}{\supseteq}}$ as showing how “more gives less”. In this sense the process of going from an event e in N to event fe in N' is analogous to weakening in Hoare logic [Apt81].

We express the notion of refinement given in Definition 4.4.8 by the slogan “more gives more”. This notion, and its converse, “less gives less” (which describes morphisms in $\mathbf{NSet}^{\text{co}}$) are found in the literature of algebraic specification, where the latter construction corresponds to a *reduct*, see [EKMP82] and [Ehr81].

For each of the categories $\mathbf{NSet}_{\frac{\subseteq}{\supseteq}}$, $\mathbf{NSet}_{\frac{\supseteq}{\subseteq}}$ and $\mathbf{NSet}^{\text{co}}$, we have an analogue of Proposition 4.4.10.

Proposition 6.2.14 Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ and $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$ be safe nets, and let f be function from \mathcal{E} to \mathcal{E}' and F a function from \mathcal{B}' to \mathcal{B} . Whenever a marking M_0 can evolve in N by a multiset of events A to a marking M_1 , then there exist markings S_0 and S_1 of N' such that S_0 can evolve by the multiset of events $f(A)$ to S_1 . Further,

1. if (f, F) is a morphism in $\mathbf{NSet}_{\frac{\subseteq}{\supseteq}}$, then $F^{-1}(M_0) \subseteq S_0$ and $S_1 \subseteq F^{-1}(M_1)$.

In particular, N' can evolve under the multiset of events fA from the marking $F^{-1}(M_0)$ to a marking S_2 , where $S_2 \subseteq F^{-1}(M_1)$.

2. If (f, F) is a morphism in $\mathbf{NSet}_{\frac{\supseteq}{\subseteq}}$, then $S_0 \subseteq F^{-1}(M_0)$ and $F^{-1}(M_1) \subseteq S_1$.

3. If (f, F) is a morphism in $\mathbf{NSet}^{\text{co}}$, then $S_0 \subseteq F^{-1}(M_0)$ and $S_1 \subseteq F^{-1}(M_1)$. In particular, N' can evolve under the multiset of events fA from the marking $F^{-1}(M_0)$ to a marking S_2 , where $S_2 \subseteq F^{-1}(M_1)$.

Proof: We define S_0 and S_1 as follows:

$$S_0 = F^{-1}(M_0) + \text{pre}(fA) - F^{-1}(\text{pre}(A)) \quad \text{and}$$

$$S_1 = F^{-1}(M_1) + \text{post}(fA) - F^{-1}(\text{post}(A)).$$

The result then follows immediately from the definition of morphisms in each of the categories in question. \square

Remark 6.2.15 *As was mentioned after Proposition 4.4.10, it should be possible to extend the above proposition to the case where N and N' are any elementary nets.*

6.3 The Category $\mathbf{NC}^{\#}$

\odot Each of the categories \mathbf{NC} , $\mathbf{NC}_{\subseteq}^{\subseteq}$, $\mathbf{NC}_{\supseteq}^{\supseteq}$ and \mathbf{NC}^{co} has a subcategory $\mathbf{NC}^{\#}$ obtained by requiring that both inclusions k be identities. Observe that $\mathbf{NC}^{\#}$ has the same object set as any of \mathbf{NC} , $\mathbf{NC}_{\subseteq}^{\subseteq}$, $\mathbf{NC}_{\supseteq}^{\supseteq}$ and \mathbf{NC}^{co} . A morphism in $\mathbf{NC}^{\#}$ from $(\mathcal{E} \xleftarrow{\alpha} \mathbb{1} \mathcal{B})$ to $(\mathcal{E}' \xleftarrow{\alpha'} \mathbb{1} \mathcal{B}')$ is a pair of functions $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B}' \rightarrow \mathcal{B}$ such that for all $e \in \mathcal{E}$,

$$F^{-1}(\alpha^0(e)) = (\alpha')^0(fe) \quad \text{and} \quad F^{-1}(\alpha^1(e)) = (\alpha')^1(fe).$$

We have as yet no easy characterisation of $\mathbf{NC}^{\#}$ as a limit in \mathbf{Cat} , in the manner of Theorem 5.1.1, and so the proof that $\mathbf{NC}^{\#}$ is a sound model of linear logic is extremely tedious. We are required to prove that, for a suitable choice of base category \mathbf{C} , $\mathbf{NC}^{\#}$ has finite products and coproducts, a symmetric monoidal closed structure \otimes , and a monoidal structure $\#$. This we do by showing that all the relevant structure of \mathbf{NC} restricts to $\mathbf{NC}^{\#}$. $\mathbf{NC}^{\#}$ and \mathbf{NC} have the same

terminal and initial objects. Since $\mathbf{NC}^=$ has the same object set as \mathbf{NC} , we can again define the negation of an object A of $\mathbf{NC}^=$ to be the object $[A, \perp]$. We now prove an instructive lemma to which we shall appeal frequently.

Lemma 6.3.1 Every isomorphism in \mathbf{NC} is a morphism in $\mathbf{NC}^=$.

Proof: In the proof of Proposition 4.4.13, we saw that whenever (f, F) is an isomorphism in \mathbf{NC} , the inclusion k in the diagram

$$\begin{array}{ccccc}
 & & X' & \xrightarrow{\quad} & X \\
 & & \downarrow \lrcorner & & \downarrow \alpha \\
 & k \nearrow & & & \\
 Y' & \xrightarrow{\quad} & \mathcal{E} \times B' & \xrightarrow{id \times F} & \mathcal{E} \times B \\
 \downarrow \lrcorner & & \downarrow f \times id & & \\
 Y & \xrightarrow{\quad} & \mathcal{E}' \times B' & & \\
 & \alpha' \searrow & & &
 \end{array}$$

is the identity morphism. □

Remark 6.3.2 A map (f, F) is an isomorphism in \mathbf{NC} if and only if it is an isomorphism in $\mathbf{NC}^=$.

Proposition 6.3.3 The symmetric monoidal structures \otimes and $\#$ of \mathbf{NC} restrict to symmetric monoidal structures \otimes and $\#$ in $\mathbf{NC}^=$.

The units of \otimes and $\#$ are the same in the two categories.

Proof:

$\langle \mathbf{NC}, \otimes, I, a, \lambda, \sigma \rangle$ is a symmetric monoidal category. $\mathbf{NC}^=$ has the same object set as \mathbf{NC} , and so it is evident that \otimes restricts to a functor $\mathbf{NC} \times \mathbf{NC} \rightarrow \mathbf{NC}$, and I is an object of $\mathbf{NC}^=$. By Lemma 6.3.1, the isomorphisms $a_{x,y,z}$, λ_x and $\sigma_{x,y}$ in \mathbf{NC} are isomorphisms in $\mathbf{NC}^=$. Thus the coherence conditions satisfied by $a_{x,y,z}$, λ_x and $\sigma_{x,y}$ in \mathbf{NC} are also satisfied in $\mathbf{NC}^=$, and $\langle \mathbf{NC}^=, \otimes, I, a, \lambda, \sigma \rangle$ is a symmetric monoidal category. □

Proposition 6.3.4

The symmetric monoidal closed structure of \mathbf{NC} restricts to $\mathbf{NC}^=$.

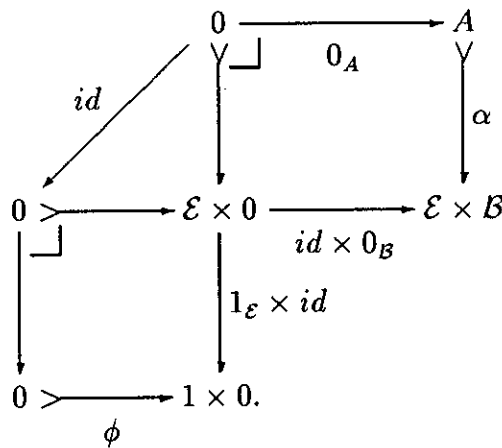
Proof: A lengthy diagram chase exactly following de Paiva's proof that the internal hom in \mathbf{GC} is adjoint to the monoidal structure, \otimes .

□

Proposition 6.3.5 $\mathbf{1}$ is terminal in $\mathbf{NC}^=$, and $\mathbf{0}$ initial.

Proof: We write $1_{\mathcal{E}}$ for the unique map from \mathcal{E} into $\mathbf{1}$, and $0_{\mathcal{B}}$ for the unique map from $\mathbf{0}$ into \mathcal{B} .

The unique map in \mathbf{NC} from an object $(\mathcal{E} \xleftarrow{\alpha} \mathbf{1} \xrightarrow{\beta} \mathcal{B})$ of $\mathbf{NC}^=$ into $\mathbf{1}$ is $(1_{\mathcal{E}}, 0_{\mathcal{B}})$, and the following diagram commutes:



Thus $(1_{\mathcal{E}}, 0_{\mathcal{B}})$ is a morphism in $\mathbf{NC}^=$.

A similar argument shows that the unique morphism $(0_{\mathcal{E}}, 1_{\mathcal{B}})$ in \mathbf{NC} from $\mathbf{0}$ to $(\mathcal{E} \xleftarrow{\alpha} \mathbf{1} \xrightarrow{\beta} \mathcal{B})$ is a morphism in $\mathbf{NC}^=$. □

We now prove three useful lemmata about the interaction between pullbacks and coproducts in a category \mathbf{C} which has disjoint, stable, finite coproducts and finite limits.

Lemma 6.3.6 Let \mathbf{C} be a locally cartesian closed category with disjoint, stable, finite coproducts, and for $i = 0, 1$ let α_i be monomorphisms as in the diagram.

Then for $i = 0, 1$, the following diagram:

$$\begin{array}{ccc}
 A_i & \xrightarrow{in_i} & A_0 + A_1 \\
 \alpha_i \downarrow & & \downarrow \alpha_0 + \alpha_1 \\
 B_i & \xrightarrow{in_i} & B_0 + B_1
 \end{array}$$

is a pullback diagram in \mathbf{C} .

Proof: We pull back $\alpha_0 + \alpha_1$ along the injections in_0 and in_1 to obtain C_0 and C_1 respectively. Since A_0 makes square (1) commute, C_0 is a subobject of $A_0 + A_1$, being the pullback of a monomorphism, and it follows that square (2) is a pullback square thus:

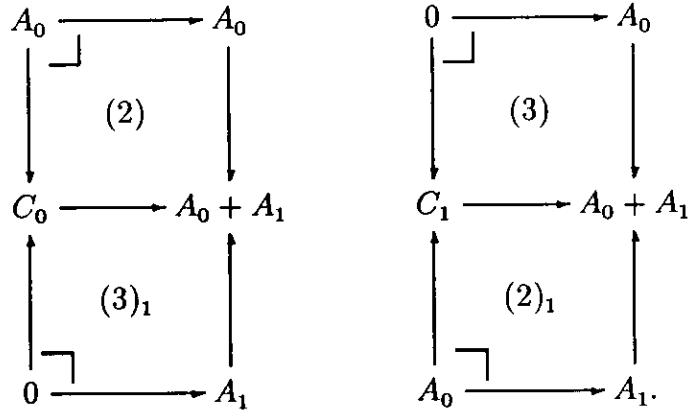
$$\begin{array}{ccccc}
 A_0 & \xrightarrow{\quad} & C_0 & \xrightarrow{\quad} & B_0 \\
 \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow in_0 \\
 & (2) & & (1) & \\
 A_0 & \xrightarrow{\quad} & A_0 + A_1 & \xrightarrow{\quad} & B_0 + B_1 \\
 \uparrow in_0 & & \uparrow & \alpha_0 + \alpha_2 & \uparrow in_1 \\
 (3) & & & & \\
 0 & \xrightarrow{\quad} & C_1 & \xrightarrow{\quad} & B_1
 \end{array}$$

To see that square (3) is a pullback square, observe that the following diagram commutes:

$$\begin{array}{ccccc}
 A_0 & \xrightarrow{\alpha_0} & B_0 & \xrightarrow{in_0} & B_0 + B_1 \\
 \uparrow & & \uparrow & & \uparrow \\
 (4) & & (5) & & \\
 0 & \xrightarrow{\quad} & 0 & \xrightarrow{\quad} & B_1
 \end{array}$$

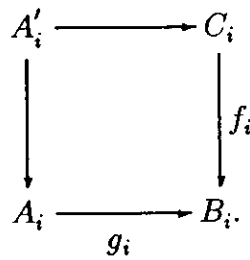
Since α_0 is a monomorphism, square (4) is a pullback. The fact that square (5) pulls back to 0 follows from the definition of disjoint co-products. It follows that square (3) is a pullback.

Applying a similar argument for A_1 , we obtain the diagrams:

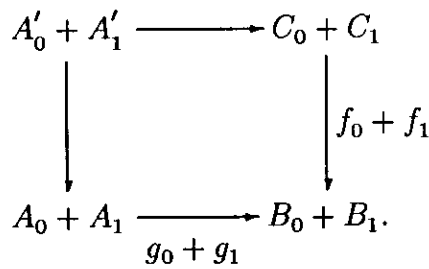


By stability of coproducts, for $i = 0, 1$ we have $C_i \cong (A_i + 0)$ and hence $C_i \cong A_i$. □

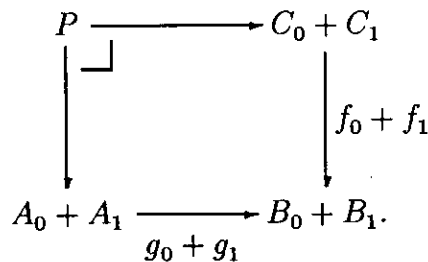
Lemma 6.3.7 Let \mathbf{C} be a category with stable, disjoint finite coproducts and finite limits. For $i = 0, 1$ let the following two diagrams be pullbacks in \mathbf{C} :



Then the following is a pullback in \mathbf{C} :



Proof: Let P be the pullback of $f_0 + f_1$ along $g_0 + g_1$, thus:



Then by stability of coproducts, $P \cong P_0 + P_1$, where the P_i are given by pullbacks as follows:

$$\begin{array}{ccc}
 P_i & \longrightarrow & C_i \\
 \downarrow & \lrcorner & \downarrow \text{in}_i \\
 P & \longrightarrow & C_0 + C_1.
 \end{array}$$

Thus we have the following two diagrams:

$$\begin{array}{ccc}
 \begin{array}{ccc}
 P_0 & \longrightarrow & C_0 \\
 \downarrow \text{in}_0 & \lrcorner & \downarrow \text{in}_0 \\
 P & \longrightarrow & C_0 + C_1 \\
 \downarrow & \lrcorner & \downarrow f_0 + f_1 \\
 A_0 + A_1 & \xrightarrow{g_0 + g_1} & B_0 + B_1
 \end{array} & \text{and} & \begin{array}{ccc}
 A'_0 & \longrightarrow & C_0 \\
 \downarrow & \lrcorner & \downarrow f_0 \\
 A_0 & \longrightarrow & B_0 \\
 \downarrow & \lrcorner & \downarrow \text{in}_0 \\
 A_0 + A_1 & \xrightarrow{g_0 + g_1} & B_0 + B_1.
 \end{array} \\
 (1) & & (3) \\
 (2) & & (4)
 \end{array}$$

Since (1), (2), (3) and (4) are all pullbacks (using Lemma 6.3.6 for (4)), both rectangles are pullbacks.

Hence $P_0 \cong A'_0$, and since we can show similarly that $P_1 \cong A'_1$, we have $P \cong A'_0 + A'_1$. \square

Notation 6.3.8 In the following lemma and proposition we shall overload the notation $[-, -]$. Thus if for $i = 0, 1$, $g_i: A_i \rightarrow B$, then the morphism $[g_0, g_1]: A_0 + A_1 \rightarrow B$ is the universal arrow of the coproduct diagram for $A_0 + A_1$. This need not be confused with the notation for the internal hom, which is applied to objects as opposed to morphisms.

Lemma 6.3.9 Let \mathbf{C} be a category with stable, disjoint, finite coproducts and finite limits. Let the following two diagrams be pullbacks in \mathbf{C} for $i = 0, 1$:

$$\begin{array}{ccc}
 A'_i & \longrightarrow & C \\
 \downarrow & \lrcorner & \downarrow f \\
 A_i & \xrightarrow{g_i} & B.
 \end{array}$$

Then the following is a pullback in \mathbf{C} :

$$\begin{array}{ccc}
 A'_0 + A'_1 & \xrightarrow{\quad} & C \\
 \downarrow & \lrcorner & \downarrow f \\
 A_0 + A_1 & \xrightarrow{\quad} & B. \\
 & [g_0, g_1] &
 \end{array}$$

Proof: Let P be the pullback of $[g_0, g_1]$ thus:

$$\begin{array}{ccc}
 P & \xrightarrow{\quad} & C \\
 \downarrow & \lrcorner & \downarrow f \\
 A_0 + A_1 & \xrightarrow{\quad} & B. \\
 & [g_0, g_1] &
 \end{array}$$

By stability of coproducts we have $P \cong P_0 + P_1$, where the P_i are given by pullbacks as follows:

$$\begin{array}{ccc}
 P_i & \xrightarrow{\quad} & P \\
 \downarrow & \lrcorner & \downarrow \\
 A_i & \xrightarrow{\quad} & A_0 + A_1.
 \end{array}$$

Thus we have the following two diagrams:

$$\begin{array}{ccccc}
 P_0 & \xrightarrow{\quad} & P & \xrightarrow{\quad} & C \\
 \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow f \\
 A_0 & \xrightarrow{\quad} & A_0 + A_1 & \xrightarrow{\quad} & B \\
 & in_0 & & [g_0, g_1] &
 \end{array}$$

(1) (2)

and

$$\begin{array}{ccc}
 A'_0 & \xrightarrow{\quad} & C \\
 \downarrow & \lrcorner & \downarrow f \\
 A_0 & \xrightarrow{\quad} & B. \\
 & g_0 &
 \end{array}$$

(3)

Since (1), (2) and (3) are each pullbacks, it follows that $P_0 \cong A'_0$.

Similarly, $P_1 \cong A'_1$ and we have $P \cong A'_0 + A'_1$. □

Notation 6.3.10 Throughout the proof of the next proposition we shall draw one diagram \mathcal{D} in \mathbf{C} and write α , α_0 and α_1 as appropriate, to stand for the pair of diagrams $\langle \mathcal{D}_0, \mathcal{D}_1 \rangle$ in \mathbf{C} where \mathcal{D}_i is the diagram \mathcal{D} with α^i , α_0^i and α_1^i substituted for α , α_0 and α_1 respectively.

Proposition 6.3.11 \mathbf{NC}^\equiv has finite products given as in \mathbf{NC} .

Proof: Throughout this proof, we shall for convenience suppress the evident structural isomorphisms.

By Proposition 6.3.5, \mathbf{NC}^\equiv has terminal object as in \mathbf{NC} . Thus it suffices to show that \mathbf{NC}^\equiv has binary products given as in \mathbf{NC} . Specifically, we must show that for objects N_0 and N_1 of \mathbf{NC}^\equiv , the projection $(\pi_i, in_i): N_0 \times N_1 \rightarrow N_i$ is a morphism in \mathbf{NC}^\equiv . Moreover, we must show that given an object $N = (\mathcal{E} \xleftarrow{\alpha} \mathcal{B})$ of \mathbf{NC}^\equiv , whenever we have morphisms $(f_i, F_i): N \rightarrow N_i$ in \mathbf{NC}^\equiv for $i = 0, 1$, then the universal arrow $((f_0, f_1), [F_0, F_1])$ from N to $N_0 + N_1$ in \mathbf{NC} is a morphism in \mathbf{NC}^\equiv . By Lemma 6.3.6, the following diagram :

$$\begin{array}{ccc}
 A_0 \times \mathcal{E}_1 & \xrightarrow{in_0} & (A_0 \times \mathcal{E}_1) + (\mathcal{E}_0 \times A_1) \\
 \alpha_0 \times id \downarrow & \lrcorner & \downarrow (\alpha_0 \times id) + (id \times \alpha_1) \\
 \mathcal{E}_0 \times \mathcal{B}_0 \times \mathcal{E}_1 & \xrightarrow{in_0} & (\mathcal{E}_0 \times \mathcal{B}_0 \times \mathcal{E}_1) + (\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_1)
 \end{array}$$

is a pullback diagram in \mathbf{C} .

It is evident that the pullback of α_0 along the projection π_{01} from $\mathcal{E}_0 \times \mathcal{B}_0 \times \mathcal{E}_1$ to $\mathcal{E}_0 \times \mathcal{B}_0$ is likewise $\alpha_0 \times id$. We regard the projection π_{01} as the product $\pi_0 \times id$ from $\mathcal{E}_0 \times \mathcal{E}_1 \times \mathcal{B}_0$, and thus have an identity

morphism making the triangle of the following diagram:

$$\begin{array}{ccccc}
 & A_0 \times \mathcal{E}_1 & \xrightarrow{\quad} & (A_0 \times \mathcal{E}_1) + (\mathcal{E}_0 \times A_1) & \\
 & \searrow^{id} & \downarrow \lrcorner & \downarrow \lrcorner & \\
 & & \alpha_0 \times id & (\alpha_0 \times id) + (id \times \alpha_1) & \\
 A_0 \times \mathcal{E}_1 & \xrightarrow{\quad} & \mathcal{E}_0 \times B_0 \times \mathcal{E}_1 & \xrightarrow{id \times in_0} & (\mathcal{E}_0 \times B_0 \times \mathcal{E}_1) + (\mathcal{E}_0 \times \mathcal{E}_1 \times B_1) \\
 \downarrow & \lrcorner & \downarrow \lrcorner & & \\
 A_0 & \xrightarrow{\alpha_0} & \mathcal{E}_0 \times B_0 & &
 \end{array}$$

commute.

Thus the projection (π_0, in_0) from $N_0 \times N_1$ to N_0 is a morphism of $\mathbf{NC}^=$.

A similar argument shows that the projection (π_1, in_1) from $N_0 \times N_1$ to N_1 is a morphism of $\mathbf{NC}^=$.

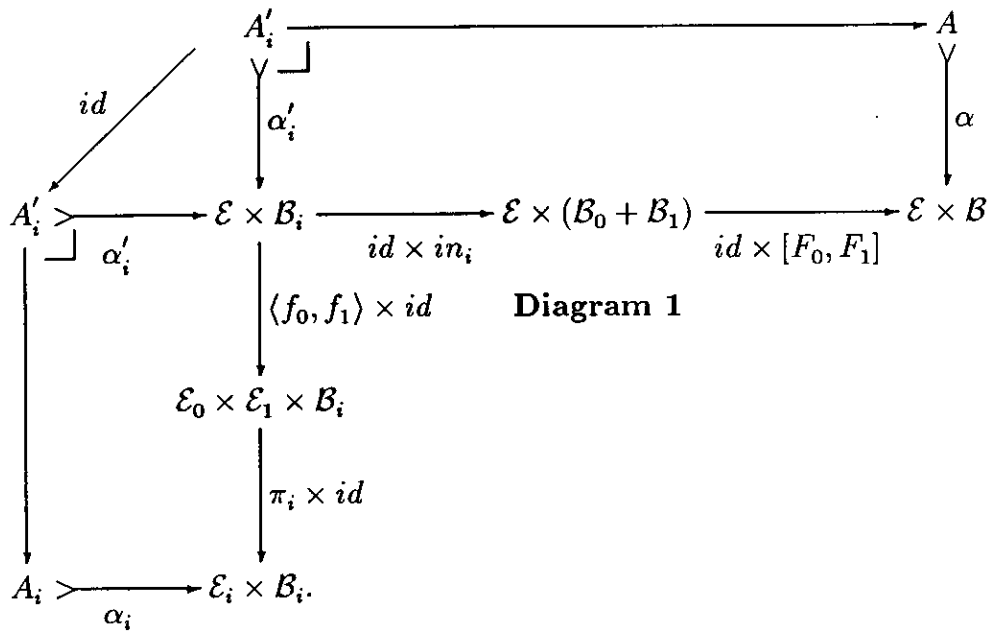
The following proof that $(\langle f_0, f_1 \rangle, [F_0, F_1])$ is the required universal arrow from N to $N_0 + N_1$ is tedious and routine.

Since

$$(\langle f_0, f_1 \rangle \times id); (\pi_i \times id) = f_i \quad \text{and} \quad (id \times in_i); (id \times [F_0, F_1]) = F_i,$$

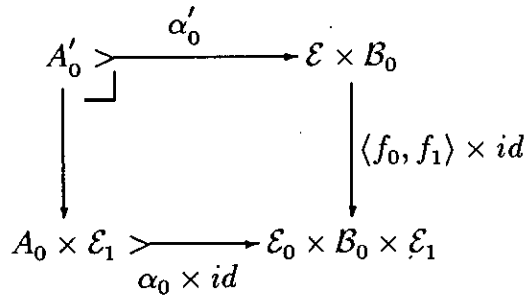
and for $i = 0, 1$ (f_i, F_i) is a morphism of $\mathbf{NC}^=$, the identity morphism

on A_i makes the triangles in the following pair of diagrams in \mathbf{C} :

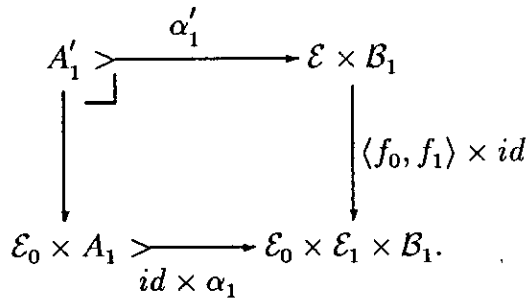


commute.

