Centrum voor Wiskunde en Informatica

Centre for Mathematics and Computer Science

J.N. Kok

Data flow semantics

Computer Science/Department of Software Technology   Report CS-R8835   August

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Data Flow Semantics

Joost N. Kok *
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

In this paper we study the semantics of data flow. A data flow net is made up of several basic nodes which are connected by lines. The behaviour of a node is specified by a relation and is in general nondeterministic. First we assign a history based operational semantics to data flow nets, which models a net as a function from inputs to sets of possible outputs. Moreover, we present an alternative definition in which we associate with each node an automaton. Brock and Ackerman showed that a history based semantics is not compositional: it is not fine enough. We introduce an intermediate semantics which is shown to be compositional by proving the intermediate semantics equivalent to a denotational semantics, which is compositional by definition. It is also proved that the denotational semantics is fully abstract with respect to the operational semantics: it generates the greatest congruence which is contained in the equivalence relation generated by the operational semantics. It is shown that an alternative definition of the denotational semantics involves an (implicit) fixed point: a fixed point of a multivalued function (contain point) is taken. Some properties of contain points are given in a metric topological setting. These properties are used to show that for a restricted class of data flow nets there exists a metric compositional semantics in which the contain points can be obtained by iteration. It is called a metric semantics because the domains are metric spaces. We show that it is correct with respect to the operational semantics. For this semantics we can abstract from delay along the lines in the net. However, this metric semantics is not fully abstract (with respect to the operational semantics).

Key words and phrases: operational semantics, denotational semantics, parallelism, data flow, full abstraction, metric topology, contain points, firing sequences.
1985 Mathematics Subject Classification: 68Q55, 68Q10.
1987 Computing Reviews Categories: D.1.3, D.3.1, F.1.2, F.3.2.

*This work has been carried out in the context of LPC: the dutch National Concurrency Project, supported by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.), grant 125-20-04

Report CS-R8835
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Contents

1 Introduction

We shall introduce several semantics for data flow nets. In [Kok 1986], [Kok 1987] and [de Bakker & Kok 1985] we already considered several models. This paper can be seen as an integration and extension of ideas present in these papers.

First we explain what is meant by a data flow net. A data flow net (in this paper) consists of some nodes that communicate by passing tokens over lines. Each node has a fixed number of input lines and a fixed number of output lines. A line either connects two different nodes or is a feedback line. A feedback line passes tokens back to the node itself. A line is a directed FIFO channel. Tokens that are passed over a line by a node arrive in an unspecified but finite time at the destination node in the same order as they are sent. If the nodes in a net behave (non)deterministically we say that a net is (non)deterministically.

We look at semantic models that describe the behaviour of such nets. Data flow semantics is often based on a history model. A history is a finite or infinite word over the alphabet of tokens. A history of a line can be seen as the tokens that have passed this line during the execution. A history tuple is a tuple of histories. A history function is a function from history tuples to sets of history tuples. A history model of a net is a history function: for a deterministic net it is a function from history tuples to history tuples and for nondeterministic nets is a function from history tuples to sets of history tuples. With a history model the external behaviour of a net is described: a net is seen as a black box in which internal behaviour is hidden from the outside world.

Already in 1974 a history model was presented by Kahn: in [Kahn 1974] a semantic model is described for deterministic nets. A history function is associated with a net. The Kahn model is a nice example of the application of (complete) partial order theory. A net is described by a set of equations. When interpreted over the domain of histories, this set has a (least) solution. This solution can be obtained by iteration of the operator associated with the set of equations from the empty history tuple (a tuple with as elements only the empty word). This solution is taken as the meaning of the net. The Kahn-model is a very general model which can be applied in a lot of situations. However, only a certain class of data flow nets is modeled. Nodes that appear in nets have to behave deterministically.

Subsequently researchers have tried to extend this model to more general classes of nets, for example to nondeterministic nets. A straightforward extension does not work in this case. One of the problems was first shown by Brock and Ackerman in [Brock & Ackerman 1981]. When we take the Kahn approach, the semantics of a deterministic net is a history function from history tuples to history tuples. For a nondeterministic net we can take a function from history tuples to sets of history tuples. Brock and Ackerman showed that such a semantics is not compositional. A semantics is called compositional if the following condition holds:

> For any context (a data flow net with a hole in it) and for any two nets which have the same semantics, whenever we place these two nets in the context the two resulting nets should have the same semantics.

In order to show that a history model for nondeterministic nets is non-compositional, Brock and Ackerman present two data flow nets that have the same history function. Then they construct a context which shows the non-compositionality of the semantics: when we place the two nets in this context the resulting nets have different history functions. This construction is shown in full detail in section 3.

If we want to design a compositional semantics, we have to add information to the history model. In the literature this is done in several ways, including:

1. the addition of special tokens ([Park 1983] 'hiatons'),

2. the use of graph like structures ([Brock & Ackerman 1981] 'scenarios', [Pratt 1984] 'partially ordered multisets'),

3. the use of traces ([Keller & Panangaden 1984] 'archives', [Jonsson 1987a] 'quiescent traces'),

4. providing oracles ([Park 1983] , [Broy 1988]).

We try to give some flavor of the different approaches. For details consult the cited articles and reports.

Hiatons are silent actions. We can use hiatons to make certain functions contracting and hence continuous in their history arguments. This enables us to use complete partial order theory as in the approach of Kahn. Such a procedure is called the extended Kahn principle by Park.

Graph like structures describe an ordering on events. Production and consumption of a token are seen as events. Events are partially ordered in a graph or in a multiset. In this way we get information on the relative timing of events. It is this kind of information that is needed to obtain a compositional semantics: Brock and Ackerman present a scenario set algebra which is compositional.

Traces provide a total ordering on events in the sense of the previous paragraph. Quiescent traces are maximal traces: in order to get a longer trace, we have to provide more input.

Oracles can be seen as a method to eliminate the nondeterminism. They provide the information about choices. Given an oracle, a nondeterministic node behaves deterministically. Fair merge nodes can be described with fair oracles.

Our approach can be seen as a combination and an extension of the first two approaches. As basic domain we use the set of finite-word vectors: vectors with as elements finite words of tokens. (In the model of Kahn the basic domain is the set of histories). A finite-word vector can contain empty words. The role of empty words in our approach can be compared with the role of hiatons in the approach of Park. Nets are modeled as finite-word vector functions: functions from tuples of finite-word vectors to sets of tuples of finite-word vectors. (Compare with the model of Kahn: he uses functions from history tuples to sets of history tuples). Whereas Kahn considers on one single line a possibly infinite word of tokens, in our model we cut this sequence in finite (possibly empty) pieces. We find it convenient to group a tuple of finite-word vectors in a finite-word matrix: a matrix with elements taken from the set of finite words over the alphabet of tokens. This is possible because we use finite-word vectors of infinite height. Hence, a net is modeled by a function from finite-word matrices to sets of finite-word matrices. From now on, we will call these functions finite-word vector functions. From such a function we can derive a certain ordering of events: Restricting ourselves to 1-1 finite-word vector functions (functions from finite-word matrices of width 1 to sets of finite-word matrices of width 1) and given a finite-word matrix $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ we will get a set of possible finite-word matrices. If $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \end{pmatrix}$ is one of the possible outcomes, then we have that the word x_1 contains enough tokens for the net to produce $\bar{x}_1$, the word $x_1 x_2$ contains enough tokens for the node to produce $\bar{x}_1 \bar{x}_2$, etc. This provides us with information about relations between input and output as is done in the scenarios of Brock and Ackerman, the traces of Jonsson or the archives of Keller and Panangaden.

It is interesting to investigate the minimal amount of information we have to add to a Kahn like semantics in order to get a compositional model: This is the question of full abstraction. A model is called fully abstract with respect to the history model if it generates the greatest congruence which is contained in the equivalence relation generated by the history model.

In the literature we find three results and/or claims:

1. Staples and Nguyen claim in [Staples & Nguyen 1985] without proof that the frameworks in [Back & Mannila 1982], [Brock & Ackerman 1981], [Keller 1978], [Kosinski 1978], [Park 1983] are not fully abstract frameworks.

2. Jonsson proves in [Jonsson 1987b] that his framework is a minimal extension of the Kahn model that is compositional.

3. In [Kearney & Staples 1987] an extensional model is given for an oracle based semantics.

We prove that our denotational semantics is fully abstract with respect to the history model. In order to prove that our denotational semantics is fully abstract, we give, for any two nets that have a different operational semantics (i.e. different history functions), a context in which they have different history functions. Jonsson makes in his proof explicit use of fairness constraints. A specification of the behaviour of a node in his framework includes fairness requirements. His proof does not apply to the (sub)set of nets with nodes which do not have fairness requirements. The proof in [Kearney & Staples 1987] is based on so-called 'wait-visualizer' nodes. A 'wait-visualizer' node makes a hiaton visible. In [Kearney & Staples 1987] it is remarked that "there may be some cause for concern over the extensionality of the model, since the 'wait-visualizing' process (*viz*) defies reasonable specification at the non-deterministic level". In our formalism it is not possible to specify such kind of nodes.

Another way to classify approaches is to look at the underlying mathematical model. We can distinguish at least:

1. order-theoretic frameworks ([Abramsky 1984], [de Bakker et al 1985], [Broy 1983], [Broy 1988], [Broy 1985], [Kahn & MacQueen 1977], [Staples & Nguyen 1985], [Kearney & Staples 1987])

2. algebraic frameworks ([Brock & Ackerman 1981], [Back & Mannila 1982], [Bergstra & Klop 1983]),

3. automata theoretic and/or transition systems ([Arnold 1981], [Jonsson 1987a], [Jonsson 1987b]),

4. metric topological models ([de Bakker & Kok 1985], [Kok 1986], [Kok 1987]),

5. category theoretic frameworks ([Abramsky 1983],[Keller & Panangaden 1984]).

In this paper we use a combination of the models mentioned in 3. and 4.

We now give an overview of the rest of the paper. Section 2 gives a history model for a nondeterministic version of data flow nets along the lines of Kahn. It is a non-compositional model. The behaviour of a node is described by its specification. A specification is a set of firing rules: A firing rule is a four tuple that has as elements an initial state, input, final state and output. If the node is in the initial state and has the specified input on its input lines it can fire: enter the final state and produce the output. Given a net t we can derive its operational meaning by considering firing sequences for the nodes in net t. The operational model serves as basic model in the sense that it describes the observational behaviour of a net. We also provide an alternative definition for the operational semantics based on automata. For each node d an automaton M is constructed from the specification δ of d. A system of several automata which can share tapes yields the semantics of a net.

Section 3 provides the details concerning the fact that the operational model is not compositional. An alternative semantic model is proposed. This model differs from the operational model: we use different domains. The semantic model is called intermediate because it is a step on the way to a denotational model: It is used because it has a close link with the operational intuition. The contents of a line is described with a finite-word vector and the semantics of a net is modeled by finite-word vector function. We give some properties of this intermediate semantics. Again, we provide an alternative definition for the intermediate semantics based on automata. We also show the correctness of the intermediate semantics with respect to the operational semantics: there exists an abstraction operator *abstr* which relates the two semantic models.

In section 4 we introduce a third model (the denotational semantics) and we prove the compositionality of the intermediate semantics: if the intermediate semantics of two nets t_1 and t_2 are equal then for any context C we have that the semantics of $C[t_1]$ and $C[t_2]$ are equal. The compositionality is proved with the help of the denotational semantics. The denotational semantics is compositional by definition, and it is shown that it equals the intermediate semantics. For an overview, see figure 1. Also an associativity result is proved: the order in which lines are connected in a net does not influence the semantics. This is approached as follows. We introduce the notion of a normal form of a net. It is shown that the semantics of a net and its normal form are the same. The notion of normal form is defined in such a way that if two nets are equal, but for the order in which the lines are connected, then they have the same normal form.

Section 5 is devoted to full abstraction. We prove that the denotational model is the minimal extension of the operational model that is compositional. For any two nets t_1 and t_2 that have a different denotational semantics, we provide a context C such that the operational semantics of $C[t_1]$ and $C[t_2]$ differ. From this result we derive the full abstraction: the equivalence relation generated by the denotational semantics is the greatest congruence contained in the equivalence relation generated by the operational semantics. In addition, we have the rather surprising fact that we can use in all cases a context which does not depend on the nets. On the other hand, depending on the fact whether the input is finite or infinite, an *amerge* (angelic merge) or *imerge* (infinity merge) node is used in the context. Suppose we have two nets that have a different denotational semantics and the same operational semantics. (If they have a different operational semantics we can take an empty context). From this we can conclude there is a timing difference between the two nets: the output is produced in a different way. We can make this difference visible in the operational semantics by tagging the output and feeding it back as soon as possible to a merge node which merges this tagged output with the original input. The resulting history on the output line of the merge node is a mixture of tagged tokens (from the output that is fed back) and tokens that are not tagged (from the original input). With a split node we make copies of all tokens that are sent along the output line of the merge node. One of these copies is delivered as output and the other copy is sent to a node that removes the tagged tokens. This node generates the original input which is sent to either t_1 or t_2. Due to the timing difference we observe (in our operational semantics) a different mixture of tagged tokens and tokens that are not tagged.

In the Kahn model for deterministic data flow there is no explicit modeling of delay along lines. In all three models that are introduced in sections 2, 3 and 4 we have an explicit modeling of delay in the sense that besides all outputs also all delays are delivered. In section 6 we investigate a class of nets (the so-called finite-choice nets) in which we can abstract from delay. This enables us to set up a metric topological framework: we can work with closed sets. A finite-choice net is a net in which all the nodes are finite-choice nodes. A finite-choice node has certain restrictions on its specification. One of these restrictions is that after a bounded amount of input a node does not have to wait any more for more input for the next firing: the

$\mathcal{O}$ operational semantics | $\mathcal{I}$ intermediate semantics | $\mathcal{D}$ denotational semantics

$\mathcal{O} = abstr \circ \mathcal{I}$ | $\mathcal{I} = \mathcal{D}$

Figure 1: the three models

arrival of new input does not influence the set of applicable firing rules any more. We define a metric semantics for finite-choice nets: the domains of this semantics are metric spaces. In fact, we assign metrics to the domains used by the denotational semantics. We use contain points in the definition of the metric semantics. These contain points can be obtained by iteration. We show the correctness of the semantics by relating it with the operational model. The abstraction operator we use is the same as the one that relates the operational semantics and the denotational semantics. Though correct, the metric semantics is not fully abstract. This is due to the fact that in the definition of the metric semantics the simultaneous integer speed up and the delay are not applied. This simultaneous integer speed up together with delay enabled us to prove the full abstraction of the denotational semantics. In the proof these two operators (simultaneous integer speed up and delay) play an important role. We can not apply them because in general they disturb the closedness. This can not be simply remedied by including them in the metric semantics since they do not preserve closedness. Hence the metric semantics distinguishes too many nets.

2 Operational semantics $\mathcal{O}$

In this section we introduce an operational semantics for nondeterministic dataflow nets. The operational semantics is a history model along the lines of Kahn: it models a net as a function from history tuples to sets of history tuples (such a function is called a history function). A history is a finite or infinite word over the alphabet of tokens. A history tuple is a tuple of histories. We use sets of history tuples because we want to model nondeterministic nets: nets with nodes that can behave nondeterministically. The definition of the operational semantics exploits the notion of firing sequences of nodes. We also provide an alternative definition of the operational semantics which uses automata to describe the behaviour of nodes and nets.

This section is divided in five subsections: first we introduce some basic domains and operations, the second subsection gives the syntax of data flow nets, in the third subsection we investigate how to specify the behaviour of nodes, in the fourth subsection we define an operational semantics $\mathcal{O}$, and the fifth subsection gives the alternative definition.

2.1 Basic domains and operators

Let **N** be the set of integers. *Throughout our paper integer will always mean positive integer unless explicitly stated otherwise.* Let A be an, in general infinite alphabet, the elements of which will be called tokens. Let a be a typical element of A. Let A^* be the set of finite words over the alphabet A, with typical element x. Let A^∞ be the set of finite and infinite words over the alphabet A, with typical element y. The empty word is denoted by ϵ.

Definition 2.1 *For any integer $n \geq 0$, $FTrace^n$ is the set of finite-word tuples with n elements which are taken from A^*. Let χ be a typical element of $FTrace = \bigcup_n FTrace^n$.*

Definition 2.2 *For any integer $n \geq 0$, $Trace^n$ is the set of word tuples with n elements which taken from A^∞. Let Γ be a typical element of $Trace = \bigcup_{n\geq 0} Trace^n$. Let $Trace^{n:m} = Trace^n \rightarrow \mathcal{P}(Trace^m)$, where $\mathcal{P}(\cdot)$ denotes subsets of $\cdot$. Elements of Trace are called history tuples and elements of $\bigcup_{n,m\geq 0} Trace^{n:m}$ are called history functions.*

Definition 2.3 *Take any integers k, n such that $0 \leq k \leq n$. Define*

$$Tup_{k.n} = \{< i_1, \ldots, i_k >:$$
$$\forall j \in \{1, \ldots, k\}[1 \leq i_j \leq n \wedge \forall l \in \{1, \ldots, k\}[l \neq j \Rightarrow i_j \neq i_l]]\}$$

1. *(projection from $Trace^n$)*

$$\downarrow: Trace^n \times Tup_{k.n} \rightarrow Trace^k$$

$$(y_1, \ldots, y_n) \downarrow < i_1, \ldots, i_k > = (y_{i_1}, \cdots, y_{i_k}),$$

2. *(projection from $FTrace^n$)*

$$\downarrow: FTrace^n \times Tup_{k.n} \rightarrow Trace^k$$

$$(x_1, \ldots, x_n) \downarrow < i_1, \ldots, i_k > = (x_{i_1}, \ldots, x_{i_k}),$$

3. *(slicing from $Trace^n$)*

$$\uparrow: Trace^n \times Tup_{k.n} \rightarrow Trace^{n-k}$$

$\Gamma \uparrow \{i_1, \ldots, i_k\}$

is the word tuple obtained by leaving out the i_1*-th* $, \ldots, i_k$*-th components of* Γ*,*

4. *(slicing from* $FTrace^n$*)*

$$\uparrow : FTrace^n \times Tup_{k.n} \rightarrow FTrace^{n-k}$$

$$\chi \uparrow \{i_1, \ldots, i_k\}$$

is the finite-word tuple obtained by leaving out the i_1*-th* $, \ldots, i_k$*-th components of* χ*.*

On A^* and A^∞ we have the normal concatenation of words which we extend to $FTrace^n$ and $Trace^n$ in

Definition 2.4

1. *(concatenation of* $FTrace^n$*)*

$$(x_1, \ldots, x_n)(\bar{x}_1, \ldots, \bar{x}_n) = (x_1\bar{x}_1, \ldots, x_n\bar{x}_n),$$

2. *(concatenation of* $Trace^n$*)*

$$(y_1, \ldots, y_n)(\bar{y}_1, \ldots, \bar{y}_n) = (y_1\bar{y}_1, \ldots, y_n\bar{y}_n).$$

We define the prefix order on words in A^* and A^∞ and extend it to $FTrace^n$ and $Trace^n$ in

Definition 2.5

1. *(prefix order on* A^* *and* A^∞*)*
We put $x_1 \leq x_2$ *if* x_1 *is a prefix of* x_2 *and* $y_1 \leq y_2$ *if* y_1 *is a prefix of* y_2*.*
2. *(ordering on* $FTrace^n$*)*

$$(x_1, \ldots, x_n) \leq (\bar{x}_1, \ldots, \bar{x}_n) \Leftrightarrow x_1 \leq \bar{x}_1 \wedge \cdots \wedge x_n \leq \bar{x}_n,$$

3. *(ordering on* $Trace^n$*)*

$$(y_1, \ldots, y_n) \leq (\bar{y}_1, \ldots, \bar{y}_n) \Leftrightarrow y_1 \leq \bar{y}_1 \wedge \cdots \wedge y_n \leq \bar{y}_n.$$

2.2 Syntax of data flow nets

Let *Node* be a set of abstract elements called nodes and let d be a typical element of this set. The set *Node* can be partitioned into subsets $Node^{n:m}$ $(n, m \geq 0)$. If d belongs to $Node^{n:m}$ we say that d has n input and m output lines.

We define the syntax of a data flow net. Adhering to the terminology of input and output lines one may understand definition 2.6 as follows: A net consists of a number of basic nodes plus a specification of the connections between these nodes. We put the nodes in a tuple and number the input and output lines from left to right. A connection is specified by two integers: $i : j$ means that we connect the i th input line to the j th output line. Once we have connected two lines they are not visible anymore: we can not make more connections to these lines. (This is no real restriction because we can use so called *split* and *merge* nodes. A *split* node splits its input line: when it receives a token it sends copies of it to all its output lines. We can split a line to which we want to make more than one connection. A *merge* node merges its two input lines into one output line. Later we shall be more specific about merge nodes where necessary.)

Definition 2.6 (Syntax of data flow nets) *Let the set Net of data flow nets have elements that can be constructed as follows: Let $k, l \geq 0$ be any integers. Let $\alpha_1, \ldots, \alpha_k$, $\beta_1, \ldots, \beta_k$ be integers. Let $i_1, \ldots, i_l$ be integers smaller than $\sum_{i=1}^{k} \alpha_i$ and let them all be different. Let $j_1, \ldots, j_l$ be integers smaller than $\sum_{i=1}^{k} \beta_i$ and let them all be different. Let $d_1, \ldots, d_k$ be nodes such that $d_1 \in Node^{\alpha_1:\beta_1}, \ldots, d_k \in Node^{\alpha_k:\beta_k}$. Then*

$$< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}$$

is an element of the set Net. As with the set Node of nodes, we can partition the set of nets Net in sets $Net^{n:m}$: the element constructed above is an element of $Net^{n_0:m_0}$ where $n_0 = (\sum_{i=1}^{k} \alpha_i) - l$ and $m_0 = (\sum_{i=1}^{k} \beta_i) - l$.

Example

Assume the following nodes: $+ \in Node^{2:1}$, *merge* $\in Node^{2:1}$, *plus1* $\in Node^{1:1}$, *split* $\in Node^{1:2}$. Consider

$$< merge, split, plus1 > \{2 : 4, 3 : 1, 4 : 3\}.$$

This is a data flow net according to definition 2.6. We can construct it in the following way. We follow the informal description above this definition: We form the tuple

$$< merge, split, plus1 >$$

and with each input and output line of a node in the tuple we associate a number. We number the lines from left to right:

	1 2		3		4		← input lines
<	*merge*	,	*split*	,	*plus1*	>	
	1		2 3		4		← output lines

Then we specify the connections:

$$< merge, split, plus1 > \{2 : 4, 3 : 1, 4 : 3\}.$$

Each data flow net has a graph-like representation. Figure 2 shows such a graph for the net

$$< merge, split, plus1 > \{2 : 4, 3 : 1, 4 : 3\} \in Net^{1:1}.$$

The graph resembles a directed graph with labeled edges and vertices, except for the fact that there are special kinds of edges. For this reason, formally these representations are not proper graphs. When we look at figure 2 we see that the graph has three vertices, labeled with *merge*, *split*, *plus1* respectively. Labels of vertices are taken from the set *Node*. There are three kinds of edges:

1. input edges,

2. output edges,

3. feedback edges.

Feedback edges are normal directed edges which connect two vertices which may coincide. Input edges are edges that are directed, but do not have a starting vertex. They only point to a vertex. Output edges are also directed edges, but they do not point to a vertex. Input and output edges are labeled with an integer. Feedback edges have labels of the form $i : j$. The graph-like structure of figure 2 is in this paper often pictured as in figure 3.

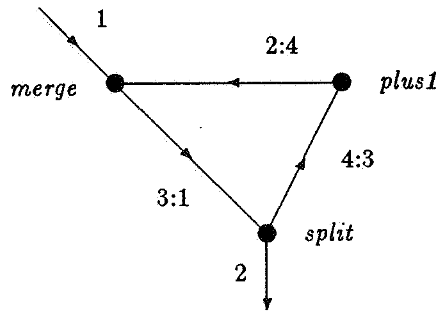

Figure 2: $< merge, split, plus1 > \{2:4, 3:1, 4:3\}$

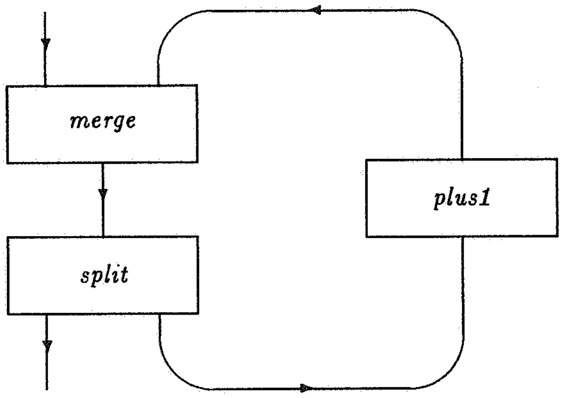

Figure 3: $< merge, split, plus1 > \{2:4, 3:1, 4:3\}$

2.3 Specification of the behaviour of nodes

The operational semantics will be defined with the help of firing sequences. For this we need a specification of the behaviour of a node which is a set of four tuples. Elements of firing sequences for a node are elements of the specification of the node. We give the formal definition of such a specification. Assume given a set of states Σ, with typical element σ. We assume a special state σ_{INIT} in Σ called the initial state.

Definition 2.7 (Specification) *The set Spec of specifications, with typical element δ, is the union of all sets $Spec^{n:m}$ $(n, m \geq 0)$, where the set $Spec^{n:m}$ of all specifications for nodes with n inputs and m outputs is the set of all subsets of $FTrace^n \times \Sigma \times FTrace^m \times \Sigma$, i.e.*

$$\mathcal{P}(FTrace^n \times \Sigma \times FTrace^m \times \Sigma).$$

Elements of a specification will be called firings.

Note that a specification may be an infinite set. We try to explain the intuitive meaning of such a specification. First we state what it means that a finite-word tuple χ is on the input lines of a node. If $\chi = (x_1, x_2, \ldots, x_n)$ then it means that we have on the first input line of the node the word x_1, on the second input line the word x_2, ..., and on the n th input line the word x_n. In a similar way we can say that $\bar{\chi}$ is written on the output lines. This notation should not be confused with the notation in the introduction: a finite-word vector used in the introduction is intended to describe the contents of a single line, wheras a finite-word tuple in a specification tells something about several lines.

Assume given a node $d \in Node^{n:m}$ and a specification $\delta_d \in Spec^{n:m}$ for this node. Suppose $(\chi_1, \sigma_1, \chi_2, \sigma_2) \in \delta_d$. The intuitive meaning is:

> After the node d in state σ_1 has read χ_1 on the input lines of d, the node d can fire: write χ_2 on its output lines and enter the state σ_2.

This intuition already anticipates on an automata theoretic intuition which will be given later.

Examples

We take in these examples the set of tokens the set of integers $\mathbf{N}$. We give the specifications of a *plus-*, a *split-*, a *merge-* and a *2buffer-*node:

1. $\delta_{plus} = \{((a_1, \bar{a}_1), \sigma_{INIT}, (a_1 + \bar{a}_1), \sigma_{INIT}) : a_1, \bar{a}_1 \in \mathbf{N}\}$,
2. $\delta_{split} = \{((a), \sigma_{INIT}, (a, a), \sigma_{INIT}) : a \in \mathbf{N}\}$,
3. $\delta_{merge} = \{((a, \epsilon), \sigma_{INIT}, (a), \sigma_{INIT}) : a \in \mathbf{N}\} \cup \{((\epsilon, a), \sigma_{INIT}, (a), \sigma_{INIT}) : a \in \mathbf{N}\}$,
4. $\delta_{2buffer} = \{((a), \sigma_{INIT}, (\epsilon), \sigma_a) : a \in \mathbf{N}\} \cup \{((\bar{a}), \sigma_a, (a\bar{a}), \sigma_{INIT}) : a, \bar{a} \in \mathbf{N}\}$.

The merge node specified here is a so called angelic merge node: if the input on one of the input lines is finite, all the input on the other input line will eventually be put on the output line. In section 5 we will see another merge node.

2.4 Operational semantics $\mathcal{O}$

We introduce the notion of a firing sequence for a node with respect to some input Γ. Such a firing sequence is a sequence of elements in the specification δ_d of the node d. The sequence can be seen as all the successive firings of a node for a given input Γ, obeying the restriction that the firings first elements χ_i are together smaller than or equal to Γ (the input available). See figure 4. Moreover if the sequence of firings is finite, it should not be possible to fire again.

$$
\begin{array}{cccccl}
(& x_{11} & ,\ldots, & x_{1k} &) = & \chi_1 \\
(& x_{21} & ,\ldots, & x_{2k} &) = & \chi_2 \\
 & \vdots & & \vdots & & \\
 & \leq & & \leq & & \\
(& y_1 & ,\ldots, & y_k &) = & \Gamma
\end{array}
$$

Figure 4: input restriction

Definition 2.8 (Firing sequence for a node) *A sequence $(\chi_i, \sigma_i, \bar{\chi}_i, \bar{\sigma}_i)_i$ in $FTrace^n \times \Sigma \times FTrace^m \times \Sigma$ is called a firing sequence for a node $d \in Node^{n:m}$ with respect to $\Gamma \in Trace^n$ if*

1. $\sigma_1 = \sigma_{INIT}$ *(the initial state),*
2. $\forall i[(\chi_i, \sigma_i, \bar{\chi}_i, \bar{\sigma}_i) \in \delta_d]$,
3. $\forall i[\sigma_{i+1} = \bar{\sigma}_i]$,
4. $\forall i[\chi_1 \cdots \chi_i \leq \Gamma]$,
5. $(\chi_i, \sigma_i, \bar{\chi}_i, \bar{\sigma}_i)_i$ *is a maximal sequence: if it is of length n then there is no* $(\chi, \sigma, \bar{\chi}, \bar{\sigma}) \in \delta_d$ *such that* $\chi_1 \cdots \chi_n \chi \leq \Gamma$ *and* $\sigma = \bar{\sigma}_n$.

A firing sequence for a tuple of nodes is a combination of firing sequences of each node in the tuple. The projection for a specific node is almost a firing sequence for that node, except for some "dummy" firings. In such a dummy firing, nothing is removed from the input lines, nor is anything put on the output lines. Also there are no state changes for that particular node. A typical situation for the input lines is given in figure 5: Take two nodes with first elements of firing sequences $(\chi_{i1})_i$ and $(\chi_{i2})_i$ respectively. In this defintion we need the notion

$$
\begin{array}{cccc}
\boxed{\chi_{11}} & (\epsilon,\ldots,\epsilon) & = & \chi_1 \\
(\epsilon,\ldots,\epsilon) & \boxed{\chi_{12}} & = & \chi_2 \\
(\epsilon,\ldots,\epsilon) & (\epsilon,\ldots,\epsilon) & = & \chi_3 \\
\boxed{\chi_{21}} & \boxed{\chi_{22}} & = & \chi_4 \\
\boxed{\chi_{31}} & (\epsilon,\ldots,\epsilon) & = & \chi_5 \\
\vdots & \vdots & & \\
\leq & \leq & & \\
\boxed{\Gamma_1} & \boxed{\Gamma_2} & = & \Gamma
\end{array}
$$

Figure 5: input for two nodes

of projection on tuples of states:

Definition 2.9 (projection from Σ^n)

$$\downarrow : \Sigma^n \times Tup_{k.n} \rightarrow \Sigma^k$$

$$(\sigma_1, \ldots, \sigma_n) \downarrow < i_1, \ldots, i_k > = (\sigma_{i_1}, \cdots, \sigma_{i_k}),$$

Definition 2.10 (Firing sequence for a tuple of nodes) *Take any tuple of nodes $< d_1, \ldots, d_n >$ and suppose for all $i \in \{1, \ldots, n\}$ that $< d_1, \ldots, d_i > \in Net^{k_i : l_i}$ and take $k_0 = l_0 = 0$. A sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ in $FTrace^{k_n} \times \Sigma^n \times FTrace^{l_n} \times \Sigma^n$ is called a firing sequence for the tuple of nodes $< d_1, \ldots, d_n >$ with respect to $\Gamma \in Trace^{k_n}$ if*

1. $w_1 = \underbrace{(\sigma_{INIT}, \ldots, \sigma_{INIT})}_{n}$ *(n times the initial state),*
2. $\forall i[\bar{w}_i = w_{i+1}]$,
3. *the sequence $(\chi_i \downarrow < k_{j-1} + 1, \ldots, k_j >, w_i \downarrow < j >, \bar{\chi}_i \downarrow < l_{j-1} + 1, \ldots, l_j >, \bar{w}_i \downarrow < j >)_i$ has a subsequence that is a firing sequence for d_j with respect to $\Gamma \downarrow < k_{j-1} + 1, \ldots, k_j >$ and all elements of the sequence that are not in the subsequence are of the form $((\underbrace{\epsilon \cdots \epsilon}_{k_i - k_{i-1}}), \sigma, (\underbrace{\epsilon \cdots \epsilon}_{l_i - l_{i-1}}), \sigma)$.*

Now we can introduce

Definition 2.11 (Operational semantics $\mathcal{O}$) *Let $\mathcal{O} : Net^{n:m} \rightarrow Trace^{n:m}$ be given by*

$$\bar{\Gamma} \in \mathcal{O}(< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\})(\Gamma)$$

$$\Leftrightarrow$$

$$\exists \Gamma_1 \in Trace^{n+l}, \Gamma_2 \in Trace^{m+l}[$$

$$\Gamma_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\Gamma} \wedge \Gamma_1 \uparrow \{i_1, \ldots, i_l\} = \Gamma \wedge$$

$$\exists (\chi_j, w_j, \bar{\chi}_j, \bar{w}_j)_j \text{ , a firing sequence for } < d_1, \ldots, d_k > \text{ w.r.t } \Gamma_1 \; [$$

$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >] \wedge$$
$$\Gamma_2 = \bar{\chi}_1 \bar{\chi}_2 \cdots$$

$$]$$

$$]$$

Remark

In the definition n, m are free. What we mean is that we have a function

$$\mathcal{O} : \bigcup_{n.m} Net^{n:m} \rightarrow \bigcup_{n.m} Trace^{n:m}$$

such that if $t \in Net^{n:m}$ then $\mathcal{O}(t) \in Trace^{n:m}$. The same remark applies to definitions of other semantics which will be given later.

We give some informal explanation of the definition of the operational semantics. In order to determine whether a pair $(\Gamma, \bar{\Gamma})$ is an input/output pair for the net $< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}$ we consider firing sequences for $< d_1, \ldots, d_k >$. A firing sequence should respect the input Γ: there should be a $\Gamma_1 \in Trace^{n+l}$ such the the firing sequence is with respect to

Γ_1 and where Γ_1 is such that $\Gamma_1 \uparrow \{i_1, \ldots, i_l\} = \Gamma$. Moreover, the feedback behaviour should be achievable: the ith-firing has on the feedback lines not more that the first $i-1$ firings at its disposal: $\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >]$. Moreover we require that all tokens produced in the firing sequence are visible in the output: we can find a $\Gamma_2 \in \mathit{Trace}^{m+l}$ such that $\Gamma_2 = \bar{\chi}_1\bar{\chi}_2 \cdots$ and $\Gamma_2 \uparrow \{j_1, \ldots, j_l\}$.

Examples

Take $t =< \mathit{merge}, \mathit{split}, \mathit{plus1} > \{2:4, 3:1, 4:3\}$.

1. $\mathcal{O}(t)((1)) = \{(12345\ldots)\}$,
2. $\mathcal{O}(t)((11)) = \{(y) : y \in \{12345\ldots n\, 1\, n+1\, 2\, n+2 \ldots : n \geq 1\}\} \cup \{12345\ldots\}$.

2.5 Alternative definition of the operational semantics $\mathcal{O}$

In this subsection we provide an alternative definition of the operational semantics couched in an automata-theoretic framework. We believe that a completly formal framework will enable us to prove the equivalence of the two definitions but we prefer not to do so in order not to detract from the main development. We start with the construction of an automaton for a node d given a specification δ for this node.

Definition 2.12 (Automaton M_δ) *For any specification $\delta \in \mathit{Spec}^{n:m}$ we define an automaton M_δ as follows. The set of states of M_δ consists of those states that appear in δ_d. The automaton M_δ has n read heads that read from n different tapes and has m write heads that write on m different tapes. All tapes may be finite or infinite. The automaton starts in the initial state σ_{INIT} and reads from its input tapes into a buffer β. The buffer is a finite-word tuple with n elements. Each element of the buffer is associated with an input tape. The last token of such an element of the tuple is the one most recently read. The operation of the automaton consists normally in an alternation of a finite number (possibly zero) of read steps and fire steps. When the automaton fires, it picks nondeterministically an element $(\chi, \sigma, \bar{\chi}, \bar{\sigma})$ from δ such that*

1. *the automaton is in state σ*
2. *$\chi \leq \beta$ (ordering as in definition 2.5)*

and

1. *removes χ from its buffer*
2. *writes $\bar{\chi}$ on the output tapes*
3. *enters state $\bar{\sigma}$.*

There is also a fairness condition on the automaton: If an automaton is enabled, it will fire after a finite number of reading steps. The automaton is enabled if there exists a $(\chi, \sigma, \bar{\chi}, \bar{\sigma})$ in δ such that

1. *the automaton is in state σ*
2. *$\chi \leq \beta$.*

The fairness condition prevents the automaton form the situation that, although it can write, it never writes. Note that the tuple that causes the enabledness is not necesarily the same as the tuple that is used in the firing. For instance, we may have the situation that both $((a), \sigma_{INIT}, (a), \sigma_{INIT})$ and $((aa), \sigma_{INIT}, (b), \sigma_{INIT})$ are members of the specification of a node. If we have on the input tape a word that starts with two a's, the automaton can be enabled by the first element of the specification, but can use the second element for the firing.

