Comparative Semantics for a Real-Time Programming Language with Integration[1]

Franck van Breugel[2]
Department of Mathematics and Computer Science, Vrije Universiteit
De Boelelaan 1081a, 1081 HV Amsterdam

Abstract

An operational and a denotational semantic model are presented for a real-time programming language incorporating the concept of integration. This concept of integration, which has been introduced by Baeten and Bergstra [4], enables us to specify a restricted form of unbounded non-determinism. For example, the execution of an action at an arbitrary moment in a time interval can be specified using integration. The operational and the denotational model are proved to be equivalent using a general method based on higher-order transformations and complete metric spaces. In this context, Banach's fixed point theorem and Michael's theorem will turn out to be the most important aspects of complete metric spaces. Banach's theorem, which states that a contraction on a complete metric space has a unique fixed point, will be used to define semantic models and to compare semantic models. Michael's theorem, which roughly states that a compact union of compact sets is compact, will be used for the definition of semantic models.

Contents

Introduction

Real-time programming can be viewed as traditional concurrency supplied with timing constraints [47]. Because these timing constraints cause more complexity, the advantages of high-level languages are even greater in real-time programming than in concurrency and sequential programming. Several languages, like RTL [35], have been designed specifically for real-time programming. Other languages are extensions of already existing languages, for example, the language TCSP [24, 25] is an extension of the language CSP [34]. In real-time programming the correctness of a program depends not only on the flow of control. The program should also meet its timing constraints [46]. Therefore new semantic models should be developed. Several models both operational [4, 28, 32] and denotational [27, 37, 43] have already been provided.

In this paper a simple real-time programming language is studied. Apart from the traditional programming constructs, this language incorporates timed atomic actions and integration.

[1]full version of [22]

[2]electronic mail address: franck@cs.vu.nl

Timed atomic actions are atomic actions each provided with some timing information. This timing information denotes when the atomic action should be executed. The concept of integration has been introduced by Baeten and Bergstra [4]. A statement is integrated over a time set, which is a subset of the time domain, i.e. a non-deterministically chosen value from the time set is passed to the statement. Integration enables us to specify the execution of an action at an arbitrary moment in a time interval, for example, $\int_{t\in[1.03,2.41]}(a,t)$ denotes the execution of the action a at an arbitrary moment in the time interval $[1.03, 2.41]$. This execution can give rise to an infinite (conceptually uncountable) number of different executions.

For this simple real-time programming language an operational and a denotational semantic model are presented. The operational model is based on a labelled transition system in the style of Hennessy and Plotkin [31]. The denotational model is by definition compositional and fixed points are exploited to handle recursion. This denotational model uses a complete metric space as its mathematical domain, which has been initiated by Nivat [40] and de Bakker and Zucker [16].

To compare these models a method based on higher-order transformations, which has been described by Kok and Rutten [36], is used. This general method for comparing different semantic models has already been applied successfully to several programming paradigms varying from notions related to concurrency [7, 13, 14] to notions related to logic programming [6, 10, 20] and object-oriented programming [15, 44]. The present paper shows another application of this technique. This method is founded on complete metric spaces. Higher-order transformations are used to define semantic operators and models. Furthermore, the higher-order transformations are also used to compare semantic models.

In the definition of semantic models and in the comparison of these models we use several aspects of complete metric spaces; Banach's theorem, which states that if X is a complete metric space and $f : X \to X$ is a contraction then f has a unique fixed point, will be used to define semantic models and to compare these models. Furthermore, in combination with fixed point induction Banach's fixed point theorem will be used to prove several properties of these semantic models; Michael's theorem, which states that if $\mathcal{X}$ is a compact set of compact sets then the set $\bigcup \mathcal{X}$ is also compact, will be used to define semantic models; Kuratowski's theorem, which states that if X is a complete metric space then $\mathcal{P}_{nc}(X)$, the set of non-empty compact subsets of X, provided with the Hausdorff metric based on the metric of X is again a complete metric space, will be used to obtain complete metric spaces as mathematical domains of the semantic models.

In this paper a denotational modelling of integration is presented. This concept of integration describes a restricted form of unbounded non-determinism. In general, the modelling of unbounded non-determinism causes serious technical problems [3, 5, 33]. Because higher-order transformations are used, infinite computations can be modelled. This combination of metrically modelling a restricted form of unbounded non-determinism and infinite computations has not been presented elsewhere [21, 41]. Because the denotational model is compared with an operational model, which captures the computational intuition, we can derive the correctness of this denotational model with respect to the operational model. Banach's fixed point theorem and Michael's theorem play a technical but eminent role as will become clear in the rest of this paper.

1 Language definition

In this section we introduce the syntax of the real-time programming language, which is studied in this paper. This programming language is an extension of one of the languages studied by de Bakker and Meyer [13]. The language is uniform, i.e. the elementary actions are left atomic [11]. The language is built from atomic actions provided with some timing information, sequential composition, non-deterministic choice, parallel composition, so-called integration and recursion.

The real-time concepts of this language are timed atomic actions and integration. With timed atomic actions we denote atomic actions each provided with an expression. The evaluation of this expression yields an element of the time domain denoting the amount of time the atomic action should be executed after its enabling. Integration of a statement over some time set, which is a subset of the time domain, gives rise to the execution of the statement with some non-deterministically chosen value from the time set passed to that statement.

Before we can introduce the syntax of expressions, which are part of timed atomic actions, we first introduce the following sets:

- the set $\mathbb{R}_>$ of positive real numbers, with typical element r, which is our time domain;
- the set $TVar$ of time variables, with typical element t;
- the set $FSym$ of function symbols, with typical element f.

With each function symbol f we associate a function $f : \mathbb{R}_>^n \to \mathbb{R}_>$. We have to restrict the functions to continuous functions in order to be able to model integration as will become clear in section 4.

Definition 1.1
The class Exp of expressions, with typical element e, is given by
$e ::= t \mid f(e_1, \ldots, e_n)$
End 1.1

Because expressions are built from function symbols and time variables, expressions themselves can also be associated with continuous functions. We will denote the value of an expression e by $\mathcal{V}(e)$. After having defined the syntax of expressions, we have to introduce the following sets in order to be able to define the class of statements:

- the (possibly infinite) set $Atom$ of atomic actions, with typical element a;
- the set $PVar$ of procedure variables, with typical element x;
- the collection of time sets, which is represented by $\mathcal{P}_{nc}(\mathbb{R}_>)$, the set of non-empty compact subsets of $\mathbb{R}_>$, with typical element T.

Integration gives rise to a non-deterministic choice of an element from a time set. To guarantee that we can always make such a choice we have to restrict time sets to non-empty subsets of $\mathbb{R}_>$. The restriction of time sets to compact subsets of $\mathbb{R}_>$ has a technical motivation and this restriction will be crucial for the modelling of integration.

Definition 1.2
The class $Stat$ of statements, with typical element s, is given by
$s ::= (a, e) \mid x \mid s_1; s_2 \mid s_1 \cup s_2 \mid s_1 \parallel s_2 \mid \int_{t \in T} s$
End 1.2

A statement s is of one of the six following forms:

- (a, e) : a timed atomic action: the atomic action a has to be executed at time $\mathcal{V}(e)$, the value of the expression e, after its enabling;
- x : a procedure call: execution of the corresponding body of the procedure x;
- $s_1; s_2$: sequential composition of the statements s_1 and s_2;
- $s_1 \cup s_2$: non-deterministic choice of the statements s_1 and s_2;
- $s_1 \parallel s_2$: parallel composition of the statements s_1 and s_2: the arbitrary interleaving of the atomic actions of both statements;
- $\int_{t \in T} s$: integration: execution of the statement s with an arbitrary element of T passed to the time variable t in s.

The execution of the statement $(a, 1.56)$ corresponds to the execution of the atomic action a. The atomic action a should be executed 1.56 after its enabling. The execution of the integration $\int_{t \in [0.82, 1.73]} (a, f(t))$ corresponds to the execution of the statement $(a, f(t))$ with a non-deterministically chosen value from the time set $[0.82, 1.73]$ passed to the time variable t in that statement, which can give rise to the execution of, for example, $(a, f(1.08))$. Because the evaluation of an expression delivers a positive real number, two successive atomic actions cannot be executed at the same time. We stipulate that the execution of atomic actions and operators takes no time. We refer to [19] for a justification of this assumption. Next we introduce the class of guarded statements, which will be used to define procedure bodies.

Definition 1.3
The class $GStat$ of guarded statements, with typical element g, is given by

$$g ::= (a, e) \mid g; s \mid g_1 \cup g_2 \mid g_1 \parallel g_2 \mid \int_{t \in T} g$$

End 1.3

Before we give the definition of the class of declarations, which bind procedure variables with their corresponding bodies, we introduce the notion of free time variables.

Definition 1.4
The mapping $tvar : Exp \cup Stat \to \mathcal{P}(TVar)$ is given by

$$tvar(t) = \{t\}$$
$$tvar(f(e_1, \ldots, e_n)) = tvar(e_1) \cup \cdots \cup tvar(e_n)$$
$$tvar((a, e)) = tvar(e)$$
$$tvar(x) = \emptyset$$
$$tvar(s_1 * s_2) = tvar(s_1) \cup tvar(s_2) \qquad * \in \{;, \cup, \parallel\}$$
$$tvar(\textstyle\int_{t \in T} s) = tvar(s) \setminus \{t\}$$

End 1.4

A statement is called closed whenever it does not contain any free time variables. In several cases we will restrict ourselves to the class of closed statements or the class of closed guarded statements, which are defined in the following definition.

Definition 1.5
The class $CStat$ of closed statements, with typical element s, is given by

$$CStat = \{s \in Stat \mid tvar(s) = \emptyset\}$$

and the class $CGStat$ of closed guarded statements, with typical element g, is given by

$CGStat = \{g \in GStat \mid tvar(g) = \emptyset\}$

End 1.5

Now we have all the ingredients to define the class of declarations.

Definition 1.6

The class *Decl* of declarations, with typical element d, consists of sets of pairs

$\{(x_i, g_i) \in PVar \times CGStat \mid 1 \leq i \leq n\}$

where x_i are distinct procedure variables.

End 1.6

All procedure bodies in a declaration are restricted to guarded statements. This requirement corresponds to the usual Greibach condition in formal language theory. There are possibilities to eliminate this restriction as is illustrated by Reed and Roscoe [41, 42, 43]. However, by eliminating this restriction we are not able to model recursion any more. The restriction of the procedure bodies to closed statements guarantees that there are no global time variables. The execution of the procedure call x, where x is declared as $(a, 1); x$, corresponds to the execution of the procedure body $(a, 1); x$. We conclude this section with the definition of the class of programs.

Definition 1.7

The class *Prog* of programs, with typical element p, consists of pairs (d, s), such that each procedure variable occurring in s or d is declared in d and $s \in CStat$.

End 1.7

2 Complete metric spaces

Before the operational and denotational model are presented, we pay some attention to some aspects of complete metric spaces. These complete metric spaces have been introduced into semantics in papers of Nivat [40] and de Bakker and Zucker [16]. In this section we present two main theorems, Banach's fixed point theorem and Michael's theorem, which will be used frequently in the rest of this paper. For further reference considering metric spaces we suggest [26].

First we show how we can compose metric spaces. In the following definition we give some possible compositions, which will be used in the rest of this paper.

Definition 2.1

Let (X, d_X), (X_1, d_{X_1}) and (X_2, d_{X_2}) be metric spaces, where $d_X : X \times X \to [0, 1]$, $d_{X_1} : X_1 \times X_1 \to [0, 1]$ and $d_{X_2} : X_2 \times X_2 \to [0, 1]$.

- We define a metric on the Cartesian product of X_1 and X_2, $X_1 \times X_2$, by

  $$d_{X_1 \times X_2}((x_1, x_2), (y_1, y_2)) = max\{d_{X_1}(x_1, y_1), d_{X_2}(x_2, y_2)\}$$

- We define a metric on the collection of functions from X_1 to X_2, $X_1 \to X_2$, by

  $$d_{X_1 \to X_2}(f_1, f_2) = sup\{d_{X_2}(f_1(x), f_2(x)) \mid x \in X_1\}$$

- We define a metric on the collection of continuous functions from X_1 to X_2, $[X_1 \to X_2]$, by

  $$d_{[X_1 \to X_2]}(f_1, f_2) = sup\{d_{X_2}(f_1(x), f_2(x)) \mid x \in X_1\}$$

- We define a metric on the disjoint union of X_1 and X_2, $X_1 \bar{\cup} X_2$, by

$$\begin{array}{ll} d_{X_1 \bar{\cup} X_2}(x,y) = d_{X_1}(x,y) & x \in X_1 \wedge y \in X_1 \\ d_{X_1 \bar{\cup} X_2}(x,y) = d_{X_2}(x,y) & x \in X_2 \wedge y \in X_2 \\ d_{X_1 \bar{\cup} X_2}(x,y) = 1 & \text{otherwise} \end{array}$$

- We define a metric on X by

$$d_{id_{\frac{1}{2}}(X)}(x,y) = \tfrac{1}{2} d_X(x,y)$$

- We define a metric, the Hausdorff metric [30], on the set of non-empty compact subsets of X, $\mathcal{P}_{nc}(X)$, by

$$d_{\mathcal{P}_{nc}(X)}(Y,Z) = max \left\{ \begin{array}{l} sup\{inf\{d_X(y,z) \mid z \in Z\} \mid y \in Y\} \\ sup\{inf\{d_X(z,y) \mid y \in Y\} \mid z \in Z\} \end{array} \right\}$$

End 2.1

The following theorem states that the compositions, which have been described in the previous definition, of complete metric spaces give us again complete metric spaces.

Theorem 2.2
If (X, d_X), (X_1, d_{X_1}) and (X_2, d_{X_2}) are complete metric spaces, where $d_X : X \times X \to [0,1]$, $d_{X_1} : X_1 \times X_1 \to [0,1]$ and $d_{X_2} : X_2 \times X_2 \to [0,1]$, then

- $(X_1 \times X_2, d_{X_1 \times X_2})$,
- $(X_1 \to X_2, d_{X_1 \to X_2})$,
- $([X_1 \to X_2], d_{[X_1 \to X_2]})$,
- $(X_1 \bar{\cup} X_2, d_{X_1 \bar{\cup} X_2})$,
- $(id_{\frac{1}{2}}(X), d_{id_{\frac{1}{2}}(X)})$ and
- $(\mathcal{P}_{nc}(X), d_{\mathcal{P}_{nc}(X)})$

are also complete metric spaces.
End 2.2

All proofs but the proof of the last case of the above theorem are straightforward. The proof of the last case, Kuratowski's theorem, can be found in [38]. The next theorem, Banach's fixed point theorem [17], states that a contraction on a complete metric space has a unique fixed point. We will use this theorem to define semantic operators and models and to compare the semantic models developed below.