It follows immediately that the following pairs of diagrams:



and



are pullbacks, and by Lemma 6.3.7, we have the following pullback in \mathbf{C} :

$$\begin{array}{ccc}
 A'_0 + A'_1 & \xrightarrow{\alpha'_0 + \alpha'_1} & \mathcal{E} \times (B_0 + B_1) \\
 \downarrow & \lrcorner & \downarrow \langle f_0, f_1 \rangle \times id \\
 (A_0 \times \mathcal{E}_1) + (\mathcal{E}_0 \times A_1) & \xrightarrow{(\alpha_0 \times id) + (\alpha_1 \times id)} & (\mathcal{E}_0 \times \mathcal{E}_1) \times (B_0 + B_1).
 \end{array} \quad (2)$$

Further, the square:

$$\begin{array}{ccc}
 A'_0 + A'_1 & \xrightarrow{\quad} & A \\
 \downarrow \alpha'_0 + \alpha'_1 & \lrcorner & \downarrow \alpha \\
 (\mathcal{E} \times B_0) + (\mathcal{E} \times B_1) & \xrightarrow{id \times [F_0, F_1]} & \mathcal{E} \times B.
 \end{array} \quad (3)$$

is a pullback, by Diagram 1 and Lemma 6.3.9.

Thus squares (2) and (3) both pull back to $A'_0 + A'_1$, and hence the universal arrow $(\langle f_0, f_1 \rangle [F_0, F_1])$ of the product in \mathbf{NC} is a morphism of $\mathbf{NC}^=$. \square

Proposition 6.3.12 $\mathbf{NC}^=$ has coproducts given as in \mathbf{NC} .

Proof: The isomorphism $\iota: \mathbf{GC}^{co}$ to \mathbf{GC}^{op} (see proof of Proposition 6.2.3) given by

$$\iota: (\mathcal{E} \xleftarrow{\alpha} \mathcal{B}) \mapsto (\mathcal{B} \xleftarrow{\alpha^{op}} \mathcal{E}) \quad \text{and}$$

$$\iota: (f, F) \mapsto (F, f)$$

restricts to an isomorphism, which we shall also call ι , from $(\mathbf{NC}^=)^{co}$ to $(\mathbf{NC}^=)^{op}$. It is evident that $(\mathbf{NC}^=)^{co} \cong \mathbf{NC}^=$. Thus, since $\mathbf{NC}^=$ has all finite products, given as in $\mathbf{GC} \times \mathbf{GC}^{op}$, it follows that $(\mathbf{NC}^=)^{op}$ has all finite coproducts, given as in $\mathbf{GC} \times \mathbf{GC}^{op}$, and so $(\mathbf{NC}^=)^{co}$ has all finite coproducts, given as in $\mathbf{GC} \times \mathbf{GC}^{op}$, and so $\mathbf{NC}^=$ has all finite coproducts, given as in $\mathbf{GC} \times \mathbf{GC}^{op}$, and so $\mathbf{NC}^=$ has all finite coproducts, given as in \mathbf{NC} . \square

Proposition 6.3.13 If we define negation in $\mathbf{NC}^=$ to be the functor $(-)^{\perp} = [-, \perp]_{\mathbf{NC}^=}$, then for any object A of $\mathbf{NC}^=$ there is a map from A to $A^{\perp\perp}$.

Proof: There is a morphism in \mathbf{NC} from A to $A^{\perp\perp}$ given by the transpose across the adjunction $\mathbf{NC}(A \otimes A^{\perp}, \perp) \cong \mathbf{NC}(A, [A^{\perp}, \perp])$ of the evaluation from $A \otimes A^{\perp}$ to \perp . \square

The above propositions together prove the following:

Theorem 6.3.14 $\mathbf{NC}^=$ is a sound model of linear logic.

6.3.1 The Category $\mathbf{NSet}^=$ of Elementary Nets

Objects of $\mathbf{NSet}^=$ are precisely elementary Petri nets. A morphism in $\mathbf{NSet}^=$ from $(\mathcal{E} \xleftarrow{\alpha} \text{---} B)$ to $(\mathcal{E}' \xleftarrow{\alpha'} \text{---} B')$ is a pair of functions $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: B' \rightarrow B$ such that for any multiset A over \mathcal{E} ,

$$F^{-1}(\text{pre}(A)) = \text{pre}'(fA) \quad \text{and} \quad F^{-1}(\text{post}(A)) = \text{post}'(fA).$$

The category $\mathbf{NSet}^=$ resembles Winskel's category **SafeNet** and the various categories of Meseguer and Montanari more closely than the categories \mathbf{NSet} , $\mathbf{NSet}^{\text{co}}$, $\mathbf{NSet}_{\subseteq}^{\supseteq}$ and $\mathbf{NSet}_{\supseteq}^{\subseteq}$ in that it has equality where morphisms in these other categories require only containments. It is precisely this, however, which makes it less flexible. Winskel's morphisms, as is shown in his Theorem 4.2.1, insist that the part of the refined net chosen to simulate a net N be intimately related to N . A precise comparison of Winskel's categories with ours is given in Section 6.5.1.

Ultimately perhaps, the category $\mathbf{NSet}^=$ will prove more useful in the study of Petri nets than the other categories discussed in this thesis. It remains interesting, however, that we can weaken the requirements on morphisms as in the categories \mathbf{NSet} , $\mathbf{NSet}^{\text{co}}$, $\mathbf{NSet}_{\subseteq}^{\supseteq}$ and $\mathbf{NSet}_{\supseteq}^{\subseteq}$, and retain considerable categorical structure. This is an advantage of our unifying aim to define categories of Petri nets which are sound models of linear logic.

6.4 The Kleisli Category \mathbf{NC}_T

We next define a monad T on \mathbf{NC} and consider the Kleisli category \mathbf{NC}_T . We can also consider T as a monad on $\mathbf{NC}^=$, $\mathbf{NC}_{\subseteq}^{\subseteq}$, $\mathbf{NC}_{\supseteq}^{\supseteq}$, \mathbf{NC}^{∞} and form the Kleisli category for T on these categories. The monad T corresponds to lifting in the first component of a morphism or object in \mathbf{NC} . Thus a morphism in \mathbf{NC}_T from N to N' consists (up to isomorphism) of a partial function f from \mathcal{E} to \mathcal{E}' , and a function F from \mathcal{B}' to \mathcal{B} . This allows us to consider morphisms which are essentially the same as the morphisms of Winskel's category **SafeNet**, as we discuss in detail in Sections 6.5 and 6.5.1.

6.4.1 Definitions

Definition 6.4.1 A monad on a category \mathcal{C} is a tuple $\langle T, \eta, \mu \rangle$ consisting of an endofunctor T on \mathcal{C} , a natural transformation η from $\text{Id}_{\mathcal{C}}$ to T and a natural transformation μ from T^2 to T such that the diagrams

$$\begin{array}{ccc}
 T^3 A & \xrightarrow{T\mu_A} & T^2 A \\
 \mu_{TA} \downarrow & & \downarrow \mu_A \\
 T^2 A & \xrightarrow{\mu_A} & TA
 \end{array}$$

and

$$\begin{array}{ccccc}
 & T\eta_A & & \eta_{TA} & \\
 TA & \xrightarrow{\quad} & T^2 A & \xleftarrow{\quad} & TA \\
 & \searrow id & \downarrow \mu_A & \swarrow id & \\
 & & TA & &
 \end{array}$$

commute.

It is well known [ML71] that any adjunction $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$ with unit η and counit ϵ gives rise to a monad $\langle GF, \eta, G\epsilon F \rangle$ on \mathcal{C} , and also that any monad arises

from an adjunction. This adjunction is not in general unique, and there are two standard constructions, the Eilenberg–Moore construction [ML71] and the Kleisli construction [ML71] which given any monad construct an adjunction defining that monad. In this section we are concerned with the Kleisli construction.

Definition 6.4.2 *Let \mathcal{C} be a category and let $\langle T, \eta, \mu \rangle$ be a monad on \mathcal{C} . The Kleisli category for T , which we denote \mathcal{C}_T , is given as follows:*

- *objects of \mathcal{C}_T are objects of \mathcal{C} ,*
- *a morphism from A to B in \mathcal{C}_T is a morphism from A to TB in \mathcal{C} , and*
- *the composition of morphisms $f: A \rightarrow B$ and $g: B \rightarrow C$ in \mathcal{C}_T is the morphism $f(Tg)\mu$ in \mathcal{C} .*

The primary result relating a category with a monad to its Kleisli category is:

Proposition 6.4.3 *Let \mathcal{C} be a category and let $\langle T, \eta, \mu \rangle$ be a monad on \mathcal{C} . There exists $F_T \dashv G_T: \mathcal{C}_T \rightarrow \mathcal{C}$ such that the monad T arises from the adjunction $F_T \dashv G_T$.*

6.4.2 A Kleisli Category on NC

In this section we define on NC a monad T , which we shall call “lifting”. We shall show that T is also a monad on $\text{NC}^=$, $\text{NC}_{\frac{\subseteq}{\supseteq}}^{\subseteq}$, $\text{NC}_{\frac{\supseteq}{\subseteq}}^{\supseteq}$ and NC^{∞} . We form the Kleisli categories for T on NC, $\text{NC}_{\frac{\subseteq}{\supseteq}}^{\subseteq}$, $\text{NC}_{\frac{\supseteq}{\subseteq}}^{\supseteq}$ and $\text{NC}^=$, and consider the categories NSet_T , $(\text{NSet}^=)_T$, $(\text{NSet}_{\frac{\subseteq}{\supseteq}}^{\subseteq})_T$, $(\text{NSet}_{\frac{\supseteq}{\subseteq}}^{\supseteq})_T$ and $(\text{NSet}^{\infty})_T$, which have the set of elementary Petri nets as object set. Finally, we show that if (f, F) is a morphism from N to N' in $(\text{NSet}^=)_T$, then (f, F) is a morphism in Winskel’s category **SafeNet** from N with marking $\mathbf{0}$ to N' with marking $\mathbf{0}$, where $\mathbf{0}$ is the empty initial marking.

The definition of the monad requires several routine lemmata, which we now prove.

Lemma 6.4.4 The assignment $T(-) = - + \perp$ defines an endofunctor T on \mathbf{NC} .

Proof: Immediate. □

Notation 6.4.5 Let N be an object of \mathbf{NC} , and let (f, F) be a morphism in \mathbf{NC} . We shall write N_\perp for $T(N)$ and $(f, F)_\perp$ for $T(f, F)$.

We shall write \mathcal{E}_\perp for $\mathcal{E} + 1$, and c_B for the canonical isomorphism in \mathbf{C} from $B \times 1$ to B .

Let $N = \langle \mathcal{E}, \mathcal{B}, \alpha^0, \alpha^1 \rangle$ be a net.

N_\perp is the net $\langle \mathcal{E}_\perp, \mathcal{B} \times 1, (\alpha^0)_\perp, (\alpha^1)_\perp \rangle$, where the subobjects $(\alpha^i)_\perp$ of $(\mathcal{E}_\perp) \times (\mathcal{B} \times 1)$ are given for $i = 0, 1$ by

$$(e, 0) (\alpha^i)_\perp \langle b, \star \rangle \quad \text{if and only if} \quad e \alpha^i b,$$

$$\text{and} \quad (\star, 1) (\alpha^i)_\perp \langle b, \star \rangle \quad \text{never.}$$

Let (f, F) be a map in \mathbf{NC} from N to N' . Let $f_\perp: \mathcal{E}_\perp \rightarrow \mathcal{E}'_\perp$ and $F_\perp: \mathcal{B} \times 1 \rightarrow \mathcal{B}' \times 1$ be the morphisms in \mathbf{C} given by

$$f_\perp(e, 0) = f(e) \quad \text{for all } e \in \mathcal{E}, \quad f_\perp(\star, 1) = \star \quad \text{and}$$

$$F_\perp(\langle b', \star \rangle) = F(b') \quad \text{for all } b' \in \mathcal{B}'.$$

Then $(f, F)_\perp = (f_\perp, F_\perp)$.

Corollary 6.4.6 Let \mathcal{NC} be any of the categories $\mathbf{NC}^=$, \mathbf{NC}^{\subseteq} , \mathbf{NC}^{\supseteq} or \mathbf{NC}^{co} . The restriction of T to \mathcal{NC} is an endofunctor of \mathcal{NC} .

Proof: It is readily verified that whenever (f, F) is a morphism in \mathcal{NC} , so too is $(f, F)_\perp$. The result follows. □

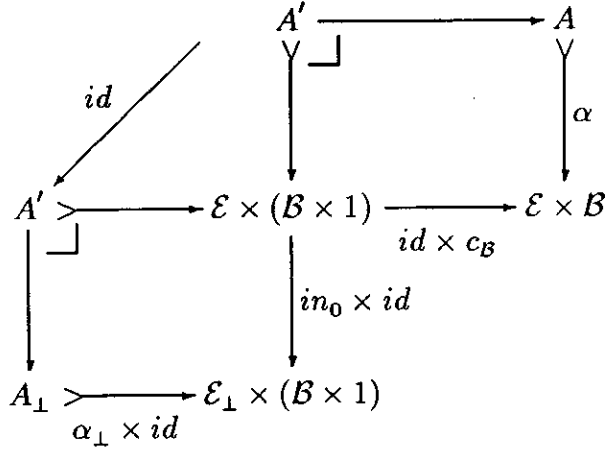
Notation 6.4.7

We shall write T for the restriction of T to any of the categories \mathcal{NC} .

Lemma 6.4.8

Let N be an object of \mathbf{NC}^\perp . Then the assignment $N \mapsto \eta_N: N \rightarrow N_\perp$ where $\eta_N = (in_0, c_B)$ defines a natural transformation from $Id_{\mathbf{NC}^\perp}$ to T .

Proof: To see that η_N is a morphism in \mathbf{NC}^\perp observe that triangle in the following the diagram:



commutes in \mathbf{C} .

Naturality of η is readily verified. □

Corollary 6.4.9 Let \mathcal{NC} be any of the categories \mathbf{NC}^\subseteq , \mathbf{NC}^\supseteq or \mathbf{NC} . Then η is a natural transformation from $Id_{\mathcal{NC}}$ to T .

Proof: Immediate, since any of the categories \mathcal{NC} has the same object set as \mathbf{NC}^\perp , and any morphism in \mathbf{NC}^\perp is a morphism in each of the categories \mathcal{NC} . □

Definition 6.4.10 Let \mathcal{E} be an object of \mathbf{C} . We define a morphism $fl_\mathcal{E}$ in \mathbf{C} from $\mathcal{E}_{\perp\perp}$ to \mathcal{E}_\perp as follows:

$$fl((e, 0), 0) = (e, 0), \quad fl((\star, 1), 0) = (\star, 1) \quad \text{and} \quad fl((\star, 1)) = (\star, 1).$$

Lemma 6.4.11

Let N be an object of \mathbf{NC}^\perp . The assignment $N \mapsto \mu_N: N_{\perp\perp} \rightarrow N_\perp$ where $\mu_N = (fl_\mathcal{E}, c_B)$, defines a natural transformation from $Id_{\mathbf{NC}^\perp}$ to T .

Proof: To see that μ_N is a morphism in $\mathbf{NC}^=$ observe that the diagram:

$$\begin{array}{ccccc}
 & & A & \xrightarrow{\quad} & A_{\perp\perp} \\
 & \nearrow id & \downarrow \lrcorner & & \downarrow \alpha_{\perp\perp} \\
 A & \xrightarrow{\quad} & \mathcal{E}_{\perp\perp} \times (\mathcal{B} \times 1) & \xrightarrow{id \times (c_{\mathcal{B}} \times 1)^{-1}} & \mathcal{E}_{\perp\perp} \times (\mathcal{B} \times 1 \times 1) \\
 \downarrow \lrcorner & & \downarrow fl_{\mathcal{E}} \times id & & \\
 A_{\perp} & \xrightarrow{\alpha_{\perp}} & \mathcal{E}_{\perp} \times (\mathcal{B} \times 1) & &
 \end{array}$$

commutes in \mathbf{C} .

Naturality of μ is readily verified. □

Corollary 6.4.12 Let \mathcal{NC} be any of the categories \mathbf{NC} , $\mathbf{NC}_{\perp}^{\subseteq}$, $\mathbf{NC}_{\subseteq}^{\supseteq}$ or \mathbf{NC}^{co} . Then μ is a natural transformation from $Id_{\mathcal{NC}}$ to T .

Proposition 6.4.13 The tuple $\langle T, \eta, \mu \rangle$ is a monad on $\mathbf{NC}^=$.

Proof: We have shown that T is an endofunctor on $\mathbf{NC}^=$, that η is a natural transformation from $Id_{\mathbf{NC}^=}$ to T , and that μ is a natural transformation from $Id_{\mathbf{NC}^=}$ to T . It remains to check that the three diagrams of Definition 6.4.1 commute in $\mathbf{NC}^=$. This is routine. □

Corollary 6.4.14 Let \mathcal{NC} be any of the categories \mathbf{NC} , $\mathbf{NC}_{\perp}^{\subseteq}$, $\mathbf{NC}_{\subseteq}^{\supseteq}$ or \mathbf{NC}^{co} . Then the tuple $\langle T, \eta, \mu \rangle$ is a monad on \mathcal{NC} .

6.4.3 Kleisli Categories with objects the Elementary Nets

Let \mathcal{NSet} be any of the categories \mathbf{NSet} , $\mathbf{NSet}^=$, $\mathbf{NSet}_{\perp}^{\subseteq}$, $\mathbf{NSet}_{\subseteq}^{\supseteq}$ or \mathbf{NSet}^{co} . The Kleisli category for T on \mathcal{NSet} is given as follows:

- objects of \mathcal{NSet}_T are objects of \mathcal{NSet} , that is, elementary Petri nets,

- a morphism from N to N' in \mathcal{NSet}_T is a morphism in \mathcal{NSet} from N to N'_\perp , and
- the composition of morphisms $(f, F): N \rightarrow N'$ and $(g, G): N' \rightarrow N''$ in \mathcal{NSet}_T is given by the composition:

$$N \xrightarrow{(f, F)} TN' \xrightarrow{T(g, G)} T^2N'' \xrightarrow{\mu} TN''.$$

Thus, making the usual identification of a partial function from \mathcal{E} to \mathcal{E}' with a total function from \mathcal{E} to \mathcal{E}_\perp , we can use the Kleisli categories \mathcal{NSet}_T to define morphisms between elementary Petri nets with the map on events a partial function.

In particular, a morphism from N to N' in \mathbf{NSet}_T^\equiv is a pair (f, F) of maps in \mathbf{Set} with $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B}' \times 1 \rightarrow \mathcal{B}$ such that, where fe is defined, we have

$$F^{-1}(\text{pre}(e)) = \text{pre}'(fe) \quad \text{and} \quad F^{-1}(\text{post}(e)) = \text{post}'(fe).$$

Let M be an initial marking of the net N , and M' an initial marking of the net N' . Then if (f, F) in \mathbf{NSet}_T^\equiv satisfies the condition

$$F^{-1}(M) = M',$$

(f, F) is a morphism in **SafeNet**. In particular, the condition is satisfied when $M = M' = \emptyset$, and so every morphism (f, F) from N to N' in \mathbf{NSet}_T^\equiv is a morphism from $\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, \emptyset \rangle$ to $\langle \mathcal{E}', \mathcal{B}', \text{pre}', \text{post}', \emptyset \rangle$ in **SafeNet**.

6.5 Comparison of the NC categories with SafeNet

Suppose (η, β) is a morphism from N to N' in **SafeNet**. Winskel's condition on morphisms is strict in the sense that for any condition b of N , the multiset $\beta(b)$ must play exactly the same rôle in N' as b plays in N , apart from its use by events not in the image of η .

Winskel obtains flexibility in his morphisms in three ways. The map η on events is partial, leading to an interpretation of morphisms as indicating a partial

simulation of N by N' . Secondly, η need not be injective, so that one event in N' may be used to simulate two events in the N . Finally, η need not be surjective, and in fact N' may include events totally unrelated to any events or conditions in N . This last property is perhaps not essential to a concept of refinement. Morphisms in \mathbf{NSet} have a similar property, but insist that no extraneous *conditions* be introduced in the refined net. Morphisms in \mathbf{NSet} also have the second property listed above, and we saw in Section 6.4.3 how morphisms in $\mathbf{NSet}_{\mathcal{T}}$ allow morphisms where the map on the event sets is a partial function.

It is useful to have the option of additional flexibility available in \mathbf{NSet} , where a net which refines a net N is not required to mirror the behaviour of N to the extent of equality. At present, however, it is difficult to see which of \mathbf{NSet} , \mathbf{NSet}^{co} , $\mathbf{NSet}_{\subseteq}^{\subseteq}$, $\mathbf{NSet}_{\supseteq}^{\supseteq}$ and \mathbf{NSet}^{\equiv} is the most useful category. I hope that further thought and more examples will lead to an understanding of the best occasions to use each category. Further, the category \mathbf{NPSet}^{\equiv} has been studied (under a different name) in [ER90], [Roz87] and [Thi87], where behavioural tools for studying the non-sequential behaviour of elementary net systems are set out. We may gain insight here from the structure of \mathbf{NPSet}^{\equiv} . Unfortunately, the work of Ehrenfeucht, Nielsen, Rozenberg and Thiagarajan has come to my attention only recently, and I cannot give herein a careful consideration of its relationship with the dialectica categories of nets.

6.5.1 Comparing \mathbf{NSet}^{\equiv} with $\mathbf{SafeNet}$

The most significant difference between Winskel's category $\mathbf{SafeNet}$ and \mathbf{NSet} is that the objects of $\mathbf{SafeNet}$ are marked safe nets, whereas objects of \mathbf{NSet} are elementary nets. The marking of a Petri net N can be regarded as an environment for N . In this part of this thesis we consider nets independent of their environments, and treat the behaviours of a net in a given environment as additional categorical structure. A first attempt at modelling behaviours in the dialectica categories of nets is given in [BG90]. At this point we merely observe that the objects of $\mathbf{SafeNet}$ may be regarded as objects of \mathbf{NSet} if we forget their initial markings.

We do not have a forgetful functor from **SafeNet** to **NSet**, because **SafeNet** has morphisms which do not correspond to any morphisms of **NSet**.

The morphisms of **SafeNet** differ from those of **NSet** in three ways. Let $N_m = \langle \mathcal{E}, \mathcal{B}, pre, post, M \rangle$ and $N'_m = \langle \mathcal{E}', \mathcal{B}', pre', post', M' \rangle$ be safe nets which are marked elementary nets. Then $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ and $N' = \langle \mathcal{E}', \mathcal{B}', pre', post' \rangle$ are objects of **NSet**. Let (f, F) be a morphism from N to N' in **NSet**, and let (η, β) be a morphism from N_m to N'_m in **SafeNet**. From the definitions of the morphisms we have

- $\eta: \mathcal{E} \rightarrow \mathcal{E}'$ is a partial function while $f: \mathcal{E} \rightarrow \mathcal{E}'$ is a total function,
- β is an arbitrary multirelation from \mathcal{B} to \mathcal{B}' , whereas F^{-1} , regarded as a multirelation from \mathcal{B} to \mathcal{B}' , arises as the inverse of a function, and
- for every multiset A over \mathcal{E} , we have the conditions

$$pre(\eta A) = \beta(pre(A)) \quad \text{while} \quad F^{-1}(pre(fA)) \subseteq pre(A), \text{ and}$$

$$post(\eta A) = \beta(post(A)) \quad \text{while} \quad F^{-1}(post(fA)) \subseteq post(A).$$

Let $(\eta, \beta): N_m \rightarrow N'_m$ be a morphism in **SafeNet**. If β is the inverse of a function, then (η, β^{-1}) is a morphism from N to N'_\perp in **NSet_T**. If in addition η is total (so that (η, β) is a morphism in **SafeNet_{syn}**) then (η, β^{-1}) is a morphism in **NSet⁼** from N to N' .

Conversely, if $(f, F): N \rightarrow N'$ is a morphism in **NSet⁼** and $F^{-1}(M) = M'$, then (f, F) is a morphism from N_m to N'_m in both **SafeNet** and **SafeNet_{syn}**.

We expect that work now in progress on dialectica categories with object set **Petri** (see **Conclusion**) will show that Winskel's category **Net** relates to our categories of general nets in ways similar to those described above.

Part III

Petri Nets and Quantales

Introduction to Part III

When Girard introduced linear logic [Gir86], he suggested that it may be a natural logic for reasoning about concurrent systems. Recent results of Asperti [Asp87], Brown [Bro89b] and Gunter and Gehlot [GG89] have shown that evolution in Petri nets corresponds to linear proof, and in fact that the simple tensorial fragment of linear logic suffices to describe Petri nets. Attempts to understand the other connectives of linear logic in terms of nets have also been made, in particular in [Bro89b] [BG90], [BG] and [MOM89].

In this Part, we construct a quantale from a Petri net, and prove it to be a sound model for linear logic. This allows us to give a meaning to all of the connectives of linear logic in terms of nets. From our interpretation of linear logic in this *net-quantale*, we develop a specification language for Petri nets which uses linear entailment to reason about the behaviour of nets. Thus we use linear logic to manipulate Petri nets, and Petri nets to model linear logic.

Quantales were introduced by Mulvey [Mul86], and have been studied by Abramsky and Vickers [AV88], Niefield and Rosenthal [NR88] and Yetter [Yet], among others. Yetter [Yet] showed that quantales are models of linear logic. Further work has been done by Engberg and Winskel [EW89]. They use a Petri net to construct a different quantale from ours. We show that their quantale can be more elegantly constructed by use of our more general approach.

Chapter 7

Quantales of Behaviours of a Net

7.1 Introduction

Section 2 of this chapter gives a general construction of a quantale from a net, which can be used to construct the quantale of [EW89], as we show. Section 3 uses the quantale to consider properties of nets with respect to a particular form of behaviour, which we shall call *traps*. Section 4 interprets linear logic in a net-quantale, and shows how to specify properties of a net using linear logic. Section 5 discusses the interpretation of linear negation, and uses it to specify *safety properties* of a net. Section 6 describes the equivalence on nets induced by their generating the same net-quantale. Section 7 constructs a finite quantale from a net. This quantale also gives a sound semantics for linear logic.