Example

Take δ_{plus1} to be

$$\delta_{plus1} = \{((a), \sigma_{INIT}, (a+1), \sigma_{INIT}) : a \in \mathbf{N}\}.$$

The automaton $M_{\delta_{plus1}}$ can behave as follows:

input tape	*output tape*
11...	22...
1234...	2345...
1	2

Let δ_{merge} be as in the previous example. The automaton $M_{\delta_{merge}}$ can behave as follows:

first input tape	*second input tape*	*output tape*
11	22	1212
1234...	2	123245...
1234...	2	12345...
1	ϵ	1

Now that we have defined automata for nodes, we want to let them work together to get the (operational) behaviour of a net. We shall describe a run of the system of automata associated with the nodes in a net, according to the connections specified by the net. Let $t \in Net^{n:m}$ be a net and suppose that

$$t = < d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}.$$

Assume given specifications δ_i and automata M_{δ_i} for each node d_i ($i \in \{1, \ldots, k\}$). Each automaton has input and output tapes. We are going to let the automata (pairwise) share certain tapes according to the connections specified by $\{i_1 : j_1, \ldots, i_l : j_l\}$. If there is a connection $i : j$ specified between nodes d and d' (an input line of d is connected to an output line of d') we identify the input tape related to i of the d-automaton with the output tape related to j of the d' automation in the following way: We take one tape on which the d'-automaton writes and from which the d-automaton reads. Hence we need $n + l + m$ tapes for the net t, from which l tapes are shared by two automata. (As a special case the two automata may coincide). Now we have related each tape to a different edge of the data flow graph of t. We call the tapes that are related to input edges input tapes. In the same way we can speak about feedback tapes and output tapes. With all input and output tapes one head is related, and with a feedback tape a read and a write head is related. Next we describe a run of the automata. To start a run of the automata, place n words of tokens on the input tapes. All the heads are put at the beginning of their tapes. At the beginning of input tapes we have read heads, at the beginning of output tapes we have write heads and at the beginning of feedback tapes we have both read and write heads. Now we start all the automata in the initial state and let them run simultaneously each according to its own specification. Note that during such a run on a feedback tape a write head is never overtaken by a read head. After the run of the automata (which can be infinite) we can collect the contents of the output lines in a history tuple with m elements. We say that this history tuple is delivered as output. Now we give

Definition 2.13 (Operational Semantics) *Let $\mathcal{O} : Net^{n:m} \rightarrow Trace^{n:m}$ be given by*

$$\Gamma_2 \in \mathcal{O}(t)(\Gamma_1)$$

if and only if there exists a run of the system of automata associcated with the nodes in net t according to the connections specified by the net t on input Γ_1 such that Γ_2 is delivered as output.

3 Intermediate semantics $\mathcal{I}$

The operational semantics of the previous section yields a function in $Trace^{n:m}$. Brock and Ackerman showed in [Brock & Ackerman 1981] that any semantics based on such functions is not compositional. They gave an example of two nets that have the same operational semantics but that behave differently when placed in a context. We give this example in subsection 3.1. In order to obtain a compositional semantics, we have to add information to our semantic domains. The operational semantics is not fine enough: it does not make all the distinctions that are necessary.

We propose richer domains in subsection 3.2. As basic domain we use finite-word vectors: vectors with as elements finite words of tokens. A finite-word vector can contain empty words. Nets are modeled as finite-word vector functions: functions from tuples of finite-word vectors to sets of tuples of finite-word vectors. Whereas the operational semantics considers on one single line a possibly infinite word of tokens, in the intermediate semantics we cut this sequence in finite (possibly empty) pieces. We find it convenient to group a tuple of finite-word vectors in a finite-word matrix: a matrix with elements taken from the set of finite words over the alphabet of tokens. This is possible because we use finite-word vectors that are of equal height, that is of infinite height. Hence, a net is modeled by a function from finite-word matrices to sets of finite-word matrices. We will call these functions finite-word vector functions. From such functions we can derive a certain ordering of events: Restricting ourselves to 1-1 finite-word vector functions (functions from finite-word matrices of width 1 to sets of finite-word matrices of width 1) and given a finite-word matrix $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ we will get a set of possible finite-word matrices. If $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \end{pmatrix}$ is one of the possible outputs, then we have that the word x_1 contains enough tokens for the net to produce (at least) $\bar{x}_1$, the word x_1x_2 contains enough tokens for the node to produce (at least) $\bar{x}_1\bar{x}_2$, etc.

In subsection 3.3 we present the intermediate semantics which is based on the domains given in subsection 3.2. The intermediate semantics $\mathcal{I}$ is defined with firing sequences. It is called intermediate because it is a step on the way to a denotational semantics, which will be defined in section 4. The intermediate semantics can (as in the operational case) also be based on an automata theoretic intuition: We associate in subsection 3.4 with each node an automaton that works on tapes with a special structure: the contents are finite-word vectors. Then we show how to connect several of them in order to get an alternative definition for the intermediate semantics. In subsection 3.5 we give some properties of $\mathcal{I}$. Subsection 3.6 describes the relation between the operational semantics $\mathcal{O}$ and the intermediate semantics $\mathcal{I}$: we introduce an abstraction operator *abstr* and we prove that $\mathcal{O} = abstr \circ \mathcal{I}$. This *abstr*-operator "forgets" about the partioning into finite words on the lines by concatenating the finite words on the lines yielding the possible infinite histories. In sections 4 and 5 we turn to the compositionality and the full abstraction of $\mathcal{I}$.

3.1 Brock-Ackerman anomaly

A semantics based on histories which are words of tokens (like our operational semantics) is not compositional. This was first shown in ([Brock & Ackerman 1981]). They give the following example. Take two nets t_1, t_2:

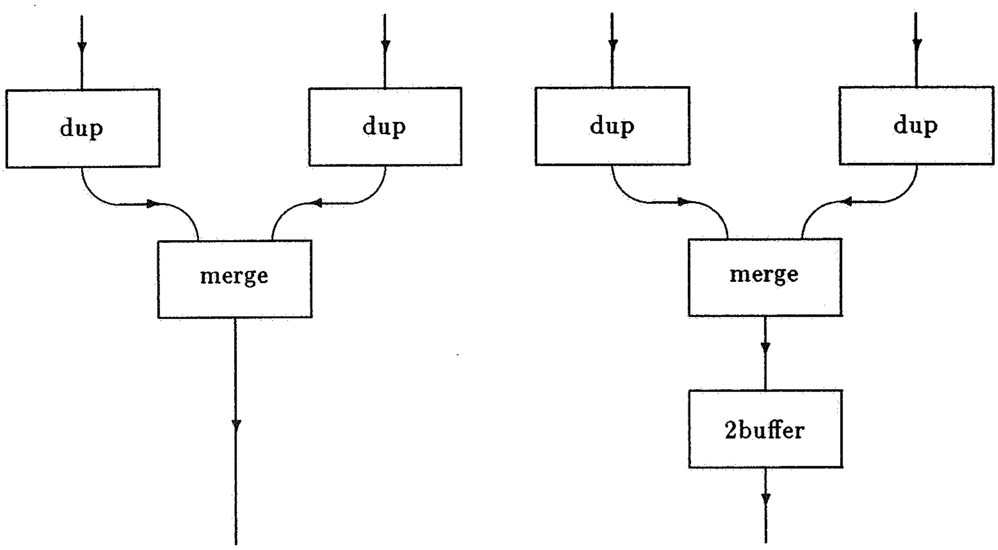

Figure 6: t_1 and t_2

$$t_1 =< dup, merge, dup > \{2:1, 3:3\},$$

$$t_2 =< dup, merge, dup, 2buffer > \{2:1, 3:3, 5:2\}.$$

The graphs of t_1, t_2 are shown in figure 6. In t_1, t_2 we have three kinds of nodes:

1. $dup \in Node^{1:1}$,
2. $merge \in Node^{2:1}$,
3. $2buffer \in Node^{1:1}$.

Brock and Ackerman take as set of tokens the set of integers. The specifications of most of these nodes were already given in section 2. We give these specifications again, together with some informal explanation. The node *dup* is a node which duplicates each token it receives and sends both to the output line. Its specification is

$$\delta_{dup} = \{((a), \sigma_{INIT}, (aa), \sigma_{INIT}) : a \text{ is a token}\}.$$

The node *2buffer* is a special kind of buffer. It waits for a second token if it has received one and then it outputs them both. The specification of *2buffer* is

$$\delta_{2buffer} = \begin{array}{l} \{(a), \sigma_{INIT}, (\epsilon), \sigma_a) : a \text{ is a token}\} \cup \\ \{(a'), \sigma_a, (aa'), \sigma_{INIT}) : a, a' \text{ are tokens}\}. \end{array}$$

The node *merge* merges its two inputs. Recall its specification:

$$\delta_{merge} = \begin{array}{l} \{((a, \epsilon), \sigma_{INIT}, (a), \sigma_{INIT}) : a \text{ is a token}\} \cup \\ \{((\epsilon, a), \sigma_{INIT}, (a), \sigma_{INIT}) : a \text{ is a token}\}. \end{array}$$

It is not difficult to see that $\mathcal{O}(t_1) = \mathcal{O}(t_2)$. The difference between the two nets is masked by the *dup* nodes.

We shall now show that the two nets can be embedded in a context such that the resulting nets have a different operational semantics. First we state informally what a context C is. (A precise definition will be given in the next section.) A context is a net with a hole in it. In this hole we can place a net by connecting some lines such that the result is again a net. The Brock-Ackerman anomaly uses the context as is shown in figure 7. When we place t_1, t_2

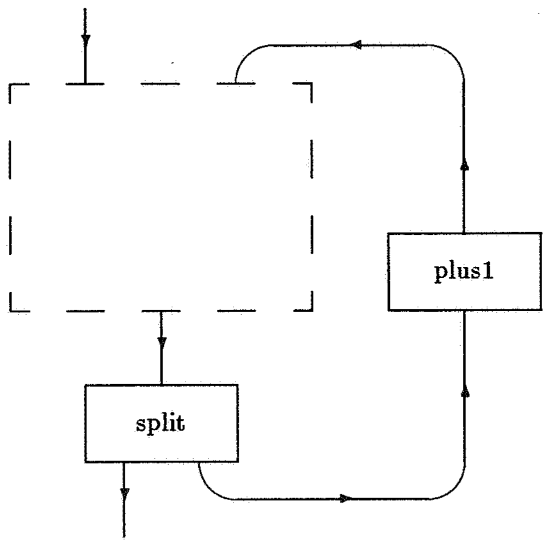

Figure 7: context C

in this context we get

$$C[t_1] =< dup, merge, dup, split\ , plus\ 1 > \{2:1, 3:3, 5:2, 6:5, 4:6\}$$

and

$$C[t_2] =< dup, merge, dup, 2buffer, split\ , plus\ 1 > \{2:1, 3:3, 4:7, 5:2, 6:4, 7:6\}.$$

The specifications of the *split* and the *plus*1 node are

$$\delta_{split} = \{((a), \sigma_{INIT}, (a, a), \sigma_{INIT}) : \text{ a is a token}\}$$

$$\delta_{plus1} = \{((a), \sigma_{INIT}, (a+1), \sigma_{INIT}) : \text{ a is a token}\}$$

The operational semantics of $C[t_1]$ and $C[t_2]$ are different:

$$\mathcal{O}(C[t_1]) \neq \mathcal{O}(C[t_2]).$$

Take for example as input 1. We have

$$\mathcal{O}(C[t_1])((1)) =$$

$$\{(y) : y \in \{$$

$$1\underbrace{22}_{2^1}\underbrace{3333}_{2^2}\ldots\underbrace{n-1\ldots n-1}_{2^{n-2}}\underbrace{n\ldots n}_{k}\ 1\ \underbrace{n\ldots n}_{2^{n-1}-k}\underbrace{n+1\ldots n+1}_{2k}\ 22\ \underbrace{n+1\ldots n+1}_{2^n-2k}\ldots :$$

$n \geq 1 \wedge k \geq 0\}$

$\}$

$$\cup\{(1\underbrace{22}_{2^1}\underbrace{3\ldots3}_{2^2}\underbrace{4\ldots4}_{2^3}\ldots\underbrace{n\ldots n}_{2^{n-1}}\ldots)\}$$

and

$$\mathcal{O}(C[t_2])((1)) = \{(1\underbrace{22}_{2^1}\underbrace{3\ldots3}_{2^2}\underbrace{4\ldots4}_{2^3}\ldots\underbrace{n\ldots n}_{2^{n-1}}\ldots)\}.$$

When we input a token, it becomes duplicated delivering a first and a second copy of it. In net t_1 we have that the second copy can spend some time between the *dup* node and the *merge* node, while the first copy can go around the feed back loop and pass the *merge* node before the token that is waiting between the *dup* and the *merge* node. In t_2 this is impossible: one may view this as the *2buffer* pulling the second token down.

It should be clear that such an example can be given in any reasonable semantics that is based on history functions. Therefore, we have to search for a semantics that gives more details (is finer, makes more distinctions) than our operational semantics $\mathcal{O}$.

3.2 Basic domains and operators

We propose a new domain. We wish to add timing information to our history tuples. Moreover, it should be the minimal amount of information. (This condition will be discussed in section 5).

Instead of using histories we use finite-word vectors. A finite-word vector is a vector of infinite height with elements taken from A^*. The counterpart of history tuples are finite-word vector tuples. Because finite-word vectors have infinite height, we can group a finite-word vector tuple in a so called finite-word matrix in the following way: Let

$$(\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \end{pmatrix}, \ldots, \begin{pmatrix} x_{1n} \\ x_{2n} \\ \vdots \end{pmatrix})$$

be a finite-word vector tuple (with n elements) then we group it in the following finite-word matrix:

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix}$$

Definition 3.1 *Define Dom^n to be the set of finite-word matrices of width n and infinite height. Let θ be a typical element of $\bigcup_{n\geq 0} Dom^n$.*

We shall sometimes use the *transpose* operator $\cdot^T$: a finite-word vector $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ is then written as $(\ x_1\ \ x_2\ \ \ldots\ )^T$.

Definition 3.2 *Define the domain* $Dom^{n:m}$ *by*

$$Dom^{n:m} = Dom^n \to \mathcal{P}(Dom^m)$$

Elements of $Dom^{n:m}$ *are denoted by* ϕ *and are called finite-word vector functions.*

We define an abstraction operator *flatten* which concatenates the (finite) words in the columns of the matrix and yields a history tuple in $Trace^n$:

Definition 3.3 $\quad flatten : Dom^n \to Trace^n$

$$flatten(\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ x_{31} & \cdots & x_{3n} \\ \vdots & & \vdots \end{pmatrix}) = (x_{11}x_{21}x_{31}\cdots, \ldots, x_{1n}x_{2n}x_{3n}\cdots).$$

Next we generalize $flatten : Dom^n \to Trace^n$ in

Definition 3.4 $\quad abstr : Dom^{n:m} \to Trace^{n:m}$

$$abstr(\phi) = \lambda\Gamma.\{flatten(\bar{\theta}) : \exists\theta[flatten(\theta) = \Gamma \wedge \bar{\theta} \in \phi(\theta)]\}$$

The operator *abstr* will relate the operational semantics $\mathcal{O}$ and the intermediate semantics $\mathcal{I}$ which will be introduced in the next subsection. We shall prove that $\mathcal{O} = abstr \circ \mathcal{I}$ in subsection 3.6. We need the notions of projection and slicing from Dom^n. (Recall that $Tup_{k,n} = \{< i_1, \ldots, i_k >: \forall j \in \{1, \ldots, k\}[1 \le i_j \le n \wedge \forall l \in \{1, \ldots, k\}[l \ne j \Rightarrow i_j \ne i_l]]\}$.)

Definition 3.5 *Define for any pair of integers* k, n *with* $0 \le k \le n$*:*

1. *(projection from* Dom^n*)*

   $\downarrow: Dom^n \times Tup_{k,n} \to Dom^k$

   $$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} \downarrow < i_1, \ldots, i_k > = \begin{pmatrix} x_{1i_1} & \cdots & x_{1i_k} \\ x_{2i_1} & \cdots & x_{2i_k} \\ \vdots & & \vdots \end{pmatrix}.$$

2. *(slicing from* Dom^n*)*

   $\uparrow: Dom^n \times Tup_{k,n} \to Dom^{n-k}$

   $\theta \uparrow \{i_1, \ldots, i_k\}$

   is the matrix obtained by removing the $i_1, \ldots, i_k$ *columns of* θ.

We define a function which combines several finite-word vector functions into one finite-word vector function:

Definition 3.6 *Take any* $\phi_i \in Dom^{n_i:m_i}$, $i = \{1, \ldots, k\}$. *Let* $n = \sum_{i=1}^{k} n_i$ *and* $m = \sum_{i=1}^{k} m_i$. *Put* $\alpha_0 = \beta_0 = 0$, $\alpha_i = \sum_{j=1}^{i} n_j$ *and* $\beta_i = \sum_{j=1}^{i} m_j$. *We define* $\phi_1 :: \cdots :: \phi_k$ *to be a function in* $Dom^{n:m}$ *such that*

$$(\phi_1 :: \cdots :: \phi_k)(\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix}) = \{\begin{pmatrix} \bar{x}_{11} & \cdots & \bar{x}_{1m} \\ \bar{x}_{21} & \cdots & \bar{x}_{2m} \\ \vdots & & \vdots \end{pmatrix}:$$

$$\forall i \in \{1,\ldots,k\}[\begin{pmatrix} \bar{x}_{1.\beta_{i-1}+1} & \cdots & \bar{x}_{1\beta_i} \\ \bar{x}_{2.\beta_{i-1}+1} & \cdots & \bar{x}_{2\beta_i} \\ \vdots & & \vdots \end{pmatrix} \in \phi_i(\begin{pmatrix} x_{1.\alpha_{i-1}+1} & \cdots & x_{1\alpha_i} \\ x_{2.\alpha_{i-1}+1} & \cdots & x_{2\alpha_i} \\ \vdots & & \vdots \end{pmatrix})]\}.$$

The next definition gives the possibility to concatenate the first k elements of an element of Dom^n (and to throw away the remaining elements):

Definition 3.7 *(concatenate first k elements) Let $k \geq 0$.*

$$[k] : Dom^n \to FTrace^n$$

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} [k] = (x_{11}x_{21}\cdots x_{k1}, \ldots, x_{1n}x_{2n}\cdots x_{kn}).$$

We also need an operator $\leadsto (\alpha_i)_i$ to be read as an integer speed up (according to $(\alpha_i)_i$):

Definition 3.8 *Let $(\alpha_i)_i$ be an infinite sequence of integers.*

$$\leadsto (\alpha_i)_i : Dom^n \to Dom^n$$

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} \leadsto (\alpha_i)_i =$$

$$\begin{pmatrix} x_{11}\cdots x_{\alpha_1.1} & \cdots & x_{1n}\cdots x_{\alpha_1.n} \\ x_{\alpha_1+1.1}\cdots x_{\alpha_1+\alpha_2.1} & \cdots & x_{\alpha_1+1.n}\cdots x_{\alpha_1+\alpha_2.n} \\ x_{\alpha_1+\alpha_2+1.1}\cdots x_{\alpha_1+\alpha_2+\alpha_3.1} & \cdots & x_{\alpha_1+\alpha_2+1.n}\cdots x_{\alpha_1+\alpha_2+\alpha_3.n} \\ \vdots & & \vdots \end{pmatrix}$$

Example

$$\begin{pmatrix} 45 & 22 \\ 9 & \epsilon \\ \epsilon & 1961 \\ 71 & 44 \\ 80 & \epsilon \\ 1 & 3 \\ 65 & 455 \\ 95 & 29 \\ \vdots & \vdots \end{pmatrix} \leadsto (3,2,1,2,\ldots) = \begin{pmatrix} 459 & 221961 \\ 7180 & 44 \\ 1 & 3 \\ 6595 & 45529 \\ \vdots & \vdots \end{pmatrix}$$

If there exists $(\alpha_i)_i$ such that $\theta_2 = \theta_1 \leadsto (\alpha_i)_i$ then we say that θ_2 is an integer speed up of θ_1. Now we introduce the important notion of delay:

Definition 3.9 (delay) *Let $\searrow : Dom^n \to \mathcal{P}(Dom^n)$ be given by*

$$\theta \searrow = \{\bar{\theta} : \forall k \exists n_k [\bar{\theta}[k] \leq \theta[k] \leq \bar{\theta}[n_k]]\}$$

An equivalent definition would be

$$\theta \searrow = \{\bar{\theta} : \forall k [\bar{\theta}[k] \leq \theta[k]] \wedge flatten(\theta) = flatten(\bar{\theta})\}.$$

If $\theta_1 \in \theta_2 \searrow$ then we say that θ_1 is a delay of θ_2 and that θ_2 is a speed up of θ_1. (See below for a discussion about the relation between integer speed up and speed up.) We have the following lemma which states some properties of delay:

Lemma 3.10

1. $\forall \theta_1, \theta_2[\theta_1 \in \theta_2 \searrow \Rightarrow \theta_1 \searrow \subseteq \theta_2 \searrow]$
2. $\forall (\alpha_i)_i \forall \theta_1, \theta_2[\theta_2 = \theta_1 \rightsquigarrow (\alpha_i)_i \Rightarrow \theta_1 \in \theta_2 \searrow]$
3. $\forall (\alpha_i)_i \forall \theta_1, \theta_2, \theta_2'[(\theta_1 \rightsquigarrow (\alpha_i)_i = \theta_2 \wedge \theta_2' \in \theta_2 \searrow) \Rightarrow \exists \theta_1' \in \theta_1 \searrow [\theta_1' \rightsquigarrow (\alpha_i)_i = \theta_2']]$
4. $\forall (\alpha_i)_i \forall \theta_1, \theta_2, \theta_2'[(\theta_1 \rightsquigarrow (\alpha_i)_i = \theta_2 \wedge \theta_2 \in \theta_2' \searrow) \Rightarrow \exists \theta_1' : \theta_1 \in \theta_1' \searrow [\theta_1' \rightsquigarrow (\alpha_i)_i = \theta_2']]$

Part 2. of the lemma shows that integer speed up implies speed up. In order to see that the reverse implication does not hold, we give the following example.

Example

Take

$$\theta_1 = \begin{pmatrix} 594 & 265 \\ 14 & 1 \\ \epsilon & 2 \\ 4 & \epsilon \\ \epsilon & 4 \\ \vdots & \vdots \end{pmatrix} \wedge \theta_2 = \begin{pmatrix} 59 & 2 \\ 41 & 65 \\ \epsilon & 12 \\ 4 & \epsilon \\ 4 & 4 \\ \vdots & \vdots \end{pmatrix}$$

We have that $\theta_2 \in \theta_1 \searrow$, but there does not exist a sequence $(\alpha_i)_i$ of integers such that $\theta_2 \rightsquigarrow (\alpha_i)_i = \theta_1$.

3.3 Intermediate semantics $\mathcal{I}$

In this subsection we give the intermediate semantics $\mathcal{I}$ which will assign to a net $t \in Net^{n:m}$ an element of $Dom^{n:m}$. Before we give the definition, we try to give some intuition. Suppose $t =< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}$ and that $t \in Net^{n:m}$. Take any $\theta \in Dom^n$. We are going to determine the set of all possible outputs $\bar{\theta}$ in $\mathcal{I}(t)(\theta)$: 'θ is on the input lines of t'. We are looking for pairs $(\theta_1, \theta_2) \in Dom^{n+l} \times Dom^{m+l}$ (which can be seen as pairs of input and output for the tuple of nodes $< d_1, \ldots, d_k >$ without connections) such that

1. it is an input/output pair according to a firing sequence for $< d_1, \ldots, d_k >$,
2. if $i : j$ is an element of $\{i_1 : j_1, \ldots, i_l : j_l\}$ then the contents of the i-th input line ($\theta_1 \downarrow < i >$) should be consistent (in a sense to be made more precise) with the contents of the j-th output line ($\theta_2 \downarrow < j >$),
3. the matrix θ_1 without the lines that are to be connected according to $\{i_1 : j_1, \ldots, i_l : j_l\}$ (i.e. $\theta_1 \uparrow \{i_1, \ldots, i_l\}$) equals θ.

If we have found such a pair (θ_1, θ_2) we take as output θ_2 without the connected lines ($\theta_2 \uparrow \{j_1, \ldots, j_l\}$).

We work out the first point: Suppose the firing sequence for the tuple of nodes is $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$. In order to perform the i-th firing $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)$ we should have on the input lines of $< d_1, \ldots, d_k >$ at least $\chi_1 \cdots \chi_i$ or, equivalently, if we can not perform the i-th firing yet (because there is not enough input) the output is at most the output of the first $i-1$ firings. In the definition of $\mathcal{I}$ we use yet another formulation (indicated with a (*) in the definition of $\mathcal{I}$ below): for every firing we can find a point j such that at this point there is enough input to fire the i-th firing and at one point before $(j-1)$ we do not notice any output of the i-th firing. This formulation is equivalent to the first two above as is remarked after the definition of $\mathcal{I}$. We also have to make a fairness assumption: all tokens that are produced by the firing sequence will eventually come out of the net: $flatten(\theta_2) = \bar{\chi}_1\bar{\chi}_2\cdots$.

The second point (about the consistency on the feedback lines) can be worked out as follows: in order to perform the i-th firing there should be enough tokens on the input lines. If we consider the lines that are to be connected we know that tokens on these lines are coming from certain output lines. Hence the first $i-1$ firings should deliver enough tokens on these lines to be able to perform the i-th firing: $\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >]$.

Definition 3.11 (Intermediate semantics $\mathcal{I}$) *Let*

$$\mathcal{I} : Net^{n:m} \rightarrow Dom^{n:m}$$

be given by

$$\bar{\theta} \in \mathcal{I}(< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\})(\theta)$$

$$\Leftrightarrow$$

$$\exists \theta_1 \in Dom^{n+l}, \theta_2 \in Dom^{m+l}[$$

$$\theta_1 \uparrow \{i_1, \ldots, i_l\} = \theta \wedge \theta_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta} \wedge$$

$$\exists (\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i \text{ firing sequence for } < d_1, \ldots, d_k > \text{ w.r.t } flatten(\theta_1) \; [$$

$$(*)\; \forall i \geq 1 \exists j \geq 1[\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}] \wedge$$
$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >] \wedge$$
$$flatten(\theta_2) = \bar{\chi}_1\bar{\chi}_2\cdots$$

$$]$$

$$]$$

Remark: we can replace the condition (*) in this definition by

$$\forall i \geq 1 \forall j \geq 1[(\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_1[j-1] \not\geq \chi_1 \cdots \chi_i) \Rightarrow \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}]$$

or by

$$\forall i \geq 1 \forall j \geq 1[\theta_1[j] \not\geq \chi_1 \cdots \chi_i \Rightarrow \theta_2[j] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}]$$

to get equivalent definitions for the intermediate semantics.

Examples

1. Let $1supply$ be a node in $Node^{1:1}$. Let us use 0 as start token. Suppose the specification is

$$\delta_{1supply} = \{((0), \sigma_{INIT}, (1), \sigma), ((\epsilon), \sigma, (1), \sigma)\}.$$

We have

$$I(< 1supply >)(\begin{pmatrix} 0 \\ \epsilon \\ \epsilon \\ \vdots \end{pmatrix}) = \{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix} : \forall i[x_i \in \{1\}^*] \wedge x_1 x_2 \cdots = 1^\omega\}.$$

2. Consider the net

$$t =< split, merge > \{2 : 1, 3 : 2\}.$$

We have that

$$I(t)(\begin{pmatrix} 1 \\ 2 \\ \epsilon \\ \vdots \end{pmatrix}) = \begin{pmatrix} 1 \\ 2 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup \begin{pmatrix} \epsilon \\ 21 \\ \epsilon \\ \vdots \end{pmatrix} \searrow$$

3. We show that the intermediate semantics is fine enough to distinguish between the two nets used in the formulation of the Brock-Ackerman anomaly. Later, we will prove that I is compositional. Take

$$t_1 =< dup, merge, dup > \{2 : 1, 3 : 3\}$$

and

$$t_2 =< dup, merge, dup, 2buffer > \{2 : 1, 3 : 3, 5 : 2\}.$$

Consider for example as input $\theta = \begin{pmatrix} 1 & \epsilon \\ \epsilon & 2 \\ \epsilon & \epsilon \\ \vdots & \vdots \end{pmatrix}$:

$$I(t_1)(\begin{pmatrix} 1 & \epsilon \\ \epsilon & 2 \\ \epsilon & \epsilon \\ \vdots & \vdots \end{pmatrix}) =$$

$$\begin{pmatrix} 11 \\ 22 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup \begin{pmatrix} 1 \\ 212 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup \begin{pmatrix} 1 \\ 221 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup$$

$$\begin{pmatrix} \epsilon \\ 2211 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup \begin{pmatrix} \epsilon \\ 2112 \\ \epsilon \\ \vdots \end{pmatrix} \searrow \cup \begin{pmatrix} \epsilon \\ 2121 \\ \epsilon \\ \vdots \end{pmatrix} \searrow$$

and

$$\mathcal{I}(t_2)(\begin{pmatrix} 1 & \epsilon \\ \epsilon & 2 \\ \epsilon & \epsilon \\ \vdots & \vdots \end{pmatrix}) =$$

$$\begin{pmatrix} 11 \\ 22 \\ \epsilon \\ \vdots \end{pmatrix}\searrow \cup \begin{pmatrix} \epsilon \\ 1212 \\ \epsilon \\ \vdots \end{pmatrix}\searrow \cup \begin{pmatrix} \epsilon \\ 1221 \\ \epsilon \\ \vdots \end{pmatrix}\searrow \cup$$

$$\begin{pmatrix} \epsilon \\ 2211 \\ \epsilon \\ \vdots \end{pmatrix}\searrow \cup \begin{pmatrix} \epsilon \\ 2112 \\ \epsilon \\ \vdots \end{pmatrix}\searrow \cup \begin{pmatrix} \epsilon \\ 2121 \\ \epsilon \\ \vdots \end{pmatrix}\searrow$$

so

$$\mathcal{I}(t_1) \neq \mathcal{I}(t_2).$$

Note that

$$\mathcal{I}(t_2)(\begin{pmatrix} 1 & \epsilon \\ \epsilon & 2 \\ \epsilon & \epsilon \\ \vdots & \vdots \end{pmatrix}) \subset \mathcal{I}(t_1)(\begin{pmatrix} 1 & \epsilon \\ \epsilon & 2 \\ \epsilon & \epsilon \\ \vdots & \vdots \end{pmatrix})$$

There are two elements in the second set that are not in the first set:

$$\begin{pmatrix} 1 \\ 212 \\ \epsilon \\ \vdots \end{pmatrix} \text{ and } \begin{pmatrix} 1 \\ 122 \\ \epsilon \\ \vdots \end{pmatrix}.$$

This can be interpreted as follows. In the case that the tokens come out of the net in the order 1212 or 1221 the first 1 can come out earlier in t_1. This timing difference is made visible in the Brock-Ackerman context.

3.4 Alternative definition of the intermediate semantics

In this subsection we provide an alternative definition for the intermediate semantics based on automata associated with the nodes in a net. For reasons similar to those discussed in section 2, we do not prove the equivalence of the two definitions. We first consider the behaviour of a single automaton. For any specification $\delta \in Spec^{n:m}$ we define an automaton as follows. The set of states of the automaton consists of those states that appear in δ. The automaton M_δ has n read heads that read from n different tapes and has m write heads that write on m different tapes which may be finite or infinite. We use a different kind of tapes than in the previous section. The refined structure of the tapes induces a refined notion of the operation of the automaton. The first step in the construction of an automaton is a closer examination

11	ϵ	456	...

Figure 8: tape with contents

of the tapes. As indicated, this structure is different from the tapes used in the alternative definition of the operational semantics $\mathcal{O}$. Each tape is divided into an infinite number of cells. Each cell may contain a finite word. Hence the contents of such a tape can be described as a finite-word vector. For example, a tape with contents $(11\ \epsilon\ 33\ 456\ \ldots)^T$ can be pictured as in figure 8.

The contents of n tapes can be described with an element of Dom^n, where a column in the matrix describes the contents of a single tape.

We sketch the run of an automaton. Let there be given a matrix θ, which gives the contents of the input tapes. The automaton reads from these input tapes. It has 2 buffers: an input buffer β_1 and an output buffer β_2. The automaton starts in the initial state σ_{INIT} and reads from its input tapes into the input buffer β_1. The buffers are finite-word tuples, the input buffer has n elements and the output buffer has m elements. Each element of the input buffer is associated with an input tape and each element of the output buffer is associated with an output tape. The last token of an element of the input buffer is the one most recently read. The operation of the automaton consists normally of performing independently read steps, fire steps and write steps. When the automaton performs a fire step, it picks nondeterministically an element $(\chi, \sigma, \bar{\chi}, \bar{\sigma})$ from δ such that

1. the automaton is in state σ
2. $\chi \leq \beta_1$ (ordering as in definition 2.5)

and

1. removes χ from its input buffer
2. puts $\bar{\chi}$ in its output buffer
3. enters state $\bar{\sigma}$.

There are independent parts of the automaton that perform read and write steps. The part that performs write steps takes tokens from the output buffer and writes them on the output tapes: it picks nondeterministically $(x'_1, \ldots, x'_m)$ such that $(x'_1, \ldots, x'_m) \leq \beta_2$ and writes $x'_1, \ldots, x'_m$ on the corresponding output tapes. The part that performs read steps takes tokens from the input tapes and puts them into the input buffer.

There is a restriction in the places where an automaton is allowed to write, which we now define: In order to formulate this restriction we introduce the notion of the position of a (read or write) head on a tape. Such a position is a number: a head is in position i if it is reading or writing in the i th cell of that tape. We formulate the restriction:

> The position of a write head should always be greater than or equal to the position of any read head of the automaton.

Note that this restriction can lead to the writing of empty words. For example, take a node $d \in Node^{1:1}$ and suppose $(x_1 x_2 \ldots)^T$ gives the contents of the input line. Assume that $(x, \sigma, \bar{x}, \bar{\sigma}) \in \delta_d$ implies that $x_1 \not\geq x$. Hence, we are not able to fire with the first word of the input. So the automaton needs to read more than the first word of input. By the restriction this implies that the write head has to write the empty word. The intuition behind the restriction is

> in order to obtain (as output) the first k words (of the finite-word vectors) it is sufficient to have as input the first k words.

There are two fairness conditions on the automaton: If an automaton is enabled, it will fire after a finite number of reading steps. The automaton is enabled if there exists a $(\chi, \sigma, \bar{\chi}, \bar{\sigma})$ in δ such that

1. the automaton is in state σ
2. $\chi \leq \beta_1$.

The second fairness condition is that if a token is put in the output buffer, it will be eventually written on an output tape.

Caution: we have introduced two notions of timing: the notion of timing associated to the finite-word vectors does not coincide with the notion of timing of the firings. Therefore in general we do not have that, if $(\; x_1 \quad x_2 \quad x_3 \quad \ldots \;)^T \in Dom^1$ is used as input for an automaton belonging to a node $d \in Node^{1:1}$ and the automaton delivers $(\; \bar{x}_1 \quad \bar{x}_2 \quad \bar{x}_3 \quad \ldots \;)^T \in Dom^1$ as output, then necessarily $(x_1, ., \bar{x}_1, .) \in \delta_d \wedge (x_2, ., \bar{x}_2, .) \in \delta_d \wedge \ldots$.

In order to get an alternative definition for the intermediate semantics $\mathcal{I}$ we combine several automata in one system. This is done in the same way as is done in the alternative definition of the operational semantics: we have that on a feedback tape a read head is never overtaken by a write head.

Definition 3.12 (Alternative definition of the intermediate semantics)
Let $\mathcal{I} : Net^{n:m} \to Dom^{n:m}$ be given by

$$\theta_2 \in \mathcal{I}(t)(\theta_1)$$

if and only if there exists a run of the system of automata associated with the nodes in net t according to the connections specified by the net t on input θ_1 such that θ_2 is delivered as output.