Theorem 2.3
If (X, d_X) is a complete metric space and $f : X \to X$ is a contraction then f has a unique fixed point x. Furthermore, we have that $\forall y \in X : \lim_{n \to \infty} f^n(y) = x$.
End 2.3

Michael's theorem [9, 39] states that a compact union of compact sets is again compact. This theorem will be used for the definition of semantic operators and semantic models.

Theorem 2.4
For all $\mathcal{X} \in \mathcal{P}_{nc}(\mathcal{P}_{nc}(X))$ we have that $\bigcup \mathcal{X} \in \mathcal{P}_{nc}(X)$.
Proof
Let $\{x_i\}_i$ be a sequence in $\bigcup \mathcal{X}$. Then there exists a sequence $\{X_i\}_i$ in $\mathcal{X}$ such that $x_i \in X_i$. Because $\mathcal{X}$ is compact, $\{X_i\}_i$ has a converging subsequence $\{X_{f(i)}\}_i$, which converges to some $X \in \mathcal{X}$. For each x_i we can find $y_i \in X$ such that $d_X(x_i, y_i) \leq 2d_{\mathcal{P}_{nc}(X)}(X_i, X)$. Because X is compact, the sequence $\{y_{f(i)}\}_i$ has a converging subsequence $\{y_{f(g(i))}\}_i$, which converges to some $y \in X$. Then we have that $d_X(x_{f(g(i))}, y) \leq d_X(x_{f(g(i))}, y_{f(g(i))}) + d_X(y_{f(g(i))}, y) \leq 2d_{\mathcal{P}_{nc}(X)}(X_{f(g(i))}, X) + d_X(y_{f(g(i))}, y)$. So we can conclude that $\{x_i\}_i$ has converging subsequence $\{x_{f(g(i))}\}_i$, which converges to $y \in \bigcup \mathcal{X}$.
End 2.4

In the remainder of this section we introduce complete metric spaces, which will be used in the rest of this paper. First of all, we define the so-called discrete metric on the class of atomic actions. We obtain a complete metric space.

Definition 2.5
The mapping $d_{Atom} : Atom \times Atom \to [0, 1]$ is given by

$$d_{Atom}(a, a') = 0 \qquad a = a'$$
$$d_{Atom}(a, a') = 1 \qquad a \neq a'$$

End 2.5

We extend our time domain $I\!R_>$ to the set of non-negative real numbers $I\!R_\geq$ in order to obtain a complete metric space. As will become clear in section 4 and 5 each mapping $d_{I\!R_\geq} : I\!R_\geq \times I\!R_\geq \to [0, 1]$ defining a complete metric space suffices. The metric $d_{I\!R_\geq}$ specifies the collection of compact subsets of the time domain: the collection of time sets. We can define a metric on $I\!R_\geq$ as follows.

Definition 2.6
The mapping $d_{I\!R_\geq} : I\!R_\geq \times I\!R_\geq \to [0, 1]$ is given by

$$d_{I\!R_\geq}(r, r') = \frac{|r - r'|}{|r - r'| + 1}$$

End 2.6

With respect to this metric closed intervals are compact sets. Furthermore, this metric and the usual metric on $I\!R_\geq$, $d_{I\!R_\geq}(r, r') = |r - r'|$, are equivalent, i.e. both induce the same converging behaviour. However, the metric of definition 2.6 is restricted to $[0, 1]$. Next we introduce the class of timed actions. These timed actions will be used to describe the execution of timed atomic actions.

Definition 2.7
The class TA of timed actions, with typical element α, is given by

$$TA = Atom \times I\!R_\geq$$

End 2.7

With the timed action $(a, 2)$ we will describe the execution of the atomic action a. The atomic action a is executed 2 after its enabling. We define a metric on timed actions by combining the metrics we have already defined on $Atom$ and $I\!R_\geq$ as described in definition 2.1. As stated in theorem 2.2 we obtain a complete metric space. To describe the execution of a sequence of timed atomic actions we introduce timed streams. We define this class of timed streams as the unique (up to isomorphism) solution of a domain equation in a certain category of complete metric spaces [1, 16].

Definition 2.8
The class TS of timed streams, with typical element σ, is given by the domain equation
$$TS \cong TA \mathbin{\bar{\cup}} TA \times id_{\frac{1}{2}}(TS)$$
End 2.8

For example, the timed stream $<(a,2),<(b,1),(c,3)>>$ describes the execution of atomic action a at 2 (after its enabling) followed by the execution of atomic action b at 1 and the execution of atomic action c at 3. Furthermore, we have that, for example,

$$d_{TS}(<(a,2),(b,1)>,<(a,3),(c,1)>)$$
$$=$$
$$max\{d_{TA}((a,2),(a,3)), \tfrac{1}{2}d_{TS}((b,1),(c,1))\}$$
$$=$$
$$max\{d_{TA}((a,2),(a,3)), \tfrac{1}{2}d_{TA}((b,1),(c,1))\}$$
$$=$$
$$max\{max\{d_{Atom}(a,a), d_{I\!R_{\geq}}(2,3)\}, \tfrac{1}{2}max\{d_{Atom}(b,c), d_{I\!R_{\geq}}(1,1)\}\}$$
$$=$$
$$max\{max\{0,\tfrac{1}{2}\}, \tfrac{1}{2}max\{1,0\}\}$$
$$=$$
$$\tfrac{1}{2}$$

Sets of sequences of timed atomic actions will be described by non-empty compact sets of timed streams. We can obtain a complete metric space on these sets of non-empty compact sets of timed streams as is described in definition 2.1 and theorem 2.2. Also on the class of statements we define the discrete metric. Again we obtain a complete metric space. Finally, we introduce the class of substitutions and define a metric on this class.

Definition 2.9
The class $Subst$ of substitutions, with typical element θ, consists of the class of homomorphisms from Exp to $I\!R_{\geq}$.
End 2.9

We will only consider substitutions θ with a finite support, i.e. there exist only finitely many time variables t such that $\theta t \neq t$. To simplify the exposition we will, without loss of generality, assume in almost all cases that substitutions θ satisfy the additional property that the set of time variables occurring in θ is exactly the set $\{t_1,\ldots,t_n\}$ of the first n time variables. Substitutions will be notated as $[t_1/r_1,\ldots,t_n/r_n]$. With ϵ we denote the empty substitution. For these substitutions we define the set of time variables occurring in those substitutions as follows.

Definition 2.10
The mapping $tvar : Subst \rightarrow \mathcal{P}(TVar)$ is given by
$$tvar(\theta) = \{t_1,\ldots,t_n \mid \theta = [t_1/r_1,\ldots,t_n/r_n]\}$$
End 2.10

We conclude this section with the definition of the metric on substitutions and the observation that this metric gives us a complete metric space.

Definition 2.11
The mapping $d_{Subst} : Subst \times Subst \rightarrow [0,1]$ is given by
$$\begin{array}{ll} d_{Subst}(\theta,\theta') = 0 & \theta = \epsilon \wedge \theta' = \epsilon \\ d_{Subst}(\theta,\theta') = max\{d_{I\!R_{\geq}}(\theta t,\theta' t) \mid t \in tvar(\theta)\} & tvar(\theta) = tvar(\theta') \wedge \theta \neq \epsilon \wedge \theta' \neq \epsilon \\ d_{Subst}(\theta,\theta') = 1 & \text{otherwise} \end{array}$$
End 2.11

3 Operational semantics

In this section we present an operational semantic model for our language. The operational semantics of a program describes the behaviour of an abstract machine running that program. The execution of a program on an abstract machine is characterised by sets of timed streams. Which timed actions and in which order the timed actions are performed by the abstract machine is described by means of a labelled transition system à la Hennessy and Plotkin [31].

Before giving a labelled transition system, we first introduce the empty statement E. This empty statement [2] is associated with termination. The class of statements $Stat$ is extended to $Stat_E$.

Definition 3.1
The class $Stat_E$, with typical element $\bar{s}$, is given by
$$Stat_E = Stat \cup \{E\}$$
End 3.1

Having extended the class of statements, we also extend the notions of free time variables and closed statements.

Definition 3.2
The mapping $tvar : Exp \cup Stat_E \to \mathcal{P}(TVar)$ is given by
$$tvar(E) = \emptyset$$
End 3.2

Definition 3.3
The class $CStat_E$, with typical element $\bar{s}$, is given by
$$CStat_E = \{\bar{s} \in Stat_E \mid tvar(\bar{s}) = \emptyset\} = CStat \cup \{E\}$$
End 3.3

It will be convenient to allow expressions of the form $\bar{s} * \bar{s}'$. This will reduce the number of rules of the labelled transition system. We define the following reasonable equivalences on these expressions.

Definition 3.4
For all $\bar{s} \in Stat_E$ and $* \in \{;, \cup, \|\}$
$$\bar{s} * E = \bar{s}$$
$$E * \bar{s} = \bar{s}$$
End 3.4

Next we present a transition relation, which induces a labelled transition system as has been described in, for example, [29].

Definition 3.5
The transition relation $\longrightarrow$ is the smallest subset of $CStat \times TA \times Decl \times CStat_E$ satisfying

$$(a, e) - (a, \mathcal{V}(e)) \to_d E$$

$$\frac{g - \alpha \to_d \bar{s} \qquad (x, g) \in d}{x - \alpha \to_d \bar{s}}$$

$$\frac{s - \alpha \rightarrow_d \bar{s}}{s; s' - \alpha \rightarrow_d \bar{s}; s'}$$

$$\frac{s - \alpha \rightarrow_d \bar{s}}{\begin{array}{l} s \cup s' - \alpha \rightarrow_d \bar{s} \\ s' \cup s - \alpha \rightarrow_d \bar{s} \end{array}}$$

$$\frac{s - \alpha \rightarrow_d \bar{s}}{\begin{array}{l} s \parallel s' - \alpha \rightarrow_d \bar{s} \parallel s' \\ s' \parallel s - \alpha \rightarrow_d s' \parallel \bar{s} \end{array}}$$

$$\frac{s[t/r] - \alpha \rightarrow_d \bar{s} \qquad r \in T}{\int_{t \in T} s - \alpha \rightarrow_d \bar{s}}$$

End 3.5

Intuitively, a rule $s - \alpha \rightarrow_d \bar{s}$ tells us that the execution of statement s consists of timed action α followed by the execution of statement $\bar{s}$. Consider the axiom for the statement (a, e). The execution of (a, e) consists of the execution of atomic action a at $\mathcal{V}(e)$, the value of expression e, after its enabling followed by termination. Because (a, e) is a closed statement, the evaluation of the expression e delivers an element of $I\!R_>$. Since the evaluation of an expression delivers an element of $I\!R_>$, we can conclude that two successive atomic actions cannot be executed at the same time. The rule for a procedure call indicates body replacement. Parallel composition is modelled by arbitrary interleaving of the atomic actions of both statements. The rule for integration states that some arbitrary element r from time set T is passed to time variable t in statement s. Using the above rules we can derive that $\int_{t \in [1.03, 2.41]} (a, t) - (a, r) \rightarrow_d E$ for all $r \in [1.03, 2.41]$.

Now we can define the operational semantics for closed statements s, related to a declaration d, such that all procedure variables occurring in statement s or declaration d are declared in declaration d.

Definition 3.6
The mapping $\mathcal{O}_d : CStat \rightarrow \mathcal{P}(TS)$ is given by
$\sigma \in \mathcal{O}_d(s)$ if and only if one of the following conditions is satisfied:

- $\exists n \in I\!N : \exists s_1, \ldots, s_n \in CStat : \exists \alpha_1, \ldots, \alpha_{n+1} \in TA :$
  $s - \alpha_1 \rightarrow_d s_1 - \alpha_2 \rightarrow_d \cdots - \alpha_n \rightarrow_d s_n - \alpha_{n+1} \rightarrow_d E \wedge$
  $\sigma = < \alpha_1, < \alpha_2, \cdots < \alpha_n, \alpha_{n+1} > \cdots >>$

- $\exists s_1, \ldots \in CStat : \exists \alpha_1, \ldots \in TA :$
  $s - \alpha_1 \rightarrow_d s_1 - \alpha_2 \rightarrow_d \cdots \wedge \sigma = < \alpha_1, < \alpha_2, \cdots >>$

End 3.6

For example, we have that $\mathcal{O}_d(\int_{t \in [1.03, 2.41]} (a, t)) = \{(a, 1.03), \ldots, (a, 2.41)\}$. We conclude this section with the definition of the operational semantics for programs.

Definition 3.7
The mapping $\mathcal{O} : Prog \rightarrow \mathcal{P}(TS)$ is given by

$$\mathcal{O}((d, s)) = \mathcal{O}_d(s)$$

End 3.7

4 Denotational semantics

After having defined an operational semantics, we give a denotational semantics for the language. This denotational semantics is by definition compositional, i.e. the meaning of a program can be derived from the meaning of its constituent parts. Fixed points are used to deal with recursion. To obtain a compositional model we define for every syntactic operator $\square$ a corresponding semantic operator $\diamond$, such that, for example, $\mathcal{D}((d, s_1 \square s_2)) = \mathcal{D}((d, s_1)) \diamond \mathcal{D}((d, s_2))$. To handle recursion we define the denotational model as a fixed point of a higher-order transformation. In this section we will use extensively Banach's fixed point theorem and Michael's theorem. Both theorems will be used for the construction of the semantic operators and the denotational model.

For the syntactic operator ; we define a corresponding semantic operator, which will also be denoted by ;. First we define the semantic operator ; on timed streams. This operator will be defined as a fixed point of a higher-order transformation $\Psi_;$. This higher-order transformation is a contraction on a complete metric space. Using Banach's fixed point theorem, we can conclude that $\Psi_;$ has a unique fixed point, which we will denote by ;. Then we lift the semantic operator ;, which has been defined on timed streams, such that we obtain a semantic operator defined on (non-empty compact) sets of timed streams. First we introduce the higher-order transformation $\Psi_;$.