7.2 Constructing a Quantale from a Net

It has been shown [AV88], [Yet] that quantales model linear intuitionistic logic just as complete Heyting algebras model intuitionistic logic.

Definition 7.2.1

A commutative quantale is a 4-tuple $\langle Q, \leq, \otimes, \mathbf{1} \rangle$ such that

- $\langle Q, \leq \rangle$ is a complete semi-lattice,
- $\langle Q, \otimes, \mathbf{I} \rangle$ is a commutative monoid with unit \mathbf{I} , and
- for any indexing set J and any $A, B_j \in Q$, $A \otimes \bigvee_{j \in J} B_j = \bigvee_{j \in J} (A \otimes B_j)$.

Remark 7.2.2

Henceforth, we shall write ‘quantale’ to mean ‘commutative quantale’.

Warning: In some literature, there appear definitions of quantales which differ from ours in two respects: some authors do not require $\langle Q, \otimes, \mathbf{I} \rangle$ to be commutative; others require $\langle Q, \otimes, \mathbf{I} \rangle$ to be idempotent. In the latter respect, we follow Abramsky and Vickers [AV88], Niefield and Rosenthal [NR88] and Yetter [Yet].

Remark 7.2.3

A quantale is a complete, co-complete symmetric monoidal closed category.

Definition 7.2.4 A homomorphism between quantales Q and Q' is a function $f: Q \rightarrow Q'$ which preserves \vee , \mathbf{I} and \otimes .

Remark 7.2.5 Quantales and quantale homomorphisms, together with the evident composition, define a category, which we shall call **Quant**.

Definition 7.2.6 Let Q be a quantale. A closure operator on Q is a function $j: Q \rightarrow Q$ such that

- $a \leq b \Rightarrow j(a) \leq j(b)$ (j is order-preserving),
- $a \leq j(a)$, (j is increasing), and
- $j(j(a)) = j(a)$ (j is idempotent).

Definition 7.2.7 A co-closure operator on a quantale Q is a function $j: Q \rightarrow Q$ which is decreasing, order-preserving and idempotent.

Definition 7.2.8 Let $j : Q \rightarrow Q$ be a closure operator on Q . Then j is a quantic nucleus on Q if

$$\text{for all } a, b \in Q, \quad j(a) \otimes j(b) \leq j(a \otimes b).$$

Remark 7.2.9 We have $I \leq j(I)$, since j is increasing.

Definition 7.2.10 Let $j : Q \rightarrow Q$ be a co-closure operator on Q . Then j is a quantic conucleus on Q if

$$\text{for all } a, b \in Q, \quad j(a) \otimes j(b) \leq j(a \otimes b) \quad \text{and}$$

$$I \leq j(I).$$

Remark 7.2.11 Since a conucleus j is decreasing, we have $j(I) = I$ for any quantic conucleus j .

Remark 7.2.12 Quantic nuclei and conuclei are monoidal functors.

Theorem 7.2.13 [Niefield and Rosenthal] Let $j : Q \rightarrow Q$ be a quantic nucleus. The image of j is a quantale with monoid operation given by $a \otimes_j b = j(a \otimes b)$. Further, $j : Q \rightarrow j(Q)$ is a quantale homomorphism.

Definition 7.2.14 Let $\langle Q, \leq, \otimes, I \rangle$ be a quantale. Let Q' be a subset of Q which contains the unit I and is closed under \otimes and \vee . Then $\langle Q', \leq, \otimes, I \rangle$ is a subquantale of Q .

Theorem 7.2.15 [Niefield and Rosenthal] Let $j : Q \rightarrow Q$ be a quantic conucleus. The image of j is a subquantale of Q .

7.2.1 A Quantale of Markings of a Petri Net

Recall that the pre- and post-condition relations of a Petri net N induce a relation on markings, the derivability relation, such that for markings m and m' of N , $m' \leq m$ if N can evolve from marking m to marking m' .

We now construct a quantale $q(N)$ which has as elements sets of markings of a net N . This quantale does not represent the behaviour of the net N in any way. Later we shall define a quantale Q_N , which is a quotient of $q(N)$ and which *does* represent the behaviour of the net N .

We extend multiset addition to sets of markings in the evident way. Thus if P and Q are subsets of $\text{Mark}(N)$,

$$P + Q = \{p + q \mid p \in P \text{ and } q \in Q.\}$$

Lemma 7.2.16 Let $N = \langle \mathcal{B}, \mathcal{E}, pre, post \rangle$ be a Petri net. Let \emptyset be the constant zero marking given by $\emptyset(b) = 0$ for all $b \in \mathcal{B}$. Let $\mathcal{P}(\text{Mark}(N))$ be the powerset of the markings of N . Then $\langle \mathcal{P}(\text{Mark}(N)), +, \emptyset \rangle$ is a commutative monoid.

Proof: Immediate. □

Lemma 7.2.17 Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ be a Petri net. Let $Q = \mathcal{P}(\text{Mark}(N))$ and let $+$ be addition of sets of markings. Then the assignment

$$N \mapsto \langle Q, \subseteq, +, \{\emptyset\} \rangle$$

is a function, which we shall call q , from **Petri** to **|Quant|**.

Proof: Routine. □

Remark 7.2.18 The top element $\mathbf{1}$ of $\langle Q, \subseteq \rangle$ is the set of all markings of the net N , and the bottom element $\mathbf{0}$ of $\langle Q, \subseteq \rangle$ is the empty set \emptyset .

From the quantale $q(N)$, we now construct a new quantale Q_N whose lattice structure is determined precisely by the derivability relation of the net N . Q_N is the image of $q(N)$ under an appropriate quantic nucleus \downarrow . Once we have shown that \downarrow is a quantic nucleus on $q(N)$, it follows from Theorem 7.2.13 that $\downarrow q(N)$ is a quantale. This approach is more elegant than proving Q_N to be a quantale

directly, and using different nuclei we can construct different quantaes from $q(N)$. For example, we show in Section 7.2.27 that the quantale of [EW89] arises in this way. Q_N is the quantale of primary interest to us.

In Section 7.7 we shall show that restricting the set of initial markings in certain ways leads to smaller, more tractable quantaes.

It follows from the linearity of evolution of Petri nets that

Lemma 7.2.19 If m_1, m'_1, m_2 and m'_2 are markings of a net such that $m_1 \leq m_2$ and $m'_1 \leq m'_2$ then $(m_1 + m'_1) \leq (m_2 + m'_2)$.

We extend the derivability relation \leq to sets of markings as follows:

Definition 7.2.20 Let A and B be sets of markings of a given net N . Then $B \leq A$ if for any $b \in B$, we can find an $a \in A$ such that $(b \leq a)$.

Definition 7.2.21 Let N be a Petri net. Forwards closure under evolution, written \downarrow , is an endofunction on Q , defined as follows. Given any subset A of $\text{Mark}(N)$,

$$\downarrow A = \{m \in \text{Mark}(N) \mid \exists a \in A.(m \leq a)\}.$$

Remark 7.2.22

- $\downarrow A$ is the downwards closure of the set A with respect to the ordering \leq .
- For any sets of markings A and B , if $A \subseteq B$ then $A \leq B$.
- If A and B are downwards closed under \leq , then $A \leq B$ if and only if $A \subseteq B$.

Proposition 7.2.23

Let N be a Petri net. Then $\downarrow: q(N) \rightarrow q(N)$ is a quantic nucleus.

Proof: \downarrow is evidently increasing and idempotent. Further, for any sets

A and B of markings of N , $A \leq B \Rightarrow A \leq \downarrow B$ (since \downarrow is increasing)

$\Rightarrow \downarrow A \leq \downarrow B$ (since $\downarrow B$ is downwards closed).

Thus \downarrow preserves order, and is a closure operator. It remains to show that for any $A, B \subseteq Q$, $\downarrow(A) + \downarrow(B) \leq \downarrow(A + B)$.

This follows essentially from Lemma 7.2.19, since

$$\begin{aligned} \downarrow(A) + \downarrow(B) &= \{m \mid \exists a \in A.(m \leq a)\} + \{m \mid \exists b \in B.(m \leq b)\} \\ &\subseteq \{p \mid \exists m \in (A + B).(p \leq m)\} \\ &= \downarrow(A + B) \end{aligned}$$

Using Remark 7.2.22, we have $\downarrow(A) + \downarrow(B) \leq \downarrow(A + B)$. □

Corollary 7.2.24 Applying Theorem 7.2.13, we see that the image of $q(N)$ under \downarrow is a quantale, in which

- elements are subsets of $\text{Mark}(N)$ closed under evolution,
- the ordering is subset inclusion, \subseteq ,
- the monoid operation \otimes is given by $A \otimes B = \downarrow(A + B)$, and
- the unit of \otimes is $\downarrow\{\emptyset\}$.

Remark 7.2.25 *Commutativity of \otimes follows from the commutativity of $+$.*

Notation 7.2.26

We denote the quantale $\downarrow q(N)$ by Q_N , and we call Q_N the net-quantale of N .

In general, we shall use Q_N to refer both to the quantale representing the net N , and to its underlying set.

Q_N expresses the derivability relation of the Petri net N in lattice form. We can use Q_N to examine the behaviour of N without reference to specific events. Certain aspects of behaviour become more apparent when the net is viewed in this way. An example is given in Section 7.3.

7.2.2 Backwards closed sets of markings

Definition 7.2.27 *Let N be a Petri net. We define an endofunction \uparrow on the powerset of $\text{Mark}(N)$ as follows. Given any element A of the powerset of $\text{Mark}(N)$,*

$$\uparrow A = \{m \in \text{Mark}(N) \mid \exists a \in A.(a \leq m)\}$$

It is routine to verify that \uparrow is a quantic nucleus on $q(N)$. The quantale $\uparrow q(N)$ is the quantale discussed in [EW89], whose objects are sets closed under backward evolution of the net. The objects of this quantale indicate what resources are needed for the net to evolve to a given marking.

Remark 7.2.28 *Let N be a Petri net. There is an alternative presentation of the quantales $q(N)$, Q_N and $\uparrow q(N)$, following Abramsky and Vickers [AV90]. Thus*

- $q(N)$ is the free commutative quantale over the set B of conditions of N ,
- Q_N is the commutative quantale with generators B , subject to the relations $\text{post}(e) \leq \text{pre}(e)$ for each event e of N , thus

$$Q_N = \text{Comm.Qu} \langle a(a \in B) \mid \text{post}(e) \leq \text{pre}(e) (e \in \mathcal{E}) \rangle \quad \text{and}$$

- $\uparrow q(N)$ is the commutative quantale with generators B , subject to the relations $\text{pre}(e) \leq \text{post}(e)$ for each event e of N , that is,

$$\uparrow q(N) = \text{Comm.Qu} \langle a(a \in B) \mid \text{pre}(e) \leq \text{post}(e) (e \in \mathcal{E}) \rangle .$$

7.3 Traps

In this section we consider a simple instance of structure in a net-quantale Q_N which elegantly expresses a behavioural feature of N which we shall call *traps*.

Consider the atoms of the lattice $\langle Q_N, \leq \rangle$: that is, those elements A of the lattice for which $X \leq A \Rightarrow (X = \perp \text{ or } X = A)$.

Definition 7.3.1 *Let N be a Petri net. A trap for N is a finite subset T of $\text{Mark}(N)$ such that whenever $m \in T$ and $m' \leq m$, it follows that $m' \in T$.*

Thus traps are finite elements of Q_N . We shall usually be interested in very small traps, to study which we introduce the concept of *basic trap*, defined below. Traps allow us to study certain safety properties of a net N , since a trap identifies a set of markings which is stable under all future evolution of the net. Conversely, there may be traps which we wish to avoid, for instance infinite cycles. In that case, identifying the trap will enable us to refine the net in such a way that it no longer has this loop.

We now characterise traps succinctly using some elementary definitions from graph theory (see [Bol79]). The definition of directed graph we repeat here for convenience.

Definition 7.3.2 *A directed graph is a 4-tuple $\langle V, E, \text{Source}, \text{Target} \rangle$ where V and E are disjoint sets, and Source and Target are functions from E to V .*

Definition 7.3.3 *A subgraph of a directed graph $\langle V, E, \text{Source}, \text{Target} \rangle$ is a directed graph $\langle V', E', \text{Source}', \text{Target}' \rangle$ such that $V' \subseteq V$, $E' \subseteq E$, and Source' and Target' are respectively the restriction of Source and Target to E' .*

Let $G = \langle V, E, \text{Source}, \text{Target} \rangle$ be a directed graph and let $V' \subseteq V$. The subgraph of G on vertices V' is the maximal subgraph of G with vertex set V' .

Definition 7.3.4

Let $G = \langle V, E, Source, Target \rangle$ be a directed graph, and let $v_0, v_n \in V$. A directed path in G from v_0 to v_n is a subgraph $P = \langle V', E', Source, Target \rangle$ of G with finite vertex set $V' = \{v_0, \dots, v_n\}$ and edge set $E = \{e_1, e_2, \dots, e_n\}$ such that

- for $i \in \{1, \dots, n\}$, $Source(e_i) = v_{i-1}$ and
- for $i \in \{1, \dots, n\}$, $Target(e_i) = v_i$.

Definition 7.3.5 A directed graph $G = \langle V, E, Source, Target \rangle$ is totally connected if for every $v, w \in V$, there is a directed path in G from v to w .

Definition 7.3.6 A component of a directed graph G is a maximal, totally connected subgraph of G .

Definition 7.3.7 Let $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ be a Petri net, and let $E = \mathbf{N}^{\mathcal{E}}$, the set of multisets over \mathcal{E} . The transition graph of N , written $\mathcal{T}(N)$, is the directed graph $\langle \text{Mark}(N), \mathcal{E}, \bullet(-), (-)\bullet \rangle$, where $\bullet(-)$ and $(-)\bullet$ are the extensions of the functions pre and $post$ respectively to multisets over \mathcal{E} .

Definition 7.3.8 A basic trap Tr of a Petri net N is a subset of $\text{Mark}(N)$ such that the subgraph of $\mathcal{T}(N)$ on vertices Tr is totally connected.

Remark 7.3.9 Any trap is a finite union of finite basic traps.

Definition 7.3.10 A cycle in a net N is a sequence $C = m_0, \dots, m_n$ of at least two distinct markings of N such that $m_0 \leq m_1 \leq \dots \leq m_n = m_0$.

Remark 7.3.11 A cycle in N is a directed path in $\mathcal{T}(N)$ for which $v_0 = v_n$.

Example 7.3.12 Let $C = m_0, \dots, m_n$ be a cycle in N . The subgraph of $\mathcal{T}(N)$ on vertices $\{m_0, \dots, m_n\}$ is a component of $\mathcal{T}(N)$. Thus $\{m_0, \dots, m_n\}$ is a finite basic trap.

Example 7.3.13 Let m be a marking from which a net N cannot evolve (so that the net N deadlocks if it reaches marking m). There is no directed path in $\mathcal{T}(N)$ from m to any other vertex, and so the subgraph of $\mathcal{T}(N)$ on vertices $\{m\}$ is a component of $\mathcal{T}(N)$. Thus $\{m\}$ is a finite basic trap.

Proposition 7.3.14 Let N be a Petri net. A subset A of $\text{Mark}(N)$ is a lattice atom of Q_N if the subgraph of $\mathcal{T}(N)$ on vertices A is a component of $\mathcal{T}(N)$.

Proof: An atom is a downwards closed set of markings which has no non-trivial downwards closed subset.

Thus if $\mathcal{M} = \{m_0, \dots, m_n\}$, we have $\mathcal{M} = \downarrow m_i$ for each $i = 0, \dots, n$.

Hence the subgraph of $\mathcal{T}(N)$ on vertices \mathcal{M} is totally connected. That this subgraph is a *maximal* totally connected subgraph follows from the fact that \mathcal{M} is downwards closed.

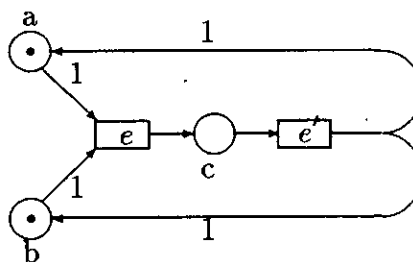
Thus the subgraph of $\mathcal{T}(N)$ on vertices \mathcal{M} is a component of $\mathcal{T}(N)$.

□

Corollary 7.3.15 The lattice atoms of Q_N with finitely many elements are precisely the finite basic traps of N .

If all the lattice atoms of Q_N are finite, then all basic traps of N are finite. The necessary conditions for this to be the case are not immediately apparent. We define a loop in a net N to be a finite sequence of events e_0, e_1, \dots, e_n in N such that $\text{pre}(e_0) \cap \text{post}(e_n) \neq \emptyset$. A sufficient condition for a net to have only finite basic traps is that it has no events with empty pre- or post-conditions, and no loops.

Example 7.3.16 Consider the marked net N with two events shown below:



The net N cycles between the two markings c and $a+b$. Therefore the set $\{a+b, c\}$ is an atom in $\langle Q_N, \subseteq \rangle$, and $\{a+b, c\}$ is a basic trap for N . The atom $\{a\}$ is a marking at deadlock. The atom $\{a+b, c\}$ is a cycle.

For convenience, we shall say that when a net has evolved to an element of a trap, it has reached that trap.

In general, examining the atoms of a net-quantale Q_N enables us to make statements about the possible behaviour of the net N . For instance, if we let A range over the set of all atoms of the lattice, then for any marking m of N , we can show that

$$\downarrow\{m\} \wedge A = \begin{cases} A & \text{if } \exists m' \in A. (m' \leq m) \\ \phi & \text{otherwise.} \end{cases}$$

This allows us to establish algebraically whether N can reach a particular trap A from m . Further,

$$\bigwedge\{A \mid \downarrow\{m\} \wedge A \neq \phi\} = \begin{cases} p & \text{if } p \text{ is the only basic trap} \\ & \text{which } N \text{ can reach from } m \\ \phi & \text{if } N \text{ can reach more than one} \\ & \text{basic trap from } m \end{cases}$$

and

$$\bigvee\{A \mid \downarrow\{m\} \wedge A \neq \phi\} = \bigcup\{p \mid p \text{ a basic trap which } N \text{ can reach from } m\}$$

Thus we can establish algebraically whether a marked net N has one basic trap, or several.

7.4 Interpreting Linear Logic in a Net–Quantale

Let N be a Petri net. We can interpret the linear logic connectives \oplus , \wedge , \otimes , \multimap , $(-)^{\perp}$ and $!$, and also the constants $\mathbf{1}$, \perp , \mathbf{I} and $\mathbf{0}$ in Q_N .

We shall assume that tokens at a condition interpret the atomic propositions of the linear calculus. Our interpretation is then parametric in the interpretation of these atomic propositions. We shall write m_A for the marking which consists of a single token at the condition A . Formulae of linear logic are denoted by sets of markings of N as follows:

- $\llbracket A \rrbracket = \{m \mid m \leq m_A\}$ for an atomic proposition A
- $\llbracket \mathbf{1} \rrbracket = \{m \mid m \text{ a possible marking of the net } N\} = \mathbf{1}_{Q_N}$
- $\llbracket \mathbf{0} \rrbracket = \phi = \mathbf{0}_{Q_N}$
- $\llbracket \mathbf{I} \rrbracket = \downarrow \{\emptyset\}$
- $\llbracket A \otimes B \rrbracket = \llbracket A \rrbracket \otimes \llbracket B \rrbracket = \downarrow \{a + b \mid a \in \llbracket A \rrbracket \text{ and } b \in \llbracket B \rrbracket\}$
- $\llbracket A \wedge B \rrbracket = \llbracket A \rrbracket \cap \llbracket B \rrbracket$
- $\llbracket A \oplus B \rrbracket = \llbracket A \rrbracket \cup \llbracket B \rrbracket$
- $\llbracket A \multimap B \rrbracket = \cup \{\llbracket C \rrbracket \mid \llbracket C \otimes A \rrbracket \subseteq \llbracket B \rrbracket\}$

With any interpretation of the linear logic constant \perp as an element of Q_N , we can now interpret A^{\perp} in the quantale in the usual intuitionistic way by putting

$$\llbracket A^{\perp} \rrbracket = \llbracket A \multimap \perp \rrbracket = \cup \{\llbracket C \rrbracket \mid \llbracket C \otimes A \rrbracket \subseteq \llbracket \perp \rrbracket\}.$$

This interpretation is exactly analogous to the interpretation of intuitionistic logic using Heyting algebras, in that we interpret linear entailment by the ordering

\subseteq on the net-quantale of \mathbf{N} , we interpret $\mathbf{1}$ and $\mathbf{0}$ by the top and bottom elements of the lattice respectively, and implication (\multimap) as

$$A \multimap B \triangleq \bigvee \{C \mid C \otimes A \leq B\}.$$

In particular, we have the usual adjunction

$$C \otimes A \leq B \text{ if and only if } C \leq A \multimap B$$

which we expect, since linear \otimes is here playing the role of the intuitionistic *and*.

We define semantic entailment in the quantale by

$$A_1 \otimes \cdots \otimes A_n \models A \text{ if and only if } \llbracket A_1 \rrbracket \otimes \cdots \otimes \llbracket A_n \rrbracket \subseteq \llbracket A \rrbracket.$$

The motivation for this interpretation is that a marking m is denoted by its consequences, or in other words, by the set of all resources we could gain from the resource m . Thus anything gained by having an element of $\llbracket A \otimes B \rrbracket$ must be a possible gain when we have some $a \in \llbracket A \rrbracket$ and some $b \in \llbracket B \rrbracket$ *at the same time*.

Also, any consequence of having some resource which came from a non-deterministic choice between A and B must be either a consequence of having some element of A or of having some element of B . Accordingly, we interpret $A \oplus B$ as the union of consequences of A and consequences of B .

Similarly, whenever we have a consequence x of $A \wedge B$, we can make a determined choice of A and we know that x will be a consequence of A . Similarly, we know that if we choose B , x must be a consequence of our choice. We must therefore insist that x be a consequence of both A and B , and so we interpret $A \wedge B$ by the intersection of consequences of A and B .

Our interpretation of $A \multimap B$ expresses the property of implication that no consequence of $A \multimap B$ can give any more gain when taken in conjunction (\otimes) with some consequence a of A than could be gained from an appropriate b in $\llbracket B \rrbracket$.

The interpretations of $\mathbf{1}$, $\mathbf{1}$ and $\mathbf{0}$ follow simply from their required behaviour as constants of the logic. We can gain nothing more from the set of all possible

markings than what was already possible, and this explains the choice of interpretation for $\mathbf{1}$. Also, if we have an element of $\llbracket \mathbf{0} \rrbracket$, we can deduce even impossible markings—as there can be no such element, $\mathbf{0}$ is interpreted by the empty set. As we expect, $\llbracket \mathbf{1} \rrbracket$ is the set of resources which can be gained from nothing.

Remark 7.4.1 In [EW89], Winskel and Engberg show that the interpretation of “of course” A should be

$$\llbracket !A \rrbracket = \bigcup \{C \in Q_N \mid C \text{ is a postfix point of } f_A\},$$

where $f_A : Q_N \rightarrow Q_N$ is the function given by

$$x \mapsto \mathbf{1} \wedge \llbracket A \rrbracket \wedge (x \otimes x).$$

This follows a suggestion of Girard in [GL87].

We abbreviate $\mathbf{1} \models A$ by $\models A$.

Proposition 7.4.2

1. $\models A$ if and only if $\emptyset \in \llbracket A \rrbracket$
2. $A \models B$ if and only if $\llbracket A \rrbracket \subseteq \llbracket B \rrbracket$ if and only if $\models A \multimap B$, and
3. $\models m \multimap m'$ if and only if N can evolve from marking m' to marking m .

Proof: Immediate from the definitions. □

Theorem 7.4.3 The quantale Q_N with the above interpretation is sound with respect to the single-conclusion sequent calculus for linear logic without the rules for \sharp , i.e.

$$\Gamma \vdash A \Rightarrow \Gamma \models A.$$

Proof: By case analysis.

For example, the proof of soundness with respect to the (Cut) rule

$$\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta}$$

is as follows:

By hypothesis, $\Gamma \models A$ and $\Delta, A \models B$. Hence

$$\llbracket \Gamma \rrbracket \subseteq \llbracket A \rrbracket \quad \text{and} \quad \llbracket \Delta \rrbracket \otimes \llbracket A \rrbracket \subseteq \llbracket B \rrbracket.$$

Hence $\llbracket \Delta \rrbracket \otimes \llbracket \Gamma \rrbracket \subseteq \llbracket B \rrbracket$, and we have $\Gamma, \Delta \models B$.

Thus our interpretation is sound with respect to the (Cut) rule. \square

This semantics allows us to make assertions about the behaviour of the net N whose behaviour has been encoded in the quantale. For example,

- $\models m$ asserts that marking m can evolve to the empty marking, \emptyset ,
- $\models (A \otimes B) \multimap C$ asserts that from a marking of condition C , the net N can evolve to the marking $A + B$, and
- $\models (m_1 \wedge m_2) \multimap m$ asserts that the marking m can evolve to marking m_1 and also to marking m_2 .

7.5 Linear Negation

In this section, we suggest a choice for the interpretation of the logical constant \perp , and show how it can be used to make further assertions about the behaviour of a net. In general, we use negation to assert things which a net *cannot* do, rather than things which it can do.

Definition 7.5.1 *Let N be a Petri net and M_F a subset of $\text{Mark}(N)$. Define*

$$\perp = \{m \in \text{Mark}(N) \mid \nexists m' \in M_F. (m' \leq m)\}.$$

Remark 7.5.2 \perp is evidently downwards closed, and hence an element of Q_N .