3.5 Properties of the intermediate semantics $\mathcal{I}$

In this subsection we give some properties of the intermediate semantics. We start with

Lemma 3.13

$$\forall n \forall d_1, \ldots, d_n [\mathcal{I}(< d_1, \ldots, d_n >) = \mathcal{I}(d_1) :: \cdots :: \mathcal{I}(d_n)]$$

The proof of this lemma is omitted. All lemmas in the rest of this section are of the following form: given that $\bar{\theta} \in \mathcal{I}(t)(\theta)$ we can obtain from $(\theta, \bar{\theta})$ other input/output pairs $(\theta', \bar{\theta}')$. The next lemma tells us that we may delay the output and speed up the input:

Lemma 3.14 (delay lemma)

$$\forall t \in Net^{n:m}\ \forall \theta, \theta' \in Dom^n \forall \bar{\theta}, \bar{\theta}' \in Dom^m\ [$$

$$\bar{\theta} \in \mathcal{I}(t)(\theta) \wedge \bar{\theta}' \in \bar{\theta} \searrow \wedge \theta \in \theta' \searrow$$

$$\Rightarrow$$

$$\bar{\theta}' \in \mathcal{I}(t)(\theta')$$

$$]$$

Proof

Take any $t \in Net^{n:m}$, $\theta \in Dom^n$, $\bar{\theta}, \bar{\theta}' \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{I}(t)(\theta)$$

and

$$\bar{\theta}' \in \bar{\theta} \searrow .$$

Suppose

$$t = < d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}.$$

By the definition of $\mathcal{I}$ we can find a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t *flatten*(θ) and $\theta_1 \in Dom^{n+l}$, $\theta_2 \in Dom^{m+l}$ such that

$$\forall i \exists j [\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}],$$

$$flatten(\theta_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots,$$

$$\forall i \geq 1 [(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >],$$

$$\theta_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta}$$

and

$$\theta_1 \uparrow \{i_1, \ldots, i_l\} = \theta.$$

Define $\theta_1' \in Dom^{n+l}$ and $\theta_2' \in Dom^{m+l}$ such that

$$\theta_1' \uparrow \{i_1, \ldots, i_l\} = \theta'$$

$$\theta_1' \downarrow < i_1, \ldots, i_l > = \theta_1 \downarrow < i_1, \ldots, i_l > .$$

$$\theta_2' \uparrow \{j_1, \ldots, j_l\} = \bar{\theta}'$$

and

$$\theta_2' \downarrow < j_1, \ldots, j_l > = \theta_2 \downarrow < j_1, \ldots, j_l > .$$

We have

$$\bar{\theta}' \in \bar{\theta} \searrow \Rightarrow \theta_2' \in \theta_2 \searrow \Rightarrow flatten(\theta_2') = flatten(\theta_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots$$

Take any integer $i \geq 1$. Define an integer j such that

$$\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1},$$

We have

$$\theta_1'[j] \geq \theta_1[j] \geq \chi_1 \cdots \chi_i$$

$$\theta_2'[j-1] \leq \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}$$

so we can conclude by the definition of $\mathcal{I}$ that $\bar{\theta}' \in \mathcal{I}(t)(\theta)$. □

Lemma 3.15 *Take any pair of integers n, m. Take $t \in Net^{n:m}$, $\theta, \theta_1 \in Dom^n$, $\bar{\theta}, \theta_2 \in Dom^m$ such that*

$$\theta_2 \in \mathcal{I}(t)(\theta_1),$$

$$\forall i \geq 1 \exists N, M \geq 1[N \geq M \wedge \theta_1[N] \leq \theta[i] \wedge \bar{\theta}[i] \leq \theta_2[M]], (*)$$

$$\mathit{flatten}(\theta) = \mathit{flatten}(\theta_1),$$

$$\mathit{flatten}(\bar{\theta}) = \mathit{flatten}(\theta_2).$$

Then we have

$$\bar{\theta} \in \mathcal{I}(t)(\theta).$$

Remark: we can replace the condition $(*)$ by one of the following equivalent conditions:

$$\forall N, M[(\theta[N] \geq \theta_1[M] \wedge \theta[N-1] \not\geq \theta_1[M]) \Rightarrow \bar{\theta}[N] \leq \theta_2[M]]$$

or

$$\forall N \exists M[\theta[N] \geq \theta_1[M] \wedge \bar{\theta}[N] \leq \theta_2[M]].$$

Proof

Take any $t \in Net^{n:m}$, $\theta, \theta_1 \in Dom^n$, $\bar{\theta}, \theta_2 \in Dom^m$ such that

$$\theta_2 \in \mathcal{I}(t)(\theta_1)$$

$$\forall i \exists N, M[N \geq M \wedge \theta_1[N] \leq \theta[i] \wedge \bar{\theta}[i] \leq \theta_2[M]]$$

$$\mathit{flatten}(\bar{\theta}) = \mathit{flatten}(\theta_2)$$

$$\mathit{flatten}(\theta) = \mathit{flatten}(\theta_1)$$

Assume $t =< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}$. By the definition of $\mathcal{I}$ we can find $\theta_4 \in Dom^{m+l}$, $\theta_3 \in Dom^{n+l}$ and a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t $\mathit{flatten}(\theta_3)$ such that

$$\forall i \exists j[\theta_3[j] \geq \chi_1 \cdots \chi_i \wedge \theta_4[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}],$$

$$\mathit{flatten}(\theta_4) = \bar{\chi}_1 \bar{\chi}_2 \cdots,$$

$$\theta_4 \uparrow \{j_1, \ldots, j_l\} = \theta_2,$$

$$\theta_3 \uparrow \{i_1, \ldots, i_l\} = \theta_1$$

and

$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >].$$

We construct $\theta_5 \in Dom^{n+l}$, $\theta_6 \in Dom^{m+l}$ as follows. Let $(\alpha_i)_i$ be a sequence of integers such that

$$\theta[i] \geq (\chi_1 \cdots \chi_{\alpha_i}) \uparrow \{i_1, \ldots, i_l\}$$

$$\theta[i] \not\geq (\chi_1 \cdots \chi_{\alpha_i+1}) \uparrow \{i_1, \ldots, i_l\}$$

From the definition of $(\alpha_i)_i$ it follows that $(\alpha_i)_i$ is monotonically nondecreasing. We consider only the case that $\lim_{i\to\infty} \alpha_i$ is infinite. (other cases can be handled by a similar argument). Define

$$\theta_5 = \begin{pmatrix} \boxed{\chi_1 \cdots \chi_{\alpha_1}} \\ \vdots \\ \boxed{\chi_{\alpha_{k-1}+1} \cdots \chi_{\alpha_k}} \\ \vdots \end{pmatrix} \wedge \theta_6 = \begin{pmatrix} \boxed{\bar{\chi}_1 \cdots \bar{\chi}_{\alpha_1}} \\ \vdots \\ \boxed{\bar{\chi}_{\alpha_{k-1}+1} \cdots \bar{\chi}_{\alpha_k}} \\ \vdots \end{pmatrix}$$

Let us explain the notation: The rows of the matrices above are the finite-word tuples in the boxes. We have that

$$(\theta_6 \uparrow \{j_1, \ldots, j_l\}) \in \mathcal{I}(t)(\theta_5 \uparrow \{i_1, \ldots, i_l\})$$

Hence by lemma 3.14

$$(\theta_6 \uparrow \{j_1, \ldots, j_l\}) \in \mathcal{I}(t)(\theta)$$

We show that $\bar{\theta} \in (\theta_6 \uparrow \{j_1, \ldots, j_l\}) \searrow$: Take any integer i. By the condition we can find N, M such that $\bar{\theta}[i] \leq \theta_2[M]$, $\theta[i] \geq \theta_1[N]$ and $N \geq M$. $\theta[i] \geq \theta_1[N]$ implies $\theta_3[N] \not\geq \chi_1 \cdots \chi_{\alpha_i+1}$. This implies that $\theta_4[N] \leq \bar{\chi}_1 \cdots \bar{\chi}_{\alpha_i}$ so $\theta_4[M] \leq \bar{\chi}_1 \cdots \bar{\chi}_{\alpha_i}$. Because $\bar{\theta}[i] \leq \theta_2[M]$ we have $\bar{\theta}[i] \leq (\theta_4 \uparrow \{j_1, \ldots, j_l\})[M] \leq (\bar{\chi}_1 \cdots \bar{\chi}_{\alpha_i}) \uparrow \{j_1, \ldots, j_l\}$ Hence $\bar{\theta} \in (\theta_6 \uparrow \{j_1, \ldots, j_l\}) \searrow$.

Now apply lemma 3.14: $\bar{\theta} \in \mathcal{I}(t)(\theta)$. □

The lemma below tells us that we can take the same integer speed up in input and output simultaneously:

Lemma 3.16

$$\forall t \in Net^{n:m}\ \forall \theta \in Dom^n \forall \bar{\theta} \in Dom^m \ [$$

$$\bar{\theta} \in \mathcal{I}(t)(\theta) \Rightarrow \forall (\alpha_i)_i [(\bar{\theta} \rightsquigarrow (\alpha_i)_i) \in \mathcal{I}(t)(\theta \rightsquigarrow (\alpha_i)_i)]$$

$$]$$

Proof

Take any $t \in Net^{n:m}$ and $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that $\bar{\theta} \in \mathcal{I}(t)(\theta)$. Assume $t =< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}$. By the definition of $\mathcal{I}$ we can find a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t *flatten*(θ) and $\theta_1 \in Dom^{n+l}$, $\theta_2 \in Dom^{m+l}$ such that

$$\forall i \exists j [\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}],$$

$$flatten(\theta_2) = \bar{\chi}_1\bar{\chi}_2\cdots,$$

$$\forall i \geq 1[(\chi_1\cdots\chi_i)\downarrow<i_1,\ldots,i_l>\leq(\bar{\chi}_1\cdots\bar{\chi}_{i-1})\downarrow<j_1,\ldots,j_l>],$$

$$\theta_2\uparrow\{j_1,\ldots,j_l\} = \bar{\theta}$$

and

$$\theta_1\uparrow\{i_1,\ldots,i_l\} = \theta.$$

Take any integer i. Take j such that

$$\theta_1[j] \geq \chi_1\cdots\chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1\cdots\bar{\chi}_{i-1},$$

Take any sequence of integers $(\alpha_i)_i$ (such that $\forall i[\alpha_i > 0]$). Define an integer γ such that

$$\alpha_1 + \cdots + \alpha_{\gamma-1} < j \leq \alpha_1 + \cdots + \alpha_\gamma$$

We have

$$(\theta_1 \leadsto (\alpha_i)_i)[\gamma] = \theta_1[\alpha_1 + \cdots + \alpha_\gamma] \geq \theta_1[j] \geq \chi_1\cdots\chi_i,$$

$$(\theta_2 \leadsto (\alpha_i)_i)[\gamma-1] = \theta_2[\alpha_1 + \cdots + \alpha_{\gamma-1}] \leq \theta_2[j-1] \leq \bar{\chi}_1\cdots\bar{\chi}_{i-1},$$

So for any i we can find a γ such that it satisfies a condition in the definition of the intermediate semantics, that is

$$(\theta_1 \leadsto (\alpha_i)_i)[\gamma] \geq \chi_1\cdots\chi_i$$

$$(\theta_2 \leadsto (\alpha_i)_i)[\gamma-1] \leq \bar{\chi}_1\cdots\bar{\chi}_{i-1}$$

It is not difficult to see that

$$(\theta_1 \leadsto (\alpha_i)_i)\uparrow\{i_1,\ldots,i_l\} = (\theta \leadsto (\alpha_i)_i)$$

and

$$(\theta_2 \leadsto (\alpha_i)_i)\uparrow\{j_1,\ldots,j_l\} = (\bar{\theta} \leadsto (\alpha_i)_i)$$

so by the definition of $\mathcal{I}$ we have $(\bar{\theta} \leadsto (\alpha_i)_i) \in \mathcal{I}(t)(\theta \leadsto (\alpha_i)_i)$. □

3.6 Relation between the intermediate semantics $\mathcal{I}$ and the operational semantics $\mathcal{O}$

The operational and intermediate semantics can be related with the operator *abstr*, which was defined in subsection 3.2. This is stated in

Theorem 3.17 $\mathcal{O} = abstr \circ \mathcal{I}$

Proof

Take any $t \in Net^{n:m}$. Suppose

$$t =< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\}.$$

We have

$$(abstr \circ \mathcal{I})(t) =$$

$$abstr(\mathcal{I}(t)) = [Definition\ \mathcal{I}]$$

$$abstr(\lambda\theta.\{\bar{\theta} :$$

$$\exists\theta_1 \in Dom^{n+l}, \theta_2 \in Dom^{m+l}[$$

$$\theta_1 \uparrow \{i_1, \ldots, i_l\} = \theta \wedge \theta_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta} \wedge$$

$$\exists(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i\ firing\ sequence\ for\ < d_1, \ldots, d_k >\ w.r.t\ flatten(\theta_1)\ [$$

$$\forall i \geq 1 \exists j \geq 1[\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}] \wedge$$
$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >] \wedge$$
$$flatten(\theta_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots$$
$$]$$
$$]\})$$

$$= [definition\ of\ abstr]$$

$$\lambda\Gamma.\{flatten(\bar{\theta}) : \exists\theta \in Dom^n[\Gamma = flatten(\theta) \wedge$$

$$\exists\theta_1 \in Dom^{n+l}, \theta_2 \in Dom^{m+l}[$$

$$\theta_1 \uparrow \{i_1, \ldots, i_l\} = \theta \wedge \theta_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta} \wedge$$

$$\exists(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i\ firing\ sequence\ for\ < d_1, \ldots, d_k >\ w.r.t\ flatten(\theta_1)\ [$$

$$\forall i \geq 1 \exists j \geq 1[\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}] \wedge \quad (*)$$
$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >] \wedge$$
$$flatten(\theta_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots$$
$$]$$
$$]]\}$$

$$=^{def} \Psi.$$

$(abstr \circ \mathcal{I} \subset \mathcal{O})$ Take any $\Gamma \in Trace^n$. Take any $\bar{\Gamma} \in Trace^m$ such that $\bar{\Gamma} \in (abstr \circ \mathcal{I})(t)(\Gamma)$. By the derivation above we have $\bar{\Gamma} \in \Psi(\Gamma)$. By definition of Ψ we can find $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$, $\theta_1 \in Dom^{m+k}$, $\theta_2 \in Dom^{n+k}$ and a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t $flatten(\theta)$ such that

$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >],$$

$$flatten(\theta_1) = \bar{\chi}_1 \bar{\chi}_2 \cdots,$$

$$\theta_1 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta},$$

$$\theta_2 \uparrow \{i_1, \ldots, i_l\} = \theta,$$

$$\mathit{flatten}(\bar{\theta}) = \bar{\Gamma}$$

and

$$\mathit{flatten}(\theta) = \Gamma.$$

Define $\Gamma_1 = \mathit{flatten}(\theta_1)$, $\Gamma_2 = \mathit{flatten}(\theta_2)$. We have

$$\Gamma_1 = \bar{\chi}_1\bar{\chi}_2\cdots,$$

$$\Gamma_1 \uparrow \{j_1, \ldots, j_l\} = \bar{\Gamma}$$

and

$$\Gamma_2 \uparrow \{i_1, \ldots, i_l\} = \Gamma$$

i.e. $\bar{\Gamma} \in \mathcal{O}(t)(\Gamma)$ by the definition of $\mathcal{O}$.

($\mathcal{O} \subset \mathit{abstr} \circ \mathcal{I}$) Take any $\Gamma \in \mathit{Trace}^n$. Take any $\bar{\Gamma} \in \mathit{Trace}^m$ such that $\bar{\Gamma} \in \mathcal{O}(t)(\Gamma)$. By the definition of $\mathcal{O}$ we can find $\Gamma_1 \in \mathit{Trace}^{n+l}$, $\Gamma_2 \in \mathit{Trace}^{m+l}$ and a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t Γ_1 such that

$$\forall i \geq 1[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >],$$

$$\Gamma_2 = \bar{\chi}_1\bar{\chi}_2\cdots,$$

$$\Gamma_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\Gamma}$$

and

$$\Gamma_1 \uparrow \{i_1, \ldots, i_l\} = \Gamma.$$

We only consider the case that $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ is an infinite sequence such that $\Gamma_1 = \chi_1\chi_2\cdots$. Define

$$\theta_1 = \begin{pmatrix} \boxed{\chi_1} \\ \boxed{\chi_2} \\ \vdots \end{pmatrix} \wedge \theta_2 = \begin{pmatrix} \boxed{\bar{\chi}_1} \\ \boxed{\bar{\chi}_2} \\ \vdots \end{pmatrix}$$

Take $\theta = \theta_1 \uparrow \{i_1, \ldots, i_l\}$. We have $\bar{\Gamma} \in \Psi(\Gamma)$, i.e. $\bar{\Gamma} \in (\mathit{abstr} \circ \mathcal{I})(t)(\Gamma)$.

□

4 Compositionality of the intermediate semantics $\mathcal{I}$ and the introduction of the denotational semantics $\mathcal{D}$

In this section we investigate the compositionality of the intermediate semantics $\mathcal{I}$. Normally, compositionality is defined with respect to some operators. On *Net* we have as yet no operators. Instead of defining operators on *Net* and then proving the compositionality, we proceed in this section in a more indirect way.

We want to have two kinds of operators on nets: tupling and connecting operators. Tupling is putting nets besides each other, without making connections. The connecting operators make more connections in a net. First we build up a new set *CNet* of nets based directly on these operators. A net $s \in CNet$ is either a basic node d or a tupling operator applied to subnets or a connection operator applied to a subnet.

We introduce a denotational semantics $\mathcal{D}$ for *CNet*. It is called denotational because each syntactic operator has a semantic counterpart. Hence this denotational semantics $\mathcal{D}$ is compositional by definition. We introduce a function *NF* on *CNet* which gives a normal form of a net. The normal form $NF(s)$ of a net $s \in CNet$ can be seen as the flattened version of the net s. We show, for any $s \in CNet$, that $\mathcal{D}(s) = \mathcal{D}(NF(s))$. A corollary of this normal form result is an associativity result for $\mathcal{D}$ which states that if two nets are equal but for the order in which connections are made, then they have the same denotational semantics. We also introduce the notion of context on *CNet*. A context is a net with one hole (or, occasionally, just a net, i.e., a net with no holes). For any context C and nets $s_1, s_2 \in CNet$ we have that if $\mathcal{D}(s_1) = \mathcal{D}(s_2)$ then also $\mathcal{D}(C(s_1)) = \mathcal{D}(C(s_2))$. This follows directly from the compositionality of the $\mathcal{D}$ with respect to the operators of tupling and connecting.

We return to the issue of compositionality of the intermediate semantics $\mathcal{I}$. We have that $Net \subset CNet$ and hence the denotational semantics $\mathcal{D}$ is also defined on *Net*. We show for any $t \in Net$ that $\mathcal{I}(t) = \mathcal{D}(t)$. Given a net $s \in CNet$ the flattened version $NF(s)$ is an element of *Net*. With the help of the denotational semantics $\mathcal{D}$ (using the compositionality of $\mathcal{D}$, its equality to $\mathcal{I}$ on *Net* and the normal form result) we are able to show the compositionality of the intermediate semantics $\mathcal{I}$: We show that for any context C (on *CNet*) and for any two nets $t_1, t_2 \in Net$ that if $\mathcal{I}(t_1) = \mathcal{I}(t_2)$ then also $\mathcal{I}(NF(C(t_1))) = \mathcal{I}(NF(C(t_2)))(*)$. The notion of context will also play an important role in section 5. The formal definition of a context will be given in subsection 4.6.

We give an overview of the rest of this section. In subsection 4.1 we introduce the new syntax for the class of data flow nets *CNet*. The set of nets *CNet* introduced by this syntax is a superset of the set of nets *Net* used in the previous sections. To this syntax we assign in 4.2 the denotational semantics $\mathcal{D}$. Subsection 4.3 provides some properties of the denotational semantics $\mathcal{D}$. In 4.4 we relate $\mathcal{I}$ and $\mathcal{D}$ by showing that

$$\forall t \in Net[\mathcal{I}(t) = \mathcal{D}(t)].$$

In subsection 4.5 we give the relation between the two kinds of syntax: each net $s \in CNet$ has a normal form $NF(s)$ which is a net in *Net*. We then show the normal form result:

$$\forall s \in CNet[\mathcal{D}(NF(s)) = \mathcal{D}(s))].$$

Using these results we derive in 4.6 the compositionality of $\mathcal{I}$, i.e., the implication (*). At the end of subsection 4.6, we comment on the relationship between the notion of compositionality involving a context and the (usual) notion involving (corresponding) syntactic and semantic operators.

4.1 Compositional syntax

If we look at the syntax of data flow nets given in section 2.2 we see that a data flow net is described as a flat structure. All nodes of the net are listed and some connections are made. In this subsection we propose a different kind of syntax. We build the nets in a compositional manner: a net is either a basic node or a number of nets put in parallel (without connections) or is a net to which some connections are added.

Definition 4.1 (Compositional Syntax) *Let the set CNet with typical element s, be defined as* $\bigcup_{n,m\geq 0} CNet^{n:m}$ *where the sets* $CNet^{n:m}$ *are given by*

$$
\begin{aligned}
s \in CNet^{n:m} ::= \; & d \in Node^{n:m} \\
& \mid < s_1, \ldots, s_k >: \; k \geq 1 \wedge \\
& \qquad \forall n_1, \ldots, n_k, m_1, \ldots, m_k[\\
& \qquad\quad \forall i \in \{1, \ldots, k\}[s_i \in CNet^{n_i:m_i}] \\
& \qquad\quad \Rightarrow \\
& \qquad\quad n = \textstyle\sum_{i=1}^{k} n_i \wedge m = \sum_{i=1}^{k} m_i \\
& \qquad] \\
& \mid s_1\{i_1 : j_1, \ldots, i_k : j_k\} : \; k \geq 1 \wedge \\
& \qquad s_1 \in CNet^{n+k:m+k} \wedge \\
& \qquad \forall 1 \leq N, M \leq k[N \neq M \Rightarrow i_N \neq i_M] \wedge \\
& \qquad \forall 1 \leq N, M \leq k[N \neq M \Rightarrow j_N \neq j_M] \wedge \\
& \qquad \forall i \in \{i_1, \ldots, i_k\}[1 \leq i \leq n + k] \wedge \\
& \qquad \forall j \in \{j_1, \ldots, j_k\}[1 \leq j \leq m + k].
\end{aligned}
$$

The restrictions in the definition of the syntax imply

- s has indeed n input and m output lines,
- it is only possible to connect existing lines
- each input line is connected to at most one output line and each output line is connected to at most one input line

Connecting lines decreases the number of input and output lines. This decrease is sometimes called hiding: the connected lines are said to be hidden.
A net defined in section 2 is also a net in the sense of the previous definition. Hence $Net^{n:m} \subset CNet^{n:m}$.

Example

We show in four figures (figures 9, 10, 11 and 12) how the net

$$<< merge, split > \{3:1\}, plus1 > \{2:3, 3:2\}.$$

is built up. This net is not an element of *Net*.

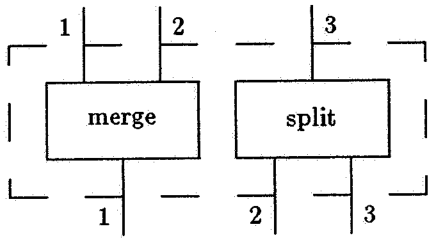

Figure 9: < *merge*, *split* >

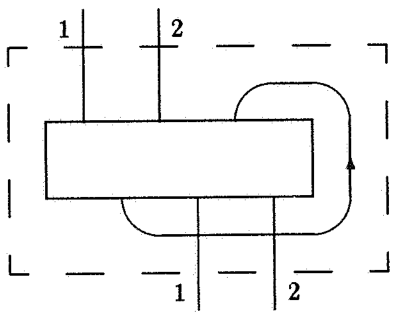

Figure 10: < *merge*, *split* > {3 : 1}

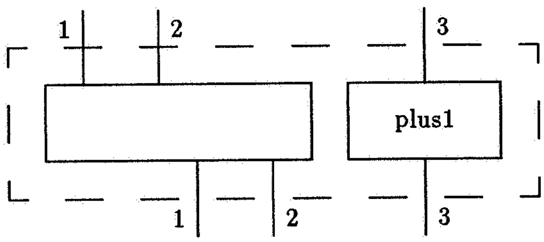

Figure 11: << *merge*, *split* > {3 : 1}, *plus1* >

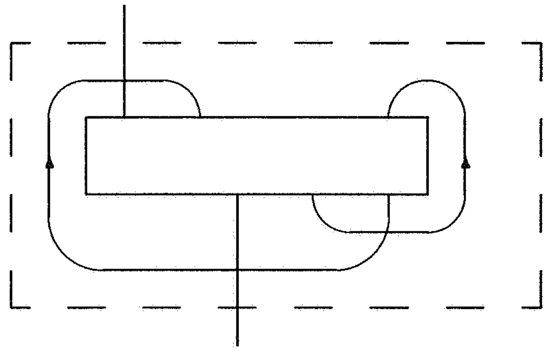

Figure 12: $<< merge, split > \{3:1\}, plus1 > \{2:3, 3:2\}$

4.2 The denotational semantics $\mathcal{D}$

In the previous subsection we have introduced a compositional syntax and in this section we assign a denotational semantics $\mathcal{D}$ to it. For the basic nodes this is easy. Inspired by the definition of the intermediate semantics we define for any node d,

$$\mathcal{D}(d) = \phi_d$$

where ϕ_d is given in the following definition:

Definition 4.2 *Let $d \in Node^{n:m}$. Define $\phi_d \in Dom^{n:m}$ by $\phi_d = \mathcal{I}(d)$.*

The case that a net consists of n subnets is also not difficult because the executions of the nets do not interact with each other. We can use the operator $:: \cdots ::$ (see definition 3.6): $\mathcal{D}(< s_1, \ldots, s_n >) = \mathcal{D}(s_1) :: \cdots :: \mathcal{D}(s_n)$.

The complicated case is the case that a net is made up of a subnet s in which some more connections are to be made. Suppose we have the meaning of this subnet (which is a function in $Dom^{n:m}$). We want to make more connections: for example, suppose we want to connect the i th input line to the j th output line (and no other connections). Because we hide all connections, the new net (with the $i : j$ connection) has 'lost' one input and one output line. We consider what happens for a fixed input $\theta \in Dom^{n-1}$. There should be a relation between the i th input and the j th output. They have to be almost the same, except for the fact that we have to guarantee that nothing is consumed before it is produced. The first idea is that the input on the i-the input line should be an (epsilon) shift of the output on the j-th output line. We first give the definition of an epsilon shift:

Definition 4.3 (ϵ-shift) *Define the operator*

$$\epsilon\square : Dom^n \rightarrow Dom^n$$

by

$$\epsilon\square \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} = \begin{pmatrix} \epsilon & \cdots & \epsilon \\ x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix}$$

We continue with the $i:j$-connection: we look for pairs $\theta_1 \in Dom^n$, $\theta_2 \in Dom^m$ such that

- $\theta_2 \in \phi(\theta_1)$ where $\phi \in Dom^{n:m}$ is the meaning of the subnet s,
- θ_1 without the i th line equals θ, i.e. $\theta_1 \uparrow \{i\} = \theta$,
- if θ_2 projected on the j th line equals $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ then θ_1 projected on the i th line equals $\begin{pmatrix} \epsilon \\ x_1 \\ x_2 \\ \vdots \end{pmatrix}$.

This turns out too restrictive as can be seen from the following example: Let $I \in Node^{1:1}$ be the identity node:

$$\delta_I = \{((a), \sigma_{INIT}, (a), \sigma_{INIT}) : a \textit{ is a token}\}$$

We do not want to make a semantic difference between the following two nets:

$$t_1 =< I >,$$

$$t_2 =< I, I > \{2:1\}.$$

If we follow the definition as outlined above we have

$$\mathcal{D}(t_1)(\theta) = \theta \searrow$$

and

$$\mathcal{D}(t_2)(\theta) = (\epsilon \Box \theta) \searrow .$$

We can use the $\rightsquigarrow (\alpha_i)_i$ operator to solve this problem. Informally, if we apply this operator as in the next definition, we do not require that the feedback lines of θ_1 are an ϵ shift of the feedback lines of θ_2. We make a weaker assumption: there exist θ_1' and θ_2' with $\theta_2' \in \phi(\theta_1')$ such that the feedback lines of θ_1' and θ_2' differ only by an ϵ shift and such that θ, $\bar{\theta}$ are integer speed ups (with the same sequence of integers) of the input lines of θ_1' and the output lines of θ_2' respectively. As we will see in remark 2 below, the application of the $\rightsquigarrow (\alpha_i)_i$ operator which concatenates words has as a consequence that it intuitively removes finite numbers of empty words. After the definition of the denotational semantics we show that the nets t_1 and t_2 are identified by the denotational semantics.

Definition 4.4 (Denotational semantics) *Let the semantics*

$$\mathcal{D} : CNet^{n:m} \rightarrow Dom^{n:m}$$

be given by

1. $\mathcal{D}(d) = \phi_d$
2. $\mathcal{D}(< s_1, \ldots, s_n >) = \mathcal{D}(s_1) :: \cdots :: \mathcal{D}(s_n)$

3. $\mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\}) =$

$$\lambda\theta.\{\bar{\theta} : \exists\theta_1 \in Dom^{n+k}, \theta_2 \in Dom^{m+k}\exists(\alpha_i)_i[$$

$$\begin{array}{l} \theta_2 \in \mathcal{D}(s)(\theta_1) \wedge \\ (\theta_1 \downarrow < i_1, \ldots, i_k >) = \epsilon\Box(\theta_2 \downarrow < j_1, \ldots, j_k >) \wedge \\ \bar{\theta} = (\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\alpha_i)_i \wedge \\ \theta = (\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\alpha_i)_i \\ \qquad] \end{array}$$

$\}$.

Remarks

1. We have that $\mathcal{D}(< s_1, \ldots, s_k >)$ is a function of $\mathcal{D}(s_1), \ldots, \mathcal{D}(s_k)$ and that $\mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})$ is a function of $\mathcal{D}(s)$ and $i_1, \ldots, i_k, j_1, \ldots, j_k$, so $\mathcal{D}$ is a compositional semantics.

2. Take (as in the previous example) $t_1 =< I >$ and $t_2 =< I, I > \{2 : 1\}$. We show that $\mathcal{D}(t_1) = \mathcal{D}(t_2)$:

$$\mathcal{D}(t_1) = \phi_I = \lambda\theta.\theta \searrow,$$

$$\mathcal{D}(t_2) =$$

$$\begin{array}{l} \lambda\theta.\{\bar{\theta} \in Dom^1 : \exists\theta_1, \theta_2 \in Dom^2, (\alpha_i)_i[\\ \quad \theta_2 \in (\phi_I :: \phi_I)(\theta_1) \wedge \\ \quad \theta_1 \downarrow < 2 >= \epsilon\Box\theta_2 \downarrow < 1 > \wedge \\ \quad \theta_1 \uparrow \{2\} \rightsquigarrow (\alpha_i)_i = \theta \wedge \theta_2 \uparrow \{1\} \rightsquigarrow (\alpha_i)_i = \bar{\theta}]\} \end{array}$$

Suppose that $\bar{\theta} \in \mathcal{D}(t_2)(\theta)$. Hence we can find $\theta_1, \theta_2 \in Dom^2$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta = (\theta_1 \uparrow \{2\}) \rightsquigarrow (\alpha_i)_i,$$

$$\bar{\theta} = (\theta_2 \uparrow \{1\}) \rightsquigarrow (\alpha_i)_i,$$

$$\theta_1 \downarrow < 2 >= \epsilon\Box(\theta_2 \downarrow < 1 >),$$

$$\theta_2 \in \theta_1 \searrow.$$

From $\theta_2 \in \theta_1 \searrow$ we derive

$$\theta_2 \downarrow < 2 > \in (\theta_1 \downarrow < 2 >) \searrow.$$

Hence, by $\theta_1 \downarrow < 2 >= \epsilon\Box(\theta_2 \downarrow < 1 >)$, we have

$$\theta_2 \downarrow < 2 > \in (\epsilon\Box(\theta_2 \downarrow < 1 >)) \searrow$$

so (because $\theta_2 \in \theta_1 \searrow$)

$$\theta_2 \downarrow < 2 > \in (\epsilon \Box (\theta_1 \downarrow < 1 >)) \searrow$$

so

$$\theta_2 \downarrow < 2 > \in (\theta_1 \downarrow < 1 >) \searrow .$$

This implies

$$(\theta_2 \downarrow < 2 >) \rightsquigarrow (\alpha_i)_i \in ((\theta_1 \downarrow < 1 >) \rightsquigarrow (\alpha_i)_i) \searrow$$

i.e.

$$(\theta_2 \uparrow \{1\}) \rightsquigarrow (\alpha_i)_i \in ((\theta_1 \uparrow \{2\}) \rightsquigarrow (\alpha_i)_i) \searrow$$

so

$$\bar{\theta} \in \theta \searrow$$

and hence $\bar{\theta} \in \mathcal{D}(t_1)(\theta)$. Now suppose that $\bar{\theta} \in \mathcal{D}(t_1)(\theta)$ and that $\theta = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ and $\bar{\theta} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \end{pmatrix}$. Take $\theta_1 = \begin{pmatrix} x_1 & \epsilon \\ \epsilon & \bar{x}_1 \\ x_2 & \epsilon \\ \epsilon & \bar{x}_2 \\ \vdots & \vdots \end{pmatrix}$ and $\theta_2 = \begin{pmatrix} \bar{x}_1 & \epsilon \\ \epsilon & \bar{x}_1 \\ \bar{x}_2 & \epsilon \\ \epsilon & \bar{x}_2 \\ \vdots & \vdots \end{pmatrix}$ and $(\alpha_i)_i = (2, 2, 2, \ldots)$.

We have

$$\theta_2 \in (\phi_I :: \phi_I)(\theta_1),$$

$$\theta_1 \downarrow < 2 > = \epsilon \Box (\theta_2 \downarrow < 1 >),$$

$$(\theta_1 \uparrow \{2\}) \rightsquigarrow (\alpha_i)_i = \theta,$$

$$(\theta_2 \uparrow \{1\}) \rightsquigarrow (\alpha_i)_i = \bar{\theta}.$$

3. We cannot omit the ϵ-shift: Take for example

$$t = < merge, split, 2buffer > \{2:4, 3:1, 4:3\}$$

If we omit the ϵ shift we have that

$$\begin{pmatrix} 111 \\ \epsilon \\ \epsilon \\ \vdots \end{pmatrix} \in \mathcal{D}(t)(\begin{pmatrix} 1 \\ \epsilon \\ \epsilon \\ \vdots \end{pmatrix})$$

This behaviour is impossible according to our operational semantics.

4.3 Properties of the denotational semantics $\mathcal{D}$

As with the intermediate semantics, we devote a section to properties of $\mathcal{D}$. Again, all lemmas point out that if we have a certain input/output pair, we can obtain from this other input/output pairs.

Before we give these properties, we first give two lemmas: The first lemma gives a property of $\epsilon\Box$ and $\rightsquigarrow (\alpha_i)_i$ (the proof of which we omit):

Lemma 4.5 $\forall\theta_1,\theta_2\forall(\alpha_i)_i[\epsilon\Box(\theta_2 \rightsquigarrow (\alpha_i)_i) = \theta_1 \rightsquigarrow (\alpha_i)_i \Rightarrow \theta_1 \in (\epsilon\Box\theta_2)\searrow]$

Lemma 4.6 *Let $\theta_1 \in Dom^n$, $\theta_2 \in Dom^m$ and let $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ be a firing sequence for a tuple of nodes w.r.t. flatten(θ_2) such that*

$$\forall i\exists j[\theta_1[j] \geq \chi_1\cdots\chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1\cdots\bar{\chi}_{i-1}].$$

Assume that for some $\{i_1,\ldots,i_k,j_1,\ldots,j_k\}$ we have

$$\epsilon\Box(\theta_2 \downarrow< j_1,\ldots,j_k >) = \theta_1 \downarrow< i_1,\ldots,i_k > .$$

We have

$$\forall i \geq 1[\chi_1\cdots\chi_i \downarrow< i_1,\ldots,i_k > \leq \bar{\chi}_1\cdots\bar{\chi}_{i-1} \downarrow< j_1,\ldots,j_k >].$$

Proof

$$(\chi_1\cdots\chi_i)\downarrow< i_1,\ldots,i_k > \leq \theta_1[j]\downarrow< i_1,\ldots,i_k > =$$

$$(\epsilon\Box\theta_2\downarrow< j_1,\ldots,j_k >)[j] = \theta_2\downarrow< j_1,\ldots,j_k >)[j-1] \leq$$

$$(\bar{\chi}_1\cdots\bar{\chi}_{i-1})\downarrow< j_1,\ldots,j_k > .$$

□

We have a delay lemma for $\mathcal{D}$:

Lemma 4.7 (delay lemma)

$$\forall s \in CNet^{n:m}\forall\theta,\theta' \in Dom^n\forall\bar{\theta},\bar{\theta}' \in Dom^m[$$

$$\bar{\theta} \in \mathcal{D}(s)(\theta) \wedge \bar{\theta}' \in \bar{\theta}\searrow \wedge \theta \in \theta'\searrow$$
$$\Rightarrow$$
$$\bar{\theta}' \in \mathcal{D}(s)(\theta')$$

$$]$$

Proof

Induction on s.

(d) Follows directly from the delay lemma 3.14 for $\mathcal{I}$.

$(< s_1,\ldots,s_k >)$ Take any $\bar{\theta},\bar{\theta}' \in Dom^m$, $\theta,\theta' \in Dom^n$ such that $\bar{\theta} \in \mathcal{D}(< s_1,\ldots,s_k >)(\theta)$, $\bar{\theta}' \in \bar{\theta}\searrow$. and $\theta \in \theta'\searrow$. Suppose $\forall i \in \{1,\ldots,k\}[< s_1,\ldots,s_i > \in CNet^{n_i:m_i}]$. Put $n_0 = m_0 = 0$. Define

$$\theta_i = \theta \downarrow< n_{i-1}+1,\ldots,n_i >,$$

$$\bar{\theta}_i = \bar{\theta} \downarrow < m_{i-1}+1, \ldots, m_i >$$

and

$$\bar{\theta}'_i = \bar{\theta}' \downarrow < m_{i-1}+1, \ldots, m_i >,$$

$$\theta'_i = \theta' \downarrow < n_{i-1}+1, \ldots, n_i > .$$

We have

$$\forall i \in \{1, \ldots, k\}[\bar{\theta}_i \in \mathcal{D}(s_i)(\theta_i) \wedge \bar{\theta}'_i \in \bar{\theta}_i \searrow \wedge \theta_i \in \theta'_i \searrow]$$

so by induction

$$\forall i \in \{1, \ldots, k\}[\bar{\theta}'_i \in \mathcal{D}(s_i)(\theta'_i)]$$

so by definition of $\mathcal{D}$:

$$\bar{\theta}' \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta').$$

$(s\{i_1 : j_1, \ldots, i_k : j_k\})$ Take any $\bar{\theta}, \bar{\theta}' \in Dom^m$, $\theta, \theta' \in Dom^n$ such that $\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$, $\theta \in \theta' \searrow$ and $\bar{\theta}' \in \bar{\theta} \searrow$. By the definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k}$ and $\theta_1 \in Dom^{n+k}$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta_2 \in \mathcal{D}(s)(\theta_1),$$

$$\theta_1 \downarrow < i_1, \ldots, i_k > = \epsilon\Box(\theta_1 \downarrow < j_1, \ldots, j_k >),$$

$$(\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\alpha_i)_i = \theta$$

and

$$(\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\alpha_i)_i = \bar{\theta}.$$

We use lemma 3.10, 3 to see that it is possible to define $\theta'_2 \in Dom^{m+k}$ such that

$$\theta'_2 \downarrow < j_1, \ldots, j_k > = \theta_2 \downarrow < j_1, \ldots, j_k >,$$

$$\theta'_2 \uparrow \{j_1, \ldots, j_k\} \in (\theta_2 \uparrow \{j_1, \ldots, j_k\}) \searrow$$

and

$$(\theta'_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\alpha_i)_i = \bar{\theta}'.$$

We use lemma 3.10, 4 to see that it is possible to define $\theta'_1 \in Dom^{n+k}$ such that

$$\theta'_1 \downarrow < i_1, \ldots, i_k > = \theta_1 \downarrow < i_1, \ldots, i_k >,$$

$$\theta_1 \uparrow \{i_1, \ldots, i_k\} \in (\theta'_1 \uparrow \{j_1, \ldots, j_k\}) \searrow$$

and

$$(\theta'_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\alpha_i)_i = \theta'.$$

By induction we have

$$\theta'_2 \in \mathcal{D}(s)(\theta'_1)$$

so by definition of $\mathcal{D}$:

$$\bar{\theta}' \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta').$$

□

The next lemma states that we can take the same integer speed up $\rightsquigarrow (\alpha_i)_i$ in input and output.