Definition 4.1
The mapping $\Psi_; : (TS \times TS \to \mathcal{P}_{nc}(TS)) \to (TS \times TS \to \mathcal{P}_{nc}(TS))$ is given by

$$\Psi_;(F)(\alpha, \tau) = \{< \alpha, \tau >\}$$
$$\Psi_;(F)(< \alpha, \sigma >, \tau) = \{< \alpha, \rho >| \, \rho \in F(\sigma, \tau)\}$$

End 4.1

It is obvious that the mapping $\Psi_;$ is well-defined. Next we prove that this mapping is a contraction.

Property 4.2
The mapping $\Psi_;$ is a contraction.
Proof
We distinguish the following two cases.

1 $d_{\mathcal{P}_{nc}(TS)}(\Psi_;(F)(\alpha, \tau), \Psi_;(G)(\alpha, \tau))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\{< \alpha, \tau >\}, \{< \alpha, \tau >\})$
$=$
0
$\leq$
$\frac{1}{2} d_{TS \times TS \to \mathcal{P}_{nc}(TS)}(F, G)$

2 $d_{\mathcal{P}_{nc}(TS)}(\Psi_;(F)(< \alpha, \sigma >, \tau), \Psi_;(G)(< \alpha, \sigma >, \tau))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\{< \alpha, \rho >| \, \rho \in F(\sigma, \tau)\}, \{< \alpha, \rho >| \, \rho \in G(\sigma, \tau)\})$
$=$
$\frac{1}{2} d_{\mathcal{P}_{nc}(TS)}(F(\sigma, \tau), G(\sigma, \tau))$
$\leq$
$\frac{1}{2} d_{TS \times TS \to \mathcal{P}_{nc}(TS)}(F, G)$

End 4.2

Because $\Psi_;$ is a contraction on a complete metric space, we can deduce using Banach's fixed point theorem that $\Psi_;$ has a unique fixed point, which will be denoted by ;.

Corollary 4.3
The operator $; : TS \times TS \to \mathcal{P}_{nc}(TS)$ given by

$$\alpha;\tau = \{<\alpha,\tau>\}$$
$$<\alpha,\sigma>;\tau = \{<\alpha,\rho> \mid \rho \in \sigma;\tau\}$$

is well-defined.
End 4.3

Before we can lift the semantic operator ;, we have to prove that this operator is continuous. We prove that the operator ; is non-distance increasing in its first argument and contracting with factor $\frac{1}{2}$ in its second argument using fixed point induction and Banach's fixed point theorem without using the definition of the metric $d_{I\!R_{\geq}}$.

Property 4.4
For all $\sigma,\sigma',\tau,\tau' \in TS$ and $\varepsilon \in I\!R_{\geq}$

$$d_{TS}(\sigma,\sigma') \leq \varepsilon \wedge d_{TS}(\tau,\tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma;\tau,\sigma';\tau') \leq \varepsilon$$

Proof
We first prove

$$\forall n \geq 0 : d_{TS}(\sigma,\sigma') \leq \varepsilon \wedge d_{TS}(\tau,\tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma,\tau),\Psi_;^n(F)(\sigma',\tau')) \leq \varepsilon$$

where $F(\sigma,\tau) = \{\sigma\}$, with induction on n.

1 Let $n = 0$.

$$\begin{aligned}
&d_{\mathcal{P}_{nc}(TS)}(\Psi_;^0(F)(\sigma,\tau),\Psi_;^0(F)(\sigma',\tau'))\\
=\;&\\
&d_{\mathcal{P}_{nc}(TS)}(F(\sigma,\tau),F(\sigma',\tau'))\\
=\;&\\
&d_{\mathcal{P}_{nc}(TS)}(\{\sigma\},\{\sigma'\})\\
=\;&\\
&d_{TS}(\sigma,\sigma')\\
\leq\;&\\
&\varepsilon
\end{aligned}$$

2 Let $n > 0$.
We distinguish four cases.

2.1 Let $\sigma = (a,r)$ and $\sigma' = (a',r')$.

2.1.1. Let $a = a'$.
Then we have that $d_{TS}(\sigma,\sigma') = d_{I\!R_{\geq}}(r,r')$.

$$\begin{aligned}
&d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma,\tau),\Psi_;^n(F)(\sigma',\tau'))\\
=\;&\\
&d_{\mathcal{P}_{nc}(TS)}(\{<(a,r),\tau>\},\{<(a',r'),\tau'>\})\\
=\;&\\
&max\{d_{I\!R_{\geq}}(r,r'),\tfrac{1}{2}d_{TS}(\tau,\tau')\}\\
\leq\;&\\
&\varepsilon
\end{aligned}$$

2.1.2. Let $a \neq a'$.
Then we have that $d_{TS}(\sigma, \sigma') = 1$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau'))$
$\leq$
ε

2.2 Let $\sigma =< (a, r), \sigma'' >$ and $\sigma' = (a', r')$.
Then we have that $d_{TS}(\sigma, \sigma') = 1$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau'))$
$\leq$
ε

2.3 Let $\sigma = (a, r)$ and $\sigma' =< (a', r'), \sigma'' >$.
This case is similar to the second case.

2.4 Let $\sigma =< (a, r), \sigma'' >$ and $\sigma' =< (a', r'), \sigma''' >$.

2.4.1. Let $a = a'$.
Then we have that $d_{TS}(\sigma, \sigma') = max\{d_{\mathbb{R}_\geq}(r, r'), \frac{1}{2} d_{TS}(\sigma'', \sigma''')\}$.
Furthermore, the induction hypothesis gives us that
$d_{\mathcal{P}_{nc}(TS)}(\Psi_;^{n-1}(F)(\sigma'', \tau), \Psi_;^{n-1}(F)(\sigma''', \tau')) \leq 2\varepsilon$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau'))$
$=$
$d_{\mathcal{P}_{nc}(TS)} \begin{pmatrix} \{< (a, r), \rho >\mid \rho \in \Psi_;^{n-1}(F)(\sigma'', \tau)\} \\ \{< (a', r'), \rho >\mid \rho \in \Psi_;^{n-1}(F)(\sigma''', \tau')\} \end{pmatrix}$
$=$
$max\{d_{\mathbb{R}_\geq}(r, r'), \frac{1}{2} d_{\mathcal{P}_{nc}(TS)}(\Psi_;^{n-1}(F)(\sigma'', \tau), \Psi_;^{n-1}(F)(\sigma''', \tau'))\}$
$\leq$
ε

2.4.2. Let $a \neq a'$.
Then we have that $d_{TS}(\sigma, \sigma') = 1$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau'))$
$\leq$
ε

Now we can proceed as follows.
$\forall n \geq 0 : d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau')) \leq \varepsilon$
$\Rightarrow$
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow \forall n \geq 0 : d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau')) \leq \varepsilon$
$\Rightarrow$
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow \lim_{n\to\infty} d_{\mathcal{P}_{nc}(TS)}(\Psi_;^n(F)(\sigma, \tau), \Psi_;^n(F)(\sigma', \tau')) \leq \varepsilon$
$\Rightarrow$
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\lim_{n\to\infty} \Psi_;^n(F)(\sigma, \tau), \lim_{n\to\infty} \Psi_;^n(F)(\sigma', \tau')) \leq \varepsilon$
$\Rightarrow$ theorem 2.3
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma;\tau, \sigma';\tau') \leq \varepsilon$
End 4.4

The operator ; is lifted to sets of timed streams in the following definition.

Definition 4.5
The operator $; : \mathcal{P}_{nc}(TS) \times \mathcal{P}_{nc}(TS) \to \mathcal{P}_{nc}(TS)$ is given by
$S;T = \bigcup\{\sigma;\tau \mid \sigma \in S \wedge \tau \in T\}$
End 4.5

Next we prove the well-definedness of the lifted operator ; using Michael's theorem and the fact that the operator ; is continuous.

Property 4.6
The operator ; is well-defined.
Proof
By definition $\sigma;\tau \in \mathcal{P}_{nc}(TS)$ for all $\sigma \in S$ and $\tau \in T$. Because S and T are compact sets and the operator ; is continuous, the set $\{\sigma;\tau \mid \sigma \in S \wedge \tau \in T\}$ is compact. Theorem 2.4 tells us that the set $\bigcup\{\sigma;\tau \mid \sigma \in S \wedge \tau \in T\}$ is compact.
End 4.6

Also for the lifted operator ; we prove that it is non-distance increasing in its first argument and contracting with factor $\frac{1}{2}$ in its second argument. This property will be used to prove the well-definedness of the denotational semantics.

Property 4.7
For all $S, S', T, T' \in \mathcal{P}_{nc}(TS)$ and $\varepsilon \in I\!R_{\geq}$
$$d_{\mathcal{P}_{nc}(TS)}(S, S') \leq \varepsilon \wedge d_{\mathcal{P}_{nc}(TS)}(T, T') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(S;T, S';T') \leq \varepsilon$$
Proof
We first prove $(*)$
$$\forall \delta > 0 : \forall R \in \{\sigma;\tau \mid \sigma \in S \wedge \tau \in T\} : \exists R' \in \{\sigma';\tau' \mid \sigma' \in S' \wedge \tau' \in T'\} :$$
$$d_{\mathcal{P}_{nc}(TS)}(R, R') \leq \varepsilon + \delta$$
Take some $\delta > 0$ and some $\sigma;\tau$ where $\sigma \in S$ and $\tau \in T$. Because $d_{\mathcal{P}_{nc}(TS)}(S, S') \leq \varepsilon$ and $d_{\mathcal{P}_{nc}(TS)}(T, T') \leq 2\varepsilon$, we have that $\exists \sigma' \in S' : d_{TS}(\sigma, \sigma') \leq \varepsilon{+}\delta$ and $\exists \tau' \in T' : d_{TS}(\tau, \tau') \leq 2\varepsilon{+}2\delta$. From property 4.4 we can conclude that $d_{\mathcal{P}_{nc}(TS)}(\sigma;\tau, \sigma';\tau') \leq \varepsilon + \delta$. Having proved $(*)$ we can deduce that $d_{\mathcal{P}_{nc}(\mathcal{P}_{nc}(TS))}(\{\sigma;\tau \mid \sigma \in S \wedge \tau \in T\}, \{\sigma';\tau' \mid \sigma' \in S' \wedge \tau' \in T'\}) \leq \varepsilon$. Using property 4.9 we can conclude that $d_{\mathcal{P}_{nc}(TS)}(S;T, S';T') \leq \varepsilon$.
End 4.7

Next we define two semantic operators which correspond to the syntactic notions of non-deterministic choice and integration.

Definition 4.8
The operator $\cup : \mathcal{P}_{nc}(TS) \times \mathcal{P}_{nc}(TS) \to \mathcal{P}_{nc}(TS)$ is defined as the set-theoretic union and the operator $\bigcup : \mathcal{P}_{nc}(\mathcal{P}_{nc}(TS)) \to \mathcal{P}_{nc}(TS)$ is defined as the generalised set-theoretic union.
End 4.8

From Michael's theorem we can conclude that the semantic operator $\bigcup$ is well-defined. Both operators are non-distance increasing.

Property 4.9
For all $\mathcal{S}, \mathcal{T} \in \mathcal{P}_{nc}(\mathcal{P}_{nc}(TS))$ and $\varepsilon \in I\!R_{\geq}$
$$d_{\mathcal{P}_{nc}(\mathcal{P}_{nc}(TS))}(\mathcal{S}, \mathcal{T}) \leq \varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\bigcup \mathcal{S}, \bigcup \mathcal{T}) \leq \varepsilon$$
Proof
We have to prove that $\forall \delta > 0 : \forall \sigma \in \bigcup \mathcal{S} : \exists \tau \in \bigcup \mathcal{T} : d_{TS}(\sigma, \tau) \leq \varepsilon + \delta$. Take some $\delta > 0$ and $\sigma \in \bigcup \mathcal{S}$. Then $\sigma \in S$ for some $S \in \mathcal{S}$. We have that $\forall \delta' > 0 : \forall S' \in \mathcal{S} : \exists T' \in \mathcal{T} : d_{\mathcal{P}_{nc}(TS)}(S', T') \leq \varepsilon + \delta'$. So we have that $\exists T' \in \mathcal{T} : d_{\mathcal{P}_{nc}(TS)}(S, T') \leq \varepsilon + \frac{\delta}{2}$ thus $\exists T' \in \mathcal{T} : \forall \delta'' > 0 : \forall \sigma' \in S : \exists \tau' \in T' : d_{TS}(\sigma', \tau') \leq \varepsilon + \frac{\delta}{2} + \delta''$. We conclude that $\exists \tau \in \bigcup \mathcal{T} : d_{TS}(\sigma, \tau) \leq \varepsilon + \delta$.
End 4.9

The semantic counterpart of the syntactic operator $\|$ is also defined as the unique fixed point of a higher-order transformation. This higher-order transformation $\Psi_{\|}$ is defined by means of the higher-order transformation $\Psi_{;}$.

Definition 4.10
The mapping $\Psi_{\|} : (TS \times TS \to \mathcal{P}_{nc}(TS)) \to (TS \times TS \to \mathcal{P}_{nc}(TS))$ is given by

$$\Psi_{\|}(F)(\sigma, \tau) = \Psi_{;}(F)(\sigma, \tau) \cup \Psi_{;}(F)(\tau, \sigma)$$

End 4.10

The well-definedness of the mapping $\Psi_{\|}$ follows from the well-definedness of $\Psi_{;}$. From the fact that $\Psi_{;}$ is a contraction we can deduce that $\Psi_{\|}$ is also a contraction as is illustrated in the proof of the following property.