Now

$$\llbracket A^\perp \rrbracket = \{m \in \text{Mark}(N) \mid \forall a \in \llbracket A \rrbracket \nexists m' \in M_F. (m' \leq (m + a))\}$$

Thus, the denotation of A^\perp is the set of all markings which, when added to any marking in the denotation of A , can never evolve to a marking in M_F . We think of M_F as a set of “forbidden markings”, since we are concerned with proving that they cannot be reached.

In particular, $\models A^\perp$ if and only if for all $a \in \llbracket A \rrbracket$, there is no $m' \in M_F$ such that $(m' \leq a)$. This in turn is true if and only if $\llbracket A \rrbracket \cap M_F = \phi$. Thus whenever $\models A^\perp$, the net N can never evolve from any marking $a \in \llbracket A \rrbracket$ to a marking in M_F .

This enables us to make negative assertions about a net’s behaviour, and hence to specify safety properties of a net. Thus it is possible to assert that there is no marking reachable from A in which a particular multiset is marked.

Example 7.5.3 If we put

$$\tilde{M}_F = \{b^n \in \text{Mark}(N) \mid n \text{ a positive integer}\}$$

then $\models A^\perp$ asserts that there is no marking reachable from A in which the condition b is marked in any non-zero multiplicity.

7.6 Equivalences on Nets

We have seen that every net N generates a quantale Q_N . In this section, we consider the circumstances in which two nets generate the same net-quantale.

Notation 7.6.1 We shall write $\models_N A$ to mean that A is valid in the quantale Q_N , in the sense of Section 7.4.

Definition 7.6.2 Let $N_0 = \langle \mathcal{E}_0, \mathcal{B}_0, \text{pre}_0, \text{post}_0 \rangle$ and $N_1 = \langle \mathcal{E}_1, \mathcal{B}_1, \text{pre}_1, \text{post}_1 \rangle$ be Petri nets such that $\mathcal{B}_0 = \mathcal{B}_1$. Then the nets N_0 and N_1 are equivalent, written $N_0 \sim N_1$ if for all propositions A ,

$$\models_{N_0} A \quad \text{if and only if} \quad \models_{N_1} A.$$

Proposition 7.6.3 Let N and N' be Petri nets.

- If N_0 and N_1 have the same derivability relation, then $N_0 \sim N_1$.
- If $N_0 \sim N_1$ then $Q_{N_0} = Q_{N_1}$.

Proof: If N_0 and N_1 have the same derivability relation then the relations \models_{N_0} and \models_{N_1} are equal and the result follows by definition of \sim .

We have seen that $\models_N (m \rightarrow m')$ if and only if the net N can evolve from a marking m' to a marking m . If $\mathcal{B}' = \mathcal{B}$, the underlying sets of the net-quantales Q_{N_0} and Q_{N_1} are equal. Further, if $N_0 \sim N_1$, then the nets N_0 and N_1 have the same derivability relation. Hence N_0 and N_1 generate the same net-quantale. \square

Definition 7.6.4

An identity event is an event e of a net such that $\text{pre}(e) = \text{post}(e)$.

Definition 7.6.5

An

event s is a short cut if whenever the net N can evolve under s from marking m to marking m' , there exists some sequence of events $s_0; s_1; \dots; s_n$ which is disjoint from s , under which N can evolve from the marking m to the marking m' .

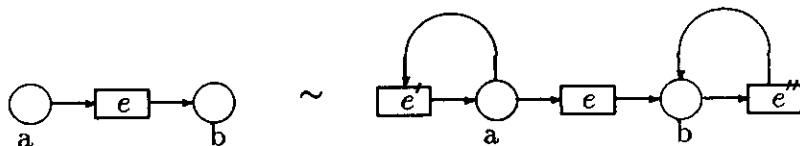
Since the derivability relation \leq is reflexive and transitive, Proposition 7.6.3 shows that two nets N_0 and N_1 are equivalent if they differ only in the presence or absence of identity events and short-cuts.

From a computational point of view, identity events are not distinguished because we are not interested in specifying actions which do not alter the net's state or environment. Also, failing to distinguish between a net with short-cuts and an otherwise identical net with all short-cuts removed only affects issues of computational complexity, which do not concern us here.

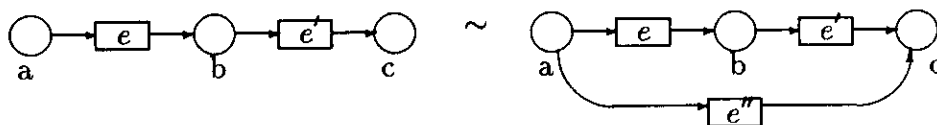
Remark 7.6.6 The quantale Q_N does not establish the order in which the events of a cycle in N occur, as is illustrated by Example 7.6.9 below.

7.6.1 Examples of Equivalent Nets

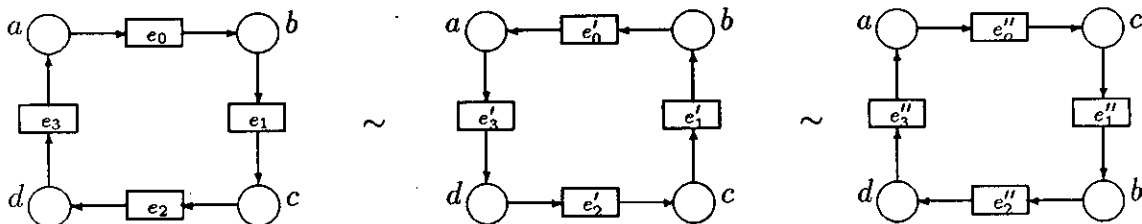
Example 7.6.7 Augmenting with identity events:



Example 7.6.8 Augmenting with short-cuts:



Example 7.6.9 Cycles:



In view of the discussion above, we see that a net-quantale determines its corresponding net up to the equivalence defined above. In particular, we have the following result:

Proposition 7.6.10

Let N and N' be nets without cycles or short-cuts. If $Q_N = Q_{N'}$, then $N = N'$.

7.7 Restrictions on the top element of a net–quantale

The top element of the quantale Q_N constructed from the net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ above is very large. Specifically, if $|\mathcal{B}| = \alpha$ and N is β -bounded, then the set $\mathbf{1}$ of possible markings of the net is of cardinality β^α . Thus $\mathbf{1}$ may be *finite* (if α and β are both finite), *countable* (if α is finite and β countable) or *uncountable* (if α is countable and $\beta \geq 2$).

In this section, we define a quantale with a smaller top element. Such a smaller quantale corresponds to a net’s behaviour on a subset P of markings, where elements of P are called “permitted markings”.

There are various ways in which the notion of permitted marking may be chosen. Some possibilities to consider are the restriction of markings to those which have:

- no more than n tokens on any one condition at once,
- no more than n tokens on a particular condition at once,
- no more than n_a tokens on condition A, n_b on condition B, and so on,
- no more than n tokens shared between some specified set of conditions,
- no more than n tokens altogether,
- no fewer than n tokens on any condition,
- or no fewer than n tokens in the marking altogether.

It turns out that all except the last two of these notions of permitted marking are suitable for constructing quantales.

In [Bro89a] a slightly different presentation was given of the following construction. The approach here applies a general theorem about quantales.

Definition 7.7.1 A notion of permitted marking on a net N is a subset P of $\text{Mark}(N)$ which contains the constant zero marking \emptyset . We say a marking of N is permitted if it is an element of P .

The first five classes of markings suggested at the start of this section contain \emptyset , and hence are notions of permitted marking. Thus one notion of permitted marking is the set of markings which have no more than k tokens on a condition at any time, for some integer k (this notion gives rise to quantales corresponding to the k -bounded nets). An example is the set of markings of a *safe* net (see [Rei85]), which have no more than one token on a condition at any time.

Definition 7.7.2 Let $N = \langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post} \rangle$ be a net and let P be a notion of permitted marking for N . A marking m of N has a one-step permitted evolution to marking m' , written $m' \leq_{1,P} m$, if m and m' are both permitted, and $m' \leq_1 m$.

Definition 7.7.3 There exists a permitted derivation of m_1 from m_2 if for $i \in \{1, \dots, n\}$, there exist permitted markings p_i such that $m_1 = p_1 \leq_{1,P} p_2 \cdots \leq_{1,P} p_n = m_2$.

Notation 7.7.4

We write $m_1 \leq_P m_2$ if there exists a permitted derivation of m_1 from m_2 .

As in section 7.2.1, we extend the definition of \leq_P to sets of markings, and define the operation \downarrow^P of closure under permitted evolution.

Lemma 7.7.5 Let Q_N be the net-quantale of a net N , and let P be a notion of permitted marking on N . Then $j(-) = (-) \cap P$ is a quantic conucleus on Q_N .

Proof: It is evident that j is a co-closure operator on Q_N .

Further,

$$(A \otimes B) \cap P$$

$$\begin{aligned}
&= \{m \in P \mid \exists a \in A \exists b \in B. ((a + b) \in P \text{ and } m \leq (a + b))\} \\
&\subseteq \{m \in P \mid \exists a \in (A \cap P) \exists b \in (B \cap P). ((a + b) \in P \text{ and } (m \leq (a + b)))\} \\
&= (A \cap P) \otimes (B \cap P).
\end{aligned}$$

Also, $I \subseteq (I \cap P)$ since P was defined to contain I . \square

Corollary 7.7.6 Let Q_N be the net-quantale of a net N , and let P be a notion of permitted marking on N . Let $j(-) = (-) \cap P$. Applying Theorem 7.2.15, we see that the image of Q_N under j is a subquantale of Q_N , in which

- elements are subsets of P closed under evolution of N ,
- the ordering is subset inclusion, \subseteq ,
- the monoid operation \otimes is given by $A \otimes_j B = (A \otimes B) \cap P$, and
- the unit of \otimes_j is I .

Notation 7.7.7 Let N be a net and let P be a notion of permitted marking on N . We have a conucleus on Q_N defined by $j(-) = (-) \cap P$.

We shall write \otimes_P for the monoid operation of the sub-quantale $j(Q_N)$ of Q_N .

We extend the operation \downarrow^P to sets of sets of markings in the evident way, writing $\downarrow^P \mathcal{P}(P)$ for the set of all subsets of P which are closed under permitted forwards evolution.

We shall denote the sub-quantale $j(Q_N)$ of Q_N by Q_N^P . Thus

$$Q_N^P = \langle \downarrow^P \mathcal{P}(P), \subseteq, \otimes_P, \downarrow^P \emptyset \rangle.$$

Q_N^P is the quantale corresponding to the permitted evolutions of the net N from permitted initial markings.

Remark 7.7.8 Whenever P has finitely many downwards closed subsets, the underlying set of Q_N^P has finitely many elements.

Example 7.7.9 Let N be a net which is safe when marked with any of the markings in a set S containing the constant zero marking \emptyset . Then $Q_N^{\downarrow S}$ is the net-quantale representing the behaviour of N on markings S .

Definition 7.7.10 Let N be a net and let P be a notion of permitted marking on N . An irrelevant event is an event e of N which is not enabled in any evolution of N from a marking in P .

A net N is irrelevance-free if it has no irrelevant events

Remark 7.7.11 A net-quantale Q_N^P does not indicate the presence in N of irrelevant events in any way. If the net N_0 with notion of permitted marking P_0 and the net N_1 with notion of permitted marking P_1 differ from one another only in the presence or absence of irrelevant events, then $Q_{N_0}^{P_0} = Q_{N_1}^{P_1}$.

Proposition 7.7.12 Let P be a notion of permitted marking on a net N . The quantale Q_N^P allows a sound interpretation of linear logic in the manner of Section 7.4.

Proof: As in Section 7.4. □

7.7.1 Arbitrary Sets of Initial Markings

In this section, we show how to construct a quantale which represents the behaviour of net N on an arbitrary set S_0 of initial markings. The natural object to take as the top element of such a quantale is $\downarrow S_0$. We wish to take $\downarrow S_0$ to be our notion of permitted marking for N . A notion of permitted marking must contain the constant zero marking \emptyset , and so the smallest notion of permitted marking containing the set S_0 is the set P_0 , given by

$$P_0 = \downarrow (S_0 \cup \{\emptyset\}).$$

Taking P_0 as our notion of permitted marking, we can construct in the manner of Section 7.7 the net-quantale $Q_N^{P_0}$ which corresponds to the behaviour of the net N on the set of markings S_0 .

Remark 7.7.13 Let N be a net and let S_0, S_1 be subsets of $\text{Mark}(N)$ with $S_0 \subseteq S_1$. If for $i = 0, 1$ we define $P_i = \downarrow(S_i \cup \{\emptyset\})$, then since $P_0 \subseteq P_1$ there is a conucleus j on $Q_N^{P_1}$ given by $j(-) = (-) \cap P_0$, and $Q_N^{P_0} = j(Q_N^{P_1})$. Thus $Q_N^{P_0}$ is a subquantale of $Q_N^{P_1}$.

Part IV

Evolution as Linear Proof

Introduction to Part IV

In Part IV, we establish a connection between Petri nets and certain formulae of linear logic, which was first presented in [Bro89b]. A corresponding connection between Petri nets and *theories* of the tensorial fragment of linear logic was explored in [Asp87], [GG89], and [MOM89]. One of the drawbacks of this approach is the lack of a definitive concept of linear theory and especially of the composition of linear theories. Formulae are preferred here to theories with the aim of better exploring problems of compositionality. It is hoped that extensions of the ideas set out here may lead to a better understanding of refinement, implementation of specifications, and composition of nets. Our slogan for this part, which epitomises the connection between Petri nets and linear logic, is

“Reachability in Petri nets corresponds to linear provability.”

It is a triviality that the tensorial fragment of linear logic corresponds to evolution of Petri nets. We could have described this correspondence with less machinery than we use here: if we restrict the relation S of Definition 8.4.7 to clauses (I) and (VI) and use a more restricted fragment of linear logic, we recapture the simple approach. The aim of this work, however, is to achieve a deeper understanding of applications of the major part of linear logic to Petri net description. An attempt has been made by Marti-Oliet and Meseguer [MOM89] to interpret connectives other than \otimes in the context of Petri nets. There, the intended meanings of these connectives are illustrated without definition or investigation of their applications. For a fuller understanding of these connectives, we should consider the proof theory of linear logic, which is indeed where many of its beauties lie. In Section 9.3.3 we consider these issues in greater depth.

The approach of Gunter and Gehlot in [GG90] has more profitable application to net theory in that it makes use of a result of proof theory to maximise the parallelism in a net's evolution. This is a genuine benefit of the translation of a problem to a different area, and illustrates our purpose in pursuing such a translation.

Chapter 8

Expressing Nets as Canonical Formulae

8.1 Chapter Summary

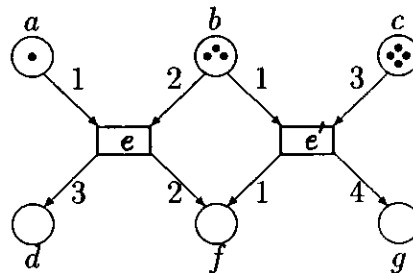
In this chapter, we work with the set \mathcal{F} of linear logic formulae involving connectives \otimes , \wedge , \multimap and $!$. We are concerned particularly with a subset **Can** of \mathcal{F} whose elements we show to be in bijection with the isomorphism classes of **MPetri**.

We generate an equivalence $=_S$ on \mathcal{F} using the reduction relation \longrightarrow_S . Intuitively, $F \longrightarrow_S G$ if the formulae F and G represent the same net, and G is shorter than F . We show that S is Church–Rosser and strongly normalising, and deduce that every formula in \mathcal{F} has a unique normal form with respect to \longrightarrow . Every formula in **Can** is in S -normal form.

8.2 The Formula Representing a Net

We shall first consider translating the Petri net of Example 8.2.1 into a formula of linear logic, and then generalise the process to any net.

Example 8.2.1 Our example of a net is taken from [MM88a]:



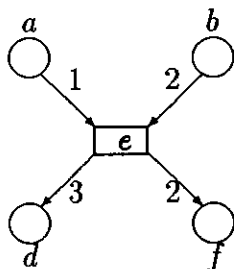
Let us regard a token in condition a as being a resource of type A . By convention we shall label conditions by small italic letters “ a ” and the corresponding resources by the corresponding italic capitals “ A ”. Then the net shows that we need one item of type A , and two of type B for a firing of e to occur. Such a combination of resources is represented by the linear logic formula

$$A \otimes B \otimes B.$$

We shall abbreviate this to $A \otimes 2B$. The net tells us that these resources can be used to produce 3 tokens of type D (that is, 3 tokens at condition d), and 2 tokens of type F . We could write this as the following formula of linear logic :

$$(A \otimes 2B) \multimap (3D \otimes 2F).$$

But this would represent the possibility of just one firing of e , whereas in fact we can fire e as often as we like, provided that each time the pre-conditions for e are satisfied. To express the persistence of the ability to fire e , we shall translate the net



by the formula

$$!((A \otimes 2B) \multimap (3D \otimes 2F)).$$

For ease of reading we omit the brackets around $A \otimes 2B$ and write the above instead as

$$!(A \otimes 2B \multimap 3D \otimes 2F).$$

In the same way, the other half of the Petri net of Example 8.2.1 can be expressed by :

$$!(B \otimes 3C \multimap F \otimes 4G).$$

These two formulae express the fact that firings of e and e' can occur whenever their pre-conditions are satisfied. In a similar way, we can represent the initial

marking by the formula

$$A \otimes 3B \otimes 4C.$$

Thus we can describe the entire net of Example 8.2.1, both its possible events and its initial marking, by the formula

$$A \otimes 3B \otimes 4C \otimes !(A \otimes 2B \multimap 3D \otimes 2F) \otimes !(B \otimes 3C \multimap F \otimes 4G)$$

8.3 Representing Arbitrary Nets as Formulae

We now define the set \mathcal{F} of linear logic formulae which will be of primary interest to us in this chapter.

Definition 8.3.1 *The set \mathcal{F} comprises the equivalence classes under permutation of those formulae in \mathbf{Lin} which are words over the following alphabet:*

parentheses “(” and “)”,

atoms a_0, a_1, \dots ,

the constant I ,

binary operators $\otimes, \wedge, \multimap$, and

a unary operator $!$.

Example 8.2.1 illustrated a way in which we can associate a formula of linear logic with a Petri net. Following this approach, we now define a function *form* from \mathbf{MPetri} to \mathcal{F} .

Definition 8.3.2 *A canonical formula is a formula $F \in \mathcal{F}$ of the form*

$$F = M \otimes \bigotimes_{j \in J} !(M_j \multimap M'_j) \otimes \bigotimes_{k \in K} !(M_k),$$

where J and K are (possibly empty) finite indexing sets, M, M_j, M'_j and M_k are multisets over $At(F)$ such that for each j , M_j is non-empty, and F has no repeated factors of the form $!(X \multimap Y)$.

Definition 8.3.3

Can is the set of equivalence classes of canonical formulae under permutation.

Convention 8.3.4 We have assumed that the set of linear atoms is countably infinite. We shall also assume the existence of a specified injective function $\ulcorner - \urcorner$, from any countable set \mathcal{B} to the set of linear atoms. We require that for any two sets \mathcal{B}_0 and \mathcal{B}_1 with non-empty intersection, the specified functions $\ulcorner - \urcorner^{\mathcal{B}_0}$ and $\ulcorner - \urcorner^{\mathcal{B}_1}$ associated with \mathcal{B}_0 and \mathcal{B}_1 respectively are equal on $\mathcal{B}_0 \cap \mathcal{B}_1$. To avoid cumbersome notation, we shall always write B for the atom $\ulcorner b \urcorner$, where $b \in \mathcal{B}$ (similarly $\ulcorner b_i \urcorner = B_i$, and so on).

The requirement that any element b in our universe of sets has a unique name $\ulcorner b \urcorner$ is necessary to ensure that Definition 9.3.1 be well-defined.

Remark 8.3.5 Let $N \in \text{MPetri}$. The function $\ulcorner - \urcorner$ from \mathcal{B} to the linear atoms determines uniquely a canonical name for each event in \mathcal{E} , since an event of a net in MPetri is determined uniquely by its pre- and post-condition sets.

Lemma 8.3.6 Let α be a finite multiset over a countable set \mathcal{B} . The assignment

$$\text{Multi}: \alpha \mapsto \bigotimes_{b \in \mathcal{B}} \alpha(b) \ulcorner b \urcorner$$

extends $\ulcorner - \urcorner$ linearly to an injective function from finite multisets over countable sets to the set of formulae of \mathcal{F} which are tensor sums of atoms.

Proof: Evident. □

Notation 8.3.7 Since *Multi* is an injection, in accordance with Convention 8.3.4 we shall often identify a multiset $M = \sum_i n_i a_i$ with the formula $\text{Multi}(M) = \bigotimes_i n_i A_i$.

Lemma 8.3.8 Let N be a Petri net. If $\mathcal{E} = \emptyset$ then we define

$$F = \text{Multi}(M_0).$$

Otherwise, $\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_1$, $\mathcal{E}_1 = \{e \in \mathcal{E} \mid \text{pre}(e) = \emptyset\}$, and we define

$$F = \text{Multi}(M_0) \otimes \bigotimes_{e \in \mathcal{E}_0} !(\text{Multi}(\text{pre}(e)) \multimap \text{Multi}(\text{post}(e))) \otimes \bigotimes_{e \in \mathcal{E}_1} !(\text{Multi}(\text{post}(e))).$$

The assignment

$$\langle \mathcal{E}, \mathcal{B}, \text{pre}, \text{post}, M_0 \rangle \mapsto F$$

defines a function, which we shall call *form*, from \mathbf{MPetri} to \mathbf{Can} .

Proof: Evident. □

Definition 8.3.9 Let N and N' be Petri nets. Then N is isomorphic to N' if and only if there exist bijections $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B} \rightarrow \mathcal{B}'$ such that for each $e \in \mathcal{E}$,

$$\text{pre}'(fe) = F(\text{pre}(e)), \quad \text{post}'(fe) = F(\text{post}(e)) \quad \text{and} \quad F(M) = M',$$

where F is extended linearly to multisets over \mathcal{B} .

Remark 8.3.10 Observe that this extends the notion of net isomorphism in the categories \mathbf{NSet} , $\mathbf{NSet}^{\text{co}}$, $\mathbf{NSet}_{\underline{2}}$ and $\mathbf{NSet}_{\underline{2}}^{\supseteq}$ of Part II of this thesis.

Lemma 8.3.11

The function *form* is surjective. Further, if $\text{form}(N) = \text{form}(N')$ then N is isomorphic to N' .

Proof: We show that any $F \in \mathbf{Can}$ is in the image of *form*. Further, we show that F determines up to net isomorphism a net N such that $\text{form}(N) = F$.

Let $F \in \mathbf{Can}$. Then F has the form

$$F = M \otimes \bigotimes_{j \in J} !(M_j \multimap M'_j) \otimes \bigotimes_{k \in K} !(M_k),$$

where M, M_j, M'_j and M_k are multisets over $\text{At}(F)$, the M_j are non-empty, and F has no repeated factors of the form $!(X \multimap Y)$.

For each factor $!(M_j \multimap M'_j)$ of F , the event label e_j of the event with pre-condition set M_j and post-condition set M'_j is uniquely determined, as we observed in Remark 8.3.5. Let $\mathcal{E}_0 = \{e_j \mid j \in J\}$.

For each factor $!(M_k)$ of F , the event label e_k of the event with empty pre-condition set and post-condition set M_k is uniquely determined. Let $\mathcal{E}_1 = \{e_k \mid k \in K\}$.

The event set \mathcal{E} of N is $\mathcal{E}_0 \cup \mathcal{E}_1$.

The condition set \mathcal{B} of N is $\mathcal{A}t(F)$.

Put $pre(e_j) = M_j$ and $post(e_j) = M'_j$ for $j \in J$ and

$$pre(e_k) = \emptyset \text{ and } post(e_k) = M_k \text{ for } k \in K.$$

This determines the pre- and post-condition relations of N .

The initial marking M of N is the multiset M .

The tuple $\langle \mathcal{E}, \mathcal{B}, pre, post, M \rangle$ is a Petri net N such that $form(N) = F$. F determines N up to the labelling of events and the assignment $\ulcorner - \urcorner$ of names to atoms. Hence F determines N up to net isomorphism. \square

Notation 8.3.12

We put $F_N = form(N)$ and call F_N the canonical formula representing N .

If F is a canonical formula, then we write N_F for the specified element of the equivalence class of the net N generated by F in the proof of Lemma 8.3.11. We call N_F the net represented by F .

Proposition 8.3.13 Two canonical formulae are isomorphic in the sense of Definition 3.4.10 precisely when the nets which they represent are isomorphic in the sense of Definition 8.3.9.

Proof: Follows routinely from the definitions. \square

Remark 8.3.14 *The set of canonical formulae forms a hierarchy in the following way (where the a_i are atoms, and for finite indexing sets I and J , M_i , M_j , and M'_j are tensor sums of atoms, the M_j being non-empty):*

$$\begin{aligned} \text{Can} ::= & a_i \mid \bigotimes_{i \in I} a_i \mid \bigotimes_{i \in I} !(M_i) \mid \bigotimes_{j \in J} (M_j \multimap M'_j) \mid \bigotimes_{i \in I} !(M_i) \otimes \bigotimes_{j \in J} (M_j \multimap M'_j) \\ & \mid M \otimes \bigotimes_{i \in I} !(M_i) \otimes \bigotimes_{j \in J} (M_j \multimap M'_j) . \end{aligned}$$

In terms of nets, this hierarchy amounts to

$$\begin{aligned} & \text{conditions} \mid \text{markings} \mid \text{events with empty pre-conditions} \\ & \mid \text{events with non-empty pre-conditions} \mid \text{nets} \mid \text{marked nets} . \end{aligned}$$

Thus approaching the theory of Petri nets from the point of view of their analogy with linear logic formulae suggests that we can regard nets and markings as examples of objects of the same nature. It is then feasible to consider nets as markings, that is, as resources which can be changed by the evolution of a controlling net.