Lemma 4.8

$$\forall s \in CNet^{n:m} \forall \theta \in Dom^n \forall \bar{\theta} \in Dom^m [$$

$$\bar{\theta} \in \mathcal{D}(s)(\theta)$$
$$\Leftrightarrow$$
$$\forall (\alpha_i)_i [(\bar{\theta} \rightsquigarrow (\alpha_i)_i) \in \mathcal{D}(s)(\theta \rightsquigarrow (\alpha_i)_i)]$$
$$]$$

Proof

If we take $(\alpha_i)_i = (1)_i$, we immediately have $(\Leftarrow)$. For $(\Rightarrow)$ take any $s \in CNet^{n:m}$, $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that $\bar{\theta} \in \mathcal{D}(s)(\theta)$. Take any sequence of integers $(\alpha_i)_i$. We use induction on s.

(d) We have that $\mathcal{I}(d) = \mathcal{D}(d) = \phi_d$ by definition. Apply lemma 3.16.

$(< s_1, \ldots, s_k >)$ Suppose

$$\forall i \in \{1, \ldots, k\}[< s_1, \ldots, s_i > \in CNet^{n_i \cdot m_i}].$$

Put $n_0 = m_0 = 0$. Define

$$\forall i \in \{1, \ldots, k\}[\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >]$$

and

$$\forall i \in \{1, \ldots, k\}[\bar{\theta}_i = \bar{\theta} \downarrow < m_{i-1} + 1, \ldots, m_i >].$$

We have

$$\forall i \in \{1, \ldots, k\}[\bar{\theta}_i \in \mathcal{D}(s_i)(\theta_i)]$$

so by induction

$$\forall i \in \{1, \ldots, k\}[(\bar{\theta}_i \rightsquigarrow (\alpha_i)_i) \in \mathcal{D}(s_i)(\theta_i \rightsquigarrow (\alpha_i)_i)]$$

so by definition of $\mathcal{D}$

$$(\bar{\theta} \rightsquigarrow (\alpha_i)_i) \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta \rightsquigarrow (\alpha_i)_i).$$

$(s\{i_1 : j_1, \ldots, i_k : j_k\})$ By definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k}$, $\theta_1 \in Dom^{n+k}$ and a sequence of integers $(\beta_i)_i$ such that

$$\theta_2 \in \mathcal{D}(s)(\theta_1),$$

$$(\theta_1 \downarrow < i_1, \ldots, i_k >) = \epsilon\Box(\theta_2 \downarrow < j_1, \ldots, j_k >),$$

$$\bar{\theta} = (\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\beta_i)_i$$

and

$$\theta = (\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\beta_i)_i.$$

Define $(\gamma_i)_i$ such that

$$\gamma_1 = \beta_1 + \cdots + \beta_{\alpha_1}$$

$$\gamma_2 = \beta_{\alpha_1+1} + \cdots + \beta_{\alpha_1+\alpha_2}$$

$$\ldots$$

We have

$$\begin{aligned}(\bar{\theta} \rightsquigarrow (\alpha_i)_i) &= ((\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\beta_i)_i) \rightsquigarrow (\alpha_i)_i \\ &= (\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\gamma_i)_i\end{aligned}$$

and

$$\begin{aligned}(\theta \rightsquigarrow (\alpha_i)_i) &= ((\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\beta_i)_i) \rightsquigarrow (\alpha_i)_i \\ &= (\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\gamma_i)_i\end{aligned}$$

so by definition of $\mathcal{D}$ we have

$$(\bar{\theta} \rightsquigarrow (\alpha_i)_i) \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta \rightsquigarrow (\alpha_i)_i).$$

□

The next lemma makes use of the following definition:

Definition 4.9 *Let $\theta \in Dom^n$ and let $(\alpha_i)_i$ be a sequence of integers. Define $\theta \rhd (\alpha_i)_i$ by*

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} \rhd (\alpha_i)_i = \begin{pmatrix} \left.\begin{matrix} \epsilon & \ldots & \epsilon \\ \vdots & & \vdots \\ \epsilon & \ldots & \epsilon \end{matrix}\right\} \alpha_1 - 1 \\ \begin{matrix} x_{11} & \ldots & x_{1n} \end{matrix} \\ \left.\begin{matrix} \epsilon & \ldots & \epsilon \\ \vdots & & \vdots \\ \epsilon & \ldots & \epsilon \end{matrix}\right\} \alpha_2 - 1 \\ \begin{matrix} x_{21} & \ldots & x_{2n} \end{matrix} \\ \vdots \end{pmatrix}$$

We list some properties of $\rhd$, which are easily verified:

Lemma 4.10

1. $\theta = \theta \rhd (1, 1, 1, \ldots)$,
2. $\epsilon \Box \theta = \theta \rhd (2, 1, 1, \ldots)$,
3. $(\theta \rhd (\alpha_i)_i) \rightsquigarrow (\alpha_i)_i = \theta$,
4. $\mathit{flatten}(\theta \rhd (\alpha_i)_i) = \mathit{flatten}(\theta)$,
5. $((\epsilon \Box \theta) \rhd (\alpha_i)_i) \in (\epsilon \Box (\theta \rhd (\alpha_i)_i)) \searrow$.

Proof

We only take a look at some steps in the proof of point 3: we have that

$$\theta \rhd (\alpha_1 + \alpha_2 - 2, \alpha_3, \alpha_4, \ldots) \in (\theta \rhd (\alpha_i)_i) \searrow .$$

This implies

$$\theta \rhd (\alpha_1 + \alpha_2, \alpha_3, \alpha_4, \ldots) \in (\theta \rhd (\alpha_1 + 1, \alpha_2, \alpha_3, \ldots)) \searrow$$

and hence

$$((\epsilon \Box \theta) \rhd (\alpha_i)_i) \in (\epsilon \Box (\theta \rhd (\alpha_i)_i)) \searrow .$$

□

We use $\rhd$ to state the following lemma:

Lemma 4.11

$$\forall s \in \mathit{CNet}^{n:m} \forall \theta \in \mathit{Dom}^n \forall \bar{\theta} \in \mathit{Dom}^m [$$
$$\bar{\theta} \in \mathcal{D}(s)(\theta)$$
$$\Leftrightarrow$$
$$\forall (\alpha_i)_i [(\bar{\theta} \rhd (\alpha_i)_i) \in \mathcal{D}(s)(\theta \rhd (\alpha_i)_i)]$$
$$]$$

Proof

($\Leftarrow$) is trivial: take $(\alpha_i)_i = (1)_i$. For ($\Rightarrow$) we use induction on s:

(d) Take any $\bar{\theta} \in \mathit{Dom}^m$, $\theta \in \mathit{Dom}^n$ such that $\bar{\theta} \in \phi_d(\theta)$. Because $\bar{\theta} \in \phi_d(\theta)$ we can find a firing sequence $(\chi_i, \sigma_i, \bar{\chi}_i, \bar{\sigma}_i)_i$ for d w.r.t $\mathit{flatten}(\theta)$ and a sequence of integers $(\beta_i)_i$ such that

$$\forall i [\theta[\beta_i] \geq \chi_1 \cdots \chi_i \wedge \bar{\theta}[\beta_i - 1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}]$$

and

$$\mathit{flatten}(\bar{\theta}) = \bar{\chi}_1 \bar{\chi}_2 \cdots .$$

Define $(\xi_i)_i = (\alpha_1 + \cdots + \alpha_{\beta_i})_i$. We have

$$\forall i [(\theta \rhd (\alpha_i)_i)[\xi_i] = \theta[\beta_i] \geq \chi_1 \cdots \chi_i],$$

$$\forall i[(\bar\theta \rhd (\alpha_i)_i)[\xi_i - 1] = \bar\theta[\beta_i - 1] \le \bar\chi_1 \cdots \bar\chi_{i-1}]$$

and

$$\mathit{flatten}(\bar\theta \rhd (\alpha_i)_i) = \bar\chi_1\bar\chi_2\cdots$$

so $(\bar\theta \rhd (\alpha_i)_i) \in \phi_d(\theta \rhd (\alpha_i)_i)$.

($< s_1, \ldots, s_k >$) Take any $\bar\theta \in Dom^m$, $\theta \in Dom^n$ such that $\bar\theta \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta)$. Suppose $\forall i \in \{1, \ldots, k\}[< s_1, \ldots, s_i > \in CNet^{n_i:m_i}]$. Put $n_0 = m_0 = 0$. Define for $i \in \{1, \ldots, k\}$:

$$\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >$$

and

$$\bar\theta_i = \bar\theta \downarrow < m_{i-1} + 1, \ldots, m_i > .$$

We have

$$\forall i\{1, \ldots, k\}[\bar\theta_i \in \mathcal{D}(s_i)(\theta_i)]$$

so by induction

$$\forall i \in \{1, \ldots, k\}[(\bar\theta_i \rhd (\alpha_i)_i) \in \mathcal{D}(s_i)(\theta_i \rhd (\alpha_i)_i)]$$

so by definition of $\mathcal{D}$

$$(\bar\theta \rhd (\alpha_i)_i) \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta \rhd (\alpha_i)_i).$$

($s\{i_1 : j_1, \ldots, i_k : j_k\}$) Take any $\bar\theta \in Dom^m$, $\theta \in Dom^n$ such that $\bar\theta \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$. By the definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k}$ and $\theta_1 \in Dom^{n+k}$ and a sequence of integers $(\beta_i)_i$ such that

$$\theta_2 \in \mathcal{D}(s)(\theta_1),$$

$$\theta_1 \downarrow < i_1, \ldots, i_k > = \epsilon\Box(\theta_2 \downarrow < j_1, \ldots, j_k >),$$

$$(\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\beta_i)_i = \theta$$

and

$$(\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\beta_i)_i = \bar\theta.$$

Define

$$(\gamma_i)_i = (\alpha_1, \underbrace{1, \ldots, 1}_{\beta_1 - 1}, \alpha_2, \underbrace{1, \ldots, 1}_{\beta_2 - 1}, \ldots).$$

By induction we have

$$\theta_2 \rhd (\gamma_i)_i \in \mathcal{D}(s)(\theta_1 \rhd (\gamma_i)_i).$$

By lemma 4.10, 3 we have that

$$(\epsilon\square(\theta_2 \downarrow< j_1, \dots j_k >)) \rhd (\gamma_i)_i \in (\epsilon\square((\theta_2 \downarrow< j_1, \dots j_k >) \rhd (\gamma_i)_i)) \searrow$$

and hence

$$(\theta_1 \downarrow< i_1, \dots i_k >) \rhd (\gamma_i)_i \in (\epsilon\square((\theta_2 \downarrow< j_1, \dots j_k >)(\gamma_i)_i)) \searrow .$$

Define $\theta'_1 \in Dom^{n+k}$ such that

$$\theta'_1 \downarrow< i_1, \dots, i_k >= \epsilon\square(\theta_2 \rhd (\gamma_i)_i)_i) \downarrow< j_1, \dots, j_k >$$

and

$$\theta'_1 \uparrow \{i_1, \dots, i_k\} = (\theta_1 \uparrow \{i_1, \dots, i_k\}) \rhd (\gamma_i)_i.$$

We have

$$(\theta_1 \rhd (\gamma_i)_i) \in \theta'_1 \searrow .$$

Because

$$(\theta_2 \rhd (\gamma_i)_i)_i) \in \mathcal{D}(s)(\theta_1 \rhd (\gamma_i)_i)_i)$$

we have by lemma 4.7

$$(\theta_2 \rhd (\gamma_i)_i) \in \mathcal{D}(s)(\theta'_1).$$

Define

$$(\xi_i)_i = (\underbrace{1, \dots, 1}_{\alpha_1 - 1}, \beta_1, \underbrace{1, \dots, 1}_{\alpha_2 - 1}, \beta_2, \dots).$$

We have

$$((\theta_1 \rhd (\gamma_i)_i) \uparrow \{i_1, \dots, i_k\}) \rightsquigarrow (\xi_i)_i = \theta \rhd (\alpha_i)_i$$

and

$$((\theta_2 \rhd (\gamma_i)_i) \uparrow \{j_1, \dots, j_k\}) \rightsquigarrow (\xi_i)_i = \bar{\theta} \rhd (\alpha_i)_i.$$

Recall that

$$(\theta'_1 \rhd (\gamma_i)_i) \downarrow< i_1, \dots, i_k >= \epsilon\square((\theta_2 \rhd (\gamma_i)_i) \downarrow< j_1, \dots, j_k >)$$

so we can conclude

$$(\bar{\theta} \rhd (\alpha_i)_i) \in \mathcal{D}(s\{i_1 : j_1, \dots, i_k : j_k\})(\theta \rhd (\alpha_i)_i).$$

□

4.4 Relation between the intermediate semantics $\mathcal{I}$ and the denotational semantics $\mathcal{D}$

The semantics $\mathcal{I}$ is only defined for nets in $Net^{n:m}$. We show in this subsection that $\mathcal{I}$ and $\mathcal{D}$ coincide on these nets.

Theorem 4.12 $\quad \forall t \in Net^{n:m}[\mathcal{I}(t) = \mathcal{D}(t)]$

Proof

Take any $t \in Net^{n:m}$. Assume

$$t =< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\},$$

$$\forall i \in \{1, \ldots, k\}[< d_1, \ldots, d_i > \in Net^{n_i:m_i}]$$

and

$$n_0 = m_0 = 0.$$

Take any $\theta \in Dom^{n:m}$.

$(\mathcal{D}(t)(\theta) \subseteq \mathcal{I}(t)(\theta))$

$$\mathcal{D}(t)(\theta) =$$

$$\mathcal{D}(< d_1, \ldots, d_k > \{i_1 : j_1, \ldots, i_l : j_l\})(\theta) =$$

$$\{\bar{\theta} : \exists \theta_2 \in Dom^{m_k}, \theta_1 \in Dom^{n_k} \exists (\alpha_i)_i[$$

$$\begin{array}{l} \theta_2 \in (\phi_{d_1} :: \cdots :: \phi_{d_k})(\theta_1) \wedge \\ \theta_1 \downarrow < i_1, \ldots, i_l >= \epsilon\Box(\theta_2 \downarrow < j_1, \ldots, j_l >) \wedge \\ (\theta_1 \uparrow \{i_1, \ldots, i_l\}) \leadsto (\alpha_i)_i = \theta \wedge \\ (\theta_2 \uparrow \{j_1, \ldots, j_l\}) \leadsto (\alpha_i)_i = \bar{\theta} \\ \qquad] \end{array}$$

$$\}.$$

Take any $\bar{\theta} \in \mathcal{D}(t)(\theta)$. Choose $\theta_2 \in Dom^{m_k}$, $\theta_1 \in Dom^{n_k}$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta_2 \in (\phi_{d_1} :: \cdots :: \phi_{d_k})(\theta_1),$$

$$\theta_1 \downarrow < i_1, \ldots, i_l >= \epsilon\Box(\theta_2 \downarrow < j_1, \ldots, j_l >),$$

$$(\theta_1 \uparrow \{i_1, \ldots, i_l\}) \leadsto (\alpha_i)_i = \theta$$

and

$$(\theta_2 \uparrow \{j_1, \ldots, j_l\}) \leadsto (\alpha_i)_i = \bar{\theta}.$$

By the definitions of $\mathcal{D}$, $\mathcal{I}$ and lemma 3.13 we have that

$$\mathcal{D}(< d_1, \ldots, d_k >) = \mathcal{D}(d_1) :: \cdots :: \mathcal{D}(d_k) = \mathcal{I}(d_1) :: \cdots :: \mathcal{I}(d_k) = \mathcal{I}(< d_1, \ldots, d_k >)$$

so

$$\theta_2 \in \mathcal{I}(< d_1, \ldots, d_k >)(\theta_1)$$

By the definition of $\mathcal{I}$ we can find a firing sequence $(\chi_i, w_i, \bar{\chi}_i, \bar{w}_i)_i$ for $< d_1, \ldots, d_k >$ w.r.t. $\mathit{flatten}(\theta_1)$ such that

$$\forall i \exists j [\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}]$$

and

$$\mathit{flatten}(\theta_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots$$

Now apply lemma 4.6 to see that

$$\forall i \geq 1 [\chi_1 \cdots \chi_i \downarrow < i_1, \ldots, i_l > \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1} \downarrow < j_1, \ldots, j_l >]$$

so by the definition of $\mathcal{I}$ we have

$$\theta_2 \uparrow \{j_1, \ldots, j_l\} \in \mathcal{I}(t)(\theta_1 \uparrow \{i_1, \ldots, i_l\})$$

By lemma 3.16 we have

$$(\theta_2 \uparrow \{j_1, \ldots, j_l\}) \rightsquigarrow (\alpha_i)_i \in \mathcal{I}(t)((\theta_1 \uparrow \{i_1, \ldots, i_l\}) \rightsquigarrow (\alpha_i)_i)$$

and hence

$$\bar{\theta} \in \mathcal{I}(t)(\theta).$$

$(\mathcal{D}(t)(\theta) \supseteq \mathcal{I}(t)(\theta))$ Take any $\bar{\theta} \in \mathcal{I}(t)(\theta)$. By definition of $\mathcal{I}$ we can find $\theta_2 \in \mathit{Dom}^{m_k}$, $\theta_1 \in \mathit{Dom}^{n_k}$, a firing sequence $(\chi_j, w_j, \bar{\chi}_j, \bar{w}_j)_j$ for $< d_1, \ldots, d_k >$ w.r.t $\mathit{flatten}(\theta_1)$ such that

$$\forall i \exists j [\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}],$$

$$\forall i [(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >],$$

$$\theta_2 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta}$$

and

$$\theta_1 \uparrow \{i_1, \ldots, i_l\} = \theta.$$

Define $\theta_1' \in \mathit{Dom}^{m_k}$, $\theta_2' \in \mathit{Dom}^{n_k}$ such that

$$\theta_1' = \begin{pmatrix} \boxed{\chi_1} \\ \boxed{\chi_2} \\ \vdots \end{pmatrix} \wedge \theta_2' = \begin{pmatrix} \boxed{\bar{\chi}_1} \\ \boxed{\bar{\chi}_2} \\ \vdots \end{pmatrix}$$

We have (because $\forall i[(\chi_1 \cdots \chi_i) \downarrow < i_1, \ldots, i_l > \leq (\bar{\chi}_1 \cdots \bar{\chi}_{i-1}) \downarrow < j_1, \ldots, j_l >],)$

$$\forall i \in \{1, \ldots, k\}[\theta'_2 \downarrow < m_{i-1}+1, \ldots, m_i > \in \phi_{d_i}(\theta'_1 \downarrow < n_{i-1}+1, \ldots, n_i >)]$$

i.e.

$$\theta'_2 \in (\phi_{d_1} :: \cdots :: \phi_{d_k})(\theta'_1).$$

Define $\tilde{\theta}_1 \in Dom^{n_k}$ such that

$$\tilde{\theta}_1 \uparrow \{i_1, \ldots, i_l\} = \theta'_1 \uparrow \{i_1, \ldots, i_l\}$$

$$\tilde{\theta}_1 \downarrow < i_1, \ldots, i_l > = \epsilon\Box(\theta'_2 \downarrow < j_1, \ldots, j_l >).$$

We have

$$\theta'_1 \in \tilde{\theta}_1 \searrow$$

so by lemma 4.7 (because $\mathcal{D}(< d_1, \ldots, d_k >) = \phi_{d_1} :: \cdots :: \phi_{d_k}$)

$$\theta'_2 \in (\phi_{d_1} :: \cdots :: \phi_{d_k})(\tilde{\theta}_1)$$

i.e.

$$\theta'_2 \uparrow \{j_1, \ldots, j_l\} \in \mathcal{D}(t)(\theta'_1 \uparrow \{i_1, \ldots, i_l\})$$

We only consider the case that

$$flatten(\theta) = flatten(\theta'_1 \uparrow \{i_1, \ldots, i_l\}).$$

Define for all i, integers $\alpha_i \geq 0$ such that

$$\theta[i] \geq (\chi_1 \cdots \chi_{\alpha_1 + \cdots + \alpha_i}) \uparrow \{i_1, \ldots, i_l\}$$

$$\theta[i-1] \not\geq (\chi_1 \cdots \chi_{\alpha_1 + \cdots + \alpha_i}) \uparrow \{i_1, \ldots, i_l\}$$

Note that in general α_i can be zero. Let $(\beta_i)_i$ be the sequence of integers which is obtained by removing all zeros in $(\alpha_i)_i$. By lemma 4.8 we have

$$(\theta'_2 \uparrow \{j_1, \ldots, j_l\}) \rightsquigarrow (\beta_i)_i \in \mathcal{D}(t)(\theta'_1 \uparrow \{i_1, \ldots, i_l\} \rightsquigarrow (\beta_i)_i)$$

We have $(\theta'_1 \uparrow \{i_1, \ldots, i_l\} \rightsquigarrow (\beta_i)_i) \in \theta \searrow$ because for all i we have

$$(\theta'_1 \uparrow \{i_1, \ldots, i_l\} \rightsquigarrow (\beta_i)_i)[i] = (\chi_1 \cdots \chi_{\alpha_1 + \cdots + \alpha_i}) \uparrow \{i_1, \ldots, i_l\} \leq \theta[i]$$

Take any i. We can find a j such that

$$\theta[j] \geq (\chi_1 \cdots \chi_{\alpha_1 + \cdots + \alpha_i}) \uparrow \{i_1, \ldots, i_l\}$$

and

$\bar{\theta}[j] \geq (\bar{\chi}_1 \cdots \bar{\chi}_{\alpha_1+\cdots+\alpha_i}) \uparrow \{j_1,\ldots,j_l\}$

Because

$\theta[i] \geq (\chi_1 \cdots \chi_{\alpha_1+\cdots+\alpha_i}) \uparrow \{i_1,\ldots,i_l\}$

and

$\theta[i-1] \not\geq (\chi_1 \cdots \chi_{\alpha_1+\cdots+\alpha_i}) \uparrow \{i_1,\ldots,i_l\}$

we have $j \geq i$. Hence we have

$(\theta_2' \uparrow \{j_1,\ldots,j_l\}) \leadsto (\beta_i)_i[i] =$

$(\bar{\chi}_1 \cdots \bar{\chi}_{\alpha_1+\cdots+\alpha_i}) \uparrow \{j_1,\ldots,j_l\} \geq$

$\bar{\theta}[j] \geq$

$\bar{\theta}[i]$.

We can conclude that

$\bar{\theta} \in (\theta_2' \uparrow \{j_1,\ldots,j_l\}) \leadsto (\beta_i)_i \searrow$

so by the delay lemma for $\mathcal{D}$ we have $\bar{\theta} \in \mathcal{D}(t)(\theta)$.

□

4.5 Normal form of compositional nets and the order of connections

In this subsection we define a function *NF*: $CNet^{N:M} \rightarrow Net^{N:M}$. This function relates the two kinds of syntax we have introduced thusfar.

Definition 4.13 (Normal Form) *Define*

$NF : CNet^{N:M} \rightarrow Net^{N:M}$

inductively as follows:

1. $NF(d) = d$,
2. $NF(< s_1,\ldots,s_k >) =$

$$<d_{11},\ldots,d_{1n_1}, \ldots, d_{k1},\ldots,d_{kn_k}>$$

$$\{\alpha_0 + i_{11} : \beta_0 + j_{11},\ldots,\alpha_0 + i_{1m_1} : \beta_0 + j_{1m_1}, \ldots, \alpha_{k-1} + i_{k1} : \beta_{k-1} + j_{k1},\ldots,\alpha_{k-1} + i_{km_k} : \beta_{k-1} + j_{km_k}\},$$

where,

$$\forall l \in \{1,\ldots,k\}[NF(s_l) =< d_{l1},\ldots,d_{ln_l} > \{i_{l1} : j_{l1},\ldots,i_{lm_l} : j_{lm_l}\}],$$

$$\forall l \in \{1,\ldots,k\}[< d_{11},\ldots,d_{1n_1},\ldots,d_{l1},\ldots,d_{ln_l} >\in Net^{\alpha_l:\beta_l}]$$

and

$$\alpha_0 = \beta_0 = 0.$$

3. $NF(s\{i_1 : j_1,\ldots,i_k : j_k\}) =$

$$< d_1,\ldots,d_n > \{i'_1 : j'_1,\ldots,i'_m : j'_m, \alpha_{i_1} : \beta_{j_1},\ldots,\alpha_{i_k} : \beta_{j_k}\}$$

where

$$NF(s) =< d_1,\ldots,d_n > \{i'_1 : j'_1,\ldots,i'_m : j'_m\},$$

and $\alpha_1,\alpha_2,\ldots,\alpha_{N+k}$ *are choosen such that*

$$\alpha_1 < \alpha_2 < \ldots < \alpha_{N+k}$$

$$\{1,\ldots,N+k+m\}\backslash\{i'_1,\ldots,i'_m\} = \{\alpha_1,\alpha_2,\ldots,\alpha_{N+k}\}$$

and $\beta_1,\beta_2,\ldots,\beta_{M+k}$ *are choosen such that*

$$\beta_1 < \beta_2 < \ldots < \beta_{M+k}$$

$$\{1,\ldots,M+k+m\}\backslash\{j'_1,\ldots,j'_m\} = \{\beta_1,\beta_2,\ldots,\beta_{M+k}\}$$

Remarks

1. We have in case 3. of the definition above that $\{\alpha_{i_1},\ldots,\alpha_{i_k}\}$ is a subset of $\{\alpha_1,\alpha_2,\ldots,\alpha_{N+k}\}$ and that $\{\beta_{j_1},\ldots,\beta_{j_k}\}$ is a subset of $\{\beta_1,\beta_2,\ldots,\beta_{M+k}\}$.
2. Note that $Net^{n:m} = \{NF(s) : s \in CNet^{n:m}\}$.

Examples

1. For any node $d \in Node^{n:m}$ $(n,m \geq 2)$ we have

$$NF((d\{1:1\})\{1:1\}) = d\{1:1,2:2\}$$

2. $$NF(<< merge,split > \{3:1\}, plus1 > \{2:3,3:2\}) =$$

$$< merge, split, plus1 > \{2:4,3:1,4:3\},$$

Next we present an associativity result. Informally, we show that the order in which connections are made is not important: making all connections simultaneously is the same as doing it in an iterative way. For example consider a node with three inputs and three outputs. As suggested in figures 13, 14 and 15 there are three different ways to connect two lines. The order in which we connect lines should not make any difference for the semantics: we like to have that

$$\mathcal{D}((d\{3:3\})\{2:2\}) = \mathcal{D}((d\{2:2\})\{2:2\}) = \mathcal{D}(d\{2:2,3:3\}).$$

The equalities are consequences of the next theorem. Similar results are shown in [Staples & Nguyen 1985] and [de Bakker et al 1985].

Theorem 4.14 $\forall s \in CNet^{n:m}[\mathcal{D}(s) = \mathcal{D}(NF(s))]$.

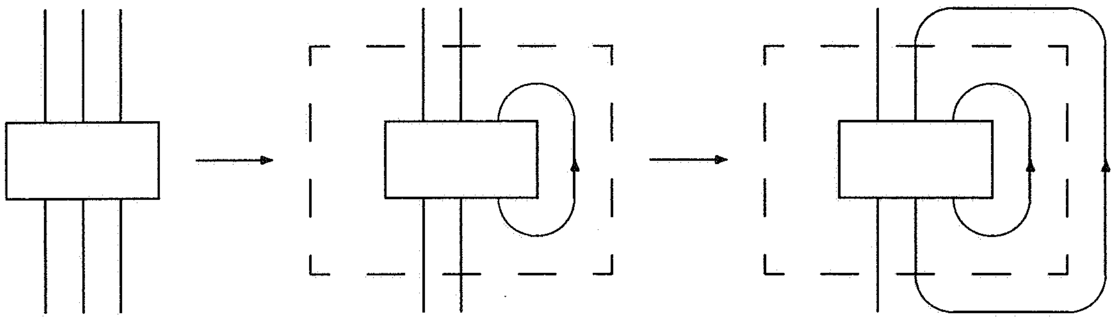

Figure 13: $(d\{3:3\})\{2:2\}$

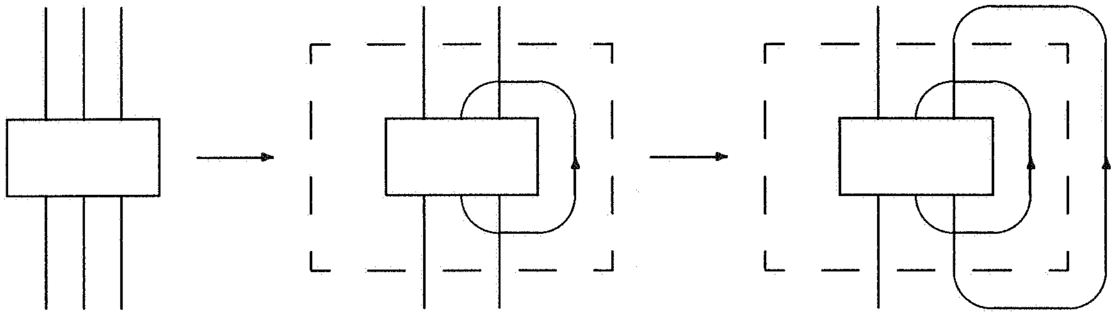

Figure 14: $(d\{2:2\})\{2:2\}$

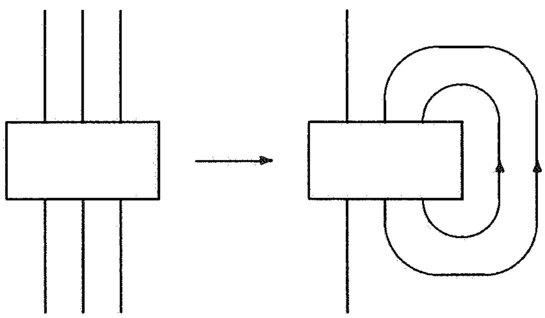

Figure 15: $d\{2:2, 3:3\}$

Proof

We use induction on the complexity of s.

(d) trivial.

$(< s_1, \ldots, s_k >)$ Assume

$$\forall l \in \{1, \ldots, k\}[NF(s_l) =< d_{l1}, \ldots, d_{ln_l} > \{i_{l1} : j_{l1}, \ldots, i_{lm_l} : j_{lm_l}\}],$$

$$\forall l \in \{1, \ldots, k\}[< s_1, \ldots, s_l >\in CNet^{\alpha_l:\beta_l}].$$

and

$$\alpha_0 = \beta_0 = 0.$$

We prove the two inclusions:

$(\mathcal{D}(< s_1, \ldots, s_k >) \subseteq \mathcal{D}(NF(< s_1, \ldots, s_k >)))$

$$\mathcal{D}(< s_1, \ldots, s_k >= [\textit{definition of } \mathcal{D}]$$

$$\mathcal{D}(s_1) :: \cdots :: \mathcal{D}(s_k) = [\textit{induction}]$$

$$\mathcal{D}(NF(s_1)) :: \cdots :: \mathcal{D}(NF(s_k)) = [\textit{definition of } ::]$$

$$\begin{aligned}&\lambda\theta.\{\bar{\theta} \ : \forall i \in \{1, \ldots, k\}[\\ &\quad \bar{\theta} \downarrow < \beta_{i-1}+1, \ldots, \beta_i > \in \mathcal{D}(NF(s_i))(\theta \downarrow < \alpha_{i-1}+1, \ldots, \alpha_i >)\\ &\quad]\\ &\} =\end{aligned}$$

$$\begin{aligned}&\lambda\theta.\{\bar{\theta} : \forall l \in \{1, \ldots, k\}\exists\theta_{l1}, \theta_{l2}\exists(\alpha_{li})_i[\\ &\quad \theta_{l2} \in (\phi_{d_{l1}} :: \cdots :: \phi_{d_{ln_l}})(\theta_{l1}) \wedge\\ &\quad \theta_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} >= \epsilon\Box\theta_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} > \wedge\\ &\quad (\theta_{l1} \uparrow \{i_{l1}, \ldots, i_{lm_l}\}) \rightsquigarrow (\alpha_{lj})_j = \theta \downarrow < \alpha_{l-1}+1, \ldots, \alpha_l > \wedge\\ &\quad (\theta_{l2} \uparrow \{j_{l1}, \ldots, j_{lm_l}\}) \rightsquigarrow (\alpha_{lj})_j = \bar{\theta} \downarrow < \beta_{l-1}+1, \ldots, \beta_l >\\ &\quad]\\ &\}.\end{aligned}$$

Take any θ, $\bar{\theta}$ such that

$$\bar{\theta} \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta).$$

By the derivation above we can find, for each $l \in \{1, \ldots, k\}$, θ_{l1}, θ_{l2} and a sequence of integers $(\alpha_{li})_i$ such that

$$\theta_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} >= \epsilon\Box\theta_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} >,$$

$$(\theta_{l1} \uparrow \{i_{l1}, \ldots, i_{lm_l}\}) \rightsquigarrow (\alpha_{lj})_j = \theta \downarrow < \alpha_{l-1}+1, \ldots, \alpha_l >$$

and

$$(\theta_{l2} \uparrow \{j_{l1}, \ldots, j_{lm_l}\}) \rightsquigarrow (\alpha_{lj})_j = \bar{\theta} \downarrow < \beta_{l-1}+1, \ldots, \beta_l > .$$

Define the sequence $(\tilde{\beta}_j)_j$ such that

$$\forall j[\tilde{\beta}_j = \max\{\alpha_{1j}, \ldots, \alpha_{kj}\}]$$

and sequences (for $l \in \{1, \ldots, k\}$) $(\xi_{lj})_j$ such that

$$\forall l \in \{1, \ldots, k\}[(\xi_{lj})_j = (\underbrace{1, \ldots, 1}_{\alpha_{l1}}, \underbrace{\tilde{\beta}_1 - \alpha_{l1}}_{if > 0}, \underbrace{1, \ldots, 1}_{\alpha_{l2}}, \underbrace{\tilde{\beta}_2 - \alpha_{l2}}_{if > 0}, \ldots)].$$

By lemma 4.11 we have

$$\forall l \in \{1, \ldots, k\}[\theta_{l2} \triangleright (\xi_{lj})_j \in (\phi_{d_{l1}} :: \cdots :: \phi_{d_{ln_l}})(\theta_{l1} \triangleright (\xi_{lj})_j)].$$

We have that for any $l \in \{1, \ldots, k\}$

$$\theta_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} > = \epsilon\Box(\theta_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} >)$$

so

$$(\theta_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} >) \triangleright (\xi_{lj})_j = (\epsilon\Box(\theta_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} >)) \triangleright (\xi_{lj})_j$$

and this implies by lemma 4.10

$$(\epsilon\Box(\theta_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} >)) \triangleright (\xi_{lj})_j \in \epsilon\Box(\theta_{l2} \triangleright (\xi_{lj})_j \downarrow < j_{l1}, \ldots, j_{lm_l} >) \searrow$$

and hence

$$(\theta_{l1} \triangleright (\xi_{lj})_j) \downarrow < i_{l1}, \ldots, i_{lm_l} > \in (\epsilon\Box(\theta_{l2} \triangleright (\xi_{lj})_j) \downarrow < j_{l1}, \ldots, j_{lm_l} >) \searrow$$

Define θ'_{l1}, θ'_{l2} such that (for $l \in \{1, \ldots, k\}$)

$$\theta'_{l2} = \theta_{l2} \triangleright (\xi_{lj})_j,$$

$$\theta'_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} > = \epsilon\Box(\theta_{l2} \triangleright (\xi_{lj})_j) \downarrow < j_{l1}, \ldots, j_{lm_l} >$$

and

$$\theta'_{l1} \uparrow \{i_{l1}, \ldots, i_{lm_l}\} = (\theta_{l1} \triangleright (\xi_{lj})_j) \uparrow \{i_{l1}, \ldots, i_{lm_l}\}.$$