Property 4.11
The mapping $\Psi_{\|}$ is a contraction.
Proof
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\|}(F)(\sigma, \tau), \Psi_{\|}(G)(\sigma, \tau))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{;}(F)(\sigma, \tau) \cup \Psi_{;}(F)(\tau, \sigma), \Psi_{;}(G)(\sigma, \tau) \cup \Psi_{;}(G)(\tau, \sigma))$
$\leq$ property 4.2 and property 4.9
$\frac{1}{2} d_{TS \times TS \to \mathcal{P}_{nc}(TS)}(F, G)$
End 4.11

We denote the unique fixed point of the higher-order transformation $\Psi_{\|}$ by $\|$. We can define $\|$ by means of a so-called left-merge $\lfloor\!\lfloor$ [18] which expresses a merge where the first element is taken from the left argument. We have that

$$\sigma \parallel \tau = \Psi_{\|}(\|)(\sigma, \tau) = \Psi_{;}(\|)(\sigma, \tau) \cup \Psi_{;}(\|)(\tau, \sigma)$$

and

$$\Psi_{;}(\|)(\alpha, \tau) = \{< \alpha, \tau >\}$$

and

$$\Psi_{;}(\|)(< \alpha, \sigma >, \tau) = \{< \alpha, \rho > \mid \rho \in \sigma \parallel \tau\}$$

Thus we can characterise the operators $\|$ and $\lfloor\!\lfloor$ as follows.

Corollary 4.12
The operator $\|: TS \times TS \to \mathcal{P}_{nc}(TS)$ given by

$$\sigma \parallel \tau = \sigma \lfloor\!\lfloor \tau \cup \tau \lfloor\!\lfloor \sigma$$

and the operator $\lfloor\!\lfloor: TS \times TS \to \mathcal{P}_{nc}(TS)$ given by

$$\alpha \lfloor\!\lfloor \tau = \{< \alpha, \tau >\}$$
$$< \alpha, \sigma > \lfloor\!\lfloor \tau = \{< \alpha, \rho > \mid \rho \in \sigma \parallel \tau\}$$

are well-defined.
End 4.12

Also for the semantic operator $\|$ we prove a continuity property. This operator is non-distance increasing in both arguments as is stated in the following property.

Property 4.13
For all $\sigma, \sigma', \tau, \tau' \in TS$ and $\varepsilon \in I\!R_{\geq}$

$$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq \varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma \parallel \tau, \sigma' \parallel \tau') \leq \varepsilon$$

Proof
From the proof of property 4.4 we know that
$\forall n \geq 0 : d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\Psi^n_;(F)(\sigma, \tau), \Psi^n_;(F)(\sigma', \tau')) \leq \varepsilon$
$\Rightarrow$
$\forall n \geq 0 : d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq \varepsilon \Rightarrow$
$d_{\mathcal{P}_{nc}(TS)}(\Psi^n_;(F)(\sigma, \tau), \Psi^n_;(F)(\sigma', \tau')) \leq \varepsilon \wedge d_{\mathcal{P}_{nc}(TS)}(\Psi^n_;(F)(\tau, \sigma), \Psi^n_;(F)(\tau', \sigma')) \leq \varepsilon$
$\Rightarrow$ property 4.9
$\forall n \geq 0 : d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq \varepsilon \Rightarrow$
$d_{\mathcal{P}_{nc}(TS)}(\Psi^n_;(F)(\sigma, \tau) \cup \Psi^n_;(F)(\tau, \sigma), \Psi^n_;(F)(\sigma', \tau') \cup \Psi^n_;(F)(\tau', \sigma')) \leq \varepsilon$
$\Rightarrow$
$\forall n \geq 0 : d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq \varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\Psi^n_{\|}(F)(\sigma, \tau), \Psi^n_{\|}(F)(\sigma', \tau')) \leq \varepsilon$
Using the arguments as in the proof of property 4.4 we can conclude that
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq \varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma \parallel \tau, \sigma' \parallel \tau') \leq \varepsilon$
End 4.13

The operator $\lfloor\!\lfloor$ is non-distance increasing in its first argument and contracting with factor $\frac{1}{2}$ in its second argument.

Property 4.14
For all $\sigma, \sigma', \tau, \tau' \in TS$ and $\varepsilon \in I\!R_{\geq}$

$$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma \lfloor\!\lfloor \tau, \sigma' \lfloor\!\lfloor \tau') \leq \varepsilon$$

Proof
From property 4.13 we can deduce that
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma \parallel \tau, \sigma' \parallel \tau') \leq \varepsilon$
Using property 4.4 we can deduce that
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\Psi_;(\|)(\sigma, \tau), \Psi_;(\|)(\sigma', \tau')) \leq \varepsilon$
So we can conclude that
$d_{TS}(\sigma, \sigma') \leq \varepsilon \wedge d_{TS}(\tau, \tau') \leq 2\varepsilon \Rightarrow d_{\mathcal{P}_{nc}(TS)}(\sigma \lfloor\!\lfloor \tau, \sigma' \lfloor\!\lfloor \tau') \leq \varepsilon$
End 4.14

The operators $\|$ and $\lfloor\!\lfloor$ are lifted to sets of timed streams.

Definition 4.15
The operator $\| : \mathcal{P}_{nc}(TS) \times \mathcal{P}_{nc}(TS) \rightarrow \mathcal{P}_{nc}(TS)$ is given by

$$S \parallel T = S \lfloor\!\lfloor T \cup T \lfloor\!\lfloor S$$

and the operator $\lfloor\!\lfloor : \mathcal{P}_{nc}(TS) \times \mathcal{P}_{nc}(TS) \rightarrow \mathcal{P}_{nc}(TS)$ is given by

$$S \lfloor\!\lfloor T = \bigcup\{\sigma \lfloor\!\lfloor \tau \mid \sigma \in S \wedge \tau \in T\}$$

End 4.15

Because the operator $\lfloor\!\lfloor$ is continuous, we can conclude that the lifted operators $\|$ and $\lfloor\!\lfloor$ are well-defined.

Property 4.16
The operators $\|$ and $\lfloor\!\lfloor$ are well-defined.
Proof
Similar to property 4.6.
End 4.16

Also the lifted operator $\|$ is non-distance increasing in both arguments.

Property 4.17
For all $S, S', T, T' \in TS$ and $\varepsilon \in I\!R_{\geq}$

$$d(S, S') \leq \varepsilon \wedge d(T, T') \leq \varepsilon \Rightarrow d(S \parallel T, S' \parallel T') \leq \varepsilon$$

Proof
Similar to property 4.7.
End 4.17

Having defined the semantic operators, we define the denotational semantics for statements related to a declaration, such that all procedure variables occurring in the statement or the declaration are declared in the declaration. To obtain a compositional model we introduce substitutions, which record the choices made for the time variables with respect to integration. For example, to derive the meaning of the statement $\int_{t \in [1.03, 2.41]}(a, t)$ from the meaning of the statement (a, t), we record the choice made for the time variable t in the substitutions $[t/r]$ for $r \in [1.03, 2.41]$.

We define the denotational semantics $\mathcal{D}_d : Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)$ as a partial function such that $\mathcal{D}_d(s)(\theta)$ is defined whenever $tvar(s) \subseteq tvar(\theta)$. The substitution θ should bind all free variables of statement s. We define the denotational semantics $\mathcal{D}_d$ as the fixed point of a higher-order transformation $\Psi_{\mathcal{D}}$. We have to impose a restriction on this mapping $\Psi_{\mathcal{D}}$ in order to obtain a well-defined mapping. We restrict $\Psi_{\mathcal{D}}$ to a mapping from $[Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)]$, the collection of continuous mappings $Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)$, to $[Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)]$.

Definition 4.18
The mapping $\Psi_{\mathcal{D}} : [Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)] \rightarrow [Stat \rightarrow Subst \rightarrow_p \mathcal{P}_{nc}(TS)]$ is given by

$$\begin{aligned}
&\Psi_{\mathcal{D}}(F)((a, e))(\theta) = \{(a, \mathcal{V}(e\theta))\} \\
&\Psi_{\mathcal{D}}(F)(x)(\theta) = \Psi_{\mathcal{D}}(F)(g)(\theta) && (x, g) \in d \\
&\Psi_{\mathcal{D}}(F)(s_1; s_2)(\theta) = \Psi_{\mathcal{D}}(F)(s_1)(\theta); F(s_2)(\theta) \\
&\Psi_{\mathcal{D}}(F)(s_1 * s_2)(\theta) = \Psi_{\mathcal{D}}(F)(s_1)(\theta) * \Psi_{\mathcal{D}}(F)(s_2)(\theta) && * \in \{\cup, \parallel\} \\
&\Psi_{\mathcal{D}}(F)(\textstyle\int_{t \in T} s)(\theta) = \bigcup\{\Psi_{\mathcal{D}}(F)(s)(\theta[t/r]) \mid r \in T\}
\end{aligned}$$

whenever $tvar(s) \subseteq tvar(\theta)$.
End 4.18

In several of the forthcoming proofs, we will use induction on the structure of statements. Therefore we introduce a complexity function on statements associated with a declaration d in the usual way.

Definition 4.19
The mapping $cf_d : Stat \rightarrow I\!N$ is given by

$$\begin{aligned}
&cf_d((a, e)) = 1 \\
&cf_d(x) = cf_d(g) + 1 && (x, g) \in d \\
&cf_d(s_1; s_2) = cf_d(s_1) + 1 \\
&cf_d(s_1 * s_2) = cf_d(s_1) + cf_d(s_2) && * \in \{\cup, \parallel\} \\
&cf_d(\textstyle\int_{t \in T} s) = cf_d(s) + 1
\end{aligned}$$

End 4.19

For each guarded statement the complexity function cf_d is well-defined, which follows immediately from the definition of the complexity function and the form of guarded statements. We can conclude that the complexity function is well-defined for all statements. In the case of sequential composition only induction on the first argument can be applied. From the definition of the complexity function we can derive that $cf_d(s[t/e]) = cf_d(s)$.

Using this induction principle, we will prove that the higher-order transformation $\Psi_{\mathcal{D}}$ is well-defined. Because expressions are continuous functions, time sets are non-empty compact sets and the semantic operators are continuous, we can prove that $\Psi_{\mathcal{D}}$ is well-defined using Michael's theorem.

Property 4.20
The mapping $\Psi_{\mathcal{D}}$ is well-defined.
Proof
We have to prove $\forall F \in [\mathit{Stat} \to \mathit{Subst} \to_p \mathcal{P}_{nc}(TS)] : \Psi_{\mathcal{D}}(F) \in [\mathit{Stat} \to \mathit{Subst} \to_p \mathcal{P}_{nc}(TS)]$. The mapping $\Psi_{\mathcal{D}}(F)$ is continuous if and only if $\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s)(\theta_i) = \Psi_{\mathcal{D}}(F)(s)(\theta)$ whenever $\lim_{i\to\infty} \theta_i = \theta$. We prove this property with induction on the complexity of statement s. We assume $\lim_{i\to\infty} \theta_i = \theta$.

1 Let $s \equiv (a, e)$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)((a,e))(\theta_i)$
$=$
$\lim_{i\to\infty}\{(a, \mathcal{V}(e\theta_i))\}$
$=$ $\qquad$ e is continuous
$\{(a, \mathcal{V}(e\theta))\}$
$=$
$\Psi_{\mathcal{D}}(F)((a,e))(\theta)$
The set $\{(a, \mathcal{V}(e\theta))\}$ is an element of $\mathcal{P}_{nc}(TS)$.

2 Let $s \equiv x$ and $(x, g) \in d$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(x)(\theta_i)$
$=$
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(g)(\theta_i)$
$=$
$\Psi_{\mathcal{D}}(F)(g)(\theta)$
$=$
$\Psi_{\mathcal{D}}(F)(x)(\theta)$
Because $\Psi_{\mathcal{D}}(F)(g)(\theta)$ is an element of $\mathcal{P}_{nc}(TS)$, $\Psi_{\mathcal{D}}(F)(x)(\theta)$ is an element of $\mathcal{P}_{nc}(TS)$.

3 Let $s \equiv s_1; s_2$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s_1; s_2)(\theta_i)$
$=$
$\lim_{i\to\infty}(\Psi_{\mathcal{D}}(F)(s_1)(\theta_i); F(s_2)(\theta_i))$
$=$
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s_1)(\theta_i); \lim_{i\to\infty} F(s_2)(\theta_i)$
$=$
$\Psi_{\mathcal{D}}(F)(s_1)(\theta); F(s_2)(\theta)$
$=$
$\Psi_{\mathcal{D}}(F)(s_1; s_2)(\theta)$
Because both $\Psi_{\mathcal{D}}(F)(s_1)(\theta)$ and $F(s_2)(\theta)$ are elements of $\mathcal{P}_{nc}(TS)$ and the operator ; is well-defined, $\Psi_{\mathcal{D}}(F)(s_1; s_2)(\theta)$ is also an element of $\mathcal{P}_{nc}(TS)$.

4 Let $s \equiv s_1 * s_2$ and $* \in \{\cup, \|\}$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s_1 * s_2)(\theta_i)$
$=$
$\lim_{i\to\infty}(\Psi_{\mathcal{D}}(F)(s_1)(\theta_i) * \Psi_{\mathcal{D}}(F)(s_2)(\theta_i))$

$=$
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s_1)(\theta_i) * \lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s_2)(\theta_i)$
$=$
$\Psi_{\mathcal{D}}(F)(s_1)(\theta) * \Psi_{\mathcal{D}}(F)(s_2)(\theta)$
$=$
$\Psi_{\mathcal{D}}(F)(s_1 * s_2)(\theta)$
Because both $\Psi_{\mathcal{D}}(F)(s_1)(\theta)$ and $\Psi_{\mathcal{D}}(F)(s_2)(\theta)$ are elements of $\mathcal{P}_{nc}(TS)$ and the operator $*$ is well-defined, $\Psi_{\mathcal{D}}(F)(s_1 * s_2)(\theta)$ is also an element of $\mathcal{P}_{nc}(TS)$.

5 Let $s \equiv \int_{t\in T} s$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(\int_{t\in T} s)(\theta_i)$
$=$
$\lim_{i\to\infty} \bigcup\{\Psi_{\mathcal{D}}(F)(s)(\theta_i[t/r]) \mid r \in T\}$
$=$
$\bigcup\{\lim_{i\to\infty} \Psi_{\mathcal{D}}(F)(s)(\theta_i[t/r]) \mid r \in T\}$
$=$
$\bigcup\{\Psi_{\mathcal{D}}(F)(s)(\theta[t/r]) \mid r \in T\}$
$=$
$\Psi_{\mathcal{D}}(F)(\int_{t\in T} s)(\theta)$
For all $r \in T$ the set $\Psi_{\mathcal{D}}(F)(s)(\theta[t/r])$ is an element of $\mathcal{P}_{nc}(TS)$. Because $\Psi_{\mathcal{D}}(F)$ is continuous and T is non-empty and compact, the set $\{\Psi_{\mathcal{D}}(F)(s)(\theta[t/r]) \mid r \in T\}$ is an element of $\mathcal{P}_{nc}(TS)$. From theorem 2.4 we can conclude that $\bigcup\{\Psi_{\mathcal{D}}(F)(s)(\theta[t/r]) \mid r \in T\}$ is also an element of $\mathcal{P}_{nc}(TS)$.