8.4 A Reduction Relation S on \mathcal{F}

We now define an equivalence $=_S$ on formulae in \mathcal{F} .

The development of this section follows [Bar85].

Definition 8.4.1 *Let F be a formula in \mathcal{F} . We define a function $\| - \|$ from \mathcal{F} to \mathbb{N} inductively as follows:*

- $\|a\| = 1$ for atoms a ,
- $\|I\| = 1$,
- $\|(f \otimes f')\| = \|f\| + \|f'\| + 1$,
- $\|(f \wedge f')\| = \|f\| + \|f'\| + 1$,

- $\|(f \circ f')\| = \|f\| + \|f'\| + 1$ and
- $\|!f\| = \|f\| + 1$.

We call $\|F\|$ the length of F .

Remark 8.4.2

$\|F\|$ is the number of symbols in F which are not parentheses.

Definition 8.4.3 A binary relation R on \mathcal{F} is compatible with the set of operators $\{\otimes, \wedge, \circ, !\}$ if for all $M, N, M', N' \in \mathcal{F}$,

$$\frac{(M, M') \in R \quad (N, N') \in R}{((M \circ N), (M' \circ N')) \in R} \quad (\circ \text{ one of } \otimes, \wedge, \text{ and } \circ) \quad \text{and}$$

$$\frac{(M, M') \in R}{(!M, !M') \in R}$$

Definition 8.4.4 A reduction relation on \mathcal{F} is a binary relation on \mathcal{F} which is reflexive, transitive and compatible with the operators $\{\otimes, \wedge, \circ, !\}$.

Definition 8.4.5 Let R be a binary relation on \mathcal{F} . Then R induces three binary relations on \mathcal{F} . These are

\longrightarrow_R one step R -reduction,

\twoheadrightarrow_R R -reduction and

$=_R$ R -equivalence

defined inductively as follows:

\longrightarrow_R is the compatible closure of R , that is,

$$\frac{(M, N) \in R}{M \longrightarrow_R N}, \quad \frac{M \longrightarrow_R M'}{!M \longrightarrow_R !M'} \quad \text{and}$$

$$\frac{M \longrightarrow_R M' \quad N \longrightarrow_R N'}{(M \circ N) \longrightarrow_R (M' \circ N')} \quad (\circ \text{ one of } \otimes, \wedge, \text{ and } \circ).$$

\longrightarrow_R is the reflexive, transitive closure of \rightarrow_R , that is,

$$\frac{M \rightarrow_R N}{M \longrightarrow_R N}, \quad \frac{}{M \longrightarrow_R M} \quad \text{and} \quad \frac{M \rightarrow_R N \quad N \longrightarrow_R L}{M \longrightarrow_R L}.$$

$=_R$ is the equivalence relation generated by \longrightarrow_R , that is,

$$\frac{M \longrightarrow_R N}{M =_R N}, \quad \frac{M =_R N}{N =_R M} \quad \text{and} \quad \frac{M =_R N \quad N =_R L}{M =_R L}.$$

Lemma 8.4.6 Let R be a binary relation on \mathcal{F} . Then \longrightarrow_R is compatible with the operators \otimes , \wedge , \rightarrow and $!$.

Proof: \longrightarrow_R is compatible with the operators by definition.

That \longrightarrow_R is also compatible follows from the compatibility of \rightarrow_R , by structural induction. \square

We are here interested in one particular relation S on \mathcal{F} . As we shall see in Section 9.4, S is chosen such that whenever $M \longrightarrow_S N$, then M and N are linear logic formulae representing the same Petri net, and $\|N\| < \|M\|$.

Definition 8.4.7 S is a binary relation on \mathcal{F} consisting of pairs

Form (I) $(!(M) \otimes !(M) , !(M)),$

Form (II) $(!(M) \otimes (M \wedge I) , !(M)),$

Form (III) $(!(M \rightarrow N) \otimes !(((M \rightarrow N) \wedge I) \otimes M) \rightarrow N , !(M \rightarrow N)),$

Form (IV) $(!((M \wedge I) \rightarrow I) , I),$

Form (V) $(M \wedge M , M)$

Form (VI) $(M \otimes I , M)$

Form (VII) $(I \rightarrow M , M)$

Form (VIII) $(!(M) \otimes !((M \wedge I) \rightarrow M) , !(M))$

Form (IX) (!I) , I)

for all $M, N \in \mathcal{F}$, and no other pairs.

Notation 8.4.8 Henceforth we shall omit the subscript or prefix S . Thus we write \longrightarrow and \longrightarrow for \longrightarrow_S and \longrightarrow_S respectively, understanding that the intended relation is the relation S of Definition 8.4.7.

The intuition behind our definition of S is that formulae, like Petri nets, represent various means of transforming resources. For most of the forms of Definition 8.4.7, it is evident that the two formulae convey the same information about the transformation of resources. For example, in the case of Form (V), a deterministic choice between M and M transforms resources in exactly the same way as M . Proposition 8.4.12 shows that such an interpretation can be given to any of the elements of S .

Notation 8.4.9 Let $X, X', F \in \text{Lin}$. Let X be an occurrence of a subformula in F . We write $F[X'/X]$ for the formula obtained from F by substituting X' for X in F . Notice that we only substitute for one occurrence of the subformula.

Lemma 8.4.10 If $(P, Q) \in S$ then $P \dashv\vdash Q$ in \mathcal{L} .

Proof: If (P, Q) is of Form (I), we derive $P \vdash Q$ as follows:

$$\frac{}{!M, !M \vdash !M \otimes !M} \text{ (Id)}$$

$$\frac{}{!M \vdash !M \otimes !M} \text{ (Cont)}$$

We derive $Q \vdash P$ as follows:

$$\frac{}{!M \vdash !M} \text{ (Id)}$$

$$\frac{}{!M \otimes !M \vdash !M} \text{ (Weak), } (\otimes L)$$

If (P, Q) is of Form (II) we derive $P \vdash Q$ as follows:

$$\begin{array}{c}
 \frac{}{!(M) \vdash !(M)} \text{ (Id)} \quad \frac{}{I \vdash} \text{ (IL)} \\
 \hline
 \frac{}{!(M), I \vdash !(M)} \text{ (\otimes R)} \\
 \hline
 \frac{}{!(M), I \vdash !(M)} \text{ (\wedge L2), (\otimes L)} \\
 \hline
 !(M) \otimes (M \wedge I) \vdash !(M)
 \end{array}$$

We derive $Q \vdash P$ as follows:

$$\begin{array}{c}
 \frac{}{M \vdash M} \text{ (Id)} \quad \frac{}{\vdash I} \text{ (IR)} \\
 \hline
 \frac{}{!(M) \vdash M} \text{ (Der)} \quad \frac{}{!(M) \vdash I} \text{ (Weak)} \\
 \hline
 \frac{}{!(M) \vdash !(M)} \text{ (Id)} \quad \frac{}{!(M) \vdash (M \wedge I)} \text{ (\wedge R)} \\
 \hline
 \frac{}{!(M) \vdash !(M)} \quad \frac{}{!(M) \vdash (M \wedge I)} \text{ (\otimes R)} \\
 \hline
 \frac{}{!(M), !(M) \vdash !(M) \otimes (M \wedge I)} \text{ (Cont)} \\
 \hline
 !(M) \vdash !(M) \otimes (M \wedge I)
 \end{array}$$

If (P, Q) is of Form (III), we derive $P \vdash Q$ by weakening.

We derive $Q \vdash P$ as follows:

$$\begin{array}{c}
 \frac{}{} \text{ (Imp)} \\
 \hline
 (M \multimap N) \otimes M \vdash N \quad \text{ (\wedge L1)} \\
 \hline
 ((M \multimap N) \wedge I) \otimes N \vdash N \quad \text{ (\multimap R)} \\
 \hline
 \vdash !(((M \multimap N) \wedge I) \otimes M) \multimap N \quad \text{ (!R)} \quad \frac{}{} \text{ Id} \\
 \hline
 \vdash !(((M \multimap N) \wedge I) \otimes M) \multimap N \quad \frac{}{!(M \multimap N) \vdash !(M \multimap N)} \text{ (\otimes R)} \\
 \hline
 !(M \multimap N) \vdash !(M \multimap N) \otimes !(((M \multimap N) \wedge I) \otimes M) \multimap N
 \end{array}$$

If (P, Q) is of Form (IV), we derive $P \vdash Q$ by weakening.

We derive $Q \vdash P$ as follows:

$$\begin{array}{c}
\frac{}{I \vdash I} \text{Id} \\
\frac{I \vdash I}{M \wedge I \vdash I} (\wedge \text{L2}) \\
\frac{M \wedge I \vdash I}{\vdash (M \wedge I) \multimap I} (\multimap \text{R}) \\
\frac{\vdash (M \wedge I) \multimap I}{\vdash !((M \wedge I) \multimap I)} (!\text{R}) \quad \frac{}{I \vdash} (\text{IL}) \\
\frac{\vdash !((M \wedge I) \multimap I) \quad I \vdash}{I \vdash !((M \wedge I) \multimap I)} (\otimes \text{R})
\end{array}$$

If (P, Q) is of Form (V), the derivations are immediate, using the rules $(\wedge \text{L1})$, $(\wedge \text{R})$ and (Id) .

If (P, Q) is of Form (VI) the derivations are immediate, using the rules (IL) , (IR) , $(\otimes \text{L})$ and $(\otimes \text{R})$.

If (P, Q) is of Form (VII), we derive $P \vdash Q$ using the rule $(\multimap \text{L})$, and $Q \vdash P$ using the rule $(\multimap \text{R})$.

If (P, Q) is of Form (VIII), we derive $P \vdash Q$ using weakening, and $Q \vdash P$ as follows:

$$\begin{array}{c}
\frac{}{M \vdash M} (\text{Id}) \\
\frac{M \vdash M}{M \wedge I \vdash M} (\wedge \text{L2}) \\
\frac{M \wedge I \vdash M}{\vdash (M \wedge I) \multimap M} (\multimap \text{R}) \\
\frac{}{\vdash (M \wedge I) \multimap M} (\text{Derel}) \\
\frac{\vdash (M \wedge I) \multimap M}{\vdash !((M \wedge I) \multimap M)} (!\text{R}) \\
\frac{\vdash !((M \wedge I) \multimap M) \quad !M \vdash !M}{!M \vdash !((M \wedge I) \multimap M)} (\otimes \text{R})
\end{array}$$

If (P, Q) is of Form (IX), the derivations follow from weakening and dereliction. This completes the proof. \square

Lemma 8.4.11 If $X, F \in \mathbf{Lin}$, with $X \leq F$ and $X \longrightarrow X'$ then $F \dashv\vdash F[X'/X]$.

Proof:

We assume the result holds for formulae shorter than F , and proceed by structural induction on F .

The base case is proved in Lemma 8.4.10.

As an example of the inductive step, if it is the case that $F = M \multimap N$ and $X \leq M$, then under the assumption that $M[X'/X] \vdash M$, we derive $F \vdash F[M'/M]$ as follows:

$$\begin{array}{c}
 \frac{}{M[X'/X] \vdash M} \quad \frac{}{N \vdash N} \quad (\text{Id}) \\
 \hline
 \frac{}{M[X'/X] \vdash M} \quad \frac{}{N \vdash N} \quad (\multimap\text{L}) \\
 \hline
 \frac{(M \multimap N) \otimes M[X'/X] \vdash N}{M \multimap N \vdash M[X'/X] \multimap N} \quad (\multimap\text{R})
 \end{array}$$

Similarly, the assumption of $M \vdash M[X'/X]$ allows us to derive $F[M'/M] \vdash F$.

The other cases are similar. \square

Corollary 8.4.12 If $F \longrightarrow G$ then $F \dashv\vdash G$ in \mathcal{L}_1 .

and reflexive,

Proof: Since \vdash is transitive, it suffices to prove the result for \longrightarrow . Lemma 8.4.11 shows that it is sufficient that if $(P, Q) \in S$, then $P \dashv\vdash Q$. This was shown in Lemma 8.4.10. \square

Remark 8.4.13 We shall define S -equivalent formulae to represent the same net. As we shall show in Section 9.6.2, linear logic proof corresponds precisely to Petri net evolution. Thus the result of Corollary 8.4.12 is fundamental to our approach.

8.4.1 S is Strongly Normalising

Definition 8.4.14

If $M \longrightarrow_R N$ then we say M R -reduces to N or N is an R -reduct of M .

If $M \longrightarrow_R N$ then we say M R -reduces to N in one step.

If $M =_R N$ we say M is R -equivalent to N .

Definition 8.4.15

Let R be a relation on \mathcal{F} . Then

- an R -redex is a formula M such that $(M, N) \in R$. Then N is the contractum of M ,
- a formula M is in R -normal form if no subformula of M is an R -redex, and
- a formula M has R -normal form N if N is in R -normal form and $M =_R N$.

We call the process of going from an R -redex to its contractum *contraction*.

Remark 8.4.16 Every canonical formula is in S -normal form, since it contains no redexes.

Remark 8.4.17 Two formulae which are inter-derivable in \mathcal{L} may have different normal forms. For example, for any formula X of \mathcal{F} which is in S -normal form, the formulae

$$!(A \multimap B) \quad \text{and} \quad !(A \multimap B) \otimes !((X \wedge I) \otimes A) \multimap B$$

are interderivable in \mathcal{L}_1 , and both are in S -normal form.

Thus $=_S$ is a finer equivalence than $\dashv\vdash_{\mathcal{L}}$.

Definition 8.4.18

1. Let Δ and Δ' be S -redexes, and let F be a formula in \mathcal{F} such that $\Delta \leq F$ and $\Delta' \leq F$. An occurrence of Δ and an occurrence of Δ' overlap if

- the occurrence of Δ is not a subformula of the occurrence of Δ' ,
- the occurrence of Δ' is not a subformula of the occurrence of Δ , and

- the occurrences of Δ and Δ' do not appear in the form $\Delta \otimes \Delta'$.

2. Further, we say an occurrence of Δ and an occurrence of Δ' overlap at G if the occurrence of Δ and the occurrence of Δ' overlap, and G is a subformula of F such that $G|\Delta$ and $G|\Delta'$.

Example 8.4.19

Let $F = !(X \multimap Y) \otimes !(X \multimap Y) \otimes (Z \wedge Z) \otimes ((X \multimap Y) \wedge I)$.

Let $\Delta = !(X \multimap Y) \otimes !(X \multimap Y)$ and let $\Delta' = !(X \multimap Y) \otimes ((X \multimap Y) \wedge I)$.

Then Δ and Δ' overlap at $!(X \multimap Y)$, whichever occurrences of the redexes we consider.

The redex $(Z \wedge Z)$ does not overlap with any other redex occurrence in F . Notice that, even if we replace Z by $!(X \multimap Y)$, the redex $(!(X \multimap Y) \wedge !(X \multimap Y))$ does not overlap with any other redex occurrence in F .

Recall that we are identifying a formula with its equivalence class under permutation, by convention 3.4.12. If we do not do so, then we must interpret $P \longrightarrow Q$ as meaning that Q is obtained from P by a finite (possibly empty) series of contractions and transpositions.

Lemma 8.4.20 Let M be a formula in R -normal form. Then

- there exists no N such that $M \longrightarrow_R N$, and
- if $M \longrightarrow_R N$ then M is a permutation of N .

Proof: Immediate. □

Definition 8.4.21

1. Let Δ be a subformula of a formula M . If Δ is an R -redex with contractum Δ' and N is the formula obtained by replacing a single occurrence of Δ in M by Δ' , then we shall write

$$M \xrightarrow{\Delta}_R N.$$

2. A reduction path is a finite or infinite sequence

$$\sigma = M_0 \xrightarrow{\Delta_0}_R M_1 \xrightarrow{\Delta_1}_R M_2 \xrightarrow{\Delta_2}_R \dots$$

Convention 8.4.22

- σ, τ range over reduction paths.
- The reduction path of Definition 8.4.21.2 starts with M_0 . If there is a last term M_n in σ then σ ends with M_n . In this case we say that σ is a reduction path from M_0 to M_n .
- The labels $\Delta_0, \Delta_1 \dots$ may be left out in denoting a reduction path.
- We shall write $\sigma: M_0 \longrightarrow M_1 \longrightarrow \dots$ to indicate that σ is the path $M_0 \xrightarrow{\Delta_0}_R M_1 \xrightarrow{\Delta_1}_R \dots$.
- If σ is a reduction path, then $\|\sigma\|$ is its length, that is, the number of $\xrightarrow{\Delta}_R$ steps in it. Note that $\|\sigma\|$ may be infinite.

Example 8.4.23

Let Δ and Γ be formulae in \mathcal{F} . Let Δ be an S -redex with contractum Δ' , and let M be the formula $(\Delta \wedge \Gamma) \otimes I$. Then

$$M \xrightarrow{M} \Delta \wedge \Gamma, \quad \text{and} \quad M \xrightarrow{\Delta} (\Delta' \wedge \Gamma) \otimes I$$

Remark 8.4.24 We cannot always recover Δ from the M and N in $M \xrightarrow{\Delta}_R N$. For example,

$$X \otimes I \otimes I \xrightarrow{X \otimes I} X \otimes I \quad \text{and} \quad X \otimes I \otimes I \xrightarrow{I \otimes I} X \otimes I.$$

Definition 8.4.25 Let M be a formula in \mathcal{F} and R a binary relation on \mathcal{F} .

- M R -strongly normalises if every R -reduction starting with M has finite length.
- R is strongly normalising if every $M \in \mathcal{F}$ R -strongly normalises.

Proposition 8.4.26 S is strongly normalising.

Proof: Let M be a formula in \mathcal{F} . We proceed by induction on $\|M\|$.

If $\|M\| = 1$ then M is either an atom or a constant, and so is not a redex. Hence the maximum possible length of a reduction path starting with M is 0, and M strongly normalises.

Suppose every formula N with $\|N\| < n$ strongly normalises. By the definition of S , for every $L \in \mathcal{F}$ such that $M \longrightarrow L$, we have $\|L\| < \|M\|$. By the inductive hypothesis, L strongly normalises.

Hence M strongly normalises.

□

8.4.2 S is Church–Rosser

Definition 8.4.27

- Let R be a binary relation on \mathcal{F} . Then R satisfies the diamond property if for any $M \in \mathcal{F}$, whenever $(M, M_1) \in R$ and $(M, M_2) \in R$, then there exists a formula $L \in \mathcal{F}$ such that $(M_1, L) \in R$ and $(M_2, L) \in R$.
- A binary relation R on \mathcal{F} is Church–Rosser if \longrightarrow_R satisfies the diamond property.

Definition 8.4.28

- Let R be a binary relation on \mathcal{F} . Then R satisfies the weak diamond property if for any $M \in \mathcal{F}$, whenever $(M, M_1) \in R$ and $(M, M_2) \in R$, there exists a formula $L \in \mathcal{F}$ such that $(M_1, L) \in R^*$ and $(M_2, L) \in R^*$, where R^* is the reflexive, transitive closure of R .
- A binary relation R on \mathcal{F} is weakly Church–Rosser if \longrightarrow_R satisfies the weak diamond property.

Theorem 8.4.29 S is weakly Church–Rosser.

Proof:

We show that for all $M \in \mathcal{F}$,

$$(M \longrightarrow N_0 \text{ and } M \longrightarrow N_1) \Rightarrow \exists L.(N_0 \longrightarrow L \text{ and } N_1 \longrightarrow L)$$

We consider the possible pairs of contractions which could have produced N_0 and N_1 . We suppose $M \xrightarrow{\Delta_0} N_0$ and $M \xrightarrow{\Delta_1} N_1$, and that $\Delta_0 \longrightarrow \Delta'_0$ and $\Delta_1 \longrightarrow \Delta'_1$.

There are three possible situations:

- (i) Δ_0 and Δ_1 occur in the form $\Delta_0 \otimes \Delta_1$,
- (ii) Δ_0 and Δ_1 overlap, or
- (iii) we have $\Delta_0 \leq \Delta_1$ or $\Delta_1 \leq \Delta_0$.

If we are in the first situation, the two contractions are independent of one another and we have

$$\begin{array}{ccc} M & \xrightarrow{\Delta_1} & N_1 \\ \Delta_0 \downarrow & & \downarrow \Delta_0 \\ N_0 & \xrightarrow{\Delta_1} & L \end{array}$$

If we are in the second situation, neither redex can be of Form (IV), (V) or (VII), as none of these can overlap with another redex. For most other pairs or reductions, the overlap makes little difference and as before, we have

$$\begin{array}{ccc} M & \xrightarrow{\Delta_1} & N_1 \\ \Delta_0 \downarrow & & \downarrow \Delta_0 \\ N_0 & \xrightarrow{\Delta_1} & L \end{array}$$

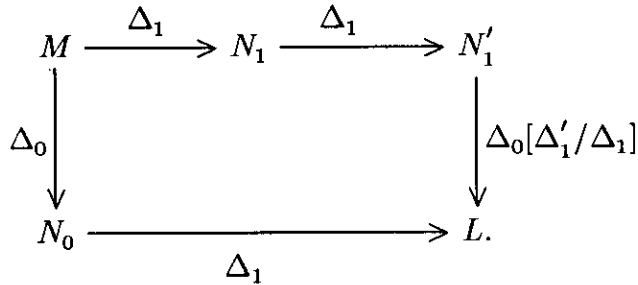
There are certain pairs of redexes which interact in a slightly more complex way whenever the formula M of one redex is an instance of the constant I . This can only occur if one redex is of Form (II) and one of Form (III), or one of Form (I) and one of Form (IX), or one of Form (VIII) and one of Form (IX). For instance if $\Delta_0 = !(I) \otimes !(I)$ and $\Delta_1 = !(I) \otimes !((I \wedge I) \otimes I \multimap I)$, then the redexes are of Forms (II) and (III) respectively, and overlap at I . It is readily seen that both Δ_0 and Δ_1 reduce by a finite contraction path to I , and thus we have

$$M \xrightarrow{\Delta_0} N_0 \twoheadrightarrow I \quad \text{and} \quad M \xrightarrow{\Delta_1} N_1 \twoheadrightarrow I.$$

We put $L = M[I/\Delta_0]$.

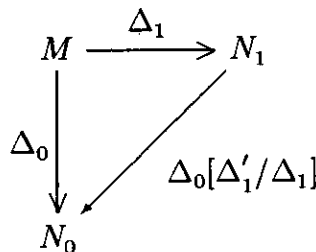
In the third situation, we can assume without loss of generality that $\Delta_1 \leq \Delta_0$. Δ_0 cannot be of Form (IX).

If Δ_0 is of Form (I) or (V), then we have



The above diagram also describes the situation where Δ_0 is of Form (II) and Δ_1 is of any Form other than (IX). In this last case, $\Delta_0 = !(I) \otimes !(I \wedge I \multimap I)$, and it can be shown that $\Delta_0 \twoheadrightarrow I$ and $\Delta_0[I/!(I)] \twoheadrightarrow I$. We put $L = M[I/\Delta_0]$.

If Δ_0 is of Form (IV), then we have



If Δ_0 is of Form (VI) or (VII), then we have

$$\begin{array}{ccc}
 M & \xrightarrow{\Delta_1} & N_1 \\
 \Delta_0 \downarrow & & \downarrow \Delta_0[\Delta'_1/\Delta_1] \\
 N_0 & \xrightarrow{\Delta_1} & L.
 \end{array}$$

If Δ_0 is of Form (III) or Form (VIII) then in all cases but two we have

$$\begin{array}{ccccccc}
 M & \xrightarrow{\Delta_1} & N_1 & \xrightarrow{\Delta_1} & N'_1 & \xrightarrow{\Delta'_1} & N''_1 \\
 \Delta_0 \downarrow & & & & & & \downarrow \Delta_0[\Delta'_1/\Delta_1] \\
 N_0 & \xrightarrow{\Delta_1} & & & & & L.
 \end{array}$$

In the case where $\Delta_0 = !(I \multimap M) \otimes !(((I \multimap M) \wedge I) \otimes I) \multimap M$, the any one-step reduction of Δ_0 can be extended to a reduction path from Δ_0 to $!(M)$, and we put $L = M[!(M)/\Delta_0]$.

In the case where $\Delta_0 = !(I) \otimes !(I \wedge I \multimap I)$, then we put $L = M[I/\Delta_0]$.

The result follows. □

We now mention a result of Newman [New42].

Proposition 8.4.30 Let R be a binary relation on \mathcal{F} . If R is strongly normalising and weakly Church–Rosser, then R is Church–Rosser.

Corollary 8.4.31 S is Church–Rosser.

Corollary 8.4.32 Every formula in \mathcal{F} has a unique normal form, and so is equivalent to at most one canonical formula.

Chapter 9

Reachability and Provability

9.1 Introduction

We have shown that the canonical formulae **Can** correspond precisely to the marked Petri nets **MPetri**. We shall use this correspondence to give a semantics to canonical formulae in terms of marked Petri nets, and show that this semantics is sound and complete with respect to the fragment \mathcal{L}_0 of linear logic. We can extend the semantics to the set of formulae in \mathcal{F} which are S -equivalent to canonical formulae, and it remains sound and complete. We consider extending the semantics to a larger subset of \mathcal{F} , and show that, while the interpretation of linear implication in terms of nets is problematic, we can give a semantics to \wedge which is complete and sound where defined. We show further that if F_N and $F_{N'}$ are canonical formulae interpreted by the nets N and N' respectively, and $F_N \vdash F_{N'}$ in \mathcal{L}_0 , there is a canonical proof of $F_N \vdash F_{N'}$ in \mathcal{L}_0 in which applications of the (Cut) rule always reflect a causal dependency in the net N .