We have for all $l \in \{1, \ldots, k\}$ by lemma 4.7

$$\theta'_{l2} \in \phi_{d_{ln_l}}(\theta'_{l1}).$$

Moreover, for all $l \in \{1, \ldots, k\}$ we have

$$\theta'_{l1} \downarrow < i_{l1}, \ldots, i_{lm_l} > = \epsilon\Box(\theta'_{l2} \downarrow < j_{l1}, \ldots, j_{lm_l} >),$$

$$(\theta'_{l2} \uparrow \{j_{l1}, \ldots, j_{lm_l}\}) \rightsquigarrow (\tilde{\beta}_i)_i = \bar{\theta} \downarrow < \beta_{l-1} + 1, \ldots, \beta_l >$$

and

$$(\theta'_{l1} \uparrow \{i_{l1}, \ldots, i_{lm_l}\}) \rightsquigarrow (\tilde{\beta}_i)_i = \theta \downarrow < \alpha_{l-1} + 1, \ldots, \alpha_l >$$

i.e.

$$\bar{\theta} \in \mathcal{D}(NF(< s_1, \ldots, s_k >))(\theta).$$

($\mathcal{D}(< s_1, \ldots, s_k >) \supseteq \mathcal{D}(NF(< s_1, \ldots, s_k >))$) Take any $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{D}(NF(< s_1, \ldots, s_k >))(\theta).$$

Define $(\varsigma_l)_{l=0}^k$ and $(\eta_l)_{l=0}^k$ such that

$$\varsigma_0 = \eta_0 = 0$$

and

$$\forall l \in \{1,\ldots,k\}[< d_{11},\ldots,d_{1n_1},\ldots,d_{l1},\ldots,d_{ln_l} >\in Net^{\varsigma_l:\eta_l}].$$

We have by the definition of NF

$$NF(< s_1, \ldots, s_k >) = <d_{11},\ldots,d_{1n_1}, \ldots, d_{k1},\ldots,d_{kn_k}> \{ \varsigma_0 + i_{11} : \eta_0 + j_{11},\ldots,\varsigma_0 + i_{1m_1} : \eta_0 + j_{1m_1}, \ldots, \varsigma_{k-1} + i_{k1} : \eta_{k-1} + j_{k1},\ldots,\varsigma_{k-1} + i_{km_k} : \eta_{k-1} + j_{km_k}\}.$$

Because $\bar{\theta} \in \mathcal{D}(NF(< s_1,\ldots,s_k >))(\theta)$ and by definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k}$, $\theta_1 \in Dom^{n+k}$ and a sequence of integers $(\xi_i)_i$ such that

$$\theta_2 \in (\phi_{d_{11}} :: \cdots :: \phi_{d_{1n_1}} :: \cdots :: \phi_{d_{k1}} :: \cdots :: \phi_{d_{kn_k}})(\theta_1),$$

$$\theta_1 \downarrow <\varsigma_0 + i_{11},\ldots,\varsigma_0 + i_{1m_1}, \ldots, \varsigma_{k-1} + i_{k1},\ldots,\varsigma_{k-1} + i_{km_k} >= \epsilon\Box(\theta_2 \downarrow<\eta_0 + j_{11},\ldots,\eta_0 + j_{1m_1}, \ldots, \eta_{k-1} + j_{k1},\ldots,\eta_{k-1} + j_{km_k} >),$$

$$(\theta_2 \uparrow\{\eta_0 + j_{11},\ldots,\eta_0 + j_{1m_1}, \ldots, \eta_{k-1} + j_{k1},\ldots,\eta_{k-1} + j_{km_k}\}) \rightsquigarrow (\xi_i)_i = \bar{\theta}$$

and

$$(\theta_1 \uparrow\{\varsigma_0 + i_{11},\ldots,\varsigma_0 + i_{1m_1}, \ldots, \varsigma_{k-1} + i_{k1},\ldots,\varsigma_{k-1} + i_{km_k}\}) \rightsquigarrow (\xi_i)_i = \theta.$$

Define

$$\forall l \in \{1,\ldots,k\}[\theta_{l2} = \theta_2 \downarrow< \eta_{l-1} + 1,\ldots,\eta_l >]$$

and

$$\forall l \in \{1,\ldots,k\}[\theta_{l1} = \theta_1 \downarrow< \varsigma_{l-1} + 1,\ldots,\varsigma_l >].$$

We have

$$\forall l \in \{1,\ldots,k\}[\theta_{l2} \in (\phi_{d_{l1}} :: \cdots :: \phi_{d_{ln_l}})(\theta_{l1})],$$

$$\forall l \in \{1,\ldots,k\}[\theta_{l1} \downarrow< i_{l1},\ldots,i_{lm_l} >= \epsilon\Box\theta_{l2} \downarrow< j_{l1},\ldots,j_{lm_l} >],$$

$$\forall l \in \{1,\ldots,k\}[(\theta_{l2} \uparrow \{j_{l1},\ldots,j_{lm_l}\}) \rightsquigarrow (\xi_i)_i = \bar{\theta} \downarrow< \beta_{l-1} + 1,\ldots,\beta_l >]$$

and

$$\forall l \in \{1,\ldots,k\}[(\theta_{l1} \uparrow \{i_{l1},\ldots,i_{lm_l}\}) \rightsquigarrow (\xi_i)_i = \theta \downarrow < \alpha_{l-1}+1,\ldots,\alpha_l >]$$

so

$$\forall l \in \{1,\ldots,k\}[(\bar{\theta} \downarrow < \beta_{l-1}+1,\ldots,\beta_l >) \in \mathcal{D}(NF(s_l))(\theta \downarrow < \alpha_{l-1}+1,\ldots,\alpha_l >)].$$

By induction

$$\forall l \in \{1,\ldots,k\}[(\bar{\theta} \downarrow < \beta_{l-1}+1,\ldots,\beta_l >) \in \mathcal{D}(s_l)(\theta \downarrow < \alpha_{l-1}+1,\ldots,\alpha_l >)].$$

i.e

$$\bar{\theta} \in \mathcal{D}(< s_1,\ldots,s_k >)(\theta).$$

$(s\{i_1 : j_1,\ldots,i_k : j_k\})$ We prove the two inclusions:

$(\mathcal{D}(s\{i_1 : j_1,\ldots,i_k : j_k\}) \subseteq (\mathcal{D}(NF(s\{i_1 : j_1,\ldots,i_k : j_k\})))$ Take any $\theta \in Dom^n$ and $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1,\ldots,i_k : j_k\})(\theta).$$

By definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k}$, $\theta_1 \in Dom^{n+k}$ and a sequence of integers $(\xi_i)_i$ such that

$$\theta_2 \in \mathcal{D}(s)(\theta_1),$$

$$\theta_1 \downarrow < i_1,\ldots,i_k > = \epsilon \Box \theta_2 \downarrow < j_1,\ldots,j_k >,$$

$$(\theta_1 \uparrow \{i_1,\ldots,i_k\}) \rightsquigarrow (\xi_i)_i = \theta$$

and

$$(\theta_2 \uparrow \{j_1,\ldots,j_k\}) \rightsquigarrow (\xi_i)_i = \bar{\theta}.$$

By induction we have

$$\theta_2 \in \mathcal{D}(NF(s))(\theta_1).$$

Assume

$$NF(s) = < d_1,\ldots,d_p > \{i'_1 : j'_1,\ldots,i'_q : j'_q\}.$$

By definition of $\mathcal{D}$ we can find $\theta_4 \in Dom^{m+k+q}$, $\theta_3 \in Dom^{n+k+q}$ and a sequence of integers $(\nu_i)_i$ such that

$$\theta_4 \in (\phi_{d_1} :: \cdots :: \phi_{d_p})(\theta_3),$$

$$(\theta_3 \downarrow < i'_1,\ldots,i'_q >) = \epsilon \Box (\theta_4 \downarrow < j'_1,\ldots,j'_q >),$$

$$(\theta_4 \uparrow \{j'_1,\ldots,j'_q\}) \rightsquigarrow (\nu_i)_i = \theta_2$$

and

$$(\theta_3 \uparrow \{i'_1,\ldots,i'_q\}) \rightsquigarrow (\nu_i)_i = \theta_1.$$

We derive

$$\theta_2 \downarrow < j_1,\ldots,j_k > =$$

$$((\theta_4 \uparrow \{j'_1,\ldots,j'_q\}) \rightsquigarrow (\nu_i)_i) \downarrow < j_1,\ldots,j_k > =$$

$$((\theta_4 \uparrow \{j'_1,\ldots,j'_q\}) \downarrow < j_1,\ldots,j_k >) \rightsquigarrow (\nu_i)_i =$$

$$(\theta_4 \downarrow < \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >) \rightsquigarrow (\nu_i)_i$$

where $\bar\beta_{j_1},\ldots,\bar\beta_{j_k}$ are as in definition 4.13, case 3. We derive

$$\theta_1 \downarrow < i_1,\ldots,i_k >=$$

$$((\theta_3 \uparrow \{i'_1,\ldots,i'_q\}) \rightsquigarrow (\nu_i)_i) \downarrow < i_1,\ldots,i_k >=$$

$$((\theta_3 \uparrow \{i'_1,\ldots,i'_q\}) \downarrow < i_1,\ldots,i_k >) \rightsquigarrow (\nu_i)_i =$$

$$(\theta_3 \downarrow < \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >) \rightsquigarrow (\nu_i)_i$$

where $\bar\alpha_{i_1},\ldots,\bar\alpha_{i_k}$ are as in definition 4.13, case 3. Note that

$$NF(s\{i_1 : j_1,\ldots,i_k : j_k\}) =$$

$$< d_1,\ldots,d_p > \{i'_1 : j'_1,\ldots,i'_q : j'_q, \bar\alpha_{i_1} : \bar\beta_{j_1},\ldots,\bar\alpha_{i_k} : \bar\beta_{j_k}\}.$$

We have

$$(\theta_3 \downarrow < \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >) \rightsquigarrow (\nu_i)_i$$

$$= \epsilon\Box((\theta_4 \downarrow < \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >) \rightsquigarrow (\nu_i)_i)$$

$$\Leftrightarrow$$

$$(\theta_3 \rightsquigarrow (\nu_i)_i) \downarrow < \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >$$

$$= \epsilon\Box(\theta_4 \rightsquigarrow (\nu_i)_i) \downarrow < \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >$$

$\Rightarrow$ by lemma 4.5

$$(\theta_3 \downarrow < \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >) \in (\epsilon\Box\theta_4 \downarrow < \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >) \searrow .$$

Define $\theta'_3 \in Dom^{n+k+q}$ such that

$$\theta'_3 \uparrow \{\bar\alpha_{i_1},\ldots,\bar\alpha_{i_k}\} = \theta_3 \uparrow \{\bar\alpha_{i_1},\ldots,\bar\alpha_{i_k}\} \textit{ and}$$

$$\theta'_3 \downarrow < \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >= \epsilon\Box(\theta_4 \downarrow < \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >).$$

We have by lemma 4.7

$$\theta_4 \in (\phi_{d_1} :: \cdots :: \phi_{d_p})(\theta'_3)$$

$$\theta'_3 \downarrow < i'_1,\ldots,i'_q, \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k} >= \epsilon\Box(\theta_4 \downarrow < j'_1,\ldots,j'_q, \bar\beta_{j_1},\ldots,\bar\beta_{j_k} >$$

Define $(\gamma_i)_i$ such that

$$\gamma_1 = \nu_1 + \cdots + \nu_{\xi_1}$$

$$\gamma_2 = \nu_{\xi_1+1} + \cdots + \nu_{\xi_1+\xi_2}$$

$$\ldots$$

then we have

$$(\theta'_4 \uparrow \{i'_1,\ldots,i'_q, \bar\alpha_{i_1},\ldots,\bar\alpha_{i_k}\}) \rightsquigarrow (\gamma_i)_i = \theta$$

$$(\theta_3 \uparrow \{j'_1,\ldots,j'_q, \bar\beta_{j_1},\ldots,\bar\beta_{j_k}\}) \rightsquigarrow (\gamma_i)_i = \bar\theta$$

i.e.

$$\bar{\theta} \in \mathcal{D}(< d_1, \ldots, d_p > \{i'_1 : j'_1, \ldots, i'_q : j'_q, \bar{\alpha}_{i_1} : \bar{\beta}_{j_1}, \ldots, \bar{\alpha}_{i_k} : \bar{\beta}_{j_k}\})(\theta)$$

so

$$\bar{\theta} \in \mathcal{D}(NF(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

$(\mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\}) \supseteq (\mathcal{D}(NF(s\{i_1 : j_1, \ldots, i_k : j_k\})))$ Take any $\theta \in Dom^n$ and $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

Assume

$$NF(s) =< d_1, \ldots, d_p > \{i'_1 : j'_1, \ldots, i'_q : j'_q\}$$

so

$$NF(s\{i_1 : j_1, \ldots, i_k : j_k\}) =$$

$$< d_1, \ldots, d_p > \{i'_1 : j'_1, \ldots, i'_q : j'_q, \bar{\alpha}_{i_1} : \bar{\beta}_{j_1}, \ldots, \bar{\alpha}_{i_k} : \bar{\beta}_{j_k}\}$$

where $\bar{\alpha}_{i_1}, \ldots, \bar{\alpha}_{i_k}$ and $\bar{\beta}_{j_1}, \ldots, \bar{\beta}_{j_k}$ are as in definition 4.13, case 3. By definition of $\mathcal{D}$ we can find $\theta_2 \in Dom^{m+k+q}$, $\theta_1 \in Dom^{n+k+q}$ and a sequence of integers $(\xi_i)_i$ such that

$$\theta_2 \in (\phi_{d_1} :: \cdots :: \phi_{d_p})(\theta_1),$$

$$\theta_1 \downarrow < i'_1, \ldots, i'_q, \bar{\alpha}_{i_1}, \ldots, \bar{\alpha}_{i_k} >=$$

$$\epsilon\Box(\theta_2 \downarrow < j'_1, \ldots, j'_q, \bar{\beta}_{j_1}, \ldots, \bar{\beta}_{j_k} >),$$

$$(\theta_1 \uparrow \{i'_1, \ldots, i'_q, \bar{\alpha}_{i_1}, \ldots, \bar{\alpha}_{i_k}\}) \leadsto (\xi_i)_i = \theta$$

and

$$(\theta_2 \uparrow \{j'_1, \ldots, j'_q, \bar{\beta}_{j_1}, \ldots, \bar{\beta}_{j_k}\}) \leadsto (\xi_i)_i = \bar{\theta}.$$

We have

$$\theta_1 \downarrow < i'_1, \ldots, i'_q >= \epsilon\Box(\theta_2 \downarrow < j'_1, \ldots, j'_q >)$$

and

$$(\theta_1 \uparrow \{i'_1, \ldots, i'_q\}) \downarrow < i_1, \ldots, i_k >=$$

$$\epsilon\Box((\theta_2 \uparrow \{j'_1, \ldots, j'_q\}) \downarrow < j_1, \ldots, j_k >).$$

Define

$$\theta_4 = \theta_2 \uparrow \{j'_1, \ldots, j'_q\}$$

and

$$\theta_3 = \theta_1 \uparrow \{i'_1, \ldots, i'_q\}.$$

We have

$$\theta_4 \in \mathcal{D}(< d_1, \ldots, d_p > \{i'_1 : j'_1, \ldots, i'_q : j'_q\})(\theta_3)$$

i.e.

$$\theta_4 \in \mathcal{D}(NF(s))(\theta_3)$$

so by induction

$$\theta_4 \in \mathcal{D}(s)(\theta_3).$$

By this result and from

$$\theta_3 \downarrow < i_1, \ldots, i_k > = \epsilon \Box (\theta_4 \downarrow < j_1, \ldots, j_k >),$$

$$(\theta_3 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\xi_i)_i = \theta$$

and

$$(\theta_4 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\xi_i)_i = \bar{\theta}$$

we derive

$$\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

□

4.6 Compositionality of the intermediate semantics $\mathcal{I}$

In this subsection we introduce the notion of a context. Intuitively, a context is a net with a hole in it. If we have a net $s \in CNet^{n:m}$, we can place it in a context $C \in Context^{u:v}_{n:m}$ and we get a net $C(s) \in CNet^{u:v}$. Before we introduce the sets $Context^{u:v}_{n:m}$ we first give the definition of the set $ContExp^{u:v}_{n:m}$ of context expressions. Assume for each n, m a set of variables $Var^{n:m}$ and let X be a typical element of this set.

Definition 4.15 *Let the sets $ContExp^{u:v}_{n:m}$ $(u, v, n, m \geq 0)$ be defined as follows:*

$$\begin{array}{lll}
h \in ContExp^{u:v}_{n:m} & ::= & d \in Node^{u:v} \\
& | & < s_1, \ldots, s_k, h, s_{k+1}, \ldots, s_l > \\
& & \\
& & k, l \geq 0 \\
& & \forall i \in \{1, \ldots, l\}[s_i \in CNet^{n_i:m_i}] \\
& & h \in ContExp^{n':m'}_{n:m} \\
& & n = n_1 + \cdots + n_k + n' + n_{k+1} + \cdots + n_l \\
& & m = m_1 + \cdots + m_k + m' + m_{k+1} + \cdots + m_l \\
& & \\
& | & X \in Var^{n:m} \\
& & \\
& | & h\{i_1 : j_1, \ldots, i_k : j_k\} \\
& & \\
& & k \geq 0 \\
& & h \in ContExp^{u+k:v+k}_{n:m} \\
& & \forall i \in \{i_1, \ldots, i_k\}[1 \leq i \leq u + k] \\
& & \forall j \in \{j_1, \ldots, j_k\}[1 \leq j \leq v + k] \\
& & \forall N, M \in \{1, \ldots, k\}[N \neq M \Rightarrow i_N \neq i_M] \\
& & \forall N, M \in \{1, \ldots, k\}[N \neq M \Rightarrow j_N \neq j_M]
\end{array}$$

Now we come to the definition of a context.

Definition 4.16 *Let the sets $Context^{u:v}_{n:m}$ $(u, v, n, m \geq 0)$ be defined as follows:*

$C \in Context^{u:v}_{n:m} ::= \lambda X.h$ *where* $h \in ContExp^{u:v}_{n:m}$

Note that if $t \in Net^{n:m}$ and $C \in Context^{u:v}_{n:m}$ then $C(t) \in CNet^{u:u}$ but not necessarily $C(t) \in Net^{u:v}$. Hence we introduce the notation $C[t]$ for $NF(C(t))$.

Now we can show the compositionality of $\mathcal{I}$ in

Theorem 4.17 (Compositionality of $\mathcal{I}$)

$$\forall t_1, t_2 \in Net^{n:m}[\mathcal{I}(t_1) = \mathcal{I}(t_2) \Rightarrow \forall C \in Context^{u:v}_{n:m}[\mathcal{I}(C[t_1]) = \mathcal{I}(C[t_2])]]$$

Proof

Take any $t_1, t_2 \in Net^{n:m}$ such that $\mathcal{I}(t_1) = \mathcal{I}(t_2)$. We have

$$\mathcal{I}(t_1) = \mathcal{I}(t_2)$$

$\Rightarrow$ [theorem 4.4]

$$\mathcal{D}(t_1) = \mathcal{D}(t_2)$$

$\Rightarrow$ [compositionality of $\mathcal{D}$]

$$\forall C \in Context^{u:v}_{n:m}[\mathcal{D}(C(t_1)) = \mathcal{D}(C(t_2))]$$

$\Rightarrow$ [theorem 4.14]

$$\forall C \in Context^{u:v}_{n:m}[\mathcal{D}(NF(C(t_1))) = \mathcal{D}(NF(C(t_2)))]$$

$\Rightarrow$ [theorem 4.4]

$$\forall C \in Context^{u:v}_{n:m}[\mathcal{I}(C[t_1]) = \mathcal{I}(C[t_2])].$$

□

We provide a remark on the relationship between the notion of compositionality involving a context and the (usual) notion involving (corresponding) syntactic and semantic operators:

Remark Assume for the duration of this remark that the n-tupling operator on *CNet* is denoted by tup_n and the connecting operator (which connects the input line i_l to the output line j_l for $l \in \{1, \ldots, k\}$) is denoted by $con_{\{i_1:j_1,\ldots,i_k:j_k\}}$. Recall that $Net \subset CNet$. Given a net $s \in CNet$ the flattened version $NF(s)$ is an element of *Net*. In order to introduce the operators for tupling and connecting on *Net* we cannot restrict the operators tup_n and $con_{\{i_1:j_1,\ldots,i_k:j_k\}}$ to *Net*: when we apply them to nets in *Net* they in general do not yield a net in *Net* (but in *CNet*). Hence we apply the operator *NF* to the result to obtain a net in *Net*: the operator $nftup_n$ (n-tupling on *Net*) is defined as $NF \circ tup_n$ and $nfcon_{\{i_1:j_1,\ldots,i_k:j_k\}}$ is defined as $NF \circ con_{\{i_1:j_1,\ldots,i_k:j_k\}}$.

With the help of the result

$$(*)\ \forall t_1, t_2 \in Net[\mathcal{I}(t_1) = \mathcal{I}(t_2) \Rightarrow \forall C \in Context[\mathcal{I}(C(t_1)) = \mathcal{I}(C(t_2))]]$$

we are able to show the compositionality of the intermediate semantics $\mathcal{I}$ with respect to $nftup_n$ and $nfcon_{\{i_1:j_1,\ldots,i_k:j_k\}}$ as follows.

Firstly we show it for $nfcon_{\{i_1:j_1,\ldots,i_k:j_k\}}$. Take any $t_1, t_2 \in Net$ such that $\mathcal{I}(t_1) = \mathcal{I}(t_2)$. Put

$$C = \lambda X.con_{\{i_1:j_1,\ldots,i_k:j_k\}}(X).$$

By (*) we have that

$$\mathcal{I}(NF(con_{\{i_1:j_1,\ldots,i_k:j_k\}}(t_1))) = \mathcal{I}(NF(con_{\{i_1:j_1,\ldots,i_k:j_k\}}(t_2)))$$

i.e. $\mathcal{I}(nfcon_{\{i_1:j_1,\ldots,i_k:j_k\}}(t))$ is a function of $\mathcal{I}(t)$.

Secondly we consider the case of the operator $nftup_n$. We take the case that $n = 2$. Other cases are treated in a similar way. Take any $t_1, t_1', t_2, t_2' \in Net$ such that $\mathcal{I}(t_1) = \mathcal{I}(t_1')$ and $\mathcal{I}(t_2) = \mathcal{I}(t_2')$. By (*) we have (put $C = \lambda X.tup_2(X, t_2)$)

$$\mathcal{I}(t_1) = \mathcal{I}(t_1') \Rightarrow \mathcal{I}(NF(tup_2(t_1, t_2))) = \mathcal{I}((NF(tup_2(t_1', t_2)))$$

and (put $C = \lambda X.tup_2(t_1', X)$)

$$\mathcal{I}(t_2) = \mathcal{I}(t_2') \Rightarrow \mathcal{I}(NF(tup_2(t_1', t_2))) = \mathcal{I}((NF(tup_2(t_1', t_2'))).$$

Combining the two equalities, we observe that $\mathcal{I}(NF(tup_2(t, t')))$ is a function of $\mathcal{I}(t)$ and $\mathcal{I}(t')$.

5 Full abstraction of the denotational semantics $\mathcal{D}$

In the previous section we gave a semantics $\mathcal{D}$ that makes enough distinctions for it to be compositional. Now a natural question is

> is $\mathcal{D}$ the most suitable refinement (that is, does it not make too many distinctions) of $\mathcal{O}$ that is compositional?

We prove that the denotational model is the minimal extension of the operational model that is compositional. For any two nets t_1 and t_2 that have a different denotational semantics, we provide a context C such that the operational semantics of $C[t_1]$ and $C[t_2]$ differ. From this result we derive the full abstraction property of $\mathcal{D}$: the equivalence relation generated by the denotational semantics is the greatest congruence contained in the equivalence relation generated by the operational semantics. In addition, we have the rather surprising fact that we can use in all cases one same context which does not depend on the nets.

We present a brief sketch of the proof outline. Suppose we have two nets that have a different denotational semantics and the same operational semantics. (If they have a different operational semantics we can take an empty context). From this we can conclude that there is a timing difference between the two nets: the output is produced in a different way. We can make this difference visible in the operational semantics by tagging the output and feeding it back as soon as possible to a merge node which merges this tagged output with the original input. The resulting history on the output line of the merge node is a mixture of tagged tokens (from the output that is fed back) and tokens that are not tagged (from the original input). With a split node we make copies of all tokens that are sent along the output line of the merge node. One of these copies is delivered as output and the other copy is sent to a node that removes the tagged tokens This node generates the original input which is sent to either t_1 or t_2. Due to the timing difference we observe (in our operational semantics) a different mixture of tagged tokens and tokens that are not tagged.

We start by giving some basic definitions:

Definition 5.1 *A relation R in a set A is called an equivalence relation if and only if it satisfies:*

1. $\forall a \in A[(a,a) \in R]$
2. $\forall a,b \in A[(a,b) \in R \Rightarrow (b,a) \in R]$
3. $\forall a,b,c \in A[(a,b) \in R \wedge (b,c) \in R \Rightarrow (a,c) \in R]$

Definition 5.2 *An equivalence relation R_1 in a set A is said to be contained in an equivalence relation R_2 in a set A if $R_1 \subseteq R_2$.*

Definition 5.3 *An equivalence relation R on Net is called a congruence if it satisfies*

$$\forall t_1, t_2 \in Net^{n:m}[R(t_1,t_2) \Rightarrow \forall C \in Context^{u:v}_{n:m}[R(C[t_1], C[t_2])]]$$

A semantic function $\mathcal{A}$ (a function with domain the set *Net*) generates an equivalence relation $R_{\mathcal{A}}$: $t_1, t_2 \in R_{\mathcal{A}}$ if and only if $\mathcal{A}(t_1) = \mathcal{A}(t_2)$. Now we can formulate what it means that a semantics is the most suitable compositional refinement of another semantics:

Definition 5.4 *A semantics $\mathcal{A}$ is called fully abstract with respect to a semantics $\mathcal{B}$ if $R_{\mathcal{A}}$ is the greatest congruence contained in $R_{\mathcal{B}}$.*

The rest of this section is devoted to the proof of

Theorem 5.5 *The denotational semantics $\mathcal{D}$ is fully abstract with respect to the operational semantics $\mathcal{O}$.*

We proceed in the following way. We first proof a lemma that gives a sufficent condition for the full abstractness. Then we show in a second lemma that this condition holds.

Lemma 5.6 *The denotational semantics $\mathcal{D}$ is fully abstract with respect to the operational semantics $\mathcal{O}$ if*

$$\forall t_1, t_2 \in Net^{n:m}[$$

$$\mathcal{D}(t_1) = \mathcal{D}(t_2)$$
$$\Leftrightarrow$$
$$\forall C \in Context^{u:v}_{n:m}[\mathcal{O}(C[t_1]) = \mathcal{O}(C[t_2])]$$
$$]$$

Proof
We show that $R_\mathcal{D}$ is the greatest congruence contained in $R_\mathcal{O}$:

1. $R_\mathcal{D}$ is a congruence by the compositionality of $\mathcal{D}$
2. $R_\mathcal{D}$ is contained in $R_\mathcal{O}$ because $\mathcal{O} = abstr \circ \mathcal{D}$ (or take the empty context)
3. Suppose there exists a congruence R such that $R_\mathcal{D} \subseteq R \subseteq R_\mathcal{O}$. For any $t_1, t_2 \in Net^{n:m}$ we have

   $$(t_1, t_2) \in R \Rightarrow$$

   (R is a congruence)

   $$\forall C \in Context^{u:v}_{n:m}[(C[t_1], C[t_2]) \in R] \Rightarrow$$

   ($R \subseteq R_\mathcal{O}$)

   $$\forall C \in Context^{u:v}_{n:m}[(C[t_1], C[t_2]) \in R_\mathcal{O}] \Rightarrow$$

   (condition in lemma)

   $$(t_1, t_2) \in R_\mathcal{D}$$

   Hence $R \subseteq R_\mathcal{D}$, i.e. $R = R_\mathcal{D}$.

□

Lemma 5.7

$$\forall t_1, t_2 \in Net^{n:m}[$$

$$\mathcal{D}(t_1) = \mathcal{D}(t_2)$$
$$\Leftrightarrow$$
$$\forall C \in Context^{u:v}_{n:m}[\mathcal{O}(C[t_1]) = \mathcal{O}(C[t_2])]$$
$$]$$

Proof

($\Rightarrow$) Take any $t_1, t_2 \in Net^{n:m}$. Assume $\mathcal{D}(t_1) = \mathcal{D}(t_2)$ Take an arbitrary context $C \in Context^{u:v}_{n:m}$. By the compositionality of $\mathcal{D}$ we derive $\mathcal{D}(C[t_1]) = \mathcal{D}(C[t_2])$. Because $\mathcal{O} = abstr \circ \mathcal{D}$ we have $\mathcal{O}(C[t_1]) = \mathcal{O}(C[t_2])$.

($\Leftarrow$) Take any $t_1, t_2 \in Net^{n:m}$ such that for all contexts $C \in Context^{u:v}_{n:m}$ we have $\mathcal{O}(C[t_1]) = \mathcal{O}(C[t_2])$. First assume that both $t_1, t_2 \in Net^{1:1}$. From the symmetry, it suffices to show

$$\forall \theta \in Dom^1[\mathcal{D}(t_1)(\theta) \subset \mathcal{D}(t_2)(\theta)].$$

Take any $\bar{\theta} \in \mathcal{D}(t_1)(\theta)$. We show $\bar{\theta} \in \mathcal{D}(t_2)(\theta)$. Let $C_1, C_2 \in Context^{1:1}_{1:1}$ be the following contexts:

$$C_1 = \lambda X. < tagger, amerge, split, remove, X > \{1:6, 3:1, 4:2, 5:4, 6:5\}$$

$$C_2 = \lambda X. < tagger, imerge, split, remove, X > \{1:6, 3:1, 4:2, 5:4, 6:5\}$$

See figure 16 and 17 for pictures of these contexts. As will become clear, C_1 will be used if the input is finite and C_2 will be used if the input is infinite.

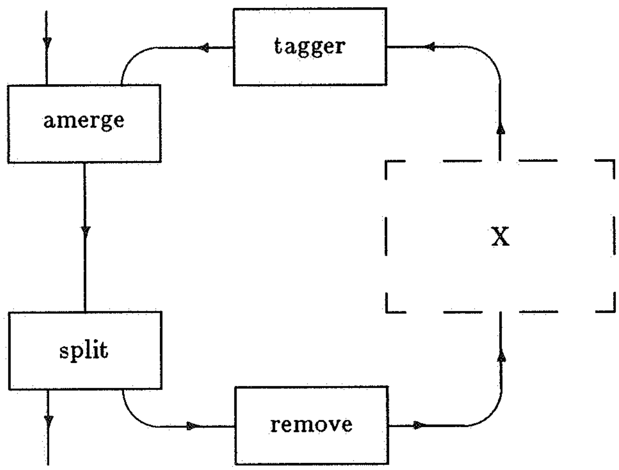

Figure 16: the context C_1

We give a short description and the specification of the nodes in this context.

tagger A node which tags all the tokens that pass this node. We assume that we can observe that a token is tagged and that for each net t we can find a tagger that does not appear in the net t. The tagged version of a token a is denoted by a^t. The specification of the node is:

$$\delta_{tagger} = \{((a), \sigma_{INIT}, (a^t), \sigma_{INIT}) : a \text{ is a token } \}$$

remove a node which removes tagged tokens. All tokens that are not tagged pass this node unchanged. Its specification is

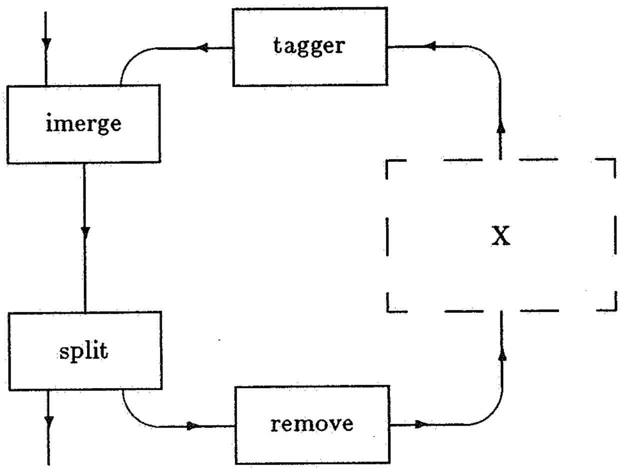

Figure 17: the context C_2

$\delta_{remove} =$

$\{((a^t), \sigma_{INIT}, (\epsilon), \sigma_{INIT}) : a \text{ is a token }\} \cup \{((a), \sigma_{INIT}, (a), \sigma_{INIT}) : a \text{ is a token }\}.$

amerge The *amerge* node is the same node as the previously introduced *merge* node:

$\delta_{amerge} =$

$\{((a, \epsilon), \sigma_{INIT}, (a), \sigma_{INIT}) : a \text{ is a token }\} \cup \{((\epsilon, a), \sigma_{INIT}, (a), \sigma_{INIT}) : a \text{ is a token }\}.$

Note that we renamed the *merge* node to *amerge* node (angelic merge) in order to distinguish it from the *imerge* node (infinity merge).

imerge $\delta_{imerge} =$

$\{((x, \epsilon), \sigma_{INIT}, (x), \sigma) : x \neq \epsilon\} \cup \{((\epsilon, x), \sigma_{INIT}, (x), \sigma) : x \neq \epsilon\} \cup$

$\{((x_1, x_2), \sigma, (x), \sigma) : x_1 \neq \epsilon \wedge x_2 \neq \epsilon \wedge x \text{ is a shuffle of } x_1 \text{ and } x_2\}.$

An alternative specification would be

$\{((x, \epsilon), \sigma_{INIT}, (x), \sigma_R) : x \neq \epsilon\} \cup \{((\epsilon, x), \sigma_{INIT}, (x), \sigma_L) : x \neq \epsilon\} \cup$

$((x, \epsilon), \sigma_L, (x), \sigma_R) : x \neq \epsilon\} \cup \{((\epsilon, x), \sigma_R, (x), \sigma_L) : x \neq \epsilon\}.$

The *amerge* and *imerge* nodes as defined here are the same nodes as in [Panangaden & Stark 1988], to which the reader is also referred for a discussion on the expressive power (in some sense they are weaker than a fair merge node). When there is a finite amount of input on one of the input lines of an *amerge* node the input on the other line is guaranteed to appear on the output line and when there is a infinite amount of input on one of the input lines of an *imerge* node the input on the other line is guaranteed to appear on the output line. We use the context C_1 (with the *amerge* node) when there is an finite amount of input on the input line and we use C_2 when there is an infinite amount of input.

split The split node has as its specification

$$\delta_{split} = \{((a), \sigma_{INIT}, (a,a), \sigma_{INIT}) : a \textit{ is a token } \}.$$

Besides tagged tokens we use tagged words of tokens and tagged elements of $FTrace^n$, $Trace^n$ and Dom^n:

$$(a_1 \ldots a_n)^t = a_1^t \ldots a_n^t \wedge (a_1 a_2 \ldots)^t = a_1^t a_2^t \ldots,$$

$$(x_1, \ldots, x_n)^t = (x_1^t, \ldots, x_n^t),$$

$$(y_1, \ldots, y_n)^t = (y_1^t, \ldots, y_n^t)$$

and

$$\begin{pmatrix} x_{11} & \ldots & x_{1n} \\ x_{21} & \ldots & x_{2n} \\ \vdots & & \vdots \end{pmatrix}^t = \begin{pmatrix} x_{11}^t & \ldots & x_{1n}^t \\ x_{21}^t & \ldots & x_{2n}^t \\ \vdots & & \vdots \end{pmatrix}.$$

Do not confuse θ^t (the tagged version of θ) with θ^T (the transpose of θ).

Recall that we assumed that both $\theta, \bar{\theta} \in Dom^1$. Suppose

$$\theta = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix} \wedge \bar{\theta} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \end{pmatrix}.$$

Define

$$\tilde{\theta} = \begin{pmatrix} x_1 \bar{x}_1^t \\ x_2 \bar{x}_2^t \\ \vdots \end{pmatrix},$$

$$\chi = \textit{flatten}(\theta),$$

$$\bar{\chi} = \textit{flatten}(\bar{\theta})$$

and

$$\tilde{\chi} = \textit{flatten}(\tilde{\theta}).$$

We prove the theorem in two stages: (choosing $i = 1$ if $\chi \in A^*$ and $i = 2$ if $\chi \in A^\omega$)

1. $\bar{\theta} \in \mathcal{D}(t_1)(\theta) \Rightarrow \tilde{\chi} \in \mathcal{O}(C_i[t_1])(\chi)$
2. $\tilde{\chi} \in \mathcal{O}(C_i[t_2])(\chi) \Rightarrow \bar{\theta} \in \mathcal{D}(t_2)(\theta)$

The result follows from 1. and 2. and from the assumption that $\forall C[\mathcal{O}(C[t_1]) = \mathcal{O}(C[t_2])]$ (this implies that $\tilde{\chi} \in \mathcal{O}(C_i[t_1])(\chi) \Rightarrow \tilde{\chi} \in \mathcal{O}(C_i[t_2])(\chi)$, $i = 1, 2$). In the rest of the proof we assume that $\chi \in A^*$. We will show below the point in the proof where we make use of this assumption. At that point we also show that in the case that $\chi \in A^\omega$ we can use the *imerge* node.