End 4.20

Next we prove that $\Psi_{\mathcal{D}}$ is a contraction. The contractivity of $\Psi_{\mathcal{D}}$ follows from the contractivity properties of the semantic operators.

Property 4.21
The mapping $\Psi_{\mathcal{D}}$ is a contraction.
Proof
We prove for all $s \in Stat$ and $\theta \in Subst$ such that $tvar(s) \subseteq tvar(\theta)$ that
$$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(s)(\theta), \Psi_{\mathcal{D}}(G)(s)(\theta)) \leq \tfrac{1}{2} d_{[Stat\to Subst\to_p \mathcal{P}_{nc}(TS)]}(F,G)$$
using induction on the complexity of statement s.

1 Let $s \equiv (a, e)$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)((a,e))(\theta), \Psi_{\mathcal{D}}(G)((a,e))(\theta))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\{(a, \mathcal{V}(e\theta))\}, \{(a, \mathcal{V}(e\theta))\})$
$=$
0
$\leq$
$\frac{1}{2} d_{[Stat\to Subst\to_p \mathcal{P}_{nc}(TS)]}(F,G)$

2 Let $s \equiv x$ and $(x, g) \in d$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(x)(\theta), \Psi_{\mathcal{D}}(G)(x)(\theta))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(g)(\theta), \Psi_{\mathcal{D}}(G)(g)(\theta))$
$\leq$
$\frac{1}{2} d_{[Stat\to Subst\to_p \mathcal{P}_{nc}(TS)]}(F,G)$

3 Let $s \equiv s_1; s_2$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(s_1;s_2)(\theta), \Psi_{\mathcal{D}}(G)(s_1;s_2)(\theta))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(s_1)(\theta); F(s_2)(\theta), \Psi_{\mathcal{D}}(G)(s_1)(\theta); G(s_2)(\theta))$
$\leq$ property 4.7
$\frac{1}{2} d_{[Stat \to Subst \to_p \mathcal{P}_{nc}(TS)]}(F, G)$

4 Let $s \equiv s_1 * s_2$ and $* \in \{\cup, \|\}$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(s_1 * s_2)(\theta), \Psi_{\mathcal{D}}(G)(s_1 * s_2)(\theta))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(s_1)(\theta) * \Psi_{\mathcal{D}}(F)(s_2)(\theta), \Psi_{\mathcal{D}}(G)(s_1)(\theta) * \Psi_{\mathcal{D}}(G)(s_2)(\theta))$
$\leq$ property 4.9 and property 4.17
$\frac{1}{2} d_{[Stat \to Subst \to_p \mathcal{P}_{nc}(TS)]}(F, G)$

5 Let $s \equiv \int_{t \in T} s$.
$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{D}}(F)(\int_{t \in T} s)(\theta), \Psi_{\mathcal{D}}(G)(\int_{t \in T} s)(\theta))$
$=$
$d_{\mathcal{P}_{nc}(TS)}(\bigcup\{\Psi_{\mathcal{D}}(F)(s)(\theta[t/r]) \mid r \in T\}, \bigcup\{\Psi_{\mathcal{D}}(G)(s)(\theta[t/r]) \mid r \in T\})$
$\leq$ property 4.9
$\frac{1}{2} d_{[Stat \to Subst \to_p \mathcal{P}_{nc}(TS)]}(F, G)$

End 4.21

Because $\Psi_{\mathcal{D}}$ is a contraction on a complete metric space, this mapping has a unique fixed point, which we will denote by $\mathcal{D}_d$. We can characterise $\mathcal{D}_d$ as follows.

Corollary 4.22
The mapping $\mathcal{D}_d : Stat \to Subst \to_p \mathcal{P}_{nc}(TS)$ is given by

$\mathcal{D}_d((a, e))(\theta) = \{(a, \mathcal{V}(e\theta))\}$
$\mathcal{D}_d(x)(\theta) = \mathcal{D}_d(g)(\theta)$ $\qquad (x, g) \in d$
$\mathcal{D}_d(s_1 * s_2)(\theta) = \mathcal{D}_d(s_1)(\theta) * \mathcal{D}_d(s_2)(\theta)$ $\qquad * \in \{;, \cup, \|\}$
$\mathcal{D}_d(\int_{t \in T} s)(\theta) = \bigcup\{\mathcal{D}_d(s)(\theta[t/r]) \mid r \in T\}$

whenever $tvar(s) \subseteq tvar(\theta)$.
End 4.22

We have, for example,
$\mathcal{D}_d(\int_{t \in [1.03, 2.41]} (a, t))(\epsilon)$
$=$
$\bigcup\{\mathcal{D}_d((a, t))([t/r]) \mid r \in [1.03, 2.41]\}$
$=$
$\bigcup\{\{(a, t[t/r])\} \mid r \in [1.03, 2.41]\}$
$=$
$\{(a, r) \mid r \in [1.03, 2.41]\}$

The denotational semantics for programs is defined as follows.

Definition 4.23
The mapping $\mathcal{D} : Prog \to \mathcal{P}_{nc}(TS)$ is given by
$\mathcal{D}((d, s)) = \mathcal{D}_d(s)(\epsilon)$
End 4.23

5 Equivalence proof

Having defined both an operational and a denotational semantics for our language the question arises whether the denotational model is correct with respect to the computational intuition captured by the operational model. In this section we will show that we can relate the operational model $\mathcal{O}$ and the denotational model $\mathcal{D}$. We will prove that these models are equivalent. To prove this we will use a general method for comparing different semantic models as described by Kok and Rutten [36]: if two models are both a fixed point of a higher-order transformation and this higher-order transformation is a contraction on a complete metric space, we can conclude that those models are equivalent.

We will introduce an intermediate operational semantic model $\mathcal{O}_d^*$ and relate this model to the operational model $\mathcal{O}_d$. Furthermore, we will define an intermediate denotational semantic model $\mathcal{D}_d^*$ and relate this model to the denotational model $\mathcal{D}_d$. Finally, we will introduce a higher-order transformation $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ and prove that this mapping is a contraction on a complete metric space. We will prove that $\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{O}_d^*) = \mathcal{O}_d^*$ and $\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*) = \mathcal{D}_d^*$. From this we can conclude that $\mathcal{O}_d^*$ and $\mathcal{D}_d^*$ are equivalent. These relations will enable us to prove the equivalence of $\mathcal{O}$ and $\mathcal{D}$.

First an intermediate operational model $\mathcal{O}_d^*$, which is associated with a labelled transition system, is introduced. In this operational model statements and substitutions, which record choices made for time variables with respect to integration, are separated. Therefore we introduce the class of configurations.

Definition 5.1
The class $Conf$ of configurations, with typical element C, is given by

$$Conf = \{[s,\theta] \in Stat \times Subst \mid tvar(s) \subseteq tvar(\theta)\}$$

and the class $Conf_E$ of configurations, with typical element $\bar{C}$, is given by

$$Conf_E = \{[\bar{s},\theta] \in Stat_E \times Subst \mid tvar(\bar{s}) \subseteq tvar(\theta)\}$$

End 5.1

The transition relation, which induces the labelled transition system describing the intermediate operational semantics, is presented in the following definition.

Definition 5.2
The transition relation $\longrightarrow$ is the smallest subset of $Conf \times TA \times Decl \times Conf_E$ satisfying

$$[(a,e),\theta] - (a, \mathcal{V}(e\theta)) \rightarrow_d [E,\theta]$$

$$\frac{[g,\theta] - \alpha \rightarrow_d \bar{C} \qquad (x,g) \in d}{[x,\theta] - \alpha \rightarrow_d \bar{C}}$$

$$\frac{[s,\theta] - \alpha \rightarrow_d [\bar{s},\theta']}{[s;s',\theta] - \alpha \rightarrow_d [\bar{s};s',\theta']}$$

$$\frac{[s,\theta] - \alpha \rightarrow_d \bar{C}}{\begin{array}{l}[s \cup s',\theta] - \alpha \rightarrow_d \bar{C}\\ [s' \cup s,\theta] - \alpha \rightarrow_d \bar{C}\end{array}}$$

$$\frac{[s,\theta] - \alpha \rightarrow_d [\bar{s},\theta']}{\begin{array}{l}[s \parallel s',\theta] - \alpha \rightarrow_d [\bar{s} \parallel s',\theta']\\ [s' \parallel s,\theta] - \alpha \rightarrow_d [s' \parallel \bar{s},\theta']\end{array}}$$

$$\frac{[s[t/t_{n+1}], \theta[t_{n+1}/r]] - \alpha \rightarrow_d \bar{C} \qquad tvar(\theta) = \{t_1, \ldots, t_n\} \qquad r \in T}{[\int_{t \in T} s, \theta] - \alpha \rightarrow_d \bar{C}}$$

End 5.2

It is not obvious that the transition relation given above is well-defined, i.e. if $[s, \theta] \in Conf$ and $[s, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$ then $[\bar{s}, \theta'] \in Conf_E$, which implies $tvar(\bar{s}) \subseteq tvar(\theta')$. To prove the well-definedness of the transition relation we introduce the notion of a substitution θ' being an extension of a substitution θ, which is denoted by $\theta \sqsubseteq \theta'$.

Definition 5.3
The relation $\sqsubseteq \subseteq Subst \times Subst$ is given by
$\theta \sqsubseteq \theta'$ if and only if $tvar(\theta) \subseteq tvar(\theta')$ and $\forall t \in tvar(\theta) : t\theta = t\theta'$
End 5.3

Next we show that if $[s, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$ then θ' is an extension of θ. From this we can deduce the well-definedness of the transition relation.

Property 5.4
For all $[s, \theta] \in Conf$, $\bar{s} \in Stat_E$, $\theta' \in Subst$ and $\alpha \in TA$
$[s, \theta] - \alpha \rightarrow_d [\bar{s}, \theta'] \Rightarrow \theta \sqsubseteq \theta'$
Proof
We prove this property using induction on the complexity of statement s.

1 Let $s \equiv (a, e)$.
By inspection of the transition system, $[(a, e), \theta] - (a, \mathcal{V}(e\theta)) \rightarrow_d [E, \theta]$, and the fact that $\theta \sqsubseteq \theta$, we can deduce that the property is satisfied in this case.

2 Let $s \equiv x$ and $(x, g) \in d$.
$[x, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$[g, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$\theta \sqsubseteq \theta'$

3 Let $s \equiv s_1; s_2$.
$[s_1; s_2, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$\exists \bar{s}' \in Stat_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}', \theta']$
$\Rightarrow$
$\theta \sqsubseteq \theta'$

4 Let $s \equiv s_1 \cup s_2$.
$[s_1 \cup s_2, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$[s_1, \theta] - \alpha \rightarrow_d [\bar{s}, \theta'] \vee [s_2, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$\theta \sqsubseteq \theta'$

5 Let $s \equiv s_1 \parallel s_2$.
$[s_1 \parallel s_2, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$

$\Rightarrow$
$\exists \bar{s}' \in Stat_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \vee \exists \bar{s}' \in Stat_E : [s_2, \theta] - \alpha \rightarrow_d [\bar{s}', \theta']$
$\Rightarrow$
$\theta \sqsubseteq \theta'$

6 Let $s \equiv \int_{t \in T} s$ and $tvar(\theta) = \{t_1, \ldots, t_n\}$.
$[\int_{t \in T} s, \theta] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$\exists r \in T : [s[t/t_{n+1}], \theta[t_{n+1}/r]] - \alpha \rightarrow_d [\bar{s}, \theta']$
$\Rightarrow$
$\theta[t_{n+1}/r] \sqsubseteq \theta'$
$\Rightarrow$
$\theta \sqsubseteq \theta'$

End 5.4

The intermediate operational model $\mathcal{O}_d^*$ is defined along the lines of the definition of the operational model $\mathcal{O}_d$.

Definition 5.5
The mapping $\mathcal{O}_d^* : [Conf \rightarrow \mathcal{P}_{nc}(TS)]$ is given by
$\sigma \in \mathcal{O}_d^*(C)$ if and only if one of the following conditions is satisfied:

- $\exists n \in I\!N : \exists C_1, \ldots, C_n \in Conf : \exists \alpha_1, \ldots, \alpha_{n+1} \in TA : \exists \theta \in Subst :$
  $C - \alpha_1 \rightarrow_d C_1 - \alpha_2 \rightarrow_d \cdots - \alpha_n \rightarrow_d C_n - \alpha_{n+1} \rightarrow_d [E, \theta] \wedge$
  $\sigma = < \alpha_1, < \alpha_2, \cdots < \alpha_n, \alpha_{n+1} > \cdots >>$
- $\exists C_1, \ldots \in Conf : \exists \alpha_1, \ldots \in TA :$
  $C - \alpha_1 \rightarrow_d C_1 - \alpha_2 \rightarrow_d \cdots \wedge \sigma = < \alpha_1, < \alpha_2, \cdots >>$

End 5.5

The well-definedness of the intermediate operational semantics follows from the compactly branching property and the continuity property of the labelled transition system as is described in property A.3 of the appendix.

Lemma 5.6
The mapping $\mathcal{O}_d^*$ is well-defined.
Proof
The proof of this lemma can be found in lemma A.6 of the appendix.
End 5.6

Next we relate the operational models $\mathcal{O}_d$ and $\mathcal{O}_d^*$ via their labelled transition systems. From the following property we can deduce that each step according to the labelled transition system describing $\mathcal{O}_d$ can be mimicked by a step according to the labelled transition system describing $\mathcal{O}_d^*$ and vice versa.

Property 5.7
For all $[s, \theta] \in Conf$, $\bar{s} \in Stat_E$ and $\alpha \in TA$

$$s\theta - \alpha \rightarrow_d \bar{s} \Leftrightarrow \exists [\bar{s}', \theta'] \in Conf_E : [s, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$$

Proof

We prove this property using induction on the complexity of statement s.