9.2 A Preorder on Nets

Definition 9.2.1 *Let N_0 and N_1 be marked Petri nets. We write $N_0 \sqsupseteq N_1$ if*

- *there exists a subset \mathcal{E}_1^* of \mathcal{E}_1 such that for each element e of \mathcal{E}_1^* , $\text{pre}(e) = \emptyset$, the marking $\text{post}(e)$ is reachable in N_0 from the empty marking, and further,*

$$\mathcal{E}_1 \subseteq \mathcal{E}_0 \cup \mathcal{E}_1^*,$$

- $\mathcal{B}_1 = \mathcal{B}_0 \setminus \{b \in \mathcal{B}_0 \mid b \notin_m M_1 \ \& \ \nexists e \in \mathcal{E}_1.(b \in \text{pre}(e) + \text{post}(e))\}$ and
- the marking M_0 can evolve in N_0 to the marking M_1 of N_1 .

That is, $N_0 \sqsupseteq N_1$ if there exists some marking M reachable by N_0 such that N_1 is a subnet of an augmentation of the net N_0 with marking M by certain events with empty pre-conditions.

We extend the ordering \sqsupseteq to sets of nets in the following way,

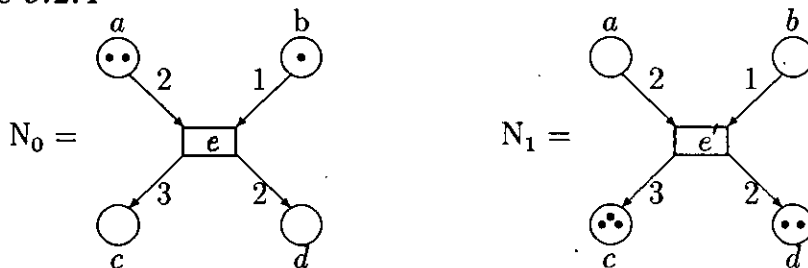
Definition 9.2.2 *If \mathcal{N}_0 and \mathcal{N}_1 are sets of Petri nets, then we write $\mathcal{N}_0 \sqsupseteq \mathcal{N}_1$ if for every $N_1 \in \mathcal{N}_1$, there exists $N_0 \in \mathcal{N}_0$ such that $N_0 \sqsupseteq N_1$.*

Remark 9.2.3 \sqsupseteq is not anti-symmetric, and thus not a partial order. Consider the net N consisting of one event, with pre-condition a and post-condition b . Let N_0 be N with marking a , and let N_1 be N with marking b . Then $N_0 \sqsupseteq N_1$ and $N_1 \sqsupseteq N_0$, but it is not the case that $N_0 = N_1$.

9.2.1 Examples of the Preorder \sqsupseteq

We give here some examples of nets related by \sqsupseteq , and the canonical formulae representing them. We also show how Theorem 9.6.2 applies to these examples. Theorem 9.6.2 shows that markings reachable by a net N correspond to formulae provable in the fragment of linear logic \mathcal{L}_0 from the canonical formula representing N .

Example 9.2.4



$$F_{N_0} = 2A \otimes B \otimes !(2A \otimes B \multimap 3C \otimes 2D) \quad \text{and}$$

$$F_{N_1} = 3C \otimes 2D \otimes !(2A \otimes B \multimap 3C \otimes 2D)$$

In this case, $N_0 \supseteq N_1$.

N_1 is obtained from N_0 by a single firing of the event e .

Notice that we can prove that

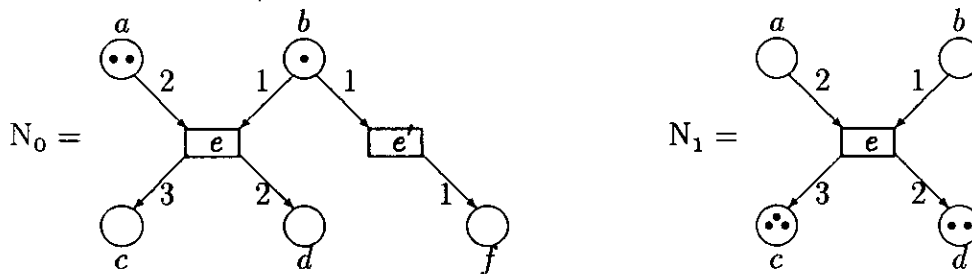
$$2A \otimes B \otimes !(2A \otimes B \multimap 3C \otimes 2D) \vdash 3C \otimes 2D \otimes !(2A \otimes B \multimap 3C \otimes 2D)$$

The proof that F_{N_1} can be derived from F_{N_0} in \mathcal{L}_0 proceeds as follows (putting E for $(2A \otimes B \multimap 3C \otimes 2A)$):

$$\frac{\frac{\frac{}{} \text{(Imp)}}{2A \otimes B \otimes (E \wedge I) \vdash 3C \otimes 2D} \quad \frac{}{} \text{(Identity)}}{!E \vdash !E} \text{(\otimes R)}}{2A \otimes B \otimes !E \otimes (E \wedge I) \vdash 3C \otimes 2D \otimes !E} \text{(Der)}$$

$$2A \otimes B \otimes E \vdash 3C \otimes 2D \otimes !E$$

Example 9.2.5



$$F_{N_0} = 2A \otimes B \otimes !(B \multimap F) \otimes !(2A \otimes B \multimap 3C \otimes 2D) \quad \text{and}$$

$$F_{N_1} = 3C \otimes 2D \otimes !((2A \otimes B) \multimap (3C \otimes 2D))$$

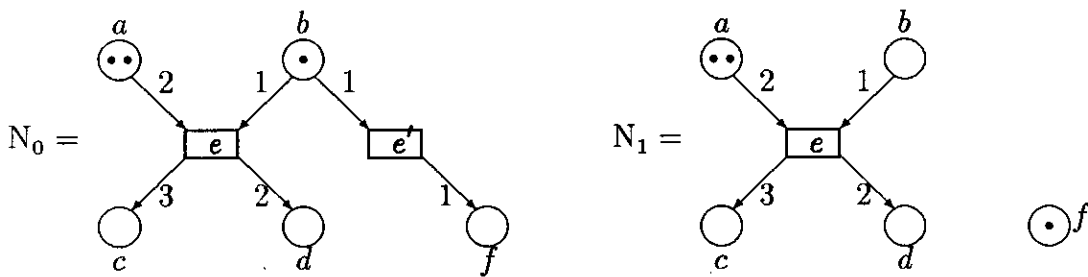
We have $N_0 \sqsupseteq N_1$.

Again, we show how F_{N_1} can be derived from F_{N_0} in \mathcal{L}_0 (as before, putting E for $(2A \otimes B \multimap 3C \otimes 2D)$):

$$\frac{\frac{\frac{}{} \text{(Imp)}}{2A \otimes B \otimes (E \wedge I) \vdash 3C \otimes 2D} \text{(Weak)} \quad \frac{}{} \text{(Identity)}}{2A \otimes B \otimes !(B \multimap F) \otimes (E \wedge I) \vdash 3C \otimes 2D} \quad !E \vdash !E \text{ (\otimes R)}}{2A \otimes B \otimes (E \wedge I) \otimes !E \otimes !(B \multimap F) \vdash 3C \otimes 2D \otimes !(B \multimap F) \otimes !E} \text{(Der)}}{2A \otimes B \otimes !E \otimes !(B \multimap F) \vdash 3C \otimes 2D \otimes !E}$$

Here, N_1 is obtained from N_0 by a firing of e , followed by the removal of the event e' .

Example 9.2.6



$$2A \otimes B \otimes !(B \multimap F) \otimes !(2A \otimes B \multimap 3C \otimes 2D) \quad \text{and}$$

$$2A \otimes F \otimes !(2A \otimes B \multimap 3C \otimes 2D)$$

We have $N_0 \sqsupseteq N_1$.

Again, it is easy to show that F_{N_1} can be derived from F_{N_0} in \mathcal{L}_0 . In this case, N_1 is obtained from N_0 by a firing of event e' , followed by the removal of e' . In this case, the condition f is not removed, because it is marked.

Example 9.2.7

$$N_0 = \boxed{} \rightarrow \bigcirc^a \quad N_1 = \bigcirc^a \bullet$$

Here, $F_{N_0} = !(A)$ and $F_{N_1} = nA$. Again, $N_0 \sqsupseteq N_1$ and $F_{N_0} \vdash_{\mathcal{L}_1} F_{N_1}$.

9.3 Composing Nets

We have expressed nets as formulae of linear logic, and hope to give a meaning in terms of nets of those connectives of linear logic which apply to such formulae. Where N_0 and N_1 are marked nets, we shall give a meaning to the formulae $F_{N_0} \otimes F_{N_1}$, $F_{N_0} \wedge F_{N_1}$ and $!(F_{N_0})$, thus defining the composite nets $N_0 \otimes N_1$, $N_0 \wedge N_1$ and $!(N_0)$. We also consider what meaning, if any, can be given to other connectives of linear logic when they are applied to nets.

9.3.1 The Action of \otimes on Nets

Definition 9.3.1 Let N_F be a net represented by the formula F , and N_G the net represented by G . Then the composite net $N = N_F \otimes N_G$ is given as follows:

- $\mathcal{E} = \mathcal{E}_F \cup \mathcal{E}_G$,
- $\mathcal{B} = \mathcal{B}_F \cup \mathcal{B}_G$,
- $pre = pre_F \cup pre_G$,
- $post = post_F \cup post_G$ and

- $M = M_F + M_G$.

Lemma 9.3.2 Let N_F be a net represented by the formula F , and N_G the net represented by G . The composite net $N_F \otimes N_G$ is the net represented by the formula $F \otimes G$. Thus

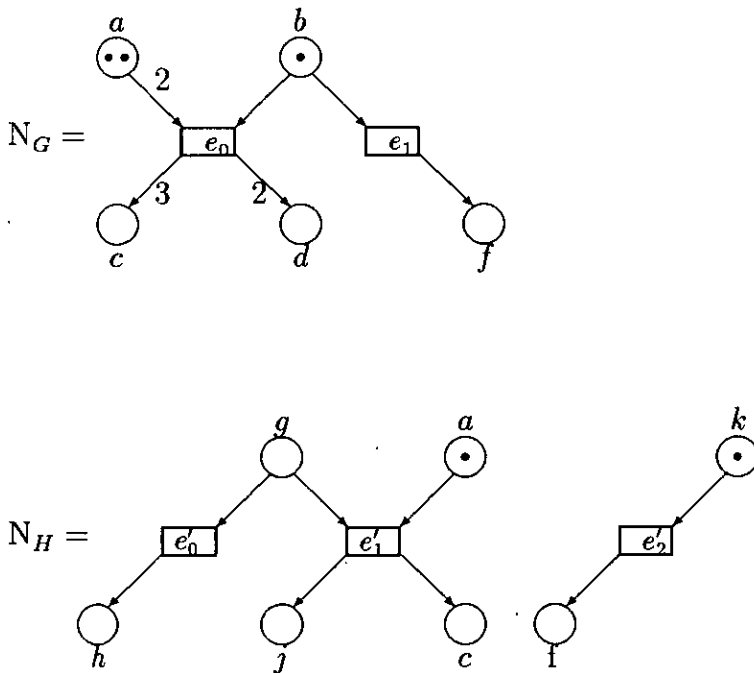
$$N_F \otimes N_G = N_{F \otimes G}.$$

Proof: Immediate from the definition. □

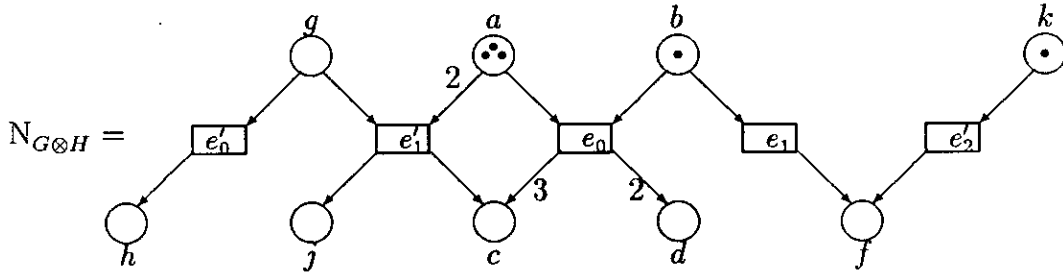
Thus $N_G \otimes N_H$ is the net formed by identifying those conditions and events which are common to the two nets N_F and N_G . This may introduce conflicts not present in either N_G or N_F .

Thus $N_{F \otimes G}$ is the quotient of two nets as defined in [Win87].

Example 9.3.3 If N_G and N_H are as shown below,



then the net $N_{G \otimes H}$ is as follows:



In the case where both the event sets and the condition sets of the two nets N_G and N_H are disjoint, the tensor product of N_G and N_H is their disjoint union.

Lemma 9.3.4 Let G and H be formulae in \mathcal{F} . If $N_G \supseteq N_H$ then $N_{F \otimes G} \supseteq N_{F \otimes H}$ for all Petri nets N_F .

Proof: By definition,

$$N_{F \otimes G} = \langle \mathcal{E}_F \cup \mathcal{E}_G, \mathcal{B}_F \cup \mathcal{B}_G, pre_F \cup pre_G, post_F \cup post_G, M_F + M_G \rangle.$$

Also, since $N_G \supseteq N_H$, we have

- $\mathcal{E}_G \supseteq \mathcal{E}_H$,
- $\mathcal{B}_G = \mathcal{B}_H \setminus \{b \in \mathcal{B}_H \mid b \notin_m M_H \ \& \ \exists e \in (\mathcal{E}_G \setminus \mathcal{E}_H). (b \in_m pre(e) + post(e))\}$, and
- M_G can evolve in N_G to M_H .

$$\text{Now } \mathcal{E}_{F \otimes G} = \mathcal{E}_F \cup \mathcal{E}_G \supseteq \mathcal{E}_F \cup \mathcal{E}_H = \mathcal{E}_{F \otimes H}.$$

Also,

$$\begin{aligned} \mathcal{B}_{F \otimes H} &= \mathcal{B}_F \cup \mathcal{B}_H \\ &= \mathcal{B}_F \cup \mathcal{B}_G \setminus \{b \in \mathcal{B}_G \mid b \notin_m M_{F \otimes H} \ \& \ \exists e \in \mathcal{E}_H. (b \in_m pre(e) + post(e))\} \\ &= \mathcal{B}_{F \otimes G} \setminus \{b \in \mathcal{B}_G \mid b \notin_m M_{F \otimes H} \ \& \ \exists e \in \mathcal{E}_H. (b \in_m pre(e) + post(e))\} \\ &= \mathcal{B}_{F \otimes G} \setminus \{b \in \mathcal{B}_G \mid b \notin_m M_{F \otimes H} \ \& \ \exists e \in \mathcal{E}_{F \otimes H}. (b \in_m pre(e) + post(e))\} \\ &= \mathcal{B}_{F \otimes G} \setminus \{b \in \mathcal{B}_{F \otimes G} \mid b \notin_m M_{F \otimes H} \ \& \ \exists e \in \mathcal{E}_{F \otimes H}. (b \in_m pre(e) + post(e))\}. \end{aligned}$$

Finally, since M_G evolves in N_G to M_H , $M_F + M_G$ evolves in $N_{G \otimes H}$ to $M_F + M_H$. □

Corollary 9.3.5 If $N_G \sqsupseteq N_H$ and $N_A \sqsupseteq N_B$ then $N_{G \otimes A} \sqsupseteq N_{H \otimes B}$.

9.3.2 The Action of \wedge on Nets

As usual, we understand $A \wedge B$ to indicate the determined choice of one of A and B . There is the same difficulty in a graphical Petri net representation of such a formula as Girard found in developing a system of proof nets for the additive connectives (see [Gir86]). We must consider $A \wedge B$ to be some entity (analogous to a proof box) which can be either A or B as required, but not both.

The concept that the determined choice between two elements of **MPetri** be a net is a simple extension of the usual definition of a Petri net.

Definition 9.3.6 $N_0 \wedge N_1$ is the determined choice between N_0 and N_1 .

The concept of determined choice is fundamental to linear logic, and its interpretation is as basic as the interpretation of intuitionistic conjunction by the “and” of natural language.

Notation 9.3.7 We write \mathcal{N} for the set defined inductively by

$$\mathcal{N} ::= P \mid N_0 \wedge N_1,$$

where $P \in \mathbf{MPetri}$.

We extend the definition of \sqsupseteq to \mathcal{N} as follows:

Definition 9.3.8

- (i) For $i = 0, 1$, $N_0 \wedge N_1 \sqsupseteq N_i$, and
- (ii) If $N \sqsupseteq N_i$ for $i = 0, 1$ then $N \sqsupseteq N_0 \wedge N_1$.

Lemma 9.3.9 If N_0 and N_1 are nets, then

$$F_{N_0 \wedge N_1} = F_{N_0} \wedge F_{N_1}.$$

Proof: Immediate from the definition of $N_0 \wedge N_1$.

□

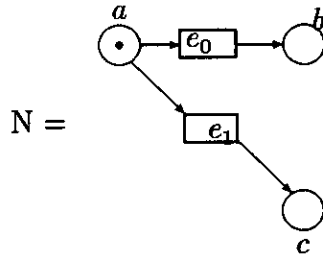
To see how the connective \wedge is used in describing nets, consider the case where a net N can evolve to two different nets, N_0 and N_1 . Then we can meaningfully state that N can evolve to the extended net $N_0 \wedge N_1$. In this way we can code up in our description of the evolution of a net as little or as much as we wish of the history of the choices to be made by the net in reaching a given state of evolution. For instance, in the above example, if we say that N evolves to N_0 , we are assuming that the choice of N_0 has been made and N has therefore lost the ability to evolve to N_1 . If we say that N has evolved to $N_0 \wedge N_1$, then we are not explicitly stating which choices have been made in the evolution of N , and have the option of making the choice at some later stage in our description of the development of the net.

9.3.3 The Issue of Choice

The usual interpretation of the additive linear connectives is as different aspects of choice. Thus $A \wedge B$ represents the determined choice between two resources A and B , since we must choose which of A and B we use when $A \wedge B$ is the premise of a sequent, only one of the possibilities corresponding to a proof. Similarly, $A \oplus B$ represents an undetermined choice between A and B , in that, when used as the premise of a sequent, both possibilities correspond to a proof. The terms “determined” and “undetermined” are here preferred to “deterministic” and “non-deterministic”, as these adjectives have defined meanings in the theory of concurrency. Their careless use can cause confusion. In [MOM89], the suggestion is made that whenever in the context of a net the marking $M_0 \wedge M_1$ can be proved, then the environment (or some idealised observer) can make an “external choice” and require that the net reach a particular preferred state. Usually, in the study of Petri nets, we understand that no observer can influence the course of events, and a conflict is resolved by the net itself. If this resolution of conflict is to constitute external choice, we need a conceptual and definitional extension of net theory.

Example 9.3.10

Consider the following example:



In the notation of [MOM89], we have $N \models A \vdash B \wedge C$: that is, given the causal dependencies of N , the marking A can evolve to either marking B or C . There is no possibility of external choice here, however: no observer outside the net can demand that it evolve to B or to C . In fact, the net makes an *internal* choice, evolving to either B or C .

Apart from this obvious objection to the use of the term “external choice” in this context, it is certainly meaningless to discuss what is external or internal without a paradigm for the internal (which I suggest should be the behaviour of the net itself) and the external (for which I suggest the environment or an observer). Marti-Oliet and Meseguer do not offer such a paradigm, nor any way of distinguishing external from internal data. We now suggest a means of achieving this in our model. Further details are required to establish a full theory: we merely outline a possible approach.

Definition 9.3.11 *The internal data of a net N are its events and initial marking.*

Definition 9.3.12 *The external data of a net N are properties of its environment.*

The definition of external data presupposes a concept of environment that goes beyond the usual one of marking.

Definition 9.3.13 *An internal choice between events of a net is a conflict whose resolution is independent of the net’s environment.*

An external choice between events is a conflict whose resolution is influenced by the net's environment.

Remark 9.3.14 *An external choice can only occur in a non-empty environment.*

We have expressed the internal data of a net as a canonical formula. We can express the external data as admissible axioms of a linear theory. We can then prove in the theory properties of the net's behaviour in a given external environment. Our purpose is to use sequents as specifications of a net's behaviour. Thus if we require a net N to be able to reach a marking B from a marking A , we require that the sequent $F_N, A \vdash B$ be derivable in \mathcal{L}_0 . If we can derive $F_N, A \vdash B$, then we say that N satisfies the specification $F_N, A \vdash B$. If we can derive $F_N, A \vdash B$ by use of an additional axiom, for example $A \vdash B$, then we say that N satisfies the specification $F_N, A \vdash B$ in the presence of external data to the effect that the environment of N can use A to produce B .

Consider the net N of Example 9.3.10. In an empty environment it can make an internal choice between the events e_0 and e_1 . Consider an environment in which the marking c is impossible. We can express the fact that c is impossible by the axiom $\vdash C^\perp$. The external data force the net to make the choice of e_0 rather than e_1 , and this is an external choice.

It is precisely because we have expressed nets as formulae that this distinction between the internal and the external is so easy to express. It is not possible to make the distinction if we represent the nets themselves as theories.

Our example relates closely to issues of choice and hiding in CCS. In [Mil89], Milner considers the processes:

$$a.E + b.F \quad \text{and} \quad \tau.a.E + \tau.b.F .$$

The first represents external choice between the summands, and the second internal choice. This is made clear if we restrict each of the processes on b . The first, an external choice, cannot deadlock, as the first summand must be chosen. The second will deadlock if the second τ action is chosen internally. This notion

of internal and external choice corresponds to our example. In an empty environment, the net N chooses internally between e_0 and e_1 . In the environment where we restrict on C , the net is forced externally to choose e_0 .

As a second example, consider the following sequents derivable in \mathcal{L}_0 :

$$(A \oplus B), !(A \multimap C), !(B \multimap C) \vdash C$$

$$(A \wedge B), !(A \multimap C) \vdash C$$

$$(A \wedge B), !(B \multimap C) \vdash C$$

$$(A \oplus B), !(A \multimap C) \not\vdash C$$

These derivations indicate that, in the empty environment, a marking of the determined choice between A and B suffices to produce C if we have only one event available. The undetermined choice between A and B does not suffice to produce C if we are restricted to one event. If the environment is such that D produces an undetermined choice between A and B , which we express by the axiom $D \vdash A \wedge B$, then we can derive the sequent:

$$D, !(A \multimap C) \vdash C.$$

If, however, the external data are such that our theory has the added axiom $D \vdash A \oplus B$, and none other, then we *cannot* derive the above sequent.

The axiom $D \vdash A \oplus B$ represents an environment in which an external choice has been made. We suggest that this is an appropriate use for the operator \oplus . Marti-Oliet and Meseguer illustrate the use of \oplus using the sequent $N \models A \vdash B \oplus \1000 . Their suggested interpretation of this sequent is that in the net N , the token A is sufficient to produce the token B , or a thousand dollars. This specification would be met by a process which always chose the first summand. While an accurate description of the meaning of \oplus , this example is not a convincing case for the application of \oplus in specifications. Nor is the process satisfying the specification “non-deterministic” fairly. Again, their intended interpretation of the term “internal” is not stated.

The specification $N \models A \vdash B \oplus \1000 indicates our reasons for omitting the connective \oplus from the conclusions of a sequent. We now consider the rôle of \oplus in the premises of a sequent. If the external data consists of the axiom $X \oplus Y \vdash C$, then in order to derive a sequent of the form

$$F_N, X \oplus Y \vdash Z,$$

we must derive *both* $F_N, X \vdash Z$ and $F_N, Y \vdash Z$. Thus our specification represents two sub-specifications: as a connective among the premises of a sequent, \oplus is a useful abbreviation. Further, if we can derive both $F_{N'} \vdash X$ and $F_{N'} \vdash Y$, we can derive $F_{N'} \vdash C$, thus eliminating the connective \oplus .

The connective \oplus might also occur on both sides of a sequent, thus:

$$F_N, X \oplus Y \vdash F_{N'}, X \oplus Y.$$

In this case, the premises are under-specified, in that we do not know which of X and Y will be used. The conclusion is also under-specified, and the choice remains to be resolved by a more detailed specification, or by an elimination.

We have shown that \oplus is applicable to specifications using external data. However, its use in describing internal data is primarily as an abbreviation. Since we are concerned in the rest of Part IV with internal data only, we shall not give an interpretation of \oplus as a combinator of Petri nets.

9.4 Nets and Equivalent formulae

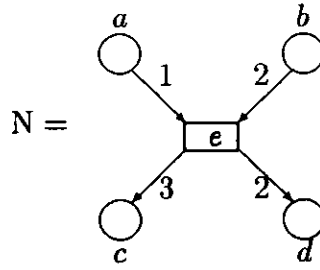
Our intention is that S -equivalent formulae represent the same net. Corollary 8.4.12 shows that

$$\text{if } F =_S G \text{ and } F \vdash_{\mathcal{L}_1} F' \text{ then } G \vdash_{\mathcal{L}_1} F'.$$

Thus S -equivalent formulae convey the same information about the transformation of resources. We now give an example of a nets and several equivalent formulae representing it.