1. Figure 18 suggests a possible behaviour of the net < *tagger, amerge, split, remove* >. From the figure it is not difficult to see that we can connect lines $\{3:1, 4:2, 5:4\}$: the contents of that lines differ an ϵ-shift. By lemma 4.8 we can take an integer speed up simultaneously in input and output together as is suggested in figure 18: take $(\alpha_i)_i = (3, 1, 4, 1, 4, 1, \ldots)$. We have that

$$\begin{pmatrix} x_1 & x_1 \\ \epsilon & \epsilon \\ \bar{x}_1^t & \epsilon \\ x_2 & x_2 \\ \epsilon & \epsilon \\ \bar{x}_2^t & \epsilon \\ \vdots & \vdots \end{pmatrix} \in \mathcal{D}(< tagger, amerge, split, remove > \{3:1, 4:2, 5:4\})(\begin{pmatrix} \epsilon & x_1 \\ \epsilon & \epsilon \\ \bar{x}_1 & \epsilon \\ \epsilon & x_2 \\ \epsilon & \epsilon \\ \bar{x}_2 & \epsilon \\ \vdots & \vdots \end{pmatrix}).$$

Define

$$\tilde{t} = < tagger, amerge, split, remove > \{3:1, 4:2, 5:4\}.$$

This behaviour can be observed in figure 19 on the first two input and output lines. The second numbering in this figure refers to the numbering of corresponding lines in C_1. Let $(\eta_i)_i = (2, 3, 3, 3, \ldots)$. By lemma 4.11 we have that

$$\bar{\theta} \rhd (\eta_i)_i \in \mathcal{D}(t_1)(\theta \rhd (\eta_i)_i),$$

i.e.

$$\begin{pmatrix} \epsilon \\ \bar{x}_1 \\ \epsilon \\ \epsilon \\ \bar{x}_2 \\ \epsilon \\ \vdots \end{pmatrix} \in \mathcal{D}(t_1)(\begin{pmatrix} \epsilon \\ \bar{x}_1 \\ \epsilon \\ \epsilon \\ \bar{x}_2 \\ \epsilon \\ \vdots \end{pmatrix}).$$

We observe also this behaviour in figure 19. The input line $\boxed{3}$ matches the output line $\boxed{2}$ and the input line $\boxed{1}$ matches the output line $\boxed{3}$ (in the sense that they differ only by an ϵ-shift). From this we derive that

$$(\ x_1 \ \ \epsilon \ \ \bar{x}_1^t \ \ x_2 \ \ \epsilon \ \ \bar{x}_2^t \ \ \ldots\ )^T \in$$
$$\mathcal{D}(C[t_1])((\ x_1 \ \ \epsilon \ \ \epsilon \ \ x_2 \ \ \epsilon \ \ \epsilon \ \ \ldots\ )^T)$$

and because $\mathcal{O} = abstr \circ \mathcal{D}$ we have

$$\tilde{\chi} \in \mathcal{O}(C_1[t_1])(\chi).$$

2. Suppose $\tilde{\chi} \in \mathcal{O}(C_1[t_2])(\chi)$. Because $\mathcal{O} = abstr \circ \mathcal{D}$ we can find $\theta'_1, \theta'_2 \in Dom^1$ such that

$$\theta'_2 \in \mathcal{D}(C_1(t_2))(\theta'_1),$$
$$flatten(\theta'_2) = \tilde{\chi}$$

and

$$flatten(\theta'_1) = \chi.$$

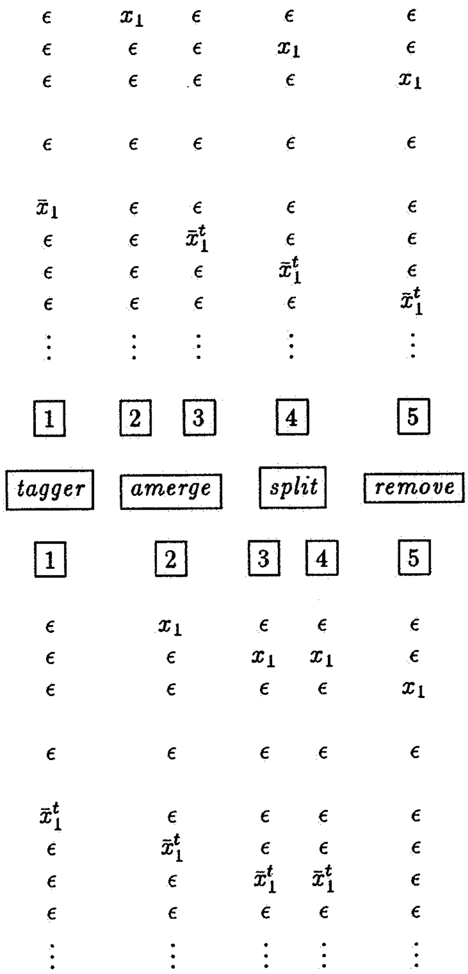

Figure 18: A behaviour of < *tagger*, *amerge*, *split*, *remove* >

$$
\begin{array}{ccc}
\epsilon & x_1 & \epsilon \\
\epsilon & \epsilon & x_1 \\
\bar{x}_1 & \epsilon & \epsilon \\
 & & \\
\epsilon & x_2 & \epsilon \\
\epsilon & \epsilon & x_2 \\
\bar{x}_2 & \epsilon & \epsilon \\
\vdots & \vdots & \vdots \\
 & & \\
(1) & (2) & (6) \\
\boxed{1} & \boxed{2} & \boxed{3} \\
 & & \\
\multicolumn{2}{c}{\tilde{t}} & t_1 \\
 & & \\
\boxed{1} & \boxed{2} & \boxed{3} \\
(3) & (5) & (6) \\
 & & \\
x_1 & x_1 & \epsilon \\
\epsilon & \epsilon & \bar{x}_1 \\
\bar{x}_1^t & \epsilon & \epsilon \\
 & & \\
x_2 & x_2 & \epsilon \\
\epsilon & \epsilon & \bar{x}_2 \\
\bar{x}_2^t & \epsilon & \epsilon \\
\vdots & \vdots & \vdots
\end{array}
$$

Figure 19: A behaviour of $\tilde{t}$ and t_1

Because

$$C_1(t_2) =< tagger, amerge, split, remove, t_2 > \{1:6, 3:1, 4:2, 5:4, 6:5\},$$

$$\mathcal{D}(C_1(t_2)) = \mathcal{D}(C_1[t_2])$$

we have that

$$\theta_2' \in \mathcal{D}(< tagger, amerge, split, remove, t_2 > \{1:6, 3:1, 4:2, 5:4, 6:5\})(\theta_1')$$

By definition of $\mathcal{D}$ we can find $\theta_1 \in Dom^6, \theta_2 \in Dom^6$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta_2 \in \mathcal{D}(< tagger, amerge, split, remove, t_2 >)(\theta_1),$$

$$\theta_1 \downarrow< 1,3,4,5,6 >= \epsilon\Box\theta_2 \downarrow< 6,1,2,4,5 >,$$

$$(\theta_1 \uparrow \{1,3,4,5,6\}) \rightsquigarrow (\alpha_i)_i = \theta_1',$$

$$(\theta_2 \uparrow \{6,1,2,4,5\}) \rightsquigarrow (\alpha_i)_i = \theta_2'.$$

Because for all θ

$$flatten(\theta \rightsquigarrow (\alpha_i)_i) = flatten(\theta)$$

we have

$$flatten(\theta_1 \uparrow \{1,3,4,5,6\}) = \chi$$

and

$$flatten(\theta_2 \uparrow \{6,1,2,4,5\}) = \tilde{\chi}.$$

From this we derive

$$\theta_1 \downarrow< 1 >= \epsilon\Box\theta_2 \downarrow< 6 >,$$

$$\theta_1 \downarrow< 2 >= \epsilon\Box\theta_2 \downarrow< 1 >,$$

$$\theta_1 \downarrow< 4 >= \epsilon\Box\theta_2 \downarrow< 2 >,$$

$$\theta_1 \downarrow< 5 >= \epsilon\Box\theta_2 \downarrow< 4 ,$$

$$\theta_1 \downarrow< 6 >= \epsilon\Box\theta_2 \downarrow< 5 >,$$

$$\theta_2 \downarrow< 1 >\in \phi_{tagger}(\theta_1 \downarrow< 1 >),$$

$$\theta_2 \downarrow< 2 >\in \phi_{amerge}(\theta_1 \downarrow< 2,3 >),$$

$$\theta_2 \downarrow< 3,4 >\in \phi_{split}(\theta_1 \downarrow< 4 >),$$

$$\theta_2 \downarrow< 5 >\in \phi_{remove}(\theta_1 \downarrow< 5 >),$$

$$\theta_2 \downarrow< 6 >\in \mathcal{D}(t_2)(\theta_1 \downarrow< 6 >),$$

$$flatten(\theta_1 \downarrow< 2 >) = \chi$$

and

$$flatten(\theta_2 \downarrow< 3 >) = \tilde{\chi}.$$

By properties of the nodes

$$flatten(\theta_1 \downarrow < 4 >) = \tilde{\chi},$$

(because $flatten(\theta_2 \downarrow < 3 >) = \tilde{\chi}$ and $flatten(\theta_1 \downarrow < 2 >) = \chi$)

$$flatten(\theta_1 \downarrow < 3 >) = \bar{\chi}^t.$$

We have the situation that χ is on one of the input lines and $\tilde{\chi}$ is on the output line. We want to conclude that $\bar{\chi}^t$ appears completely is on the other input line. This is only possible if we are sure that *all* of the second input line is put on the output. This is the place where we use the assumption that $\chi \in A^*$: by a property of the *amerge* node all of $\bar{\chi}^t$ is on the second input line. If $\chi \in A^\omega$ then we use the context C_2 which has the *imerge* node which guarantees that if the input on one of the input lines is infinite then *all* of the other input line will be appear on the output.

We continue with the proof: (because $flatten(\theta_1 \downarrow < 2 >) = \chi$ and $flatten(\theta_1 \downarrow < 4 >) = \tilde{\chi} \Rightarrow flatten(\theta_2 \downarrow < 2 >) = \tilde{\chi}$)

$$flatten(\theta_1 \downarrow < 1 >) = \bar{\chi},$$

(because $flatten(\theta_1 \downarrow < 3 >) = \bar{\chi}^t$ and all tokens that leave t_2 are not tagged)

$$flatten(\theta_1 \downarrow < 5 >) = \tilde{\chi},$$

(because $flatten(\theta_1 \downarrow < 4 >) = \tilde{\chi} \Rightarrow flatten(\theta_2 \downarrow < 4 >) = \tilde{\chi}$)

$$flatten(\theta_1 \downarrow < 6 >) = \chi,$$

(because $flatten(\theta_1 \downarrow < 5 >) = \tilde{\chi} \Rightarrow flatten(\theta_2 \downarrow < 5 >= \chi)$

$$flatten(\theta_2 \downarrow < 4 >) = \tilde{\chi},$$

(because $flatten(\theta_1 \downarrow < 4 >) = \tilde{\chi} \Rightarrow flatten(\theta_2 \downarrow < 4 >) = \tilde{\chi}$)

$$flatten(\theta_2 \downarrow < 2 >) = \tilde{\chi},$$

(because $flatten(\theta_1 \downarrow < 4 >) = \tilde{\chi}$)

$$flatten(\theta_2 \downarrow < 1 >) = \bar{\chi}^t,$$

(because $flatten(\theta_1 \downarrow < 1 >) = \bar{\chi}$)

$$flatten(\theta_2 \downarrow < 5 >) = \chi$$

(because $flatten(\theta_1 \downarrow < 5 >) = \tilde{\chi}$) and

$$flatten(\theta_2 \downarrow < 6 >) = \bar{\chi}.$$

(because $flatten(\theta_2 \downarrow < 1 >) = \bar{\chi}$).

We define ten infinite sequences of integers as follows. With each $\theta_2 \downarrow < i >$, $i = 1, 2, 4, 5, 6$ we associate a sequence $(\alpha_{ij})_{j=1}^{\infty}$ and with each $\theta_1 \downarrow < i >$, $i = 1, 3, 4, 5, 6$ we associate a sequence $(\beta_{ij})_{j=1}^{\infty}$ such that they satisfy

$$(\theta_2 \downarrow < 6 >)[\alpha_{6j} - 1] < \bar{\theta}[j] \leq (\theta_2 \downarrow < 6 >)[\alpha_{6j}],$$

$$(\theta_1 \downarrow < 1 >)[\beta_{1j} - 1] < \bar{\theta}[j] \leq (\theta_1 \downarrow < 1 >)[\beta_{1j}],$$

$$(\theta_2 \downarrow < 1 >)[\alpha_{1j} - 1] < \textit{tagged}\ (\bar{\theta}[j]) \leq (\theta_2 \downarrow < 1 >)[\alpha_{1j}],$$

$$(\theta_1 \downarrow < 3 >)[\beta_{3j} - 1] < \textit{tagged}\ (\bar{\theta}[j]) \leq (\theta_1 \downarrow < 3 >)[\beta_{3j}],$$

$$(\theta_2 \downarrow< 2 >)[\alpha_{2j} - 1] < \tilde{\theta}[j] \leq (\theta_2 \downarrow< 2 >)[\alpha_{2j}],$$

$$(\theta_1 \downarrow< 4 >)[\beta_{4j} - 1] < \tilde{\theta}[j] \leq (\theta_1 \downarrow< 4 >)[\beta_{4j}],$$

$$(\theta_2 \downarrow< 4 >)[\alpha_{4j} - 1] < \tilde{\theta}[j] \leq (\theta_2 \downarrow< 4 >)[\alpha_{4j}],$$

$$(\theta_1 \downarrow< 5 >)[\beta_{5j} - 1] < \tilde{\theta}[j] \leq (\theta_1 \downarrow< 5 >)[\beta_{5j}].$$

For $\theta_2 \downarrow< 5 >$ and $\theta_1 \downarrow< 6 >$ we have slightly different definitions of $(\alpha_{5j})_j$ and $(\beta_{6j})_j$:

$$(\theta_2 \downarrow< 5 >)[\alpha_{5j}] \leq \theta[j] < (\theta_2 \downarrow< 5 >)[\alpha_{5j} + 1]$$

and

$$(\theta_1 \downarrow< 6 >)[\beta_{6j}] \leq \theta[j] < (\theta_1 \downarrow< 6 >)[\beta_{6j} + 1].$$

By the ϵ shift we know that for all j

$$\beta_{1j} = \alpha_{6j} + 1,$$

$$\beta_{3j} = \alpha_{1j} + 1,$$

$$\beta_{4j} = \alpha_{2j} + 1,$$

$$\beta_{5j} = \alpha_{4j} + 1$$

and

$$\beta_{6j} = \alpha_{5j} + 1$$

and by properties of the nodes

$$\beta_{1j} \leq \alpha_{1j},$$

(if we do not consider the tagging of tokens, the *tagger* node behaves like an identity node)

$$\beta_{3j} \leq \alpha_{2j},$$

(if we hide all the tokens from the first input line of a *merge* node in input and output, the result is again an identity node)

$$\beta_{4j} \leq \alpha_{4j}$$

(if we only consider only the input line and the second output line of a *split* node we see the same behaviour as an identity node) and

$$\beta_{5j} - 1 \leq \alpha_{5j}$$

(the first place where all of $\tilde{\theta}[j]$ is visible (in θ_1) is smaller than the last place where not more than $\theta[j]$ is visible (in θ_2)) and we can conclude for each j

$$\alpha_{6j} < \beta_{6j}.$$

We have for all j

$$(\theta_1 \downarrow< 6 >)[\beta_{6j}] \leq \theta[j] < (\theta_1 \downarrow< 6 >)[\beta_{6j} + 1],$$

$$(\theta_2 \downarrow< 6 >)[\alpha_{6j} - 1] < \tilde{\theta}[j] \leq (\theta_2 \downarrow< 6 >)[\alpha_{6j}]$$

and

$\alpha_{6j} < \beta_{6j}$

so we can apply lemma 3.15 and derive

$\bar{\theta} \in \mathcal{D}(t_2)(\theta)$.

The next step is the generalization to cases that $t_1, t_2 \in Net^{n:m}$, $n \neq 1 \vee m \neq 1$. If $n = 0 \vee m = 0$ then we always have that $\mathcal{D}(t_1) = \mathcal{D}(t_2)$ (because we can assume $\mathcal{O}(t_1) = \mathcal{O}(t_2)$). If $n \neq 0, 1 \wedge m \neq 0, 1$ then we use the following construction. We take n copies of the loop of our context, as is shown in figure 20. If $\mathcal{D}(t_1) \neq \mathcal{D}(t_2)$ then we can always find an input θ such that there exists an output line l such that the contents of this output line (given the input θ) are different. In figure 20 we feed back n copies of this l-th output line to the original input. With the help of this context it is easy to

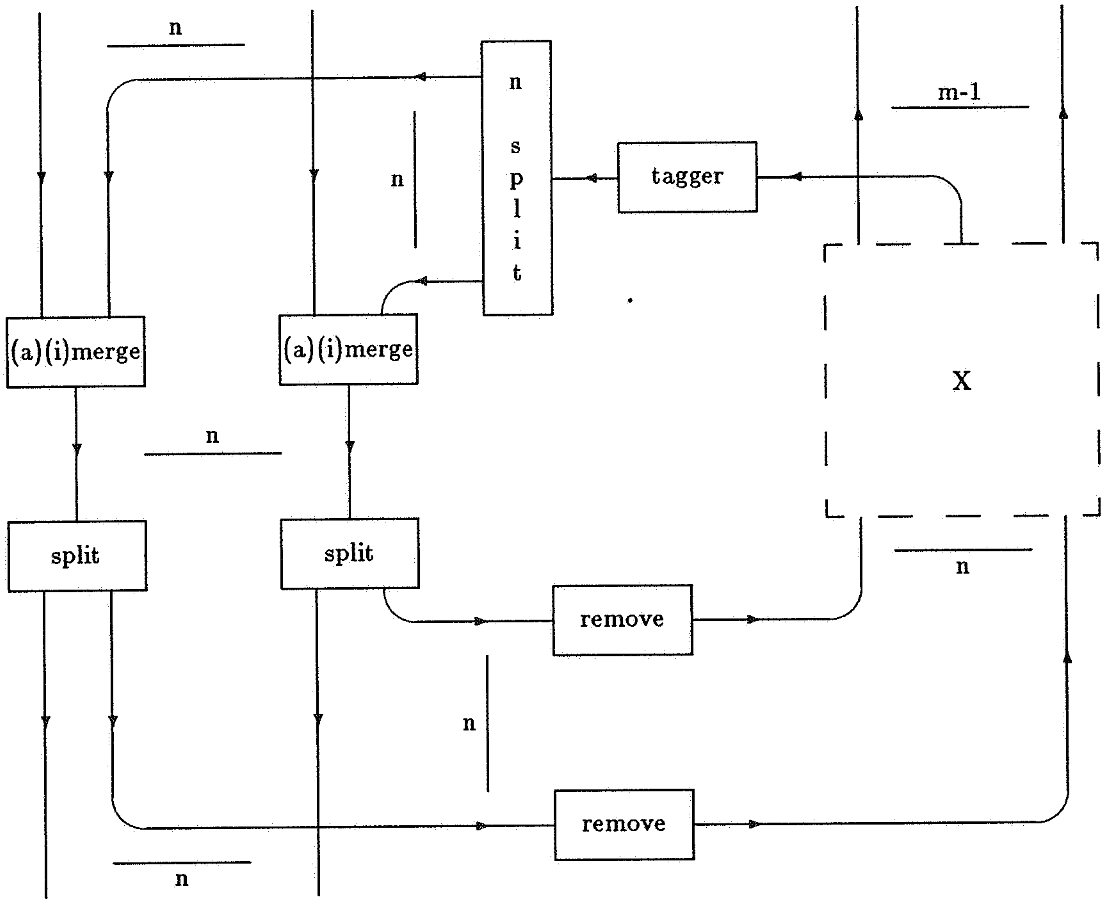

Figure 20: generalized context

mimick the arguments of the $n = m = 1$ case.

□

In the proof of the full abstraction we used two different kinds of merge nodes: the angelic merge and the infinity merge. If we are able to specify a fair merge node in our framework, then there is no need to use two different kinds of merge nodes. However, it is our conjecture that in our framework it is not possible to specify a fair merge node.

6 Metric semantics $\mathcal{N}$

In this section we define a metric semantics $\mathcal{N}$ for data flow nets. It is called metric because the domains are metric spaces (in fact, we assign a metric to a subset of $Dom^{n:m}$). The semantics $\mathcal{N}$ is a compositional semantics. It is only defined on a subset $FNet^{n:m}$ of $Net^{n:m}$. Elements of $FNet^{n:m}$ are called finite choice nets. The difference between $FNet^{n:m}$ and $Net^{n:m}$ is that for $FNet^{n:m}$ the set of nodes *Node* is restricted. This restriction enables us to set up a metric framework. We show that $\mathcal{N}$ is correct with respect to the operational semantics: we prove that $\mathcal{O} = abstr \circ \mathcal{N}$. (Recall that *abstr* also related the operational semantics $\mathcal{O}$ with denotational semantics $\mathcal{D}$.) We prove it in the following way. We define an auxiliary semantics $\mathcal{T}$. This semantics $\mathcal{T}$ which will serve as an intermediate semantics between the metric semantics $\mathcal{N}$ and the denotational semantics $\mathcal{D}$ is defined. Also operators *delay* and *close* are introduced. They can be seen as generalizations of $\searrow$ and $\rightsquigarrow (\alpha_i)_i$ to elements of $Dom^{n:m}$. Then we show in several steps that figure 21 commutes. From this result and

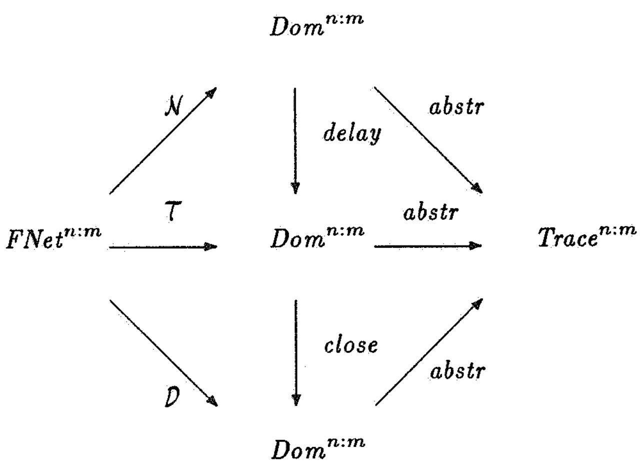

Figure 21: overview of the different semantics and operators

$\mathcal{O} = abstr \circ \mathcal{D}$ we derive that $\mathcal{O} = abstr \circ \mathcal{N}$. We informally describe the idea behind the semantics $\mathcal{T}$ and $\mathcal{N}$. If we look at the denotational semantics $\mathcal{D}$ we see that

- the denotational semantics is closed under simultaneous integer speed ups in input and output, that is

$$\bar{\theta} \in \mathcal{D}(s)(\theta) \Rightarrow \forall (\alpha_i)_i [\bar{\theta} \rightsquigarrow (\alpha_i)_i \in \mathcal{D}(s)(\theta \rightsquigarrow (\alpha_i)_i)]$$

- the denotational semantics is closed under taking delays in the output, that is

$$\bar{\theta} \in \mathcal{D}(s)(\theta) \Rightarrow \forall \bar{\theta}' \in \bar{\theta} \searrow [\bar{\theta}' \in \mathcal{D}(s)(\theta)].$$

The semantics $\mathcal{T}$ differs from $\mathcal{D}$ in the sense that the first property does not hold and the semantics $\mathcal{N}$ is in general neither closed under integer speed ups nor under delays.

We give an overview of the rest of this section. Subsection 6.1 provides the basic definitions and properties about metric spaces. In subsection 6.2 we introduce the domains for the metric semantics and the basic functions. The metric semantics itself is given in subsection 6.3 and subsection 6.4 shows a basic property of it related to the delay lemma for the denotational semantics $\mathcal{D}$. Subsection 6.5 gives the intermediate semantics $\mathcal{T}$ together with some properties. Finally, subsection 6.6 discusses the relation of the metric semantics $\mathcal{M}$ with the operational semantics $\mathcal{O}$.

6.1 Definitions and properties of metric spaces

In this subsection we give some basic definitions and properties about metric spaces.

Definition 6.1 (Metric Space) *A metric space is a pair (M, d) with M a non empty set and d a mapping $d : M \times M \to [0, 1]$ (a metric distance), which satisfies the following properties:*

1. $\forall x, y \in M[d(x, y) = 0 \Leftrightarrow x = y]$,
2. $\forall x, y \in M[d(x, y) = d(y, x)]$,
3. $\forall x, y, z \in M[d(x, y) \leq d(x, z) + d(z, y)]$.

A metric space is called an ultrametric space if we replace 3 by the stronger

$$\forall x, y, z \in M[d(x, y) \leq \max(d(x, z), d(z, y))].$$

Definition 6.2 *Let (M, d) be metric space. Let $(x_i)_i$ be a sequence in M.*

1. *We say that $(x_i)_i$ is a Cauchy sequence whenever we have*

$$\forall \epsilon > 0 \exists N \in \mathbf{N} \forall n, m > N[d(x_n, x_m) < \epsilon].$$

2. *Let $x \in M$. We say that $(x_i)_i$ converges to x and call x the limit of $(x_i)_i$ whenever we have*

$$\forall \epsilon > 0 \exists N \in \mathbf{N} \forall m > N[d(x, x_m) < \epsilon]$$

   Such a sequence we call convergent. Notation: $\lim_{i\to\infty} x_i = x$.

3. *The metric space (M, d) is called complete whenever each Cauchy sequence converges to an element of M.*

Definition 6.3 *Let (M_1, d_1), (M_2, d_2) be metric spaces.*

1. *Let $f : M_1 \to M_2$ be a function. We call f continuous whenever for each sequence $(x_i)_i$ with limit x in M_1 we have that $\lim_{i\to\infty} f(x_i) = f(x)$.*
2. *Let $c \geq 0$. With $M_1 \to^c M_2$ we denote the set of functions f from M_1 to M_2 that satisfy the following property:*

$$\forall x, y \in M[d_2(f(x), f(y)) \leq c.d_1(x, y)].$$

   Functions f in $M_1 \to^1 M_2$ we call non distance increasing, functions in $M_1 \to^c M_2$ with $0 \leq c < 1$ we call contracting.

Theorem 6.4 *Let (M_1, d_1), (M_2, d_2) be metric spaces. For every $c \geq 0$ and $f \in M_1 \to^c M_2$ we have: f is continuous.*

Theorem 6.5 (Banach's fixed point theorem) *Let (M, d) be a complete metric space and $f : M \to M$ a contracting function. Then there exists an $x \in M$ such that the following holds:*

1. $f(x) = x$ (*x is a fixed point of f*),
2. $\forall y \in M[f(y) = y \Rightarrow y = x]$ (*x is unique*),
3. $\forall x_0 \in M[\lim_{n\to\infty} f^n(x_0) = x]$, *where* $f^{n+1}(x_0) = f(f^n(x_0))$ *and* $f^0(x_0) = x_0$.

Definition 6.6 (closed subsets) *A subset X of a complete metric space* (M, d) *is called closed whenever each Cauchy sequence of elements in X converges to an element of X.*

Definition 6.7 *Let* (M, d) (M_1, d_1) (M_2, d_2) *be metric spaces.*

1. *We define a metric* d_S *on the functions in* $M_1 \to M$ *as follows. For every* $f_1, f_2 \in M_1 \to M$

$$d_S(f_1, f_2) = \sup\{d(f_1(x), f_2(x)) : x \in M_1\}$$

2. *Let* $\mathcal{P}_{closed}(M) = \{X \subset M : X \text{ is closed and non empty }\}$. *We define a metric* d_H *on* $\mathcal{P}_{closed}(M)$, *called the Hausdorff distance, as follows. For every* $X, Y \in \mathcal{P}_{closed}(M)$

$$d_H(X, Y) = \max\{\sup\{d(x, Y) : x \in X\}, \sup\{d(y, X) : y \in Y\}\}$$

where $d(x, Z) = \inf\{d(x, z) : z \in Z\}$ *for every* $Z \subset M$, $x \in M$.

Theorem 6.8 *Let* (M, d), (M_1, d_1), (M_2, d_2) *be complete metric spaces. We have that* $M_1 \to M_2$ *and* $\mathcal{P}_{closed}(M)$ *(with the metrics defined above) are complete metric spaces.*

Lemma 6.9 *Let* (M, d) *be a metric space. Consider the space of closed subsets with the Haussdorff metric* $(\mathcal{P}_{closed}(M), d_H)$. *We have*

$$\begin{aligned}
&\forall X, Y \in \mathcal{P}_{closed}(M) \forall c \geq 0[\\
&\qquad d_H(X, Y) \leq c\\
&\qquad \Leftrightarrow\\
&\qquad \forall x \in X \exists y \in Y[d(x, y) \leq c] \wedge\\
&\qquad \forall y \in Y \exists x \in X[d(x, y) \leq c]\\
&\qquad\qquad].
\end{aligned}$$

Theorem 6.10 (Hahn) *Let* $(X_i)_i$ *be a Cauchy sequence in* $\mathcal{P}_{closed}(M)$. *We have*

$$\lim_{i\to\infty} X_i = \{\lim_{i\to\infty} x_i : x_i \in X_i \wedge (x_i)_i \text{ a Cauchy sequence in } M\}.$$

Definition 6.11 (Contain Point) *For any* $\phi \in M \to \mathcal{P}(M)$ *we define*

$$CP(\phi) = \{x : x \in \phi(x)\}.$$

Remark

The set $CP(\phi)$ is usually called the fixed point of the multivalued function ϕ. We here follow Park's terminology ([Park 1983]) in calling $CP(\phi)$ the contain point of ϕ.

Given a function $\phi \in M \to \mathcal{P}(M)$ there are two (different) ways to take a fixed point:

1. take the contain point,
2. generalize the function ϕ to a function $\hat{\phi} : \mathcal{P}(M) \to \mathcal{P}(M)$ by defining $\hat{\phi}(X) = \bigcup_{x\in X} \phi(x)$. Under certain conditions there is a unique fixed point of $\hat{\phi}$ in the usual sense, that is a set Y such that $\hat{\phi}(Y) = Y$. Such a fixed point we shall denote by $FP(\phi)$.

Theorem 6.12 *If (M, d) is a complete metric space and $\phi : M \to \mathcal{P}_{closed}(M)$ is continuous then*

$$CP(\phi) = \{\lim_{i\to\infty} x_i : \forall i \in \mathbf{N}[x_{i+1} \in \phi(x_i)] \wedge (x_i)_i \text{ is a Cauchy sequence} \}.$$

If ϕ is contractive we have for all $x \in M$

$$CP(\phi) = \{\lim_{i\to\infty} x_i : \forall i \in \mathbf{N}[x_{i+1} \in \phi(x_i)] \wedge x_1 = x \wedge (x_i)_i \text{ is a Cauchy sequence} \}.$$

Proof

Suppose $\phi \in M \to \mathcal{P}_{closed}(M)$ is a continuous function. We prove

$$CP(\phi) = \{\lim_{i\to\infty} x_i : \forall i \in \mathbf{N}[x_{i+1} \in \phi(x_i)] \wedge (x_i)_i \text{ is a Cauchy sequence} \}.$$

($\supseteq$) By the continuity

$$\phi(\lim_{i\to\infty} x_i) = \lim_{i\to\infty} \phi(x_i).$$

Because $\forall i[x_{i+1} \in \phi(x_i)]$, by theorem 6.10 we have

$$\lim_{i\to\infty} x_i \in \lim_{i\to\infty} \phi(x_i)$$

($\subseteq$) Suppose $x \in \phi(x)$. Take the constant sequence $(x)_i$.

Suppose $\phi \in M \to \mathcal{P}_{closed}(M)$ is a contractive function. Take any $\hat{x} \in M$. We prove

$$CP(\phi) = \{\lim_{i\to\infty} x_i : \forall i \in \mathbf{N}[x_{i+1} \in \phi(x_i)] \wedge x_1 = \hat{x} \wedge (x_i)_i \text{ is a Cauchy sequence} \} :$$

($\supseteq$) The same as the previous case

($\subseteq$) Suppose $x \in \phi(x)$. Take $x_1 = \hat{x}$. By the contractivity of ϕ we have that there exists a c, $0 \le c < 1$, with

$$d_H(\phi(x), \phi(\hat{x})) \le c.d(x, \hat{x})$$

so (because $x \in \phi(x)$) by lemma 6.9 there exists a $x_2 \in \phi(x_1)$ such that

$$d(x, x_2) \le c.d(x, \hat{x}).$$

By the contractivity of ϕ we have

$$d_H(\phi(x), \phi(x_2)) \le c.d(x, x_2)$$

so

$$d_H(\phi(x), \phi(x_2)) \leq c^2.d(x, \hat{x})$$

and we can find a $x_3 \in \phi(x_2)$ with

$$d(x, x_3) \leq c^2.d(x, \hat{x}).$$

When we continue this way we find a sequence $(x_i)_i$ with $\forall i \in \{1, 2, \ldots\}[x_{i+1} \in \phi(x_i)]$ and

$$d(x, x_i) \leq c^{i-1}.d(x, \hat{x})$$

and hence $x = \lim_{i \to \infty} x_i$. □

Theorem 6.13 *If (M, d) is a complete metric space and $\phi : M \to \mathcal{P}_{closed}(M)$ is contracting then*

1. *$CP(\phi) \subset FP(\phi)$,*
2. *$CP(\phi)$ is closed,*
3. *$CP(\phi)$ is nonempty.*

Proof

($CP(\phi) \subset FP(\phi)$) Take an arbitrary $x \in CP(\phi)$.

$$x \in \phi(x) \Rightarrow \{x\} \subset \hat{\phi}(\{x\})$$

$$\hat{\phi}(\{x\}) \subset \hat{\phi}^2(\{x\})$$

$$\hat{\phi}^2(\{x\}) \subset \hat{\phi}^3(\{x\})$$

$$\ldots$$

so $\forall i[x \in \hat{\phi}^i(\{x\})]$. By Banach's fixed point theorem we have

$$FP(\phi) = \lim_{i \to \infty} \hat{\phi}^i(\{x\})$$

and by Hahn's theorem

$$\lim_{i \to \infty} \hat{\phi}^i(\{x\}) = \{\lim_{i \to \infty} x_i : x_i \in \hat{\phi}^i(\{x\}) \wedge (x_i)_i \text{ is a Cauchy sequence}\}$$

so we derive $x \in FP(\phi)$.

($CP(\phi)$ is closed) Let $(x_i)_i$ be a Cauchy sequence in $CP(\phi)$. By the continuity of ϕ we have

$$\phi(\lim_{i \to \infty} x_i) = \lim_{i \to \infty} \phi(x_i).$$

We have that $\forall i[x_i \in \phi(x_i)]$ and hence, by Hahn's theorem,

$$\lim_{i \to \infty} x_i \in \lim_{i \to \infty} \phi(x_i)$$

so we derive

$$\lim_{i\to\infty} x_i \in \phi(\lim_{i\to\infty} x_i)$$

i.e.

$$\lim_{i\to\infty} x_i \in CP(\phi).$$

($CP(\phi)$ is nonempty) Pick a $x_1 \in M$ and a $x_2 \in \phi(x_1)$. By the contractivity of ϕ we have that there exists a c, $0 \le c < 1$, with

$$d_H(\phi(x_1), \phi(x_2)) \le c.d(x_1, x_2)$$

so (because $x_2 \in \phi(x_1)$) by lemma 6.9 we can pick a $x_3 \in \phi(x_2)$ with

$$d(x_2, x_3) \le c.d(x_1, x_2).$$

By the contractivity of ϕ we have

$$d_H(\phi(x_2), \phi(x_3)) \le c.d(x_2, x_3) \le c^2.d(x_1, x_2)$$

so (because $x_3 \in \phi(x_2)$) we can pick a $x_4 \in \phi(x_3)$ with

$$d(x_3, x_4) \le c^2.d(x_3, x_4)$$

$$\ldots$$

In this way we obtain a Cauchy sequence $(x_i)_i$ and by theorem 6.12 we have

$$\lim_{i\to\infty} x_i \in CP(\phi).$$

and so $CP(\phi)$ is nonempty. □

Remark In general we do not have for a contraction $\phi : M \to \mathcal{P}_{closed}(M)$ that $FP(\phi) \subset CP(\phi)$. Take for example M the set of finite and infinite words over the alphabet $\{a, b\}$. Take the usual metric on such words: $d(x, y) = 2^{-\max\{x[n]=y[n]:n\in\mathbf{N}\}}$. Take $\phi(x) = a \cdot x \cup \{b\}$. This is a contraction. We have $FP(\phi) = a^* \cdot b \cup \{a^\omega\}$ and $CP(\phi) = \{a^\omega, b\}$.

Theorem 6.14 *Let (M, d) be a complete metric space and (M_1, d_1) be a complete ultrametric space. Let $f : M \times M_1 \to \mathcal{P}_{closed}(M_1)$ be such that*

1. *$\forall x \in M[\lambda y.f(x, y)$ is contractive $]$,*
2. *$\forall y \in M_1[\lambda x.f(x, y)$ is non distance increasing $]$.*

We have that $\lambda x.CP(\lambda y.f(x, y)) \in M \to \mathcal{P}_{closed}(M_1)$ is a non distance increasing function.