1 Let $s \equiv (a, e)$.
$(a, e)\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\alpha = (a, \mathcal{V}(e\theta)) \wedge \bar{s} = E$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [(a, e), \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

2 Let $s \equiv x$ and $(x, g) \in d$.
$x\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$g\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [g, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [x, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

3 Let $s \equiv s_1; s_2$.
$(s_1; s_2)\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$s_1\theta; s_2\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\exists \bar{s}' \in Stat_E : s_1\theta - \alpha \rightarrow_d \bar{s}' \wedge \bar{s} = \bar{s}'; s_2\theta$
$\Leftrightarrow$
$\exists \bar{s}' \in Stat_E : \exists[\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s}' = \bar{s}''\theta' \wedge \bar{s} = \bar{s}'; s_2\theta$
$\Leftrightarrow$
$\exists[\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = \bar{s}''\theta'; s_2\theta$
$\Leftrightarrow$ property 5.4: $\theta \sqsubseteq \theta'$
$\exists[\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = (\bar{s}''; s_2)\theta'$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [s_1; s_2, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

4 Let $s \equiv s_1 \cup s_2$.
$(s_1 \cup s_2)\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$s_1\theta \cup s_2\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$s_1\theta - \alpha \rightarrow_d \bar{s} \vee s_2\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta' \vee$
$\exists[\bar{s}', \theta'] \in Conf_E : [s_2, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$
$\Leftrightarrow$
$\exists[\bar{s}', \theta'] \in Conf_E : [s_1 \cup s_2, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

5 Let $s \equiv s_1 \parallel s_2$.
$(s_1 \parallel s_2)\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$s_1\theta \parallel s_2\theta - \alpha \rightarrow_d \bar{s}$

$\Leftrightarrow$
$\exists \bar{s}' \in Stat_E : s_1\theta - \alpha \rightarrow_d \bar{s}' \wedge \bar{s} = \bar{s}' \parallel s_2\theta \vee \exists \bar{s}' \in Stat_E : s_2\theta - \alpha \rightarrow_d \bar{s}' \wedge \bar{s} = s_1\theta \parallel \bar{s}'$
$\Leftrightarrow$
$\exists \bar{s}' \in Stat_E : \exists [\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s}' = \bar{s}''\theta' \wedge \bar{s} = \bar{s}' \parallel s_2\theta \vee$
$\exists \bar{s}' \in Stat_E : \exists [\bar{s}'', \theta'] \in Conf_E : [s_2, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s}' = \bar{s}''\theta' \wedge \bar{s} = s_1\theta \parallel \bar{s}'$
$\Leftrightarrow$
$\exists [\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = \bar{s}''\theta' \parallel s_2\theta \vee$
$\exists [\bar{s}'', \theta'] \in Conf_E : [s_2, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = s_1\theta \parallel \bar{s}''\theta'$
$\Leftrightarrow$ property 5.4: $\theta \sqsubseteq \theta'$
$\exists [\bar{s}'', \theta'] \in Conf_E : [s_1, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = (\bar{s}'' \parallel s_2)\theta' \vee$
$\exists [\bar{s}'', \theta'] \in Conf_E : [s_2, \theta] - \alpha \rightarrow_d [\bar{s}'', \theta'] \wedge \bar{s} = (s_1 \parallel \bar{s}'')\theta'$
$\Leftrightarrow$
$\exists [\bar{s}', \theta'] \in Conf_E : [s_1 \parallel s_2, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

6 Let $s \equiv \int_{t \in T} s$ and $tvar(\theta) = \{t_1, \ldots, t_n\}$.
$(\int_{t \in T} s)\theta - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\exists r \in T : s\theta[t/r] - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$ $t_{n+1} \notin tvar(\theta)$
$\exists r \in T : s[t/t_{n+1}]\theta[t_{n+1}/r] - \alpha \rightarrow_d \bar{s}$
$\Leftrightarrow$
$\exists r \in T : \exists [\bar{s}', \theta'] \in Conf_E : [s[t/t_{n+1}], \theta[t_{n+1}/r]] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$
$\Leftrightarrow$
$\exists [\bar{s}', \theta'] \in Conf_E : \exists r \in T : [s[t/t_{n+1}], \theta[t_{n+1}/r]] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$
$\Leftrightarrow$
$\exists [\bar{s}', \theta'] \in Conf_E : [\int_{t \in T} s, \theta] - \alpha \rightarrow_d [\bar{s}', \theta'] \wedge \bar{s} = \bar{s}'\theta'$

End 5.7

Having related the labelled transition systems, which describe the operational models $\mathcal{O}_d$ and $\mathcal{O}_d^*$, we can relate these models.

Lemma 5.8
For all $[s, \theta] \in Conf$
$$\mathcal{O}_d^*([s, \theta]) = \mathcal{O}_d(s\theta)$$
Proof
We can deduce this immediately from property 5.7.
End 5.8

Now we introduce an intermediate denotational model $\mathcal{D}_d^*$. This denotational model is defined as the fixed point of a higher-order transformation $\Psi_{\mathcal{D}^*}$.

Definition 5.9
The mapping $\Psi_{\mathcal{D}^*} : [Conf \rightarrow \mathcal{P}_{nc}(TS)] \rightarrow [Conf \rightarrow \mathcal{P}_{nc}(TS)]$ is given by
$$\Psi_{\mathcal{D}^*}(F)([(a, e), \theta]) = \{(a, \mathcal{V}(e\theta))\}$$
$$\Psi_{\mathcal{D}^*}(F)([x, \theta]) = \Psi_{\mathcal{D}^*}(F)([g, \theta]) \qquad (x, g) \in d$$
$$\Psi_{\mathcal{D}^*}(F)([s_1; s_2, \theta]) = \Psi_{\mathcal{D}^*}(F)([s_1, \theta]); F([s_2, \theta])$$
$$\Psi_{\mathcal{D}^*}(F)([s_1 * s_2, \theta]) = \Psi_{\mathcal{D}^*}(F)([s_1, \theta]) * \Psi_{\mathcal{D}^*}(F)([s_2, \theta]) \qquad * \in \{\cup, \parallel\}$$
$$\Psi_{\mathcal{D}^*}(F)([\textstyle\int_{t \in T} s, \theta]) = \bigcup \{\Psi_{\mathcal{D}^*}(F)([s[t/t_{n+1}], \theta[t_{n+1}/r]]) \mid r \in T\} \quad tvar(\theta) = \{t_1, \ldots, t_n\}$$
End 5.9

The well-definedness of the mapping $\Psi_{\mathcal{D}^*}$ follows from the fact that that expressions are continuous functions, time sets are non-empty compact sets and the semantic operators are continuous.

Property 5.10
The mapping $\Psi_{\mathcal{D}^*}$ is well-defined.
Proof
Similar to property 4.20 except for the continuity proof in the case of integration.

5 Let $s \equiv \int_{t\in T} s$. From definition 2.11 we can derive that there exists a subsequence $\{\theta_{i+k}\}_i$ such that $tvar(\theta_{i+k}) = \{t_1, \ldots, t_n\}$.
$\lim_{i\to\infty} \Psi_{\mathcal{D}^*}(F)([\int_{t\in T} s, \theta_i])$
$=$
$\lim_{i\to\infty} \Psi_{\mathcal{D}^*}(F)([\int_{t\in T} s, \theta_{i+k}])$
$=$
$\lim_{i\to\infty} \bigcup\{\Psi_{\mathcal{D}^*}(F)([s[t/t_{n+1}], \theta_{i+k}[t_{n+1}/r]]) \mid r \in T\}$
$=$
$\bigcup\{\lim_{i\to\infty} \Psi_{\mathcal{D}^*}(F)([s[t/t_{n+1}], \theta_{i+k}[t_{n+1}/r]]) \mid r \in T\}$
$=$
$\bigcup\{\Psi_{\mathcal{D}^*}(F)([s[t/t_{n+1}], \theta[t_{n+1}/r]]) \mid r \in T\}$
$=$
$\Psi_{\mathcal{D}^*}(F)([\int_{t\in T} s, \theta])$

End 5.10

Next we prove that $\Psi_{\mathcal{D}^*}$ is a contraction. The contractivity of $\Psi_{\mathcal{D}^*}$ follows from the contractivity properties of the semantic operators.

Property 5.11
The mapping $\Psi_{\mathcal{D}^*}$ is a contraction.
Proof
Similar to property 4.21.
End 5.11

Because $\Psi_{\mathcal{D}^*}$ is a contraction on a complete metric space, this mapping has a unique fixed point, which will be denoted by $\mathcal{D}_d^*$.

Corollary 5.12
The mapping $\mathcal{D}_d^* : [Conf \to \mathcal{P}_{nc}(TS)]$ given by

$$\begin{array}{ll}
\mathcal{D}_d^*([(a,e),\theta]) = \{(a, \mathcal{V}(e\theta))\} & \\
\mathcal{D}_d^*([x,\theta]) = \mathcal{D}_d^*([g,\theta]) & (x,g) \in d \\
\mathcal{D}_d^*([s_1 * s_2,\theta]) = \mathcal{D}_d^*([s_1,\theta]) * \mathcal{D}_d^*([s_2,\theta]) & * \in \{;, \cup, \|\} \\
\mathcal{D}_d^*([\int_{t\in T} s,\theta]) = \bigcup\{\mathcal{D}_d^*([s[t/t_{n+1}], \theta[t_{n+1}/r]]) \mid r \in T\} & tvar(\theta) = \{t_1, \ldots, t_n\}
\end{array}$$

is well-defined.
End 5.12

The denotational models $\mathcal{D}_d^*$ and $\mathcal{D}_d$ are related by proving that $\mathcal{D}_d^*$ is a fixed point of the higher-order transformation $\Psi_{\mathcal{D}}$ defining $\mathcal{D}_d$.

Lemma 5.13
For all $[s,\theta] \in Conf$

$\mathcal{D}_d^*([s,\theta]) = \mathcal{D}_d(s)(\theta)$

Proof

We prove that $\mathcal{D}_d^*$ is a fixed point of the curried version of $\Psi_{\mathcal{D}}$, with induction on the complexity of statement s.

1 Let $s \equiv (a,e)$.
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([(a,e),\theta])$
$=$
$\{(a,\mathcal{V}(e\theta))\}$
$=$
$\mathcal{D}_d^*([(a,e),\theta])$

2 Let $s \equiv x$ and $(x,g) \in d$.
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([x,\theta])$
$=$
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([g,\theta])$
$=$
$\mathcal{D}_d^*([g,\theta])$
$=$
$\mathcal{D}_d^*([x,\theta])$

3 Let $s \equiv s_1;s_2$.
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s_1;s_2,\theta])$
$=$
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s_1,\theta]);\mathcal{D}_d^*([s_2,\theta])$
$=$
$\mathcal{D}_d^*([s_1,\theta]);\mathcal{D}_d^*([s_2,\theta])$
$=$
$\mathcal{D}_d^*([s_1;s_2,\theta])$

4 Let $s \equiv s_1 * s_2$ and $* \in \{\cup, \|\}$.
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s_1 * s_2,\theta])$
$=$
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s_1,\theta]) * \Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s_2,\theta])$
$=$
$\mathcal{D}_d^*([s_1,\theta]) * \mathcal{D}_d^*([s_2,\theta])$
$=$
$\mathcal{D}_d^*([s_1 * s_2,\theta])$

5 Let $s \equiv \int_{t\in T} s$ and $tvar(\theta) = \{t_1,\ldots,t_n\}$.
$\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([\int_{t\in T} s,\theta])$
$=$
$\bigcup\{\Psi_{\mathcal{D}}(\mathcal{D}_d^*)([s,\theta[t/r]]) \mid r \in T\}$
$=$
$\bigcup\{\mathcal{D}_d^*([s,\theta[t/r]]) \mid r \in T\}$
$=$
$\bigcup\{\mathcal{D}_d^*([s[t/t_{n+1}],\theta[t_{n+1}/r]]) \mid r \in T\}$
$=$
$\mathcal{D}_d^*([\int_{t\in T} s,\theta])$

End 5.13

We relate the intermediate models $\mathcal{O}_d^*$ and $\mathcal{D}_d^*$ by proving that they are both a fixed point of the higher-order transformation $\Psi_{\mathcal{O}^*\mathcal{D}^*}$. This mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ is related to a declaration similar to the denotational models.

Definition 5.14
The mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*} : [Conf \to \mathcal{P}_{nc}(TS)] \to [Conf \to \mathcal{P}_{nc}(TS)]$ is given by

$$\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C) = \{<\alpha, \rho>\mid C - \alpha \to_d C' \wedge \rho \in F(C')\} \cup \{\alpha \mid C - \alpha \to_d [E, \theta']\}$$

End 5.14

The well-definedness of the higher-order transformation $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ follows from the compactly branching property and the continuity property of the labelled transition system.

Lemma 5.15
The mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ is well-defined.
Proof
The proof of this property can be found in lemma A.7 of the appendix.
End 5.15

To conclude that $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ has a unique fixed point, we have to prove that this mapping is a contraction.

Property 5.16
The mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ is a contraction.
Proof

$$d_{\mathcal{P}_{nc}(TS)}(\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C), \Psi_{\mathcal{O}^*\mathcal{D}^*}(G)(C))$$

$$=$$

$$d_{\mathcal{P}_{nc}(TS)}\left(\begin{array}{l}\{<\alpha, \rho>\mid C - \alpha \to_d C' \wedge \rho \in F(C')\} \cup \{\alpha \mid C - \alpha \to_d [E, \theta']\} \\ \{<\alpha, \rho>\mid C - \alpha \to_d C' \wedge \rho \in G(C')\} \cup \{\alpha \mid C - \alpha \to_d [E, \theta']\}\end{array}\right)$$

$$\leq$$

$$\tfrac{1}{2} d_{[Conf \to \mathcal{P}_{nc}(TS)]}(F, G)$$

End 5.16

First we prove that the intermediate operational model $\mathcal{O}_d^*$ is a fixed point of $\Psi_{\mathcal{O}^*\mathcal{D}^*}$.