Consider the following example:

Example 9.4.1



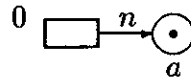
The canonical formula representing N is $F_N = !(A \otimes 2B \multimap 3C \otimes 2D)$. However, each of the formulae

- $!(A \otimes 2B \multimap 3C \otimes 2D) \otimes !(A \otimes 2B \multimap 3C \otimes 2D)$,
- $!(A \otimes 2B \multimap 3C \otimes 2D) \otimes ((A \otimes 2B \multimap 3C \otimes 2D) \wedge I)$ and
- $((A \otimes 2B \multimap 3C \otimes 2D) \wedge I) \otimes !(A \otimes 2B \multimap 3C \otimes 2D)$

conveys the same information as F_N about the transformation of resources, and so may be considered to represent the net N . Notice that all three non-canonical formulae are S -equivalent to F_N .

9.4.1 The Interpretation of $!(A)$

Example 9.4.2 An important example of S -equivalent formulae which represent the same net arises when we consider the following net N :



The canonical formula representing the net N is $form(N) = !(nA)$. A different formula representing the same transformation of resources is $!(I \multimap nA)$.

Again, we have $!(I \multimap nA) \longrightarrow_S !(nA)$.

A net of this kind might represent, for example, a stream of integers which can supply as many integers as a process requires, or a resource allocator which can make the resources it controls available an arbitrary number of times.

Lemma 9.4.3 If $N_{!(F)} \supseteq N_G$ then $N_{!(F)} \supseteq N_{!(G)}$.

Proof:

If N_G is unmarked then $!(G) = G$ and the result follows immediately.

If N_G has non-empty marking M then $N_{!(G)}$ has an event $e \notin \mathcal{E}_G$ such that $pre(e) = \emptyset$ and $post(e) = M$. Further, the net $N_{!(F)}$ can evolve to M from the empty marking. Putting $\mathcal{E}_G^* = \{e\}$ we have

$$\mathcal{E}_{!(G)} \subseteq \mathcal{E}_G^* \cup \mathcal{E}_{!(F)}, \quad \text{and}$$

$$\mathcal{B}_{!(G)} = \mathcal{B}_G,$$

while both $N_{!(G)}$ and $N_{!(F)}$ are unmarked.

By definition, we have $N_{!(F)} \supseteq N_{!(G)}$. □

Remark 9.4.4 Consider a canonical formula F . Then

$$F = M \otimes \bigotimes_{j \in J} !(M_j \multimap M'_j) \otimes \bigotimes_{k \in K} !(M_k).$$

Since $!$ is idempotent, we have

$$!(F) =_S !(M) \otimes \bigotimes_{j \in J} !(M_j \multimap M'_j) \otimes \bigotimes_{k \in K} !(M_k),$$

and so $!(F)$ is also canonical.

Thus the nets representing canonical formulae F and $!F$ have identical transitions, while all places of $N_{!F}$ are considered to have a potentially inexhaustible supply of tokens. We can regard this as indicating that the behaviour of $N_{!F}$ should be the limiting behaviour of N_F with respect to the piling up of resources. Notice that the accumulation of resources can only generate tokens in a certain proportion. For example, if $M(a) = 1$ and $M(b) = 3$, then every generation of a token at a must be accompanied by the generation of 3 tokens at b .

9.5 A Semantics for Canonical Formulae

The function *form* gives rise to a semantics for canonical formulae which expresses each formula as an isomorphism class of nets. Thus we define a function $\llbracket - \rrbracket$ from **Can** to **MPetri** as follows:

- $\llbracket A \rrbracket = \langle \phi, \{a\}, \phi, \phi, \emptyset \rangle$, for A a linear atom,
- $\llbracket I \rrbracket = \langle \phi, \phi, \phi, \phi, \emptyset \rangle$,
- $\llbracket M \rrbracket = \langle \phi, \text{At}(M_0), \phi, \phi, M_0 \rangle$, for M a tensor product of linear atoms,
- $\llbracket !(M) \rrbracket = \langle \{e\}, \text{At}(M), \text{pre}, \text{post}, \emptyset \rangle$, where the pre- and post-condition relations are given by $\text{pre}(e) = \emptyset$ and $\text{post}(e) = M$,
- $\llbracket !(M_0 \multimap M_1) \rrbracket = \langle \{e\}, \text{At}(M_0 \multimap M_1), \text{pre}, \text{post}, \emptyset \rangle$ where the pre- and post-condition relations are given by $\text{pre}(e) = M_0$ and $\text{post}(e) = M_1$.
- $\llbracket C_0 \otimes C_1 \rrbracket = \llbracket C_0 \rrbracket \otimes \llbracket C_1 \rrbracket$, where C_0, C_1 and $C_0 \otimes C_1$ are all canonical.

We can easily extend this semantics to the larger class of formulae in \mathcal{F} which have canonical S -normal forms, thus:

$$\text{if } F =_S F_N \text{ and } F_N \in \mathbf{Can}, \text{ then } \llbracket F \rrbracket = \llbracket F_N \rrbracket = N.$$

Notation 9.5.1 *If F is a formula with canonical S -normal form F_N , then we shall write N_F to stand for $\llbracket F \rrbracket$, and call N the net represented by F .*

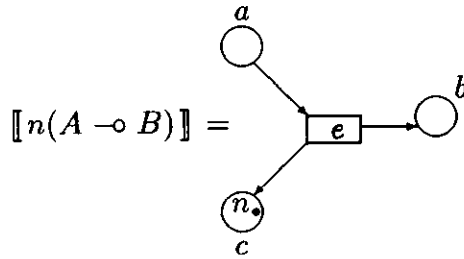
Corollary 8.4.32 ensures that $\llbracket - \rrbracket$ remains well-defined with this extension. This, however, does not greatly increase our understanding of the use of linear logic as a language for specifying Petri nets. In Section 9.3.3 we mentioned reasons for omitting an interpretation of the connective \oplus . An interpretation of the formulae of \mathcal{F} would therefore achieve our objectives of a fuller understanding of linear logic formulae as Petri nets. However, there are difficulties in giving a semantics to all of \mathcal{F} , and we here merely comment on the semantics of formulae which do not have a canonical normal form.

9.5.1 The Semantics of Linear Implication

Our major problem lies in giving a semantics to linear implication between Petri nets. In the restricted case where the implication is between markings, a possible semantics for $M_0 \multimap M_1$ is the following:

$$\llbracket n(M_0 \multimap M_1) \rrbracket = \langle \{e\}, \{c\} \cup \text{At}(M_0 \multimap M_1), \text{pre}, \text{post}, nc \rangle,$$

where C is an atom disjoint from $\text{At}(M_0 \multimap M_1)$ and the pre- and post-condition relations are given by $\text{pre}(e) = c + M_0$ and $\text{post}(e) = M_1$. This would give, for example:



Such an interpretation accords with the intuitive meaning of $n(M_0 \multimap M_1)$. However, the net $\langle \{e\}, \{c\} \cup \text{At}(M_0 \multimap M_1), \text{pre}, \text{post}, nc \rangle$ is the semantics of the canonical formula $nC \otimes !(C \otimes M_0 \multimap M_1)$. We could add a clause to the definition of S making these two formulae S -equivalent. Notice, however, that the two formulae are not interderivable, although we can prove that

$$nC \otimes !(C \otimes M_0 \multimap M_1) \vdash n(M_0 \multimap M_1).$$

Notice also that the formula $!(M_0 \multimap M_1) \otimes n(M_0 \multimap M_1)$ has normal form $!(M_0 \multimap M_1)$. Thus we have two different nets interpreting equivalent formulae. We can overcome this by defining an equivalence on nets under which $\llbracket !(M_0 \multimap M_1) \otimes n(M_0 \multimap M_1) \rrbracket$ and $\llbracket !(M_0 \multimap M_1) \rrbracket$ are equivalent. Because we ensured that S -equivalent formulae convey the same information about the transformation of resources, such an equivalence on nets is a natural one.

Thus we would need to extend our reduction system and introduce an equivalence on nets in order to give a suitable interpretation to the formula $n(M_0 \multimap M_1)$. Although rather technical, this appears to be a promising approach. In order to

interpret formulae of the form $!(M_0 \multimap M'_0) \multimap !(M_1 \multimap M'_1)$ as a net, we would have to extend our concept of net considerably, since the natural interpretation is that of an event whose pre-conditions are a net and whose post-conditions are a net. This extension appears to be straightforward.

9.5.2 The Semantics of $F \wedge G$

In order to interpret the operator \wedge on formulae in addition to the operators \otimes , \multimap and $!$, we use the powerset $\mathcal{P}(\mathbf{MPetri})$ rather than \mathbf{MPetri} itself.

Definition 9.5.2 *The set \mathbf{Can}_\wedge is the set of equivalence classes under permutation of the formulae C defined inductively by*

$$C ::= \mathbf{Can} \mid C_0 \wedge C_1.$$

We can modify our semantics so that a set of Petri nets corresponds to each formula of \mathbf{Can}_\wedge , by giving a function $\llbracket - \rrbracket'$ from \mathbf{Can}_\wedge to $\mathcal{P}(\mathbf{MPetri})$.

Lemma 9.5.3 The assignment $\llbracket - \rrbracket'$ given as follows:

$$\llbracket - \rrbracket': F \mapsto \begin{cases} \{\llbracket F \rrbracket\} & \text{if } F \in \mathbf{Can} \\ \llbracket F_0 \rrbracket \cup \llbracket F_1 \rrbracket & \text{if } F = F_0 \wedge F_1, \end{cases}$$

defines a function from \mathbf{Can}_\wedge to \mathcal{N} .

Proof: Evident. □

Remark 9.5.4 *This approach, which uses sets to give a semantics to a choice of behaviours, seems related to Plotkin's powerdomain construction [Plo76], and further consideration should be given in further work to this aspect of our semantics.*

Sections 9.5.1 and 9.5.2 suggest that, with considerably more work, we could develop a sound semantics for the formulae of \mathcal{F} as (sets of) Petri nets. As we have seen, it is relatively straightforward to extend the interpretation of the tensor fragment of linear logic, and interpret also the operators \wedge and $!$ as net constructors. The interpretation of linear implication remains unresolved.

9.6 Reachable markings as Provable Formulae

The partial function $\llbracket - \rrbracket$ from \mathcal{F} to \mathbf{MPetri} gives a semantics for certain formulae of \mathcal{F} in terms of Petri nets. We now show that this semantics is complete with respect to the fragment \mathcal{L}_0 of linear logic, and also that, where defined, it is sound. Thus we show that:

if $N \sqsupseteq N'$ then $F_N \vdash_{\mathcal{L}_0} F_{N'}$, (*completeness*) and

if $F \vdash F'$ and $\llbracket - \rrbracket$ is defined on both F and F' ,

then $\llbracket F \rrbracket \sqsupseteq \llbracket F' \rrbracket$. (*partial soundness*)

Remark 9.6.1 *When proving completeness we use only rules for which our semantics is sound, and hence the completeness is non-trivial.*

In Section 9.6.4, we show further that $\llbracket - \rrbracket'$ gives a complete and partially sound semantics for the larger fragment of linear logic \mathcal{L}_1 .

9.6.1 Soundness and Completeness for \mathcal{L}_0

Theorem 9.6.2

(1) If N and N' are both Petri nets in \mathbf{MPetri} and $N \sqsupseteq N'$ then $F_N \vdash F_{N'}$ is provable in \mathcal{L}_0 .

(2) If N is a Petri net, $F_N \vdash G$ is provable in \mathcal{L}_0 , and $G =_S F_{N'}$, then $N \sqsupseteq N'$.

Proof: Proof of (1) is straightforward.

If $N \sqsupseteq N'$ then by considering the definition of \sqsupseteq we see that there is a chain of nets $N = N_0 \sqsupseteq N_1 \sqsupseteq \cdots \sqsupseteq N_n = N'$ such that each net is related to the net above it in the chain in one of the following ways:

This is a simple consequence of the rule (Derel).

We now have that, for each i , $F_{N_i} \vdash F_{N_{i+1}}$

Applying the (Cut) rule $(n - 1)$ times, we deduce that $F_{N_0} \vdash F_{N_n}$, and hence that $F_N \vdash F_{N'}$.

Proof of (2) is by induction on the vertical depth of the derivation of $F \vdash G$.

Base Case. There are two possibilities, corresponding to the two axioms.

If this is an instance of the Identity axiom $F \vdash F$ there is nothing to prove.

If it is an instance of (IL) or (IR), then since we interpret I by the empty net $\langle \phi, \phi, \phi, \phi \rangle$, the result is immediate.

Inductive Step.

- Suppose the last rule used in the derivation was (Cut). The derivation ends:

$$\frac{F_1 \vdash G \quad F_2, G \vdash B}{F_1, F_2 \vdash B} (\text{Cut})$$

In this case $F = F_1 \otimes F_2$. By the inductive hypothesis,

$$N_{F_1} \sqsupseteq N_G \text{ and } N_{F_2 \otimes G} \sqsupseteq N_B.$$

By Lemma 9.3.4,

$$N_{F_1 \otimes F_2} \sqsupseteq N_G \otimes N_{F_2} = N_{F_2 \otimes G} \sqsupseteq N_B.$$

Hence $N_{F_1 \otimes F_2} \sqsupseteq N_B$, as required.

- Suppose that last rule used was (Exch).

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} (\text{Exch})$$

Since $\Gamma \otimes B \otimes A \otimes \Delta$ represents a net, $\Gamma \otimes A \otimes B \otimes \Delta$ represents the *same* net, that is,

$$N_{\Gamma \otimes B \otimes A \otimes \Delta} = N_{\Gamma \otimes A \otimes B \otimes \Delta}.$$

By the inductive hypothesis, $N_{\Gamma \otimes A \otimes B \otimes \Delta} \supseteq N_C$, and so $N_{\Gamma \otimes B \otimes A \otimes \Delta} \supseteq N_C$, as required.

- Suppose that last rule used was (\otimes R).

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} (\otimes R)$$

By the inductive hypothesis, $N_\Gamma \supseteq N_A$ and $N_\Delta \supseteq N_B$.

By Corollary 9.3.5,

$$N_{\Gamma \otimes \Delta} \supseteq N_A \otimes N_B = N_{A \otimes B}.$$

- If the last rule used was (\otimes L), then there is nothing to prove.
- The last rule used cannot have been (\neg L):

$$\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Delta, \Gamma, A \neg\circ B \vdash C} (\neg\circ L),$$

since the formula $\Delta \otimes \Gamma \otimes A \neg\circ B$ is not in canonical form.

- Suppose the last rule used was (Weak):

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B} (\text{Weak}).$$

By the inductive hypothesis, $N_\Gamma \supseteq N_B$, and by Lemma 9.3.4,

$N_{\Gamma \otimes !A} \supseteq N_\Gamma$, whence the required result.

- Suppose the last rule used was (Cont).

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} (\text{Cont})$$

By definition, $\Gamma \otimes !A \otimes !A$ represents the same net as $\Gamma \otimes !A$.

Thus we have $N_{\Gamma \otimes !A} = N_{\Gamma \otimes !A \otimes !A} \supseteq N_B$.

- Suppose the last rule used was (Der).

$$\frac{\Gamma, (A \wedge I), !A \vdash B}{\Gamma, !A \vdash B} (\text{Der})$$

As in the case of (Cont), this case follows immediately from the definition of equivalent formulae.

- Suppose the last rule used was (!R).

$$\frac{!(F) \vdash G}{!(F) \vdash !(G)} (!R)$$

The result follows immediately from Lemma 9.4.3.

This completes our proof of Theorem 9.6.2. □

9.6.2 Soundness and Completeness for \mathcal{L}_1

We now show that the partial function $\llbracket - \rrbracket'$ from \mathcal{F} to $\mathcal{P}(\mathbf{MPetri})$ gives rise to a semantics for \mathcal{L}_1 which is complete and partially sound. Since $\llbracket - \rrbracket'$ is defined on \mathbf{Can}_\wedge and $\mathbf{Can} \subset \mathbf{Can}_\wedge$, this is a significant extension of the result of Theorem 9.6.2.

We extend $!$, \otimes and \wedge to sets of nets pointwise. Thus if \mathcal{N}_0 and \mathcal{N}_1 are sets of nets, we have

$$\mathcal{N}_0 \otimes \mathcal{N}_1 = \{N_0 \otimes N_1 \mid N_i \in \mathcal{N}_i \text{ for } i = 0, 1\},$$

$$\mathcal{N}_0 \wedge \mathcal{N}_1 = \{N_0 \wedge N_1 \mid N_i \in \mathcal{N}_i \text{ for } i = 0, 1\}, \text{ and}$$

$$!(\mathcal{N}_0) = \{N_0 \mid N_0 \in \mathcal{N}_0\}.$$

Notation 9.6.3 Let \mathcal{N} be a finite element of $\mathcal{P}(\mathbf{MPetri})$. We write $F_{\mathcal{N}}$ for the formula $\wedge\{N \mid N \in \mathcal{N}\}$, and call $F_{\mathcal{N}}$ the formula representing the set of nets \mathcal{N} .

Theorem 9.6.4

(1) If \mathcal{N} and \mathcal{N}' are both elements of $\mathcal{P}(\mathbf{MPetri})$ and $\mathcal{N} \sqsupseteq \mathcal{N}'$ then for each $N' \in \mathcal{N}'$ there exists $N \in \mathcal{N}$ such that $F_N \vdash_{\mathcal{L}_1} F_{N'}$ is provable in \mathcal{L}_1 .

(2) If $\mathcal{N} \in \mathcal{P}(\mathbf{MPetri})$, $F_{\mathcal{N}} \vdash G$ is provable in \mathcal{L}_1 , and $G =_S F_{\mathcal{N}'}$, then $\mathcal{N} \sqsupseteq \mathcal{N}'$.

Proof: (1) is an immediate corollary of Theorem 9.6.2.

The proof of (2) is a straightforward extension of the proof of Theorem 9.6.2(2), for each rule of \mathcal{L}_0 .

We now show soundness of the semantics $\llbracket - \rrbracket'$ with respect to the rules for \wedge .

- Suppose the last rule used was ($\wedge R$):

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

By the inductive hypothesis, $\llbracket \Gamma \rrbracket' \sqsupseteq \llbracket A \rrbracket'$ and $\llbracket \Gamma \rrbracket' \sqsupseteq \llbracket B \rrbracket'$.

By Definition 9.3.8, $\llbracket \Gamma \rrbracket' \sqsupseteq \llbracket A \wedge B \rrbracket'$.

- Suppose the last rule used was ($\wedge L1$):

$$\frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C}.$$

By the inductive hypothesis, $\llbracket \Gamma \otimes A \rrbracket' \sqsupseteq \llbracket C \rrbracket'$.

By Lemma 9.3.4 and Definition 9.3.8, $\llbracket \Gamma \otimes (A \wedge B) \rrbracket' \sqsupseteq \llbracket \Gamma \otimes A \rrbracket' \sqsupseteq \llbracket C \rrbracket'$, as required.

In the case where the last rule used was ($\wedge L2$), the argument is similar.

□

9.7 Canonical Derivations

In this section, we consider only single nets, rather than sets of nets, and work with the fragment \mathcal{L}_0 rather than \mathcal{L}_1 . There may be many different ways in which a sequent $F \vdash G$ can be proved. If F_N and $F_{N'}$ are the canonical formulae representing the nets N and N' , and $N \supseteq N'$, then we can distinguish a certain class of derivations of $F_N \vdash F_{N'}$, which reflect exactly the nature of the containment $N \supseteq N'$.

From the definition of \supseteq , we know that there is a sequence of operations by which N can be transformed into N' , each operation being one of the following:

- (i) the occurrence of a firing of N
- (ii) the removal of an event of N .

It is clear that we can order these operations so that all those of the second type follow all those of the first. We then obtain a sequence of nets, as in the proof of Theorem 9.6.2(1),

$$N = N_0 \supseteq N_1 \cdots \supseteq N_n = N',$$

where there exists a k between 0 and n such that for $0 \leq i < k$, N_{i+1} is obtained from N_i by doing a firing of N_i , and for $k \leq i < n$, N_{i+1} is obtained from N_i by removing one of the events of N_i .

We build a derivation of $F_N \vdash F_{N'}$ from the bottom up in the following way¹:

- (1) For each of the k firings used in the evolution of the chain of nets, we apply the appropriate instance of the rule (Der). For convenience, we subscript the labels of rules to indicate which application of the rule is intended. Observe that in the case where an event has empty pre-conditions, we merely make a copy of a multiset.

¹Compare the normal form for natural deduction proofs

$$\frac{\Gamma \otimes !(M_k \multimap M'_k) \otimes \cdots \otimes !(M_1 \multimap M'_1) \otimes ((M_1 \multimap M'_1) \wedge I) \vdash F_{N_n}}{\text{(Der)}_1}$$

$$\vdots$$

$$\frac{\Gamma \otimes !(M_k \multimap M'_k) \otimes ((M_k \multimap M'_k) \wedge I) \otimes \cdots \otimes !(M_1 \multimap M'_1) \vdash F_{N_n}}{\text{(Der)}_k}$$

$$\Gamma \otimes !(M_k \multimap M'_k) \otimes \cdots \otimes !(M_1 \multimap M'_1) \vdash F_{N_n}$$

Here we have $F_{N_0} = \Gamma \otimes !(M_1 \multimap M'_1) \otimes \cdots \otimes !(M_k \multimap M'_k)$.

- (2) For each of the k firings used in the evolution of the chain of nets, we apply the appropriate instances of the (Imp) and (Id) rules.

$$\frac{\text{----- (Id)}_k}{M_k \vdash M_k} \quad \frac{\Gamma_1 \vdash F_{N_n}}{\text{(Imp)}_k}$$

$$\vdots$$

$$\frac{\text{----- (Id)}_1}{M_1 \vdash M_1} \quad \frac{\Gamma' \otimes ((M_k \multimap M'_k) \wedge I) \otimes \cdots \otimes M'_1 \vdash F_{N_n}}{\text{(Imp)}_2}$$

$$\frac{\Gamma' \otimes ((M_k \multimap M'_k) \wedge I) \otimes \cdots \otimes ((M_1 \multimap M'_1) \wedge I) \otimes M_1 \vdash F_{N_n}}{\text{(Imp)}_1}$$

Here, the formula

$\Gamma' \otimes ((M_k \multimap M'_k) \wedge I) \otimes \cdots \otimes ((M_1 \multimap M'_1) \wedge I) \otimes M_1$ stands for

$\Gamma \otimes !(M_k \multimap M'_k) \otimes ((M_k \multimap M'_k) \wedge I) \otimes \cdots \otimes !(M_1 \multimap M'_1) \otimes ((M_1 \multimap M'_1) \wedge I)$.

Now, since N_{i+1} is obtained from N_i simply by allowing a firing to take place, we know that, for $i = 1$ to k ,

$$F_{N_i} \otimes M_i = F_{N_{i-1}} \otimes M'_i.$$

It follows that

$$F_{N_k} \otimes M_k \cdots \otimes M_1 = F_{N_0} \otimes M'_k \cdots \otimes M'_1.$$

From the proof in (2) we see that

$$\Gamma_1 \otimes M_k \cdots \otimes M_1 = \Gamma' \otimes M_1 \otimes M'_k \cdots \otimes M'_1,$$

and since by construction of Γ' we have $F_{N_0} = \Gamma' \otimes M_1$, we see that $\Gamma_1 = F_{N_k}$.

(3) Next, for each of the $(n - k)$ events removed in the evolution of the nets, we apply the appropriate instance of the rule (Weak):

$$\frac{\Gamma'' \vdash F_{N'}}{\text{(Weak)}_{n-k}} \quad \frac{\Gamma'' \otimes !(W_{n-k} \multimap W'_{n-k}) \vdash F_{N'}}{\text{(Weak)}_{n-k-1}} \quad \vdots \quad \frac{\Gamma'' \otimes !(W_1 \multimap W'_1) \otimes \cdots \otimes !(W_{n-k} \multimap W'_{n-k}) \vdash F_{N'}}{\text{(Weak)}_1}$$

Here we have $F_{N_k} = \Gamma'' \otimes !(W_1 \multimap W'_1) \otimes \cdots \otimes !(W_{n-k} \multimap W'_{n-k})$.

Also, by construction, for $i = 1$ to $n - k$,

$$F_{N_{k+i}} \otimes !(W_i \multimap W'_i) = F_{N_{k+i-1}}.$$

Hence $\Gamma'' = F_{N_n}$, and the canonical derivation of $F_{N_0} \vdash F_{N_n}$ can be completed with an instance of the (Identity) axiom.

Notation 9.7.1 We call a proof in the form described above a canonical derivation of $F_N \vdash F_{N'}$.

Remark 9.7.2 For any two canonical formulae F_N and $F_{N'}$ where $N \supseteq N'$, such a derivation is unique up to the the number of repetitions of cycles of markings in the evolution of the net N , and up to the order in which firings are done or events removed (although of course in some cases there will be some firings which can only occur after a number of other firings has taken place).