Proof

Take any $x_1, x_2 \in M$. Take a $y \in M_1$ such that $y \in f(x_1, y)$. We show that there exists a $\tilde{y}$ with $\tilde{y} \in f(x_2, \tilde{y})$ and $d_1(y, \tilde{y}) \leq d(x_1, x_2)$. If we have this result, the theorem follows from symmetry and lemma 6.9. By 2. we have

$$(d_1)_H(f(x_1, y), f(x_2, y)) \leq d(x_1, x_2).$$

Because $y \in f(x_1, y)$, by lemma 6.9, we have that there exists a $\tilde{y}_2 \in f(x_2, y)$ with

$$d_1(y, \tilde{y}_2) \leq d(x_1, x_2).$$

From 1. we derive

$$(d_1)_H(f(x_2, y), f(x_2, \tilde{y}_2)) \leq c_1.d_1(y, \tilde{y}_2) \leq c_1.d(x_1, x_2)$$

for a certain $0 \leq c_1 < 1$. Because $\tilde{y}_2 \in f(x_2, y)$ we have that there exists a $\tilde{y}_3 \in f(x_2, \tilde{y}_2)$ with

$$d_1(\tilde{y}_2, \tilde{y}_3) \leq c_1.d(x_1, x_2)$$

and we have

$$d_1(y, \tilde{y}_3) \leq \max(d_1(y, \tilde{y}_2), d_1(\tilde{y}_2, \tilde{y}_3)) \leq d(x_1, x_2)$$

From 1. we derive

$$(d_1)_H(f(x_2, \tilde{y}_2), f(x_2, \tilde{y}_3)) \leq c_1.c_2.d(x_1, x_2)$$

for a certain $0 \leq c_2 < 1$. Because $\tilde{y}_3 \in f(x_2, \tilde{y}_3)$ we have that there exists a $\tilde{y}_4 \in f(x_2, \tilde{y}_3)$ with

$$d_1(\tilde{y}_3, \tilde{y}_4) \leq c_1.c_2.d(x_1, x_2)$$

and we have

$$d_1(y, \tilde{y}_4) \leq \max(d_1(y, \tilde{y}_3), d_1(\tilde{y}_3, \tilde{y}_4)) \leq d(x_1, x_2)$$

$$\ldots$$

Take $\tilde{y} = \lim_{i \to \infty} \tilde{y}_i$. We have

$$d_1(y, \tilde{y}) = d_1(y, \lim_{i \to \infty} \tilde{y}_i) = \lim_{i \to \infty} d_1(y, \tilde{y}_i) \leq d(x_1, x_2)$$

and, because $\forall i[\tilde{y}_{i+1} \in f(x_2, \tilde{y}_i)]$

$$\tilde{y} = \lim_{i \to \infty} \tilde{y}_i \in f(x_2, \lim_{i \to \infty} \tilde{y}_i) = f(x_2, \tilde{y}).$$

□

For more details about metric spaces consult [Dugundji 1966] or [Engelking 1977], and for an account of fixed points of multivalued functions we refer to [Nadler 1970].

6.2 Metric domains and basic functions

In this subsection we fix some metric domains. The domains are used in the definition of a semantics $\mathcal{N}$, which will be defined in the next subsection. We start with the definition of a metric on Dom^n.

Definition 6.15 (Metric on Dom^n) *Let*

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \end{pmatrix} < k > = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \\ x_{k1} & \cdots & x_{kn} \end{pmatrix}$$

Let a metric d on Dom^n be given by

$$d(\theta_1, \theta_2) = \begin{cases} 2^{-\max\{k: \theta_1 < k > = \theta_2 < k >\}} & if \theta_1 \neq \theta_2 \\ 0 & if \theta_1 = \theta_2. \end{cases}$$

Example

Let

$$\theta_1 = \begin{pmatrix} 29 & 86 & \epsilon \\ 711 & 1 & 2 \\ 44 & 40 & 59 \\ 2 & 4 & 33 \\ 3 & \epsilon & 12 \\ \vdots & \vdots & \vdots \end{pmatrix} \wedge \theta_2 = \begin{pmatrix} 29 & 86 & \epsilon \\ 711 & 1 & 2 \\ 45 & 40 & 59 \\ 2 & 14 & 13 \\ \epsilon & 9 & \epsilon \\ \vdots & \vdots & \vdots \end{pmatrix}.$$

We have $d(\theta_1, \theta_2) = \frac{1}{4}$.

Definition 6.16 *Let*

$$Dom_{cl}^{n:m} =^{def} Dom^n \rightarrow \mathcal{P}_{closed}(Dom^m)$$

and let a metric $\tilde{d}$ on $Dom_{cl}^{n:m}$ be given by

$$\tilde{d}(\phi_1, \phi_2) = \sup\{d_H(\phi_1(\theta), \phi_2(\theta)) : \theta \in Dom^n\}$$

where d_H is the Hausdorff distance on $\mathcal{P}_{closed}(Dom^m)$ induced by the metric d on Dom^m.

Theorem 6.17 *For each n, m we have that Dom^n and $Dom_{cl}^{n:m}$ are complete metric spaces.*

The next definition introduces some notation:

Definition 6.18 *Let $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$. Define $\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta})$ to be $\hat{\theta}$ where $\hat{\theta} \in Dom^{n+k}$ is such that*

$$\hat{\theta} \uparrow \{i_1, \ldots, i_k\} = \theta$$

and

$$\hat{\theta} \downarrow < i_1, \ldots, i_k > = \epsilon \Box (\bar{\theta} \downarrow < j_1, \ldots, j_k >).$$

Example

If

$$\theta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \wedge \bar{\theta} = \begin{pmatrix} 11 & 22 & 33 & 44 & 55 & 66 & 77 \\ 11 & 22 & 33 & 44 & 55 & 66 & 77 \\ 11 & 22 & 33 & 44 & 55 & 66 & 77 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

then

$$\mathcal{L}(\{1:2, 3:4\}, \theta, \bar{\theta}) = \begin{pmatrix} \epsilon & 1 & \epsilon & 2 & 3 & 4 & 5 \\ 22 & 1 & 44 & 2 & 3 & 4 & 5 \\ 22 & 1 & 44 & 2 & 3 & 4 & 5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

We give the generalizations of $\searrow$ to *delay* : $Dom^{n:m} \longrightarrow Dom^{n:m}$ and $\rightsquigarrow (\alpha_i)_i$ to *close* : $Dom^{n:m} \longrightarrow Dom^{n:m}$ in

Definition 6.19 *Define close* :$Dom^{n:m} \longrightarrow Dom^{n:m}$ *by*

$$close(\phi) = \lambda\theta.\{\bar{\theta} : \exists\theta_1 \in Dom^n, \theta_2 \in Dom^m[\theta_1 \rightsquigarrow (\alpha_i)_i = \theta \wedge \theta_2 \rightsquigarrow (\alpha_i)_i = \bar{\theta} \wedge \theta_2 \in \phi(\theta_1)]\}$$

and delay :$Dom^{n:m} \longrightarrow Dom^{n:m}$ *by*

$$delay(\phi) = \lambda\theta.\bigcup\{\bar{\theta} \searrow : \bar{\theta} \in \phi(\theta)\}$$

We first give two properties of *close* and *delay*:

Lemma 6.20

1. *abstr* $\circ$ *close* = *abstr*
2. *abstr* $\circ$ *delay* = *abstr*

Proof

We only show the first property. Take any $\phi \in Dom^{n:m}$. We have:

$close(abstr(\phi))$ = [*definition of abstr*]

$\lambda\Gamma.\{flatten(\bar{\theta}) : \exists\theta[flatten(\theta) = \Gamma \wedge \bar{\theta} \in close(\phi)(\theta)]\}$ = [*definition of close*]

$\lambda\Gamma.\{flatten(\bar{\theta} \rightsquigarrow (\alpha_i)_i) : \exists\theta\exists\theta_1[flatten(\theta) = \Gamma \wedge \bar{\theta} \in \phi(\theta_1) \wedge \theta_1 \rightsquigarrow (\alpha_i)_i = \theta]\}$ = [*substitution*]

$\lambda\Gamma.\{flatten(\bar{\theta} \rightsquigarrow (\alpha_i)_i) : \exists\theta_1[flatten(\theta_1 \rightsquigarrow (\alpha_i)_i) = \Gamma \wedge \bar{\theta} \in \phi(\theta_1)]\}$ = [*definition of flatten*]

$\lambda\Gamma.\{flatten(\bar{\theta}) : \exists\theta_1[flatten(\theta_1) = \Gamma \wedge \bar{\theta} \in \phi(\theta_1)]\}$ = [*definition of abstr*]

$abstr(\phi)$.

□

The following lemma will be used often in the sequel.

Lemma 6.21 *Let* $(\theta_i)_i$ *be a Cauchy sequence in* Dom^n. *Let* θ *be such that* $\forall i[\theta \in \theta_i \searrow]$. *Then* $\theta \in (\lim_{i\to\infty} \theta_i) \searrow$.

Proof

By definition of $\searrow$, $\forall i[\theta \in \theta_i \searrow]$ implies

$$\forall i \forall \theta_i \forall k \exists n_{k.i}[\theta[k] \leq \theta_i[k] \leq \theta[n_{k.i}]].$$

Choose a k arbitrary. We prove

$$\theta[k] \leq (\lim_{i\to\infty} \theta_i)[k] \wedge \exists n_k[(\lim_{i\to\infty} \theta_i)[k] \leq \theta[n_k]] :$$

Because $(\theta_i)_i$ is a Cauchy sequence, we can find a m_k such that for all $i \geq m_k$ we have that $\theta_i[k]$ is constant. Hence $(\lim_{i\to\infty} \theta_i)[k]$ equals $\theta_i[k]$ for $i \geq m_k$, so in particular $(\lim_{i\to\infty} \theta_i)[k] = \theta_{m_k}[k]$. We have $\theta[k] \leq \theta_{m_k}[k]$ so also $\theta[k] \leq (\lim_{i\to\infty} \theta_i)[k]$. There also exists a $n_{k.m_k}$ such that $\theta_{m_k}[k] \leq \theta[n_{k.m_k}]$ so also $(\lim_{i\to\infty} \theta_i)[k] \leq \theta[n_{k.m_k}]$. Take $n_k = n_{k.m_k}$. We now can conclude that $\theta \in (\lim_{i\to\infty} \theta_i) \searrow$. □

6.3 The metric semantics $\mathcal{N}$

In this subsection we define a metric semantics $\mathcal{N}$. We want to define it in such a way that it maps a net to $Dom_{cl}^{n:m}$. In order to do this we have to restrict the class of nets we use. We take the class of nets which have as basic nodes the so called finite choice nodes. This is a real restriction in the sense that there exist nodes that are not finite choice nodes. The semantics $\mathcal{N}$ has some interesting properties. It is formulated with the help of contain points. The contain points can be obtained by iteration. a further characteristic point of $\mathcal{N}$ is that we do not apply the *delay* operator in its definition. Not withstanding this $\mathcal{N}$ is still correct with respect to the operational semantics. This result crucially depends on the restriction to finite choice nets. There are some disadvantages too: the semantics is not fully abstract and it can handle only nets with finite choice nodes.

Definition 6.22

1. *If δ_d is a specification for a node d, we define $\hat{\delta}_d$ to be the smallest set such that*
   (a) $\delta_d \subset \hat{\delta}_d$
   (b) $(\chi_1, \sigma_1, \chi_3, \sigma) \in \hat{\delta}_d$ *and* $(\chi_2, \sigma, \chi_4, \sigma_2) \in \hat{\delta}_d$ *implies* $(\chi_1\chi_2, \sigma_1, \chi_3\chi_4, \sigma_2) \in \hat{\delta}_d$
2. *A node $d \in Node^{n:m}$ is called a finite choice node if the following two conditions hold*
   (a) $\forall \chi \in FTrace^n[\hat{\delta}_d \cap (\{\chi\} \times \{\sigma_{INIT}\} \times FTrace^m \times \Sigma)$ *is a finite set*$]$
   (b) $\forall \sigma \exists n \forall (\chi, \sigma, \bar{\chi}, \bar{\sigma}) \in \delta_d[|\chi| \leq n]$.

The first condition states that a finite amount of input on the input lines of a finite choice node can produce only a finite number of different output histories. Moreover, these histories are all finite. The second condition states that a finite choice node can make its decision which firing rule it uses after it has received a certain number of tokens. This number is fixed for each state. These first two conditions together will ensure that we can work in $Dom_{cl}^{n:m}$ (in fact, it is also possible to work with compact sets instead of closed sets in this case).

Example

Take nodes d_1, d_2 with

$$\delta_{d_1} = \{(\epsilon, \sigma_0, a, \sigma_1), (\epsilon, \sigma_1, a, \sigma_2), \ldots\},$$

$$\delta_{d_2} = \{(1^n, \sigma_0, a, \sigma_1) : n \geq 1\}.$$

Both nodes are not finite choice nodes.

Definition 6.23 *Let $FNet^{n:m} \subseteq CNet^{n:m}$ be the set of nets with n inputs and m outputs that have as basic nodes the finite choice nodes.*

Next we give meaning to the basic (finite choice) nodes. The semantics $\mathcal{N}$ will map a node d to a function $\psi_d \in Dom_{cl}^{n:m}$. The definition of ψ_d will use the notion of a firing sequence.

Definition 6.24 *With any finite choice node $d \in Node^{n:m}$ we associate a function $\psi_d \in Dom_{cl}^{n:m}$ which is given by*

$$\begin{aligned}\psi_d(\theta) = \{\bar\theta : \exists(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_i \text{ a firing sequence for } d \text{ w.r.t } flatten(\theta)\ [\\ \forall i \exists j [\ \theta[j] \geq \chi_1 \cdots \chi_i \wedge \theta[j-1] \not\geq \chi_1 \cdots \chi_i \\ \bar\theta[j] \geq \bar\chi_1 \cdots \bar\chi_i \wedge \bar\theta[j-1] \leq \bar\chi_1 \cdots \bar\chi_{i-1} \\ ] \\ \wedge\ flatten(\bar\theta) = \bar\chi_1 \bar\chi_2 \cdots \\ ] \\ \}\end{aligned}$$

It is not difficult to show that

Lemma 6.25 *For any finite choice node $d \in Node^{n:m}$ we have $\phi_d = delay \circ \psi_d$.*

Intuitively, we fire the node as soon as it is possible: we 'removed' the *delay* operator. In this way we do not get non closed sets as is the case if we use the original definition ϕ_d instead of ψ_d. We have for example

$$\bar\theta \in \phi_d(\theta) \Rightarrow \forall n[\underbrace{\epsilon \Box (\cdots \epsilon \Box}_{n} \bar\theta) \in \phi_d(\theta)]$$

but in general

$$\lim_{n\to\infty} \underbrace{\epsilon \Box (\cdots \epsilon \Box}_{n} \bar\theta) \notin \phi_d(\theta).$$

Before we prove that for any finite choice node d we have that ψ_d is well defined, we first introduce the notion of a prefix of a firing sequence and give a lemma.

Definition 6.26 *The prefix of length n of an infinite firing sequence $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_{i=1}^{\infty}$ is $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_{i=1}^{n}$ and the prefix of length n of a finite firing sequence $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_{i=1}^{m}$ is $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_{i=1}^{m}$ if $m < n$ and is $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_{i=1}^{n}$ if $m \geq n$.*

Lemma 6.27 *Suppose $\{(\chi_{ij}, \sigma_{ij}, \bar\chi_{ij}, \bar\sigma_{ij})_j : i \geq 1\}$ is a set of firing sequences for a finite choice node d with respect to Γ. Then there exists a firing sequence $(\chi_j, \sigma_j, \bar\chi_j, \bar\sigma_j)_j$ for d with respect to Γ such that for every n there is an infinite number of elements of $\{(\chi_{ij}, \sigma_{ij}, \bar\chi_{ij}, \bar\sigma_{ij})_j : i \geq 1\}$ that have the same prefix of length n as $(\chi_j, \sigma_j, \bar\chi_j, \bar\sigma_j)_j$.*

Proof

(The proof follows a standard argument for proving compactness.) Consider a set $\{(\chi_{ij}, \sigma_{ij}, \bar{\chi}_{ij}, \bar{\sigma}_{ij})_j : i \geq 1\}$ of firing sequences for a finite choice node d with respect to Γ. All firing sequences start in the initial state:

$$\{(\chi_{i1}, \sigma_{i1}, \bar{\chi}_{i1}, \bar{\sigma}_{i1}); i \geq 1\} = \{(\chi_{i1}, \sigma_{INIT}, \bar{\chi}_{i1}, \bar{\sigma}_{i1}); i \geq 1\}$$

Because $\forall i[\chi_{i1} \leq \mathit{flatten}(\theta)]$ and $\exists n[|\chi_{i1}| \leq n]$ (d is a finite choice node) there is only a finite number of possibilities for χ_{i1}. So we can conclude (because d is a finite choice node) that

$$\{(\chi_{i1}, \sigma_{INIT}, \bar{\chi}_{i1}, \bar{\sigma}_{i1}); i \geq 1\}$$

is a finite set. Hence there is an infinite number of firing sequences in $\{(\chi_{ij}, \sigma_{ij}, \bar{\chi}_{ij}, \bar{\sigma}_{ij})_j : i \geq 1\}$ that start with the same element. Consider this subset. We can continue in the same way to find the desired firing sequence. □

Next we show

Theorem 6.28 *For any finite choice node d we have that ψ_d is well defined.*

Proof

We show that for any θ we have that $\psi_d(\theta)$ is a closed set. Take any Cauchy sequence $(\theta_i)_i$ in $\psi_d(\theta)$. With each θ_i there is a firing sequence $(\chi_{ij}, \sigma_{ij}, \bar{\chi}_{ij}, \bar{\sigma}_{ij})_j$ associated. By lemma 6.27 there exists a firing sequence (for d with respect to $\mathit{flatten}(\theta)$) $(\chi_j, \sigma_j, \bar{\chi}_j, \bar{\sigma}_j)_j$ such that for every n there is an infinite number of elements of $\{(\chi_{ij}, \sigma_{ij}, \bar{\chi}_{ij}, \bar{\sigma}_{ij})_j : i \geq 1\}$ that have the same prefix of length n as $(\chi_j, \sigma_j, \bar{\chi}_j, \bar{\sigma}_j)_j$. We can use this firing sequence to show that $\lim_{i\to\infty} \theta_i \in \psi_d(\theta)$. □

Now we are ready to define the metric semantics $\mathcal{N}$.

Definition 6.29 (Metric semantics $\mathcal{N}$) *Define $\mathcal{N} : FNet^{n:m} \to Dom_{cl}^{n:m}$ as follows*

1. $\mathcal{N}(d) = \psi_d$
2. $\mathcal{N}(< s_1, \ldots, s_k >) = \mathcal{N}(s_1) :: \cdots :: \mathcal{N}(s_k)$
3. $$\mathcal{N}(s\{i_1 : j_1, \ldots, i_k : j_k\}) = \lambda\theta.(CP(\lambda\bar{\theta}.\mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}))) \tilde{\uparrow} \{j_1, \ldots, j_k\}$$

where $\tilde{\uparrow}$ is the extension of $\uparrow$ to sets.

For explanatory purposes we rewrite a part of definion of the denotational semantics $\mathcal{D}$ in a form that resembles the definition of $\mathcal{N}$: Let $s \in CNet^{n:m}$. We have

$$\begin{aligned}
&\mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\}) = \\
&\lambda\theta.\{\bar{\theta} : \exists\theta_1 \in Dom^{n+k}, \theta_2 \in Dom^{m+k}, (\alpha_i)_i[\\
&\quad \theta_2 \in \mathcal{D}(s)(\theta_1) \wedge \\
&\quad (\theta_1 \uparrow \{i_1, \ldots, i_k\}) \rightsquigarrow (\alpha_i)_i = \theta \wedge \\
&\quad (\theta_2 \uparrow \{j_1, \ldots, j_k\}) \rightsquigarrow (\alpha_i)_i = \bar{\theta} \wedge \\
&\quad \theta_1 \downarrow < i_1, \ldots, i_k >= \epsilon \Box \theta_2 \downarrow < j_1, \ldots, j_k > \\
&\quad]\} =
\end{aligned}$$

$$
\begin{aligned}
close(\lambda\theta.\{\bar\theta : \exists\theta_1 \in\ & Dom^{n+k}, \theta_2 \in Dom^{m+k}, [\\
& \theta_2 \in \mathcal{D}(s)(\theta_1) \wedge\\
& (\theta_1 \uparrow \{i_1, \ldots, i_k\}) = \theta \wedge\\
& (\theta_2 \uparrow \{j_1, \ldots, j_k\}) = \bar\theta \wedge\\
& \theta_1 \downarrow < i_1, \ldots, i_k >= \epsilon\Box\theta_2 \downarrow < j_1, \ldots, j_k >\\
&]\}) =
\end{aligned}
$$

$$
\begin{aligned}
close(\lambda\theta.\{\bar\theta : \exists\theta_1 \in\ & Dom^{n+k}, \theta_2 \in Dom^{m+k}, [\\
& \theta_2 \in \mathcal{D}(s)(\theta_1) \wedge\\
& \theta_1 = \mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \theta_2) \wedge\\
& (\theta_2 \uparrow \{j_1, \ldots, j_k\}) = \bar\theta\\
&]\}) =
\end{aligned}
$$

$$
\begin{aligned}
close(\lambda\theta.\{\theta_2 \uparrow \{j_1, \ldots, j_k\} : \exists\theta_1 \in\ & Dom^{n+k}, \theta_2 \in Dom^{m+k}, [\\
& \theta_2 \in \mathcal{D}(s)(\theta_1) \wedge\\
& \theta_1 = \mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \theta_2)\\
&]\}) =
\end{aligned}
$$

$$
\begin{aligned}
& close(\lambda\theta.\{\theta_2 \uparrow \{j_1, \ldots, j_k\} :\\
& \theta_2 \in \mathcal{D}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \theta_2)\\
& \}) =
\end{aligned}
$$

$$
\begin{aligned}
& close(\lambda\theta.\{\theta_2 :\\
& \theta_2 \in \mathcal{D}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \theta_2)\\
& \}\tilde\uparrow\{j_1, \ldots, j_k\}) =
\end{aligned}
$$

$$
\begin{aligned}
& close(\lambda\theta.CP(\lambda\theta_2.\\
& \mathcal{D}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \theta_2)\\
& \}\tilde\uparrow\{j_1, \ldots, j_k\})
\end{aligned}
$$

Lemma 6.30 *$\mathcal{N}$ is well-defined.*

Proof

We have to show that for all $s \in FNet^{n:m}$ we have $\mathcal{N}(s) \in Dom_{cl}^{n:m}$. We prove by induction on s that

1. $\mathcal{N}(s)$ is non distance increasing,
2. $\forall\theta[\mathcal{N}(s)(\theta)$is closed$]$.

(d) By definition of ψ_d we have for all $\theta \in Dom^n$ that $\mathcal{N}(d)(\theta) = \psi_d(\theta) \in \mathcal{P}_{closed}(Dom^m)$ so $\mathcal{N}(d)(\theta)$ is closed. We prove that ψ_d is non distance increasing. Take any θ_1, θ_2. We have to show $d(\psi_d(\theta_1), \psi_d(\theta_2)) \leq d(\theta_1, \theta_2)$. Due to the symmetric role of θ_1, θ_2 it suffices by lemma 6.9 to prove that

$$\forall\bar\theta_1 \in \psi_d(\theta_1)\exists\bar\theta_2 \in \psi_d(\theta_2)[d(\bar\theta_1, \bar\theta_2) \leq d(\theta_1, \theta_2)].$$

Take a $\bar\theta_1 \in \psi_d(\theta_1)$. By definition of ψ_d there exist sequences $(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i)_i$ and $(\alpha_i)_i$ such that

1. $\forall i[(\chi_i, \sigma_i, \bar\chi_i, \bar\sigma_i) \in \delta_d]$,

2. σ_1 is the initial state σ_{INIT},
3. $\forall i[\bar{\sigma}_i = \sigma_{i+1}]$,
4. $\chi_1\chi_2 \ldots \leq \mathit{flatten}(\theta_1)$,
5. if the sequence is finite (of length n) then it can not be extended: there is no $(\chi_{n+1}, \sigma_{n+1}, \bar{\chi}_{n+1}, \bar{\sigma}_{n+1})$ with $\chi_1 \ldots \chi_{n+1} \leq \mathit{flatten}(\theta_1)$ and $\bar{\sigma}_n = \sigma_{n+1}$.

Let $d(\theta_1, \theta_2) = 2^{-N}$, i.e. $\theta_1[N] = \theta_2[N]$. Because $\theta_1[N] = \theta_2[N]$ we can use the first $\max\{k : \alpha_k \leq N\}$ elements of the firing sequence of θ_1 to obtain an output of θ_2 which satisfies $\bar{\theta}_1[N] = \bar{\theta}_2[N]$.

($< s_1, \ldots, s_k >$) Take any $\theta_1, \bar{\theta}_1 \in Dom^n$. Suppose

$$\forall j \in \{1, \ldots, k\}[< s_1, \ldots, s_j > \in CNet^{n_j:m_j}]$$

and

$$n_0, m_0 = 0.$$

Take any $\theta_2 \in \mathcal{N}(< s_1, \ldots, s_k >)(\theta_1)$. We have

$$\forall j \in\{1, \ldots, k\}[\theta_2 \downarrow< m_{j-1} + 1, \ldots, m_j > \in \mathcal{N}(s_j)(\theta_1 \downarrow< n_{j-1} + 1, \ldots, n_j >)].$$

By induction there exists a $\bar{\theta}_2$ such that

$$\forall j \in\{1, \ldots, k\}[\bar{\theta}_2 \downarrow< m_{j-1} + 1, \ldots, m_j > \in \mathcal{N}(s_j)(\bar{\theta}_1 \downarrow< n_{j-1} + 1, \ldots, n_j >)],$$

$$\forall j \in \{1, \ldots, k\}[d(\theta_2 \downarrow< m_{j-1} + 1, \ldots, m_j >, \bar{\theta}_2 \downarrow< m_{j-1} + 1, \cdots, m_j >) \leq d(\theta_1 \downarrow< m_{j-1} + 1, \ldots, m_j >, \bar{\theta}_{11} \downarrow< m_{j-1} + 1, \ldots, m_j >)]$$

so $d(\theta_2, \bar{\theta}_2) \leq d(\theta_1, \bar{\theta}_1)$.

In order to prove the closedness, take any $\theta \in Dom^n$. Take a Cauchy sequence $(\bar{\theta}_i)_i$ in Dom^n such that

$$\forall i \in \mathbf{N}[\bar{\theta}_i \in \mathcal{N}(< s_1, \ldots, s_k >)(\theta)].$$

We have

$$\forall i \in \mathbf{N} \forall j \in \{1, \ldots, k\}[\bar{\theta}_i \downarrow< m_{j-1} + 1, \ldots, m_j > \in \mathcal{N}(s_j)(\theta \downarrow< n_{j-1} + 1, \ldots, n_j >)].$$

By induction

$$\forall j \in \{1,\ldots,k\}[$$
$$(\lim_{i\to\infty}\bar{\theta}_i)\downarrow< m_{j-1}+1,\ldots,m_j > \in \mathcal{N}(s_j)(\theta\downarrow< n_{j-1}+1,\ldots,n_j >)$$
$$],$$

i.e.

$$\lim_{i\to\infty}\bar{\theta}_i \in \mathcal{N}(< s_1,\ldots,s_k >)(\theta).$$

$(s\{i_1 : j_1,\ldots,i_k : j_k\})$ By induction $\mathcal{N}(s)$ is non distance increasing, so by the ϵ prefixing

$$\lambda\theta.\lambda\bar{\theta}.\mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta,\bar{\theta}))$$

is non distance increasing in θ for fixed $\bar{\theta}$ and contractive in $\bar{\theta}$ for fixed θ. We apply theorem 6.14 to see that $\mathcal{N}(s\{i_1 : j_1,\ldots,i_k : j_k\})$ is non distance increasing and theorem 6.13 to see that $\mathcal{N}(s\{i_1 : j_1,\ldots,i_k : j_k\})(\theta)$ is closed for any θ. □

By theorem 6.12 we can replace the third clause in the definition of $\mathcal{N}$ by

$$\mathcal{N}(s\{i_1 : j_1, \ldots, i_k : j_k\}) =$$
$$\lambda\theta.\{\lim_{i\to\infty}\bar{\theta}_i : \forall i[\bar{\theta}_{i+1} \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta,\bar{\theta}_i)] \wedge$$
$$\bar{\theta}_1 = \tilde{\theta} \wedge (\bar{\theta}_i)_i \textit{ is a Cauchy sequence}$$
$$\}$$

where $\tilde{\theta}$ is arbitrary. This shows that the contain points in the definition of $\mathcal{N}$ can be obtained by iteration starting from an arbitrary $\tilde{\theta}$.

6.4 Property of the metric semantics $\mathcal{N}$

The next lemma will be used when we establish a relation of $\mathcal{N}$ with the operational semantics $\mathcal{O}$. A similar lemma holds for the denotational semantics (in this case it is a direct consequence of the delay lemma: we can take $\bar{\theta}_2 = \bar{\theta}_1$).

Lemma 6.31

$$\forall s \in FNet^{n:m}\forall k\in \mathbf{N}\forall\theta_1,\theta_2 \in Dom^n\forall\bar{\theta}_1 \in Dom^m[$$

$$\bar{\theta}_1 \in \mathcal{N}(s)(\theta_1) \wedge \theta_1 \in \theta_2 \searrow \wedge\theta_1[k] = \theta_2[k]$$
$$\Rightarrow$$
$$\exists\bar{\theta}_2 \in Dom^m[\bar{\theta}_2 \in \mathcal{N}(s)(\theta_2) \wedge \bar{\theta}_1 \in \bar{\theta}_2 \searrow \wedge\bar{\theta}_1[k] = \bar{\theta}_2[k]]$$
$$]$$

Proof

The proof goes by induction on s.

(d) Take any $k \in \{1,2,\ldots\}$, $\theta_1,\theta_2 \in Dom^n$, $\bar{\theta}_1 \in Dom^m$ such that

$$\bar{\theta}_1 \in \mathcal{N}(d)(\theta_1),$$

$$\theta_1 \in \theta_2 \searrow$$

and

$\theta_1[k] = \theta_2[k].$

Because $\bar{\theta}_1 \in \psi_d(\theta_1)$ we can pick a firing sequence $(\chi_i, \sigma_i, \bar{\chi}_i, \bar{\sigma}_i)_i$ for d w.r.t *flatten*(θ) such that

$$\forall i \exists j [\theta_1[j] \geq \chi_1 \cdots \chi_i \wedge \theta_1[j-1] \not\geq \chi_1 \cdots \chi_i \wedge \bar{\theta}_1[j] \geq \bar{\chi}_1 \cdots \bar{\chi}_i \wedge \bar{\theta}_1[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1}],$$

and

$$\textit{flatten}(\bar{\theta}_1) = \bar{\chi}_1 \bar{\chi}_2 \cdots.$$

Define $\bar{\theta}_2 \in Dom^m$ such that

$$\begin{aligned} \forall i \forall j \quad & [\theta_2[j] \geq \chi_1 \cdots \chi_i \wedge \theta_2[j-1] \not\geq \chi_1 \cdots \chi_i \\ & \Rightarrow \\ & \bar{\theta}_2[j] \geq \bar{\chi}_1 \cdots \bar{\chi}_i \wedge \bar{\theta}_2[j-1] \leq \bar{\chi}_1 \cdots \bar{\chi}_{i-1} \end{aligned}$$

$$\textit{flatten}(\bar{\theta}) = \bar{\chi}_1 \bar{\chi}_2 \cdots$$

We have that if $i \leq k$ then

$$\max\{l : \exists j \leq i [\theta_2[j] \geq \chi_1 \cdots \chi_l \wedge \theta_2[j-1] \not\geq \chi_1 \cdots \chi_l]\} =$$

$$\max\{l : \exists j \leq i [\theta_1[j] \geq \chi_1 \cdots \chi_l \wedge \theta_1[j-1] \not\geq \chi_1 \cdots \chi_l]\}$$

and if $i > k$ then

$$\max\{l : \exists j \leq i [\theta_2[j] \geq \chi_1 \cdots \chi_l \wedge \theta_2[j-1] \not\geq \chi_1 \cdots \chi_l]\} \geq$$

$$\max\{l : \exists j \leq i [\theta_1[j] \geq \chi_1 \cdots \chi_l \wedge \theta_1[j-1] \not\geq \chi_1 \cdots \chi_l]\}$$

Note that for all i

$$\bar{\theta}_2[i] = \bar{\chi}_1 \cdots \bar{\chi}_{\max\{l : \exists j \leq i [\theta_2[j] \geq \chi_1 \cdots \chi_l \wedge \theta_2[j-1] \not\geq \chi_1 \cdots \chi_l]\}}$$

$$\bar{\theta}_1[i] = \bar{\chi}_1 \cdots \bar{\chi}_{\max\{l : \exists j \leq i [\theta_1[j] \geq \chi_1 \cdots \chi_l \wedge \theta_1[j-1] \not\geq \chi_1 \cdots \chi_l]\}}$$

so $\bar{\theta}_1[k] = \bar{\theta}_2[k]$ and $\forall i [\bar{\theta}_1[i] \leq \bar{\theta}_2[i]]$. Moreover, *flatten*$(\bar{\theta}_2) = \bar{\chi}_1 \bar{\chi}_2 \cdots =$ *flatten*$(\bar{\theta}_1)$, so $\bar{\theta}_1 \in \bar{\theta}_2 \searrow$.

$(< s_1, \ldots, s_l >)$ Take any $k \in \{1, 2, \ldots\}$, $\theta_1, \theta_2 \in Dom^n$, $\bar{\theta}_1 \in Dom^m$ and suppose

$$\bar{\theta}_1 \in \mathcal{N}(< s_1, \ldots, s_l >)(\theta_1),$$

$$\theta_1 \in \theta_2 \searrow$$

and

$$\theta_1[k] = \theta_2[k].$$

Suppose, for $i = 1, \ldots, l$, $< s_1, \ldots, s_i > \in CNet^{n_i:m_i}$. Put $n_0 = m_0 = 0$. Define

$$\theta_{1i} = \theta_1 \downarrow < n_{i-1} + 1, \ldots, n_i >,$$

$$\theta_{2i} = \theta_2 \downarrow < n_{i-1} + 1, \ldots, n_i >$$

and

$$\bar{\theta}_{1i} = \bar{\theta}_1 \downarrow < m_{i-1} + 1, \ldots, m_i > .$$

We have

$$\forall i[\bar{\theta}_{1i} \in \mathcal{N}(s_i)(\theta_{1i}) \wedge \theta_{1i} \in \theta_{2i} \searrow \wedge \theta_{1i}[k] = \theta_{2i}[k]]$$

so by induction

$$\forall i \exists \bar{\theta}_{2i}[\bar{\theta}_{2i} \in \mathcal{N}(s_i)(\theta_{2i}) \wedge \bar{\theta}_{1i} \in \bar{\theta}_{2i} \searrow \wedge \bar{\theta}_{1i}[k] = \bar{\theta}_{2i}[k]].$$

Take such $\bar{\theta}_{2i}$ and define $\bar{\theta}_2$ such that

$$\forall i[\bar{\theta}_2 \downarrow < m_{i-1} + 1, \ldots, m_i >= \bar{\theta}_{2i}].$$

We have

$$\bar{\theta}_2 \in \mathcal{N}(< s_1, \ldots, s_l >)(\theta_2),$$

$$\bar{\theta}_1 \in \bar{\theta}_2 \searrow$$

and

$$\bar{\theta}_1[k] = \bar{\theta}_2[k].$$

$(s\{i_1 : j_1, \ldots, i_l : j_l\})$ Take any $k \in \{1, 2, \ldots\}$, $\theta_1, \theta_2 \in Dom^n$, $\bar{\theta}_1 \in Dom^m$ and suppose

$$\bar{\theta}_1 \in \mathcal{N}(s\{i_1 : j_1, \ldots, i_l : j_l\})(\theta_1),$$

$$\theta_1 \in \theta_2 \searrow$$

and

$$\theta_1[k] = \theta_2[k].$$

By definition of $\mathcal{N}$ there exists a $\hat{\theta}_1 \in Dom^{m+l}$ such that

$$\hat{\theta}_1 \uparrow \{j_1, \ldots, j_l\} = \bar{\theta}_1$$

and

$$\hat{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_1, \hat{\theta}_1)).$$

We are going to construct a $\hat{\theta}_2$ such that

$$\hat{\theta}_2 \searrow \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_2)),$$

$$\hat{\theta}_1 \in \hat{\theta}_2 \searrow$$

and

$$\hat{\theta}_1[k] = \hat{\theta}_2[k].$$

Now if we take $\bar{\theta}_2 = \hat{\theta}_2 \uparrow \{j_1, \ldots, j_k\}$ we are done. The finite-word matrix $\hat{\theta}_2$ will be the limit of a sequence $(\tilde{\theta}_i)_i$, which is constructed in the following way.

First observe that

1. $\hat{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_1, \hat{\theta}_1))$,
2. $\theta_1 \in \theta_2 \searrow \Rightarrow \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_1, \hat{\theta}_1) \in \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1) \searrow$,
3. $\theta_1[k] = \theta_2[k]$
   $\Rightarrow$
   $\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_1, \hat{\theta}_1)[k] = \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1)[k]$

so we can apply the induction hypothesis and find a $\tilde{\theta}_1$ such that

$$\tilde{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1)),$$

$$\hat{\theta}_1 \in \tilde{\theta}_1 \searrow,$$

$$\hat{\theta}_1[k] = \tilde{\theta}_1[k].$$

Note that

1. $\tilde{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1))$,
2. $\hat{\theta}_1 \in \tilde{\theta}_1 \searrow \Rightarrow \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1) \in \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1) \searrow$,
3. $\hat{\theta}_1[k] = \tilde{\theta}_1[k]$
   $\Rightarrow$
   $\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \hat{\theta}_1)[k+1] = \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1)[k+1]$

where in the last line we have $k+1$ due to the ϵ prefixing of the $\mathcal{L}$ function. By induction we have that we can find a $\tilde{\theta}_2$ such that

$$\tilde{\theta}_2 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1)),$$

$$\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow,$$

$$\tilde{\theta}_1[k+1] = \tilde{\theta}_2[k+1].$$

We continue in this way: note that

1. $\tilde{\theta}_2 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1))$,

2. $\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow \Rightarrow \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1) \in \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_2) \searrow$,
3. $\tilde{\theta}_1[k+1] = \tilde{\theta}_2[k+1]$
$\Rightarrow$
$\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_1)[k+2] = \mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_2)[k+2]$.