Lemma 5.17
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{O}_d^*) = \mathcal{O}_d^*$
Proof

$$\sigma \in \Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{O}_d^*)(C)$$

$$\Leftrightarrow$$

$$\sigma \in \{<\alpha, \rho>\mid C - \alpha \to_d C' \wedge \rho \in \mathcal{O}_d^*(C')\} \cup \{\alpha \mid C - \alpha \to_d [E, \theta']\}$$

$$\Leftrightarrow$$

$$\exists C' \in Conf : \exists \alpha \in TA : \exists \rho \in TS : C - \alpha \to_d C' \wedge \rho \in \mathcal{O}_d^*(C') \wedge \sigma = <\alpha, \rho> \vee$$
$$\exists \theta' \in Subst : \exists \alpha \in TA : C - \alpha \to_d [E, \theta'] \wedge \sigma = \alpha$$

$$\Leftrightarrow$$

$$\sigma \in \mathcal{O}_d^*(C)$$

End 5.17

Also the intermediate denotational model $\mathcal{D}_d^*$ is a fixed point of $\Psi_{\mathcal{O}^*\mathcal{D}^*}$.

Lemma 5.18
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*) = \mathcal{D}_d^*$
Proof
We prove $\forall [s, \theta] \in Conf : \Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s, \theta]) = \mathcal{D}_d^*([s, \theta])$ using induction on the complexity of statement s.

1 Let $s \equiv (a, e)$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([(a, e), \theta])$
$=$
$\{< \alpha, \rho >| \; [(a, e), \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [(a, e), \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\{(a, \mathcal{V}(e\theta))\}$
$=$
$\mathcal{D}_d^*([(a, e), \theta])$

2 Let $s \equiv x$ and $(x, g) \in d$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([x, \theta])$
$=$
$\{< \alpha, \rho >| \; [x, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [x, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\{< \alpha, \rho >| \; [g, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [g, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([g, \theta])$
$=$
$\mathcal{D}_d^*([g, \theta])$
$=$
$\mathcal{D}_d^*([x, \theta])$

3 Let $s \equiv s_1; s_2$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1; s_2, \theta])$
$=$
$\{< \alpha, \rho >| \; [s_1; s_2, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_1; s_2, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s'; s_2, \theta'])\} \cup$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta'])\}$
$=$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s', \theta']); \mathcal{D}_d^*([s_2, \theta'])\} \cup$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta'])\}$
$=$ property 5.4: $\theta \sqsubseteq \theta'$ and $tvar(s_2) \subseteq tvar(\theta)$ so $\mathcal{D}_d^*([s_2, \theta']) = \mathcal{D}_d^*([s_2, \theta])$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s', \theta']); \mathcal{D}_d^*([s_2, \theta])\} \cup$
$\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta])\}$
$=$
$(\{< \alpha, \rho >| \; [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s', \theta'])\} \cup \{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta']\}); \mathcal{D}_d^*([s_2, \theta])$
$=$
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1, \theta]); \mathcal{D}_d^*([s_2, \theta])$
$=$
$\mathcal{D}_d^*([s_1, \theta]); \mathcal{D}_d^*([s_2, \theta])$
$=$
$\mathcal{D}_d^*([s_1; s_2, \theta])$

4 Let $s \equiv s_1 \cup s_2$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1 \cup s_2, \theta])$
$=$
$\{< \alpha, \rho >| \ [s_1 \cup s_2, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_1 \cup s_2, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\{< \alpha, \rho >| \ [s_1, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta']\} \cup$
$\{< \alpha, \rho >| \ [s_2, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_2, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1, \theta]) \cup \Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_2, \theta])$
$=$
$\mathcal{D}_d^*([s_1, \theta]) \cup \mathcal{D}_d^*([s_2, \theta])$
$=$
$\mathcal{D}_d^*([s_1 \cup s_2, \theta])$

5 Let $s \equiv s_1 \parallel s_2$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1 \parallel s_2, \theta])$
$=$
$\{< \alpha, \rho >| \ [s_1 \parallel s_2, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_1 \parallel s_2, \theta] - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\{< \alpha, \rho >| \ [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s' \parallel s_2, \theta'])\} \cup$
$\{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta'])\} \cup$
$\{< \alpha, \rho >| \ [s_2, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1 \parallel s', \theta'])\} \cup$
$\{\alpha \mid [s_2, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1, \theta'])\}$
$=$
$\{< \alpha, \rho >| \ [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s', \theta']) \parallel \mathcal{D}_d^*([s_2, \theta'])\} \cup$
$\{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta'])\} \cup$
$\{< \alpha, \rho >| \ [s_2, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1, \theta']) \parallel \mathcal{D}_d^*([s', \theta'])\} \cup$
$\{\alpha \mid [s_2, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1, \theta'])\}$
$=$ property 5.4: $\theta \sqsubseteq \theta'$ and $tvar(s_i) \subseteq tvar(\theta)$ so $\mathcal{D}_d^*([s_i, \theta']) = \mathcal{D}_d^*([s_i, \theta]$
$\{< \alpha, \rho >| \ [s_1, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s', \theta']) \parallel \mathcal{D}_d^*([s_2, \theta])\} \cup$
$\{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_2, \theta])\} \cup$
$\{< \alpha, \rho >| \ [s_2, \theta] - \alpha \rightarrow_d [s', \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1, \theta]) \parallel \mathcal{D}_d^*([s', \theta'])\} \cup$
$\{\alpha \mid [s_2, \theta] - \alpha \rightarrow_d [E, \theta'] \wedge \rho \in \mathcal{D}_d^*([s_1, \theta])\}$
$=$
$(\{< \alpha, \rho >| \ [s_1, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_1, \theta] - \alpha \rightarrow_d [E, \theta']\}) \coprod \mathcal{D}_d^*([s_2, \theta]) \cup$
$(\{< \alpha, \rho >| \ [s_2, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [s_2, \theta] - \alpha \rightarrow_d [E, \theta']\}) \coprod \mathcal{D}_d^*([s_1, \theta])$
$=$
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1, \theta]) \coprod \mathcal{D}_d^*([s_2, \theta]) \cup \Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s_1, \theta]) \coprod \mathcal{D}_d^*([s_1, \theta])$
$=$
$\mathcal{D}_d^*([s_1, \theta]) \coprod \mathcal{D}_d^*([s_2, \theta]) \cup \mathcal{D}_d^*([s_2, \theta]) \coprod \mathcal{D}_d^*([s_1, \theta])$
$=$
$\mathcal{D}_d^*([s_1, \theta]) \parallel \mathcal{D}_d^*([s_2, \theta])$
$=$
$\mathcal{D}_d^*([s_1 \parallel s_2, \theta])$

6 Let $s \equiv \int_{t \in T} s$ and $tvar(\theta) = \{t_1, \ldots, t_n\}$.
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([\int_{t \in T} s, \theta])$
$=$
$\{< \alpha, \rho >| \ [\int_{t \in T} s, \theta] - \alpha \rightarrow_d C' \wedge \rho \in \mathcal{D}_d^*(C')\} \cup \{\alpha \mid [\int_{t \in T} s, \theta] - \alpha \rightarrow_d [E, \theta']\}$

$=$

$\{<\alpha,\rho>\mid [s[t/t_{n+1}],\theta[t_{n+1}/r]] - \alpha \to_d C' \wedge \rho \in \mathcal{D}_d^*(C') \wedge r \in T\} \cup$
$\{\alpha \mid [s[t/t_{n+1}],\theta[t_{n+1}/r]] - \alpha \to_d [E,\theta'] \wedge r \in T\}$

$=$

$\bigcup\{\ \{<\alpha,\rho>\mid [s[t/t_{n+1}],\theta[t_{n+1}/r]] - \alpha \to_d C' \wedge \rho \in \mathcal{D}_d^*(C')\}\cup$
$\quad \{\alpha \mid [s[t/t_{n+1}],\theta[t_{n+1}/r]] - \alpha \to_d [E,\theta']\} \mid r \in T\}$

$=$

$\bigcup\{\Psi_{\mathcal{O}^*\mathcal{D}^*}(\mathcal{D}_d^*)([s[t/t_{n+1}],\theta[t_{n+1}/r]]) \mid r \in T\}$

$=$

$\bigcup\{\mathcal{D}_d^*([s[t/t_{n+1}],\theta[t_{n+1}/r]]) \mid r \in T\}$

$=$

$\mathcal{D}_d^*([\int_{t\in T} s,\theta])$

End 5.18

Because $\mathcal{O}_d^*$ and $\mathcal{D}_d^*$ are both fixed points of the higher-order transformation $\Psi_{\mathcal{O}^*\mathcal{D}^*}$, which is a contraction on a complete metric space, we have that $\mathcal{O}_d^*$ and $\mathcal{D}_d^*$ are equivalent due to Banach's fixed point theorem.

Corollary 5.19
$\mathcal{O}_d^* = \mathcal{D}_d^*$
End 5.19

We conclude this section by collecting all the relations between the various models into an equivalence proof.

Theorem 5.20
$\mathcal{O} = \mathcal{D}$
Proof
$\mathcal{O}((d,s))$
$=$
$\mathcal{O}_d(s)$
$=$ lemma 5.8
$\mathcal{O}_d^*([s,\epsilon])$
$=$ corollary 5.19
$\mathcal{D}_d^*([s,\epsilon])$
$=$ lemma 5.13
$\mathcal{D}_d(s)(\epsilon)$
$=$
$\mathcal{D}((d,s))$
End 5.20

Conclusions

An operational and a denotational semantic model have been presented for a real-time programming language incorporating the concept of integration. As we have seen, a restricted form of unbounded non-determinism can be specified by means of integration. Because the semantic operators and the semantic models have been defined using higher-order transformations, we were able to describe infinite behaviour. The operational and denotational semantics have been proved equivalent. Banach's fixed point theorem and Michael's theorem have been used fruitfully to define and to compare those models.

We expect it to be possible to relate the denotational semantics defined by Reed and Roscoe [41, 42, 43] to a denotational model based on the denotational model presented in this paper following the lines of [12]. We have the strong feeling that it is possible to extend the language with communication and global non-determinism [11] and to define a branching time model [8] for this language. Enriching the language with delays [37, 43] and a parameter mechanism provided to procedure variables causes no serious problems. However, extending the language with priorities [23, 45] and enforced deadlines for atomic actions may cause discontinuity of semantic operators.

Acknowledgements

I would like to thank the members of the Amsterdam Concurrency Group, consisting of Jaco de Bakker, Frank de Boer, Arie de Bruin, Eiichi Horita, Jean-Marie Jacquet, Peter Knijnenburg, Joost Kok, Jan Rutten, Erik de Vink and Jeroen Warmerdam, for comments on previous work. In particular, I thank Erik de Vink for his comments and suggestions during the evolvement of this paper. Also I would like to thank Jos Baeten for giving me the opportunity to present my work to some of his students.

References

[1] P. America and J.J.M.M. Rutten. *Solving Reflexive Domain Equations in a Category of Complete Metric Spaces.* Journal of Computer and System Sciences 39 (1989), 343-375.

[2] K.R. Apt. *Recursive Assertions and Parallel Programs.* Acta Informatica 15 (1983), 219-232.

[3] K.R. Apt and G.D. Plotkin. *Countable Non-Determinism and Random Assignment.* Journal of the ACM 33 (1986), 724-767.

[4] J.C.M. Baeten and J.A. Bergstra. *Real-Time Process Algebra.* Report P8916, Programming Research Group, University of Amsterdam, Amsterdam (1989).

[5] J.W. de Bakker. *Mathematical Theory of Program Correctness.* Prentice-Hall International (1980).

[6] J.W. de Bakker. *Comparative Semantics for Flow of Control in Logic Programming without Logic.* Report CS-8840, Centre for Mathematics and Computer Science, Amsterdam (1988). To appear in Information and Computation.

[7] J.W. de Bakker. *Designing Concurrency Semantics.* Proceedings 11th World Computer Congress (G.X. Ritter, ed.), North Holland (1989), 591-598.

[8] J.W. de Bakker, J.A. Bergstra, J.W. Klop and J.-J.Ch. Meyer. *Linear Time and Branching Time Semantics for Recursion with Merge.* Theoretical Computer Science 34 (1984), 135-156.

[9] J.W. de Bakker and J.N. Kok. *Towards a Uniform Topological Treatment of Streams and Functions on Streams.* Proceedings 12th International Colloquium on Automata, Languages and Programming (W. Brauer, eds.), Lecture Notes in Computer Science 194, Springer (1985).

[10] J.W. de Bakker and J.N. Kok. *Comparative Metric Semantics for Concurrent Prolog.* Theoretical Computer Science 75 (1990), 15-43.

[11] J.W. de Bakker, J.N. Kok, J.-J.Ch. Meyer, E.-R. Olderog and J.I. Zucker. *Contrasting Themes in the Semantics of Imperative Concurrency.* Current Trends in Concurrency: Overviews and Tutorials (J.W. de Bakker, W.P. de Roever and G. Rozenberg, eds.), Lecture Notes in Computer Science 224, Springer (1986), 51-121.

[12] J.W. de Bakker, J.-J.Ch. Meyer and E.-R. Olderog. *Infinite Streams and Finite Observations in the Semantics of Uniform Concurrency.* Theoretical Computer Science 49 (1987), 87-112.

[13] J.W. de Bakker and J.-J.Ch. Meyer. *Metric Semantics for Concurrency.* BIT 28 (1988), 504-529.

[14] J.W. de Bakker and J.J.M.M. Rutten. *Concurrency Semantics based on Metric Domain Equations.* Report CS-8954, Centre for Mathematics and Computer Science, Amsterdam (1989).

[15] J.W. de Bakker and E.P. de Vink. *CCS for OO and LP.* Proceedings TAPSOFT 91, Lecture Notes in Computer Science, Springer (1991).

[16] J.W. de Bakker and J.I. Zucker. *Processes and the Denotational Semantics of Concurrency.* Information and Control 54 (1982), 70-120.

[17] S. Banach. *Sur les Opérations dans les Ensembles Abstraits et leur Applications aux Equations Intégrales.* Fundamenta Mathematicae 3 (1922), 133-181.

[18] J.A. Bergstra and J.W. Klop. *Fixed Point Semantics in Process Algebra.* Report IW 206/82, Mathematical Centre, Amsterdam (1982).

[19] G. Berry and L. Cosserat. *The ESTEREL Synchronous Programming Language and its Semantics.* Proceedings CMU Seminar on Concurrency (S.D. Brookes, A.W. Roscoe and G. Winksel, eds.), Lecture Notes in Computer Science 197, Springer (1985), 389-448.