Schematically, a canonical derivation has the form shown below

(where $G = ((M_2 \multimap M'_2) \wedge I) \otimes \cdots \otimes ((M_k \multimap M'_k) \wedge I)$) :

$$\begin{array}{c}
 \frac{}{M_k \vdash M_k} \text{ (Id)} \quad \frac{}{F_{N_n} \vdash F_{N_n}} \text{ (Id)} \\
 \frac{}{\vdots} \text{ (Weak)} \\
 \frac{}{F_{N_{k+1}} \vdash F_{N_n}} \text{ (Weak)} \\
 \frac{}{F_{N_k} \vdash F_{N_n}} \text{ (Weak)} \\
 \frac{}{\vdots} \text{ (Imp) and (Id)} \\
 \frac{}{M_1 \vdash M_1} \text{ (Id)} \quad \frac{}{G \otimes F_{N_1} \vdash F_{N_n}} \text{ (Der)} \\
 \frac{}{\vdots} \text{ (Der)} \\
 \frac{}{((M_1 \multimap M'_1) \wedge I) \otimes F_{N_0} \vdash F_{N_n}} \text{ (Der)} \\
 \frac{}{F_{N_0} \vdash F_{N_n}}
 \end{array}$$

9.7.1 Example of a Canonical Derivation

$$\begin{array}{c}
 \frac{}{A \vdash A} \text{ (Id)} \quad \frac{}{\Gamma \otimes !(A \multimap B) \otimes B \otimes D \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Id)} \\
 \frac{}{C \vdash C} \text{ (Id)} \quad \frac{}{\Gamma \otimes !(A \multimap B) \otimes B \otimes D \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Weak)} \\
 \frac{}{\Gamma \otimes !(A \multimap B) \otimes !(C \multimap D) \otimes B \otimes D \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Imp)} \\
 \frac{}{A \vdash A} \text{ (Id)} \quad \frac{}{\Gamma \otimes !(A \multimap B) \otimes !(C \multimap D) \otimes ((C \multimap D) \wedge I) \otimes B \otimes C \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Imp)} \\
 \frac{}{\Gamma \otimes !(A \multimap B) \otimes ((A \multimap B) \wedge I) \otimes !(C \multimap D) \otimes ((C \multimap D) \wedge I) \otimes A \otimes C \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Der)} \\
 \frac{}{\Gamma \otimes !(A \multimap B) \otimes ((A \multimap B) \wedge I) \otimes !(C \multimap D) \otimes A \otimes C \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D} \text{ (Der)} \\
 \frac{}{\Gamma \otimes !(A \multimap B) \otimes !(C \multimap D) \otimes A \otimes C \vdash \Gamma \otimes !(A \multimap B) \otimes B \otimes D}
 \end{array}$$

There is a very important feature of such a canonical derivation. The only way in which $(\multimap L)$ is used in a canonical derivation (implicitly in the use of (Imp)) is to use up a copy of a firing which has been generated by an instance of the (Der) rule. Thus no false dependencies can be introduced into a formula in a canonical

proof: any subformula of the form

$$\left(\bigotimes_{j \in J} M_j \multimap \bigotimes_{k \in K} M_k \right)$$

must correspond to a event present in the net N with which we started, and so the (Imp) rule reflects a genuine causal dependency inherent in the net, rather than (as might be the case with a proof of some other form) a dependency introduced as a result of applications of the Cut rule. Compare the results of [GG89], where proofs can be limited to a certain form of (Cut) which reflects causal dependency in the nets being modelled.

9.7.2 Canonical derivations and Evolution

Theorem 9.7.3 If $F_N \vdash_{\mathcal{L}_1} G$ then there exists a net N' such that

- (i) $G =_S F_{N'}$, and
- (ii) $F_N \vdash F_{N'}$ and the derivation is canonical.

Proof: (i) follows from Theorem 9.6.2

For (ii), notice that since G has normal form $F_{N'}$ for some N' , and by Theorem 9.6.2, $N \sqsupseteq N'$, the discussion of Section 9.7 implies that there is a canonical derivation of $F_{N_1} \vdash F_{N_2}$ in \mathcal{L}_0 . □

9.7.3 A Remark about Categories

A canonical derivation between two formulae in **Can** is unique up to cycles of markings and the order of application within each of the three sets of rules corresponding to firings, to making copies of firings, and to removal of events. We shall consider canonical derivations to be equivalent if they differ only in the order of rule application and the presence of cycles. Up to this equivalence, there is at most one canonical derivation between any two canonical formulae. There is a canonical derivation of any canonical formula from itself by application of the rule

(Id). Theorem 9.7.3 suffices to show that whenever we have canonical derivations $A \vdash B$ and $B \vdash C$, there exists a canonical derivation $A \vdash C$. Thus we can regard \mathbf{Can} under equivalent canonical derivations as a preorder, and hence as a category. We shall denote this category \mathbf{Can} .

Furthermore, we can regard the preorder $(\mathbf{MPetri}, \sqsubseteq)$ as a category. We shall denote this category \mathbf{MPetri} . Theorem 9.7.3 shows that the categories \mathbf{Can} and \mathbf{MPetri} are isomorphic. This isomorphism expresses the fact that the category \mathbf{MPetri} is a sound and complete model of canonical linear logic formulae under equivalent canonical derivation. If we consider categories with object set those formulae which are S -equivalent to canonical formulae or to formulae of form $\bigwedge_{j \in J} C_j$ for a finite indexing set J and canonical formulae C_j , we can define categories of formulae representing which are equivalent to the category \mathbf{MPetri} . This allows us to consider larger classes of formulae and of derivations.

Part V

Conclusion and Further Work

Conclusion

This thesis explores three ways in which linear logic may be used to define a specification language for Petri nets, by giving precise correspondences, at different levels between linear logic and Petri nets .

In Part II, we define a collection of categories whose objects are Petri nets and whose morphisms are refinement maps. These categories are based on de Paiva's dialectica category \mathbf{GC} , and like \mathbf{GC} are sound models of linear logic. The rich structure of our categories gives rise to both new and existing constructions on Petri nets. Thus we can interpret the linear connectives \wedge , \otimes , \multimap , \oplus and $(-)^{\perp}$ as constructors on nets, and these constructors have interesting computational meanings. Restriction is readily handled in our framework, and this, together with the net constructors \otimes and \wedge , allows us to represent any binary, commutative, associative parallel composition of nets by a simple construction in the category. Several existing categories arise as special cases of our construction, and the unifying framework of the dialectica categories allows us to reason about these different categories by considering a single category. One of the aims of a categorical approach to processes is to give a compositional treatment which, combined with categorical logic, gives rise to proof systems and specification languages for parallel processes. In addition, functors between different categories of model can be used to relate the various models of concurrency.

While Part II considers the specification of net structure, Parts III and IV apply linear logic to net behaviour. In Part III, we show that the possible evolutions of a net give rise to a quantale. Since quantales are algebraic models of linear logic, we obtain a sound model of linear logic based on the dynamic behaviour of a net. We can then use linear logic to specify dynamic properties of a net, including safety and liveness properties. Linear implication expresses the possibility of an evolution, and thus acts as a modal operator. We show further that such restrictions on nets as being safe, or bounded, arise naturally as the subquantales induced by various conuclei.

In Part IV, we show that certain formulae of linear logic correspond precisely to finite, marked Petri nets. This allows us to give a semantics in terms of nets for a considerable fragment of linear logic. This semantics is complete, and sound where it is defined. We show that evolution in nets corresponds precisely to linear proof, and hence we can apply the proof theory of linear logic to the study of net behaviour. For any property of nets which can be expressed as a formula F in the calculus \mathcal{L}_0 , we establish that a net N satisfies the property by giving a derivation of F from the formula representing N . Because formulae correspond precisely to nets, we can combine nets using any of the operators with which we usually compose formulae. Thus we have a rich language for describing nets, as in Part II. The results of Part IV allow us to reason about both structural and behavioural properties of nets.

Further Work

We now indicate some of the directions in which the work of this thesis could be extended.

Extending the Results of Part II

The principal area for further work arising from the material of Part II lies in the extension of the theory to cover arbitrary nets, rather than restricting our attention to the elementary nets, as we do here. Work in progress with de Paiva will address this issue. We here outline the approach which will be taken (see also [BG90]).

We can regard a relation on sets \mathcal{E} and \mathcal{B} as a function from $\mathcal{E} \times \mathcal{B}$ into $\mathbf{2}$, and a multi-relation as a function from $\mathcal{E} \times \mathcal{B}$ into \mathbf{N} . Thus we can represent a Petri net $N = \langle \mathcal{E}, \mathcal{B}, pre, post \rangle$ by a pair of functions $\mathcal{E} \times \mathcal{B} \rightarrow \mathbf{N}$. We build a category whose objects are Petri nets as follows:

- objects are nets, denoted $\mathcal{E} \times \mathcal{B} \xrightarrow{\alpha} \mathbf{N}$,

- a morphism from $\mathcal{E} \times \mathcal{B} \xrightarrow{\alpha} \mathbf{N}$ to $\mathcal{E}' \times \mathcal{B}' \xrightarrow{\beta} \mathbf{N}$ is a pair of functions (f, F) where $f: \mathcal{E} \rightarrow \mathcal{E}'$ and $F: \mathcal{B}' \rightarrow \mathcal{B}$ such that

$$\begin{array}{ccc}
 \mathcal{E} \times \mathcal{B}' & \xrightarrow{1 \times F} & \mathcal{E} \times \mathcal{B} \\
 f \times 1 \downarrow & \geq & \downarrow \alpha \\
 \mathcal{E}' \times \mathcal{B}' & \xrightarrow{\beta} & \mathbf{N}
 \end{array}$$

- where the symbol \geq represents an ordering on functions from $\mathcal{E} \times \mathcal{B}'$ to \mathbf{N} , given pointwise, and
- composition is pointwise.

We can generalise de Paiva's constructions to this category, obtaining a sound model of linear logic.

A second important area for research is that of behaviours. Ideally, the operations \otimes , \wedge and so on which we have defined on nets should extend to categorical structure in a category of behaviours of nets. Theorem 5.3.1 indicates that this is straightforward in the case of the product of two nets. However, in the case of the coproduct of two nets the situation is more complex, and the direction to take is not clear. A preliminary approach to this issue is given in [BG90]. Further work should take into account the related results of Nielsen, Rozenberg and Thiagarajan in [ER90], [NRT90] and [Thi87].

Extending the Results of Part III

The natural extension to the semantics we give in Part III to linear logic using a net-quantale is to increase the expressiveness of our language. This involves adding fixed point operators and second order quantification. A soundness proof for such an extended calculus would enable us to make and prove more interesting assertions about nets.

Work in progress with Tofts will consider quantales generated by CCS processes.

Extending the Results of Part IV

There are two important extensions to the results of Part IV. Firstly, it is natural that the use of formulae to represent nets should lead to a notion of refinement on nets, since a formula may be considered either as an atomic *condition* in the net, or, if it has a suitable structure, may represent a whole subnet. For example, suppose $G = A \otimes B \otimes !(A \otimes B \multimap C)$. Then if G is a factor of some F_N , we can either regard G as simply a condition of N , or we can substitute $A \otimes B \otimes !(A \otimes B \multimap C)$ for G in F_N to obtain $F_{N'}$, where N' is in some sense a *refinement* of N , since the place G has been expanded into an event with a marking. The notion of refinement obtained by following this approach is closely related to that discussed in [NRT90]. In general, it is hoped that some of the ease with which we can manipulate formulae of linear logic may be transferred to Petri nets, giving us a slightly different approach to the issue of compositionality.

Secondly, as in the case of Part III, it would be useful to explore the use of second order quantifiers. When a given formula witnesses an existentially quantified formula, the corresponding net should implement the process specified by the quantified formula.

Bibliography

- [ACP86a] Proc. Advanced Course on Petri Nets, 1986, *Part 1: Petri Nets: Central Models and their Properties*.
- [ACP86b] Proc. Advanced Course on Petri Nets, 1986, *Part 2: Petri Nets: Applications and Relations to Other Models of Concurrency*, LNCS 255, Springer-Verlag.
- [Apt81] K. Apt, *Ten Years of Hoare's Logic: A Survey*, ACM Transactions on Programming Languages and Systems, 3(4):431–483, October 1981.
- [Asp87] A. Asperti, *A Logic for Concurrency*, manuscript, November 1987.
- [AV88] S. Abramsky and S. Vickers, *Linear Process Logic*, 1988, notes by Steve Vickers.
- [AV90] S. Abramsky and S. Vickers, *Quantales, Observational Logic, and Process Semantics*, Technical report, Imperial College, London, 1990.
- [Bar85] H. P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, 1985.
- [BG81] R. Burstall and J. Goguen, *An Informal Introduction to Specifications using Clear*, in R. S. Boyer and J. S. Moore, editors, *The Correctness Problem in Computer Science*, pages 185–213, Academic Press, 1981.
- [BG90] C. T. Brown and D. J. Gurr, *A Categorical Linear Framework for Petri Nets*, in Proc. LICS, 1990.

- [BG] C. T. Brown and D. J. Gurr, *A Categorical Linear Framework for Petri Nets*, Inf. and Comp., to appear.
- [Bol79] B. Bollobas, *Graph Theory: an introductory course*, Graduate texts in Mathematics, 63, Springer-Verlag, 1979.
- [Bro89a] C. T. Brown, *Petri Nets as Quantaes*, Technical Report ECS LFCS 89-96, Univ. of Edinburgh, 1989.
- [Bro89b] C. T. Brown, *Relating Petri Nets to Formulae of Linear Logic*, Technical Report ECS LFCS 89-87, Univ. of Edinburgh, June 1989.
- [DMM89] P. Degano, J. Meseguer, and U. Montanari, *Axiomatising Net Computations and Processes*, in *Proc LICS*, 1989.
- [deP87] V. C. V. De Paiva, *The Dialectica Categories*, in Proc. A.M.S. Conference on Categories in Computer Science, Boulder, 1987.
- [deP89a] V. C. V. De Paiva, *A Dialectica-like Model of Linear Logic*, in D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Proc. Category Theory and Computer Science*, Manchester, Springer-Verlag, LNCS 389, 1989.
- [deP89b] V. C. V. De Paiva, *The Dialectica Categories*, PhD thesis, University of Cambridge, 1989.
- [Ehr81] H. D. Ehrich, *On Realization and Implementation*, in Proc. 10th Symp. on Mathematical Foundations of Computer Science, Springer-Verlag, LNCS 118, 1981.
- [EKMP82] H. Ehrig, H-J. Kreowski, B. Mahr, and P. Padawitz, *Algebraic Implementation of Abstract Data Types*, TCS, 20:209-263, 1982.
- [ER90] A. Ehrenfeucht and G. Rozenberg, *Partial 2-structures; Part ii: State Spaces of Concurrent Systems*, Acta Informatica, 27:348-368, 1990.

- [EW89] U. Engberg and G. Winskel, *On Linear Logic and Petri Nets*, manuscript, September 1989.
- [Göd58] K. Gödel, *Über ein bisher noch nicht benützte Erweiterung des finiten Standpunkts*, *Dialectica*, 12:280–287, 1958.
- [Göd80] K. Gödel, *On a Hitherto Unexploited Extension of the Finitary Standpoint*, *Jou. Philosophical Logic*, 9:133–142, 1980.
- [GG89] V. Gehlot and C. Gunter, *A Proof-theoretic Operational Semantics for True Concurrency*, in *Proc. International Conference on Application and Theory of Petri Nets*, Bonn, 1989.
- [GG90] V. Gehlot and C. Gunter, *A Proof-theoretic Operational Semantics for True Concurrency*, in *Proc. LICS '90*, 1990.
- [Gir86] J-Y. Girard, *Linear Logic*, *TCS*, 46:1–102, 1986.
- [GL87] J-Y. Girard and Y. Lafont, *Linear Logic and Lazy Computation*, in *Proc. TAPSOFT 87 (Pisa)*, volume 2, pages 52–66, 1987, Springer-Verlag, LNCS 250.
- [GLT89] J-Y. Girard, Y. Lafont, and P. Taylor, *Proofs and Types*, CUP, 1989.
- [Gur90] D. J. Gurr, *Timing Nets the Poset-enriched, One-object, Lax, Symmetric Monoidal Way*, manuscript, March 1990.
- [Hoa85] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, LNM 344, 1985.
- [Hue85] G. Huet, *Deduction and Computation* in W. Bibel and Ph. Jorrand, editors, *Fundamentals in Artificial Intelligence*, Springer-Verlag, LNCS 232, 1985.
- [Joh77] P. T. Johnstone, *Topos Theory*, Academic Press, 1977.

- [Mil89] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.
- [ML71] S. A. Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, 1971, Graduate texts in Mathematics, 5.
- [MM88a] J. Meseguer and U. Montanari, *Petri nets are Monoids: A new algebraic foundation for net theory*, in Proc LICS, 1988.
- [MM88b] J. Meseguer and U. Montanari, *Petri nets are Monoids: A new algebraic foundation for net theory*, Technical Report SRI-CSL-88-3, C.S. Lab., SRI International, January 1988.
- [MOM89] N. Marti-Oliet and J. Meseguer, *From Petri Nets to Linear Logic*, in D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, Proc. Category Theory and Computer Science, Manchester, 1989.
- [Mul86] C. J. Mulvey, "&". *Suppl. ai Rend. del Circ. Mat. di Palermo, Serie II*, 12:99–104, 1986.
- [New42] M. H. A. Newman, *On Theories with a Combinatorial Definition of "Equivalence"*, Ann. of Math., 43(2):223–243, 1942.
- [NR88] S. Niefield and K. Rosenthal, *Constructing Locales from Quantales*, Math. Proc. Camb. Phil. Soc., 104:215–233, 1988.
- [NRT90] M. Nielsen, G. Rozenberg, and P. S. Thiagarajan, *Elementary Transition Systems*, Technical Report DAIMI PB – 310, DAIMI, Århus University, April 1990.
- [Plø76] G. D. Plotkin, *A Powerdomain Construction*, SIAM J. Comp., 5, 1976.
- [Rei85] W. Reisig, *Petri Nets: an Introduction*, EATCS Monographs on Theoretical Computer Science Springer-Verlag, 1985.

- [Roz87] G. Rozenberg. *Behaviour in Elementary Net Systems*, in Proc. Advanced Course on Petri nets, pages 60–94, Springer–Verlag, LNCS 254, 1987.
- [ST88] D. Sannella and A. Tarlecki, *Toward formal development of programs from Algebraic Specifications: implementations revisited*, Acta Informatica, 25:233–281, 1988.
- [Thi87] P. S. Thiagarajan, *Elementary Net Systems*, in Proc. Advanced Course on Petri Nets, pages 26–59, Springer–Verlag, LNCS 254, 1987.
- [Tro73] A. S. Troelstra, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, Springer–Verlag, LNM 344, 1973.
- [Win80] G. Winskel, *Events in Computation*, PhD thesis, University of Edinburgh, 1980.
- [Win82] G. Winskel, *Event Structure Semantics of CCS and related Languages*, in M. Nielsen and E.M. Schmidt, editors, Proc. ICALP, Springer–Verlag, LNCS 140, 1982.
- [Win84a] G. Winskel, *Categories of Models for Concurrency*, in S.D. Brookes, A.W. Roscoe, and G. Winskel, editors, Proc. Seminar on Semantics of Concurrency (CMU, Pittsburgh), Springer–Verlag, LNCS 197, 1984.
- [Win84b] G. Winskel, *Synchronisation Trees*, TCS, 1:33–82, 1984.
- [Win87] G. Winskel, *Petri Nets, Algebras, Morphisms, and Compositionality*, Information and Computation, 72:197–238, 1987.
- [Win88] G. Winskel, *A Category of Labelled Petri nets and Compositional Proof System*, in Proc. LICS, 1988.
- [Wir71] N. Wirth, *Program Development by Stepwise Refinement*, Comm. ACM, 14(4):221–227, 1971.

- [Yet] D. Yetter, *Quantales and non-commutative Linear Logic*, Jour. Symb. Logic, to appear.

Index

Index

- assigned
 - products and coproducts, 69
 - structure, 69
- asynchronous event, 88
- augmentation of a net, 22
- bipartite directed graph, 96
- bounded Petri net, 154
- canonical
 - derivation, 205, 207
 - formula, 163
- choice
 - external, 190, 191
 - internal, 190
- closure operator, 136
- co-closure operator, 136
- commutative quantale, 135
- compatible relation, 168
- concurrent event, 20
- condition, 18
- conflict, 21, 186
- constant of linear logic, 25, 26
- contractum, 174
- coproducts
 - assigned, 69
 - disjoint, 55
 - stable, 55
 - disjoint, stable 55
- cycle of markings, 143
- data
 - external, 190
 - internal, 190
- decreasing operator, 136
- dense graph, 96
- derivability relation, 21, 137
- derivation
 - canonical, 205, 207
 - permitted, 154
- dialectica, 50
 - category, 41
- diamond property, 177
- directed graph, 47, 142
- directed graph morphism, 47
- disjoint coproducts, 55
- disjoint, stable coproducts, 55
- elementary net, 58
- empty multiset, \emptyset , 17
- enabled event, 20
- event, 18
 - asynchronous, 88
 - concurrent, 20
 - enabled, 20
 - firing of, 20

- irrelevant, 156
- multiple, 19
- removal of, 22
- evolution, 21
 - backwards closure under, 141
 - forwards closure under, 139
 - one-step, 20
- external
 - choice, 190, 191
 - data, 190
- factor of a formula, 33
- firing of an event, 20
- formula
 - canonical, 163
 - factor of, 33
 - isomorphism of, 34, 166
 - length of, 168
 - linear logic, 32
 - permutation of, 34
 - substitution into, 170
 - transposition of, 34
- graph
 - bipartite directed, 96
 - dense, 96
 - density of, 96
 - directed, 47, 142
 - loop in, 96
 - order of, 95
 - size of, 95
 - sparse, 96
 - totally connected, 143
 - transition, 82
- identity event, 151
- internal
 - choice, 190
 - data, 190
- intersection
 - of multisets, 16
- irrelevance-free, 156
- irrelevant event, 156
- k -bounded Petri net, 21
- Kleisli category, 40
 - for a monad T (defn), 126
- label of an event, 46
- labelling function, 46
- length of a formula, 168
- linear extension of function, 17
- linear logic
 - LL_* , 28
 - \mathcal{L}_0 , 29
 - \mathcal{L}_1 , 28
 - classical, 26
 - constants of, 26
- linear logic formula, 32
- linear negation, 57
- locally cartesian closed, 53
- marking, 18
 - permitted, 153
- monad, 125
- multiple event, 19
- multirelation, 15

- composition, 15
- inverse function regarded as, 17
- restriction, 15
- union, 15
- multiset, 16
 - empty, 17
 - finite, 17
 - inclusion, 16
 - intersection, 16
 - linear extension of function to, 17
 - membership, 16
 - non-empty, 17
 - restriction, 17
 - subset of, 16
 - subtraction, 16
 - sum, 16
- multisubset, 16
- net represented by F , 196
- one-step evolution, 20
- opposite relation, 15
- permitted derivation, 154
- permitted marking, 153
 - notion of, 154
- permutation of a formula, 34
- Petri net, 18
 - represented by formula, 166
 - as bipartite graph, 96
 - augmentation of, 22
 - bounded, 154
 - elementary, 58
 - irrelevance-free, 156
 - isomorphism, 165
 - restriction, 22
 - safe, 21, 154
- post-condition
 - relation, 18
 - set, 18
- pre-condition
 - relation, 18
 - set, 18
- products
 - assigned, 69
- quantale (commutative), 135
- quantale homomorphism, 136
- quantic
 - conucleus, 137
 - nucleus, 137
- reduction
 - path, 176
 - relation, 168
- refinement, 60
 - relative to a function, 60
- refinement map, 60
- relation, 14
 - post-condition, 18
 - pre-condition, 18
- relative refinement, 60
- removal of an event, 22
- replacement of a formula, 34
- restriction
 - of a multiset, 17

- of a net, 22
- S , 169
- safe Petri net, 21, 154
- set
 - post-condition, 18
 - pre-condition, 18
- short cut, 151
- slice category, 53
- sparse
 - graph, 96
- stable coproducts, 55
- strongly normalising, 176
- subformula, 33
- subgraph, 142
- subnet, 21
- subobject, 14
- subquantale, 137
- symmetric monoidal
 - category, 53
 - structure, 54
- symmetric monoidal closed
 - category, 54
 - structure, 54
- synchronisation algebra, 46, 78
- synchronous morphism, 43
- tensor product, 54
- tensor sum, 17
- token, 20
- totally connected graph, 143
- transition graph, 82, 143
- transposition of a formula, 34
- trap, 142
- union of multirelations, 15
- weak diamond property, 177

Index of Notation

Sets

$\mathbf{0}$, 14
 $\mathbf{1}$, 14
 At , 33
 \mathcal{F} , 163
 \mathcal{N} , 188
 Can , 164
 Lin , 32
 Lin_{\cong} , 35
 $Petri$, 18
 $Mark(N)$, 19

Categories

Cat , 68
 C/A , 53
 GC , 51
 GC^{co} , 102
 \mathcal{NC} , 127
 NC , 58
 $NC^=$, 113
 $NC_{\subseteq}^{\subseteq}$, 108
 $NC_{\supseteq}^{\supseteq}$, 109
 NC^{co} , 109
 NC_T , 125
 Net , 42
 Net_{syn} , 43
 $NSet$, 59
 $NSet^=$, 124
 $SafeNet_{syn}$, 43
 $SafeNet$, 43
 $Quant$, 136

Notation for Petri Nets

N , 18, 19
 $N[\mathcal{E}'$, 22
 \leq , 21
 \leq_1 , 21
 \leq_P , 154
 pre , 18
 $post$, 18

Orderings

\leq , 21, 33, 139
 \leq_1 , 21
 \leq_P , 154
 \supseteq , 181, 182
 \longrightarrow , 170
 \longrightarrow , 170
 \longrightarrow_R , 168
 $\xrightarrow{\Delta}_R$, 175
 \longrightarrow_R , 168
 $=_R$, 168

Multiset Notation

\emptyset , 17
 $\alpha|_B$, 17
 $\alpha(a)$, 16
 $a \in_m \alpha$, 16
 $\beta \subseteq_m \alpha$, 16
 \emptyset , 17

Miscellaneous Notation

$F[X'/X]$, 170
 F_N , 166

form, 165

$q(N)$, 138

Q_N , 140

Q_N^P , 155

R^{op} , 15

$=_S$, 167

N_F , 166

$T(N)$, 143

$(\mathcal{E} \xleftarrow{\alpha} \parallel B)$, 58

$(\mathcal{E} \xleftarrow{\alpha} \dashv B)$, 51

$\alpha: \mathcal{E} \rightarrow_m B$, 15

$(-)^{\perp}$, 56

\downarrow , 139

\models_N , 150

\sim , 150

$\ulcorner - \urcorner$, 164