By induction we have that there exists a $\tilde{\theta}_3$ such that

$$\tilde{\theta}_3 \in \mathcal{N}(s_1)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta_2, \tilde{\theta}_2)),$$

$$\tilde{\theta}_2 \in \tilde{\theta}_3 \searrow,$$

$$\tilde{\theta}_2[k+2] = \tilde{\theta}_3[k+2].$$

We continue this process and obtain for all i

$$\tilde{\theta}_{i+1} \in \mathcal{N}(s_1)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \tilde{\theta}_i)),$$

$$\tilde{\theta}_i \in \tilde{\theta}_{i+1} \searrow,$$

$$\tilde{\theta}_i[k+i] = \tilde{\theta}_{i+1}[k+i].$$

Hence $(\tilde{\theta}_i)_i$ is a Cauchy sequence. Because $\mathcal{N}(s)$ is non distance increasing (and hence continuous) and by Hahn's theorem we have

$$(\lim_{i \to \infty} \tilde{\theta}_i) \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_l : j_l\}, \theta_2, \lim_{i \to \infty} \tilde{\theta}_i)).$$

We have

$$\hat{\theta}_1 \in \tilde{\theta}_1 \searrow,$$

$$\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow \Rightarrow \hat{\theta}_1 \in \tilde{\theta}_2 \searrow,$$

$$\tilde{\theta}_2 \in \tilde{\theta}_3 \searrow \Rightarrow \hat{\theta}_1 \in \tilde{\theta}_3 \searrow,$$

$$\ldots$$

so by lemma 6.21 we have $\hat{\theta}_1 \in (\lim_{i \to \infty} \tilde{\theta}_i) \searrow$. Moreover

$$\hat{\theta}_1[k] = \tilde{\theta}_1[k] \wedge \tilde{\theta}_1[k+1] = \tilde{\theta}_2[k+1] \wedge \ldots \wedge \tilde{\theta}_i[k+i] = \tilde{\theta}_{i+1}[k+i] \wedge \ldots$$

$$\Rightarrow$$

$$\hat{\theta}_1[k] = (\lim_{i \to \infty} \tilde{\theta}_i)[k]$$

so if we take $\bar{\theta}_2 = (\lim_{i \to \infty} \tilde{\theta}_i) \uparrow \{j_1, \ldots, j_l\}$ we have

$$\bar{\theta}_2 \in \mathcal{N}(s\{i_1 : j_1, \ldots, i_l : j_l\})(\theta_2),$$

$$\bar{\theta}_2[k] = \bar{\theta}_1[k],$$

$$\bar{\theta}_1 \in \bar{\theta}_2 \searrow.$$

□

6.5 The intermediate semantics $\mathcal{T}$

We define an intermediate semantics $\mathcal{T}$, which is used in the proof of the relation between $\mathcal{O}$ and $\mathcal{N}$.

Definition 6.32 (Semantics $\mathcal{T}$) *Define a semantics $\mathcal{T} : CNet^{n:m} \rightarrow Dom^{n:m}$ by*

1. $\mathcal{T}(d) = \phi_d$,
2. $\mathcal{T}(< s_1, \ldots, s_k >) = \mathcal{T}(s_1) :: \cdots :: \mathcal{T}(s_k)$,
3. $\mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\}) = \lambda\theta.(CP($
$$\lambda\bar{\theta}.\mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}))$$
$$)$$
$$)\tilde{\uparrow}\{j_1, \ldots, j_k\}$$

Note the differences with $\mathcal{N}$: $\mathcal{T}$ is defined on $CNet^{n:m}$ and $\mathcal{N}$ is defined only on $FNet^{n:m}$. Also $\mathcal{T}(d) = \phi_d$ whereas $\mathcal{N}(d) = \psi_d$. However, in the sequel we often restrict $\mathcal{T}$ to $FNet^{n:m}$. Intuitively, $\mathcal{T}$ does generate delays and $\mathcal{N}$ does not. We prove two properties of this semantics $\mathcal{T}$.

Lemma 6.33 (delay lemma)

$$\forall s \in CNet^{n:m} \forall \theta, \theta' \in Dom^n \forall \bar{\theta}, \bar{\theta}' \in Dom^m [$$

$$\bar{\theta} \in \mathcal{T}(s)(\theta) \wedge \bar{\theta}' \in \bar{\theta} \searrow \wedge \theta \in \theta' \searrow$$
$$\Rightarrow$$
$$\bar{\theta}' \in \mathcal{T}(s)(\theta')$$
$$].$$

Proof

The proof goes by induction on s:

(d) See lemma 4.7.

$(< s_1, \ldots, s_k >)$ Follows the same structure of the argument in lemma 4.7.

$(s\{i_1 : j_1, \ldots, i_k : j_k\})$ Again, this case is a simple version of the analogous case of the proof of lemma 4.7. The proof is here included for convenience. Take any $\theta \in Dom^n$, $\bar{\theta}, \bar{\theta}' \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$$

and

$$\bar{\theta}' \in \bar{\theta} \searrow .$$

This implies that we can take a $\tilde{\theta} \in Dom^{m+k}$ such that

$$\bar{\theta} = \tilde{\theta} \uparrow \{j_1, \ldots, j_k\}$$

and

$$\tilde{\theta} \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta})).$$

Define $\tilde{\theta}' \in Dom^{m+k}$ such that

$$\tilde{\theta}' \downarrow < j_1, \ldots, j_k >= \tilde{\theta} \downarrow < j_1, \ldots, j_k >$$

and

$$\tilde{\theta}' \uparrow \{j_1, \ldots, j_k\} = \bar{\theta}'.$$

We have

$$\tilde{\theta}' \in \tilde{\theta} \searrow$$

and

$$\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}) \in \mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta', \tilde{\theta}') \searrow$$

so by induction

$$\tilde{\theta}' \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta', \tilde{\theta}'))$$

i.e.

$$\bar{\theta}' = (\tilde{\theta}' \uparrow \{j_1, \ldots, j_k\}) \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta').$$

□

Lemma 6.34

$$\begin{aligned} &\forall s \in CNet^{n:m} \forall \theta \in Dom^n \forall \bar{\theta} \in Dom^m [\\ &\qquad \bar{\theta} \in \mathcal{T}(s)(\theta)\\ &\qquad \Leftrightarrow\\ &\qquad \forall (\alpha_i)_i [(\bar{\theta} \rhd (\alpha_i)_i) \in \mathcal{T}(s)(\theta \rhd (\alpha_i)_i)]\\ &\qquad] \end{aligned}$$

Proof

For ($\Leftarrow$) take $(\alpha_i)_i = (1)_i$. For ($\Rightarrow$) we use induction on s.

(d) See lemma 4.11.

($< s_1, \ldots, s_k >$) See lemma 4.11.

($s\{i_1 : j_1, \ldots, i_k : j_k\}$) (Cf. the comment in the proof of lemma 6.33.) Take any $\theta \in Dom^n, \bar{\theta} \in Dom^m$ such that $\bar{\theta} \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$. Take any sequence of integers $(\alpha_i)_i$. By definition of $\mathcal{T}$ we can find a $\tilde{\theta} \in Dom^{m+k}$ such that

$$\tilde{\theta} \uparrow \{j_1, \ldots, j_k\} = \bar{\theta}$$

and

$$\tilde{\theta} \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta})).$$

By induction we have

$$(\tilde{\theta} \rhd (\alpha_i)_i) \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}) \rhd (\alpha_i)_i).$$

Hence

$$(\tilde{\theta} \rhd (\alpha_i)_i) \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta \rhd (\alpha_i)_i, \tilde{\theta} \rhd (\alpha_1 + \alpha_2 - 1, \alpha_3, \ldots))).$$

Define $\tilde{\theta}' \in Dom^{m+k}$ such that:

$$\tilde{\theta}' \uparrow \{j_1, \ldots, j_k\} = (\tilde{\theta} \rhd (\alpha_i)_i) \uparrow \{j_1, \ldots, j_k\}(= \bar{\theta} \rhd (\alpha_i)_i)$$

and

$$\tilde{\theta}' \downarrow < j_1, \ldots, j_k >= \epsilon \Box (\tilde{\theta} \rhd (\alpha_1 + \alpha_2 - 1, \alpha_3, \ldots)) \downarrow < j_1, \ldots, j_k > .$$

We have

$$\epsilon \Box (\tilde{\theta} \rhd (\alpha_1 + \alpha_2 - 1, \alpha_3, \ldots)) \in (\tilde{\theta} \rhd (\alpha_i)_i) \searrow$$

Hence

$$\tilde{\theta}' \in \tilde{\theta} \rhd (\alpha_i)_i \searrow$$

$$\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta \rhd (\alpha_i)_i, \tilde{\theta} \rhd (\alpha_1 + \alpha_2 - 1, \alpha_3, \ldots)) =$$

$$\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta \rhd (\alpha_i)_i, \tilde{\theta}')$$

so by lemma 6.33

$$\tilde{\theta}' \in \mathcal{T}(s)((\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta \rhd (\alpha_i)_i, \tilde{\theta}')$$

and because

$$\tilde{\theta}' \uparrow \{j_1, \ldots, j_k\} = (\tilde{\theta} \rhd (\alpha_i)_i) \uparrow \{j_1, \ldots, j_k\} = \bar{\theta} \rhd (\alpha_i)_i$$

we have

$$(\bar{\theta} \rhd (\alpha_i)_i) \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta \rhd (\alpha_i)_i).$$

□

6.6 Relation between the metric semantics $\mathcal{N}$ and the operational semantics $\mathcal{O}$

In this subsection we show that $\mathcal{T} = delay \circ \mathcal{N}$ (note that we restricted $\mathcal{T}$ to $FNet^{n:m}$) and that $\mathcal{D} = close \circ \mathcal{T}$. Because $\mathcal{O} = abstr \circ \mathcal{D}$ and because $abstr \circ close \circ delay = abstr$ we can derive $\mathcal{O} = abstr \circ \mathcal{N}$. This shows the correctness of $\mathcal{N}$. We start with

Theorem 6.35 $\mathcal{T} = delay \circ \mathcal{N}$

Proof

We prove by induction on $s \in FNet^{n:m}$ that $\mathcal{T}(s) = (delay \circ \mathcal{N})(s)$.

(d) See lemma 6.25.

$(< s_1, \ldots, s_k >)$ Take any $\theta \in Dom^n$ and $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{T}(< s_1, \ldots, s_k >)(\theta).$$

Suppose, for $i = 1, \ldots, k$, that $< s_1, \ldots, s_i > \in FNet^{n_i:m_i}$. Put $n_0 = m_0 = 0$. Define, for $i = 1, \ldots, k$

$$\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >$$

and

$$\bar{\theta}_i = \bar{\theta} \downarrow < m_{i-1} + 1, \ldots, m_i > .$$

We have

$$\forall i[\bar{\theta}_i \in \mathcal{T}(s_i)(\theta_i)]$$

so by induction

$$\forall i[\bar{\theta}_i \in (delay \circ \mathcal{N})(s_i)(\theta_i)].$$

By definition of *delay* we can find a $\tilde{\theta}_i$ (i = 1 , ..., k) such that

$$\bar{\theta}_i \in \tilde{\theta}_i \searrow$$

and

$$\tilde{\theta}_i \in \mathcal{N}(s_i)(\theta_i).$$

Take $\tilde{\theta} \in Dom^m$ such that

$$\forall i \in \{1, \ldots, k\}[\tilde{\theta} \downarrow < m_{i-1} + 1, \ldots, m_i >= \tilde{\theta}_i]$$

so

$$\bar{\theta} \in \tilde{\theta} \searrow$$

and

$$\tilde{\theta} \in \mathcal{N}(< s_1, \ldots, s_k >)(\theta)$$

i.e.

$$\bar{\theta} \in (delay \circ \mathcal{N})(< s_1, \ldots, s_k >)(\theta).$$

For the other inclusion, take any $\theta \in Dom^n$ and $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in (delay \text{ o } \mathcal{N})(< s_1, \ldots, s_k >)(\theta).$$

By definition of *delay* we can find a $\tilde{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \tilde{\theta} \searrow$$

and

$$\tilde{\theta} \in \mathcal{N}(< s_1, \ldots, s_k >)(\theta).$$

Suppose, for $i = 1, \ldots, k$, that $< s_1, \ldots, s_i > \in FNet^{n_i:m_i}$. Put $n_0 = m_0 = 0$. Define, for $i = 1, \ldots, k$

$$\tilde{\theta}_i = \tilde{\theta} \downarrow < m_{i-1} + 1, \ldots, m_i >,$$

$$\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >$$

and

$$\bar{\theta}_i = \bar{\theta} \downarrow < m_{i-1} + 1, \ldots, m_i > .$$

We have

$$\forall i \in \{1, \ldots, k\}[\tilde{\theta}_i \in \mathcal{N}(s_i)(\theta_i)]$$

so

$$\forall i \in \{1, \ldots, k\}[\bar{\theta}_i \in (delay \text{ o } \mathcal{N})(s_i)(\theta_i)].$$

By induction we have

$$\forall i \in \{1, \ldots, k\}[\tilde{\theta}_i \in \mathcal{T}(s_i)(\theta_i)]$$

so

$$\tilde{\theta} \in \mathcal{T}(< s_1, \ldots, s_k >)(\theta)$$

and because $\bar{\theta} \in \tilde{\theta} \searrow$ we have by lemma 6.33

$$\bar{\theta} \in \mathcal{T}(< s_1, \ldots, s_k >)(\theta).$$

$(s\{i_1 : j_1, \ldots, i_k : j_k\})$ Take any $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in delay(\mathcal{N}(s\{i_1 : j_1, \ldots, i_k : j_k\}))(\theta).$$

This implies by definition of *delay* and $\mathcal{N}$ that we can take a $\tilde{\theta} \in Dom^{m+k}$ such that

$$\bar{\theta} \in (\tilde{\theta} \uparrow \{j_1, \ldots, j_k\}) \searrow$$

and

$$\tilde{\theta} \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta})).$$

By definition of *delay* we have

$$\tilde{\theta} \in delay(\mathcal{N})(s))(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}))$$

and by induction

$$\tilde{\theta} \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}))$$

i.e.

$$\tilde{\theta} \uparrow \{j_1, \ldots, j_k\} \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$$

and by lemma 6.33

$$\bar{\theta} \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

For the other inclusion, take any $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{T}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

We can pick a $\bar{\theta}' \in Dom^{m+k}$ such that

$$\bar{\theta}' \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}'))$$

and

$$\bar{\theta}' \uparrow \{j_1, \ldots, j_k\} = \bar{\theta}.$$

By induction we have

$$\bar{\theta}' \in delay(\mathcal{N})(s))(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}')).$$

By definition of *delay* we can take a $\tilde{\theta}_1 \in Dom^{m+k}$ such that

$$\bar{\theta}' \in \tilde{\theta}_1 \searrow$$

and

$$\tilde{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}')).$$

Because

1. $\bar{\theta}' \in \tilde{\theta}_1 \searrow$
2. $(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}'))[1] = (\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_1))[1]$
3. $\tilde{\theta}_1 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}'))$

by lemma 6.31 we can find $\tilde{\theta}_2 \in Dom^{m+k}$ such that

$$\tilde{\theta}_2 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_1)),$$

$$\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow$$

and

$$\tilde{\theta}_2[1] = \tilde{\theta}_1[1].$$

Because

1. $\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow$
2. $(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_1))[2] = (\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_2))[2]$
3. $\tilde{\theta}_2 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_1))$

by lemma 6.31 we can find $\tilde{\theta}_3 \in Dom^{m+k}$ such that

$$\tilde{\theta}_3 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_2)),$$

$$\tilde{\theta}_2 \in \tilde{\theta}_3 \searrow$$

and

$$\tilde{\theta}_3[2] = \tilde{\theta}_2[2].$$

We continue in this way: because

1. $\tilde{\theta}_2 \in \tilde{\theta}_3 \searrow$
2. $(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_2))[3] = (\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_3))[3]$
3. $\tilde{\theta}_3 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_2))$

by lemma 6.31 we can find $\tilde{\theta}_4 \in Dom^{m+k}$ such that

$$\tilde{\theta}_4 \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \tilde{\theta}_3)),$$

$$\tilde{\theta}_3 \in \tilde{\theta}_4 \searrow$$

and

$$\tilde{\theta}_4[3] = \tilde{\theta}_3[3]].$$

In this way we obtain a Cauchy sequence $(\tilde{\theta}_i)_i$. By the continuity of $\mathcal{N}$ and $\mathcal{L}$ and the theorem of Hahn we have

$$(\lim_{i\to\infty} \tilde{\theta}_i) \in \mathcal{N}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, (\lim_{i\to\infty} \tilde{\theta}_i))).$$

We also have

$$\bar{\theta}' \in \tilde{\theta}_1 \searrow,$$

$$\tilde{\theta}_1 \in \tilde{\theta}_2 \searrow \Rightarrow \bar{\theta}' \in \tilde{\theta}_2 \searrow,$$

$$\tilde{\theta}_2 \in \tilde{\theta}_3 \searrow \Rightarrow \bar{\theta}' \in \tilde{\theta}_3 \searrow,$$

$$\ldots$$

so

$$\forall i[\bar{\theta}' \in \tilde{\theta}_i \searrow]$$

so by lemma 6.21

$$\bar{\theta}' \in \lim_{i \to \infty} \tilde{\theta}_i \searrow .$$

By definition of $\mathcal{N}$

$$(\lim_{i \to \infty} \tilde{\theta}_i) \uparrow \{j_1, \ldots, j_k\} \in \mathcal{N}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$$

and by definition of *delay*

$$\bar{\theta}' \uparrow \{j_1, \ldots, j_k\} \in (delay \text{ o } \mathcal{N})(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta)$$

i.e.

$$\bar{\theta} \in (delay \text{ o } \mathcal{N})(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

□

We have now established a relation between $\mathcal{N}$ and $\mathcal{T}$. The next step is the relation between $\mathcal{T}$ and $\mathcal{D}$. We prove below that they are related with *close*:

Theorem 6.36 $\mathcal{D} = close \text{ o } \mathcal{T}$

Proof

We prove this by induction on s. Suppose $s \in CNet^{n:m}$.

(d) See lemma 4.8.

$(< s_1, \ldots, s_k >)$ Take any $\theta \in Dom^n, \bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{D}(< s_1, \ldots, s_k >)(\theta).$$

Assume, for $i \in \{1, \ldots, k\}$, that $< s_1, \ldots, s_i > \in CNet^{n_i:m_i}$. Put $n_0 = m_0 = 0$. Define for $i \in \{1, \ldots, k\}$

$$\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >$$

and

$$\bar{\theta}_i = \bar{\theta} \downarrow < m_{i-1}+1, \ldots, m_i > .$$

We have

$$\forall i[\bar{\theta}_i \in \mathcal{D}(s_i)(\theta_i)]$$

so by induction

$$\forall i[\bar{\theta}_i \in (close \circ \mathcal{T})(s_i)(\theta_i)].$$

By definition of *close* we can find, for $i \in \{1, \ldots, k\}$, $\bar{\theta}'_i \in Dom^{n_i - n_{i-1}}$, $\theta'_i \in Dom^{m_i - m_{i-1}}$ and sequence of integers $(\alpha_{ij})_j$ such that

$$\bar{\theta}'_i \in \mathcal{T}(s_i)(\theta'_i),$$

$$\theta'_i \rightsquigarrow (\alpha_{ij})_j = \theta_i$$

and

$$\bar{\theta}'_i \rightsquigarrow (\alpha_{ij})_j = \bar{\theta}_i.$$

Define the sequence of integers $(\beta_j)_j$ such that

$$\forall j \forall i \in \{1, \ldots, k\}[\beta_j = \max\{\alpha_{1j}, \ldots, \alpha_{kj}\}]$$

and $(\xi_{ij})_j$, for $i \in \{1, \ldots, n\}$, such that

$$\forall j[\xi_{ij} = \beta_j - \alpha_{ij} + 1].$$

By lemma 6.34 we have

$$\forall i[(\bar{\theta}'_i \rhd (\xi_{ij})_j) \in \mathcal{T}(s_i)(\theta'_i \rhd (\xi_{ij})_j)].$$

We also have for all $i \in \{1, \ldots, k\}$

$$(\theta'_i \rhd (\xi_{ij})_j) \rightsquigarrow (\beta_j)_j = \theta_i$$

and

$$(\bar{\theta}'_i \rhd (\xi_{ij})_j) \rightsquigarrow (\beta_j)_j = \bar{\theta}_i.$$

Take $\theta' \in Dom^n, \bar{\theta}' \in Dom^m$ such that, for all $i \in \{1, \ldots, k\}$,

$$\theta' \downarrow < n_{i-1}+1, \ldots, n_i >= \theta'_i \rhd (\xi_{ij})_j$$

and

$$\bar{\theta}' \downarrow < m_{i-1}+1, \ldots, m_i >= \bar{\theta}'_i \rhd (\xi_{ij})_j.$$

We have

$$\bar\theta' \in \mathcal{T}(< s_1, \ldots, s_n >)(\theta'),$$

$$\theta' \rightsquigarrow (\beta_j)_j = \theta$$

and

$$\bar\theta' \rightsquigarrow (\beta_j)_j = \bar\theta.$$

So

$$\bar\theta \in close(\mathcal{T}(< s_1, \ldots, s_n >))(\theta).$$

For the other inclusion, take any $\theta \in Dom^n, \bar\theta \in Dom^m$ such that

$$\bar\theta \in close(\mathcal{T}(< s_1, \ldots, s_k >))(\theta).$$

By the definition of the *close* operator we can find $\theta_1 \in Dom^n, \theta_2 \in Dom^m$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta_2 \in \mathcal{T}(< s_1, \ldots, s_k >)(\theta_1),$$

$$\theta_1 \rightsquigarrow (\alpha_i)_i = \theta.$$

and

$$\theta_2 \rightsquigarrow (\alpha_i)_i = \bar\theta$$

Assume, for $i \in \{1, \ldots, k\}$, $< s_1, \ldots, s_i > \in CNet^{n_i : m_i}$. Put $n_0 = m_0 = 0$. Define for $i \in \{1, \ldots, k\}$

$$\theta_i = \theta \downarrow < n_{i-1} + 1, \ldots, n_i >,$$

$$\theta_{1i} = \theta_1 \downarrow < n_{i-1} + 1, \ldots, n_i >,$$

$$\bar\theta_i = \bar\theta \downarrow < m_{i-1} + 1, \ldots, m_i >$$

and

$$\theta_{2i} = \theta_2 \downarrow < m_{i-1} + 1, \ldots, m_i > .$$

By definition of $\mathcal{T}$ we have

$$\forall i \in \{1, \ldots, k\}[\theta_{2i} \in \mathcal{T}(s_i)(\theta_{1i})].$$

By definition of *close* and because

$$\forall i \in \{1, \ldots, k\}[\theta_{1i} \rightsquigarrow (\alpha_i)_i = \theta_i \wedge \theta_{2i} \rightsquigarrow (\alpha_i)_i = \bar\theta_i]$$

we have

$$\forall i \in \{1,\ldots,k\}[\bar{\theta}_i \in (close \circ \mathcal{T})(s_i)(\theta_i)]$$

so by induction

$$\forall i \in \{1,\ldots,k\}[\bar{\theta}_i \in \mathcal{D}(s_i)(\theta_i)]$$

i.e.

$$\bar{\theta} \in \mathcal{D}(< s_1,\ldots,s_n >)(\theta).$$

$(s\{i_1 : j_1,\ldots,i_k : j_k\})$ Take any $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1,\ldots,i_k : j_k\})(\theta).$$

By the derivation directly after the definition of $\mathcal{N}$ we have

$$\bar{\theta} \in close(\lambda\theta.\ CP(\lambda\bar{\theta}.\mathcal{D}(s)\mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta,\bar{\theta}) \tilde{\uparrow}\{j_1,\ldots,j_k\})(\theta).$$

By induction we have:

$$\bar{\theta} \in close(\lambda\theta.\ CP(\lambda\bar{\theta}.close(\mathcal{T}(s))\mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta,\bar{\theta}) \tilde{\uparrow}\{j_1,\ldots,j_k\})(\theta).$$

By definition of *close* and of $\tilde{\uparrow}$ we can find $\theta_2 \in Dom^{m+k}$, $\theta_1 \in Dom^n$ and a sequence of integers $(\alpha_i)_i$ such that

$$\theta_2 \in close(\mathcal{T}(s))(\mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta_1,\theta_2),$$

$$(\theta_2 \uparrow \{j_1,\ldots,j_k\}) \rightsquigarrow (\alpha_i)_i = \bar{\theta}$$

and

$$\theta_1 \rightsquigarrow (\alpha_i)_i = \theta.$$

By definition of *close* we can find $\theta_4 \in Dom^{m+k}$, $\theta_3 \in Dom^n$ and a sequence of integers $(\beta_i)_i$ such that

$$\theta_4 \in \mathcal{T}(s)(\theta_3),$$

$$\theta_4 \rightsquigarrow (\beta_i)_i = \theta_2$$

and

$$\theta_3 \rightsquigarrow (\beta_i)_i = \mathcal{L}(\{i_1 : j_1,\ldots,i_k : j_k\},\theta_1,\theta_2).$$

Define $\theta_4' \in Dom^{m+k}$ such that

$$\theta_4' \uparrow \{j_1, \ldots, j_k\} = \theta_4 \uparrow \{j_1, \ldots, j_k\}$$

and

$$\theta_4' \downarrow < j_1, \ldots, j_k >= \tilde{\theta}$$

where $\tilde{\theta} \in Dom^k$ is such that $\theta_3 \downarrow < i_1, \ldots, i_k >= \epsilon \Box \tilde{\theta}$. We have (because $\theta_4 \leadsto (\beta_i)_i = \theta_2$ and $(\theta_4' \leadsto (\beta_i)_i) \downarrow < j_1, \ldots, j_k >= (\epsilon \Box \theta_2) \downarrow < j_1, \ldots, j_k >$)

$$\theta_4' \in \theta_4 \searrow .$$

So by lemma 6.33

$$\theta_4' \in \mathcal{T}(s)(\theta_3) = \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta_3, \theta_4')$$

so

$$\begin{aligned}(\theta_4' \uparrow \{j_1, \ldots, j_k\}) \in \lambda\theta.CP(\\ \lambda\bar{\theta}.\mathcal{T}(s)\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}) \\ )\tilde{\uparrow}\{j_1, \ldots, j_k\})(\theta_3).\end{aligned}$$

Define a sequence of integers $(\gamma_i)_i$ such that

$$\gamma_1 = \beta_1 + \cdots + \beta_{\alpha_1}$$

$$\gamma_2 = \beta_{\alpha_1+1} + \cdots + \beta_{\alpha_1+\alpha_2}$$

$$\gamma_3 = \beta_{\alpha_1+\alpha_2+1} + \cdots + \beta_{\alpha_1+\alpha_2+\alpha_3}$$

$$\ldots$$

Because

$$(\theta_4' \uparrow \{j_1, \ldots, j_k\}) \leadsto (\gamma_i)_i = \bar{\theta}$$

and

$$\theta_3 \leadsto (\gamma_i)_i = \theta$$

we have

$$\begin{aligned}\bar{\theta} \in (close(\lambda\theta.CP(\\ \lambda\bar{\theta}.\mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}))\tilde{\uparrow}\{j_1, \ldots, j_k\}))) \\ (\theta)\end{aligned}$$

i.e.

$$\bar{\theta} \in (close \circ \mathcal{T})(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

For the other inclusion, take any $\theta \in Dom^n$, $\bar{\theta} \in Dom^m$ such that

$$\bar{\theta} \in (close \circ \mathcal{T})(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

By definition of $\mathcal{T}$ we have

$$\bar{\theta} \in (close(\lambda\theta.CP(\lambda\bar{\theta}.\mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}))\tilde{\uparrow}\{j_1, \ldots, j_k\}))) (\theta).$$

By definition of *close* we can find $\theta_2 \in Dom^{m+k}$, $\theta_1 \in Dom^n$ and a sequence of integers $(\alpha_i)_i$ such that

$$(\theta_2 \uparrow \{j_1, \ldots, j_k\}) \leadsto (\alpha_i)_i = \bar{\theta},$$

$$\theta_1 \leadsto (\alpha_i)_i = \theta$$

and

$$\theta_2 \in \mathcal{T}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta_1, \theta_2)).$$

By definition of *close* we also have (take the sequence of integers $(1)_i$)

$$\theta_2 \in (close \circ \mathcal{T})(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta_1, \theta_2)).$$

and by induction

$$\theta_2 \in \mathcal{D}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta_1, \theta_2))$$

and hence

$$\bar{\theta} \in close(\lambda\theta.CP(\lambda\bar{\theta}.\mathcal{D}(s)(\mathcal{L}(\{i_1 : j_1, \ldots, i_k : j_k\}, \theta, \bar{\theta}))\tilde{\uparrow}\{j_1, \ldots, j_k\} (\theta)$$

i.e.

$$\bar{\theta} \in \mathcal{D}(s\{i_1 : j_1, \ldots, i_k : j_k\})(\theta).$$

□

Taking all the results of this subsection together we obtain:

Theorem 6.37 $\mathcal{O} = abstr \circ \mathcal{N}$

Proof

$$\mathcal{O} = abstr \circ \mathcal{D},$$

$$\mathcal{D} = close \circ \mathcal{T},$$

$$\mathcal{T} = delay \circ \mathcal{N},$$

$$abstr \circ close \circ delay = abstr.$$

□

7 Acknowledgements and references

7.1 Acknowledgements

First of all, we would like to thank Jaco de Bakker, for the support and the guidance during the writing of this paper. A paper of him ([de Bakker et al 1985]) served as a starting point for our investigation. Some of the notation is borrowed from that paper, and an equivalent of the normal form theorem (in an order theoretic context) can be found there. The careful reading of the various drafts of this paper and the suggestions to approve them are gratefully acknowledged. We also thank Jan Rutten who put a lot of effort in studying previous versions of this paper. Numerous sessions with Jaco de Bakker en Jan Rutten were spent discussing drafts of the present paper. Next we would like to thank professor Rozenberg, who suggested the automata-theoretic interpretation of our model, and the other members of LPC discussion group: professor de Roever, Ron Koymans and Iko Keesmaat for their patience and useful suggestions. We are grateful as well to the concurrency group on the CWI (Jaco de Bakker, Frank de Boer, Jan Rutten, John-Jules Meyer, Arie de Bruin and Erik de Vink) for their discussions on the contents of this paper. We would also like to thank Tom Verhoeff of Technological University Eindhoven who suggested a simplification of the proof of the full abstractness of the denotational model. We acknowledge discussions with Eugene Stark and Prakash Panangaden about the model and the full abstraction proof. Eugene Stark pointed out that in an earlier version of this paper implicitly use was made of a fair merge node. We like to thank as well Bengt Jonnson for the discussions on the semantics of data flow.

7.2 References

References

[Abramsky 1983] Abramsky, S., *On Semantic Foundations for Applicative Multiprogramming*, in Proceedings Colloquium Automata, Languages and Programming (Diaz ed.), Lecture Notes in Computer Science 154, Springer-Verlag, 1983, pp. 1-14.

[Abramsky 1984] Abramsky, S., *Reasoning About Concurrent Systems*, in Distributed Computing (Chambers, Duce, Jones eds.), 1984.

[Arnold 1981] Arnold, A., *Semantique des Processus Communicants*, in RAIRO 15, 2, 1981, pp. 103-109.

[Back & Mannila 1982] Back, R.J., Mannila, N., *A Refinement of Kahn's Semantics to Handle Nondeterminism and Communication*, in Proceedings ACM Symposium on Distributed Computing, 1982, pp.111-120.

[de Bakker et al 1985] de Bakker, J.W., Meyer, J.-J.Ch., Zucker, J., *Bringing Color into the Semantics of Nondeterministic Data Flow*, Preprint, Centre for Mathematics and Computer Science, Amsterdam, 1985.

[de Bakker & Kok 1985] de Bakker, J.W., Kok, J.N., *Towards a Topological Treatment of Streams and Functions on Streams*, in Proceedings Colloquium Automata, Languages and Programming (Brauer ed.), Lecture Notes in Computer Science 194, Springer-Verlag, 1985, pp. 140-149.

[Bergstra & Klop 1983] Bergstra, J., Klop, J.W., *Process Algebra for the Operational Semantics of Static Data Flow Networks*, Technical Report Mathematical Centre IW 222/83, Amsterdam, 1983.

[Boussinot 1982] Boussinot, F., *Proposition de semantique denotationelle pour des processues avec operateur de melange equitable*, in Theoretical Computer Science 18, 2, 1982, pp. 172-206.

[Broy 1983] Broy, M., *Fixed Point Theory for Communication and Concurrency*, in Formal Description of Programming Concepts-II (Bjoerner ed.), North-Holland, 1983, pp. 125-148.

[Broy 1985] Broy, M., *Extensional Behavior of Concurrent, Nondeterministic, Communicating Systems*, in Control Flow and Data Flow: Concepts of Distributed Programming (Broy ed.), North-Holland, pp. 229-277.

[Broy 1988] Broy, M., *Nondeterministic Data Flow Programs: How to avoid the Merge Anomaly*, in Science of Computer Programming 10, 1988, pp. 65-85.

[Brock & Ackerman 1981] Brock, J.D., Ackerman, W.B., *Scenario's: a model of nondeterminate computation*, in Formalization of Programming Concepts (Diaz, Ramos eds.), Lecture Notes in Computer Science 107, Springer-Verlag, 1981, pp. 252-259.

[Dugundji 1966] Dugundji, J., *Topology*, Allen & Bacon, 1966.

[Engelking 1977] Engelking, R., *General Topology*, Polish Scientific Publishers, 1977.

[Faustini 1982] A.A. Faustini, *An Operational Semantics for Pure Dataflow*, in Proceedings Colloquium Automata, Languages and Programming (Nielsen, Schmidt eds.), Lecture Notes in Computer Science 140, Springer-Verlag, 1982, pp. 212-224.

[Jonsson 1987a] Jonsson, B., *Compositional Verification of Distributed Systems*, Ph. D. Thesis, Department of Computer Systems, Uppsala University, Uppsala, Sweden, DoCS 87/09, 1987.

[Jonsson 1987b] Jonsson, B., *Modular Verification of Asynchronous Networks*, in Proceedings ACM Principles Of Distributed Computing, 1987.

[Kahn 1974] Kahn, G., *The semantics of a simple language for parallel programming*, in Proceedings Information Processing 1974 (Rosenfeld ed.), North-Holland, 1974, pp. 471-475.

[Kahn & MacQueen 1977] Kahn, G., MacQueen, D.B., *Coroutines and Networks of Parallel Processes*, in Proceedings Information Processing 1977 (Gilchrist ed.), North-Holland, 1977, pp. 993-998.

[Keller 1978] Keller, R.M., *Denotational Models for Parallel Programs with Indeterminate Operators*, in Formal Description of Programming Concepts (Neuhold ed.), North-Holland, 1977, pp. 337-366.

[Keller & Panangaden 1984] Keller, R.M., Panangaden, P., *Semantics of networks containing indeterminate operators*, in Proceedings Seminar on Concurrency (Brookes, Roscoe, Winskel eds.), Lecture Notes in Computer Science 197, Springer-Verlag, 1985, pp. 479-496.

[Kearney & Staples 1987] Kearney, P.F., Staples, J., *An extensional fixed point semantics for non-deterministic data flow*, draft, 1987.

[Kok 1986] Kok, J.N., *Denotational Semantics of Nets with Nondeterminism*, in Proceedings European Symposium on Programming (Wilhelm, Robinet eds.), Lecture Notes in Computer Science 206, Springer-Verlag, 1985, pp. 237-249.

[Kok 1987] Kok, J.N., *A Fully Abstract Semantics for Data Flow Nets*, in Proceedings Parallel Architectures and Languages Europe, Volume II (de Bakker, Nijman, Treleaven eds.), Lecture Notes in Computer Science 259, Springer-Verlag, 1987, pp. 351-369.

[Kosinski 1978] Kosinski, P.R., *A Straightforward Denotational Semantics for Nondeterminate Data Flow Programs*, in Proceedings ACM Principles Of Programming Languages, 1978, pp. 214-221.

[Nadler 1970] Nadler, S.B., *Some Results on Multi-Valued Contraction Mappings*, in Set-Valued Mappings, Selections and Topological Properties of 2^X (Fleischman ed.), Lecture Notes in Mathematics, pp. 64-69, 1970.

[Staples & Nguyen 1985] Staples, J., Nguyen, V.L., *A Fixpoint Semantics for Nondeterministic Dataflow*, in Journal of the ACM, 32, 2, 1985, pp. 411-445.

[Panangaden & Stark 1988] Panangaden, P., Stark, E.W., *Computations, Residuals, and the Power of Indeterminacy*, in Proceedings Colloquium Automata, Languages and Programming (Lepistö, Salomaa eds.), Lecture Notes in Computer Science 317, Springer-Verlag, 1988, pp. 439-454.

[Park 1983] Park, D., *The "fairness problem" and nondeterministic computing networks*, in Foundations of Computer Science IV, Part 2 (de Bakker, van Leeuwen eds.), Mathematical Centre Tracts 159, 1983, pp. 133-161.

[Pratt 1984] Pratt, V.R., *The pomset model of parallel processes: unifying the temporal and the spatial*, in Proceedings Seminar on Concurrency (Brookes, Roscoe, Winskel eds.), Lecture Notes in Computer Science 197, Springer-Verlag, 1985, pp. 180-196.