[20] F.S. de Boer, J.N. Kok, C. Palamidessi and J.J.M.M. Rutten. *From Failure to Success: Comparing a Denotational and a Declarative Semantics for Horn Clause Logic.* Proceedings Workshop on Semantics for Concurrency (M.Z. Kwiatkowska, M.W. Shields and R.M. Thomas, eds.), Leicester (1990), 38-60.

[21] F. van Breugel. *Semantic Models for a Language with Timed Atomic Actions.* Report IR-218, Vrije Universiteit, Amsterdam (1990).

[22] F. van Breugel. *Comparative Semantics for a Real-Time Programming Language with Integration.* Proceedings TAPSOFT 91, Lecture Notes in Computer Science, Springer (1991).

[23] R. Cleaveland and M. Hennessy. *Priorities in Process Algebra.* Information and Computation 87 (1990), 58-77.

[24] J.W. Davies and S.A. Schneider. *An Introduction to Timed CSP.* Technical Monograph PRG-75, Oxford University Computing Laboratory, Oxford (1989).

[25] J.W. Davies and S.A. Schneider. *An Extended Syntax for Timed CSP.* Report PRG-TR-4-90, Oxford University Computing Laboratory, Oxford (1990).

[26] R. Engelking. *General Topology*. Heldermann Verlag (1989).

[27] A. Goswami and M. Joseph. *Semantics of Real-Time Distributed Programs*. Proceedings International Conference on Concurrency (F.H. Vogt, ed.), Lecture Notes in Computer Science 335, Springer (1988), 292-306.

[28] J.F. Groote. *Specification and Verification of Real-Time Systems in ACP*. Proceedings 10th International IFIP WG 6.1, Symposium on Protocol Specification, Testing and Verification, Ottawa (1990), 259-270.

[29] J.F. Groote and F.W. Vaandrager. *Structured Operational Semantics and Bisimulation as a Congruence*. Proceedings 16th International Colloquium on Automata, Languages and Programming (G. Ausiello, M. Dezani-Ciancaglini and S. Ronchi Della Rocca, eds.), Lecture Notes in Computer Science 372, Springer (1989), 423-438.

[30] F. Hausdorff. *Grundzüge der Mengenlehre*. Leipzig (1914).

[31] M. Hennessy and G.D. Plotkin. *Full Abstraction for a Simple Parallel Programming Language*. Proceedings 8th Mathematical Foundations of Computer Science (J. Bečvař, ed.), Lecture Notes in Computer Science 74, Springer (1979), 108-120.

[32] M. Hennessy and T. Regan. *A Temporal Process Algebra*. Report 2-90, University of Sussex, Brighton (1990).

[33] W.H. Hesselink. *Interpretations of Recursion under Unbounded Non-Determinacy*. Theoretical Computer Science 59 (1988), 211-234.

[34] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International (1985).

[35] F. Jahanian and A. Mok. *Safety Analysis of Timing Properties in Real-Time Systems*. IEEE Transactions on Software Engineering 12 (1986), 890-904.

[36] J.N. Kok and J.J.M.M. Rutten. *Contractions in Comparing Concurrency Semantics*. Theoretical Computer Science 76 (1990), 179-222.

[37] R. Koymans, R.K. Shyamasundar, W.P. de Roever, R. Gerth and S. Arun-Kumar. *Compositional Semantics for Real-Time Distributed Computing*. Information and Computation 79 (1988), 210-256.

[38] K. Kuratowski. *Sur une Méthode de Métrisation Complète des Certains Espaces d'Ensembles Compacts*. Fundamenta Mathematicae 42 (1956), 114-138.

[39] E. Michael. *Topologies on Spaces of Subsets*. Transactions of the American Mathematical Society 71 (1951), 152-182.

[40] M. Nivat. *Infinite Words, Infinite Trees, Infinite Computations*. Foundations of Computer Science (J.W. de Bakker and J. van Leeuwen, eds.), Mathematical Centre Tracts 109, Mathematical Centre, Amsterdam (1979), 3-52.

[41] G.M. Reed. *A Hierarchy of Domains for Real-Time Distributed Computing*. Proceedings Mathematical Foundations of Programming Languages and Semantics (M. Main, A. Melton, M. Mislove and D. Schmidt, eds.), Lecture Notes in Computer Science 442, Springer (1991), 80-128.

[42] G.M. Reed and A.W. Roscoe. *Metric Spaces as Models for Real-Time Concurrency.* Proceedings Mathematical Foundations of Programming Languages and Semantics (M. Main, A. Melton, M. Mislove and D. Schmidt, eds.), Lecture Notes in Computer Science 298, Springer (1987), 331-343.

[43] G.M. Reed and A.W. Roscoe. *A Timed Model for Communicating Sequential Processes.* Theoretical Computer Science 58 (1988), 249-261.

[44] J.J.M.M. Rutten. *Semantic Correctness for a Parallel Object-Oriented Language.* SIAM Journal of Computation 19 (1990), 341-383.

[45] S.A. Smolka and B. Steffen. *Priority as Extremal Probability.* Proceedings Conference on Theories of Concurrency: Unification and Extension (J.C.M. Baeten and J.W. Klop, eds.), Lecture Notes in Computer Science 458 (1990), 456-466.

[46] J.A. Stankovic. *A Serious Problem for Next-Generation Systems.* Computer 21 (1988), 10-19.

[47] N. Wirth. *Towards a Discipline of Real-Time Programming.* Communications of the ACM 20 (1977), 577-583.

A Well-definedness proofs

In this appendix the well-definedness proofs of the mappings $\mathcal{O}_d^*$ (lemma 5.6) and $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ (lemma 5.15) are presented. Usually the well-definedness of a semantic mapping, which is related to a labelled transition system, follows from the fact that the labelled transition system is finitely branching, i.e. for all configurations C the set $\{(\alpha,\bar{C}) \mid C - \alpha \to_d \bar{C}\}$ is finite [36]. However our labelled transition system (definition 5.2) is not finitely branching. For example, the set

$$\{(\alpha,\bar{C}) \mid [\textstyle\int_{t\in[1.03,2.41]}(a,t),\epsilon] - \alpha \to_d \bar{C}\} = \{((a,r),[E,[t_1/r]]) \mid r \in [1.03,2.41]\}$$

is infinite. For our labelled transition system we have more involved properties (compactly branching property and continuity property), which are stated in the following definitions.

Definition A.1
A labelled transition system $(Conf, Label, \longrightarrow)$ is compactly branching whenever we have that for all $C \in Conf$ the set $\{(l,C') \mid C - l \to_d C'\}$ is compact.
End A.1

Note that if a labelled transition system is finitely branching then it is also compactly branching.

Definition A.2
A labelled transition system $(Conf, Label, \longrightarrow)$ is continuous whenever the mapping $Init : Conf \to \mathcal{P}(Label \times Conf)$ given by $Init(C) = \{(l,C') \mid C - l \to_d C'\}$ is continuous.
End A.2

If the configurations are endowed with a discrete metric then the corresponding labelled transition system is continuous. Our labelled transition system is compactly branching and continuous as is stated in the following property.

Property A.3
The labelled transition system defined in definition 5.2 is compactly branching and continuous.
Proof
We prove this property using induction on the complexity of statement s. We assume that $\lim_{i\to\infty}\theta_i = \theta$.

1 Let $s \equiv (a, e)$.
$\{(l, \bar{C}) \mid [(a, e), \theta] - l \rightarrow_d \bar{C}\}$
$=$
$\{((a, \mathcal{V}(e\theta)), E)\}$
This set is compact. Furthermore, we have that
$\lim_{i\to\infty}\{(l, \bar{C}) \mid [(a, e), \theta_i] - l \rightarrow_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{((a, \mathcal{V}(e\theta_i)), E)\}$
$=$ e is continuous
$\{((a, \mathcal{V}(e\theta)), E)\}$
$=$
$\{(l, \bar{C}) \mid [(a, e), \theta] - l \rightarrow_d \bar{C}\}$

2 Let $s \equiv x$ and $(x, g) \in d$.
$\{(l, \bar{C}) \mid [x, \theta] - l \rightarrow_d \bar{C}\}$
$=$
$\{(l, \bar{C}) \mid [g, \theta] - l \rightarrow_d \bar{C}\}$
This set is compact. We have that
$\lim_{i\to\infty}\{(l, \bar{C}) \mid [x, \theta_i] - l \rightarrow_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l, \bar{C}) \mid [g, \theta_i] - l \rightarrow_d \bar{C}\}$
$=$
$\{(l, \bar{C}) \mid [g, \theta] - l \rightarrow_d \bar{C}\}$
$=$
$\{(l, \bar{C}) \mid [x, \theta] - l \rightarrow_d \bar{C}\}$

3 Let $s \equiv s_1; s_2$.
$\{(l, \bar{C}) \mid [s_1; s_2, \theta] - l \rightarrow_d \bar{C}\}$
$=$
$\{(l, [\bar{s}; s_2, \theta']) \mid [s_1, \theta] - l \rightarrow_d [\bar{s}, \theta']\}$
This set is compact. Also we have that
$\lim_{i\to\infty}\{(l, \bar{C}) \mid [s_1; s_2, \theta_i] - l \rightarrow_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l, [\bar{s}; s_2, \theta']) \mid [s_1, \theta_i] - l \rightarrow_d [\bar{s}, \theta']\}$
$=$
$\{(l, [\bar{s}; s_2, \theta']) \mid [s_1, \theta] - l \rightarrow_d [\bar{s}, \theta']\}$
$=$
$\{(l, \bar{C}) \mid [s_1; s_2, \theta] - l \rightarrow_d \bar{C}\}$

4 Let $s \equiv s_1 \cup s_2$.
$\{(l, \bar{C}) \mid [s_1 \cup s_2, \theta] - l \rightarrow_d \bar{C}\}$
$=$
$\{(l, \bar{C}) \mid [s_1, \theta] - l \rightarrow_d \bar{C}\} \cup \{(l, \bar{C}) \mid [s_2, \theta] - l \rightarrow_d \bar{C}\}$

This set is compact. We have that
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [s_1 \cup s_2, \theta_i] - l \to_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [s_1, \theta_i] - l \to_d \bar{C}\} \cup \{(l,\bar{C}) \mid [s_2, \theta_i] - l \to_d \bar{C}\}$
$=$
$\{(l,\bar{C}) \mid [s_1, \theta] - l \to_d \bar{C}\} \cup \{(l,\bar{C}) \mid [s_2, \theta] - l \to_d \bar{C}\}$
$=$
$\{(l,\bar{C}) \mid [s_1 \cup s_2, \theta] - l \to_d \bar{C}\}$

5 Let $s \equiv s_1 \parallel s_2$.
$\{(l,\bar{C}) \mid [s_1 \parallel s_2, \theta] - l \to_d \bar{C}\}$
$=$
$\{(l,[\bar{s} \parallel s_2, \theta']) \mid [s_1, \theta] - l \to_d [\bar{s}, \theta']\} \cup \{(l,[s_1 \parallel \bar{s}, \theta']) \mid [s_2, \theta] - l \to_d [\bar{s}, \theta']\}$
This set is compact. Also we have that
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [s_1 \parallel s_2, \theta_i] - l \to_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l,[\bar{s} \parallel s_2, \theta']) \mid [s_1, \theta_i] - l \to_d [\bar{s}, \theta']\} \cup \{(l,[s_1 \parallel \bar{s}, \theta']) \mid [s_2, \theta_i] - l \to_d [\bar{s}, \theta']\}$
$=$
$\{(l,[\bar{s} \parallel s_2, \theta']) \mid [s_1, \theta] - l \to_d [\bar{s}, \theta']\} \cup \{(l,[s_1 \parallel \bar{s}, \theta']) \mid [s_2, \theta] - l \to_d [\bar{s}, \theta']\}$
$=$
$\{(l,\bar{C}) \mid [s_1 \parallel s_2, \theta] - l \to_d \bar{C}\}$

6 Let $s \equiv \int_{t\in T} s$ and $var(\theta) = \{t_1, \ldots, t_n\}$.
$\{(l,\bar{C}) \mid [\int_{t\in T} s, \theta] - l \to_d \bar{C}\}$
$=$
$\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C} \wedge r \in T\}$
$=$
$\bigcup\{\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$
For each $r \in T$ the set $\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\}$ is compact. Furthermore, the set $\{\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$ is compact, because T is compact and $\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\}$ is continuous in r. Michael's theorem gives us that the set $\bigcup\{\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$ is compact.
From definition 2.11 we can derive that there exists a subsequence $\{\theta_{i+k}\}_i$ such that $tvar(\theta_{i+k}) = \{t_1, \ldots, t_n\}$.
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [\int_{t\in T} s, \theta_i] - l \to_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [\int_{t\in T} s, \theta_{i+k}] - l \to_d \bar{C}\}$
$=$
$\lim_{i\to\infty}\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta_{i+k}[t_{n+1}/r]] - l \to_d \bar{C} \wedge r \in T\}$
$=$
$\lim_{i\to\infty} \bigcup\{\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta_{i+k}[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$
$=$
$\bigcup\{\lim_{i\to\infty}\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta_{i+k}[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$
$=$
$\bigcup\{\{(l,\bar{C}) \mid [s[t/t_{n+1}], \theta[t_{n+1}/r]] - l \to_d \bar{C}\} \mid r \in T\}$
$=$
$\{(l,\bar{C}) \mid [\int_{t\in T} s, \theta] - l \to_d \bar{C}\}$

End A.3

To prove the well-definedness of the mapping $\mathcal{O}_d^*$ we introduce a collection of mappings $\mathcal{O}_d^m$, which describe approximations of $\mathcal{O}_d^*$.

Definition A.4
The mapping $\mathcal{O}_d^m : [Conf \to \mathcal{P}_{nc}(TS)]$ is given by
$$\mathcal{O}_d^m(C) = \{<\alpha_1, \ldots <\alpha_{n-1}, \alpha_n> \ldots >\mid C - \alpha_1 \to_d C_1 - \alpha_2 \to_d \cdots - \alpha_n \to_d [E, \theta'] \wedge n \leq m\} \cup$$
$$\{<\alpha_1, \ldots <\alpha_{m-1}, \alpha_m> \ldots >\mid C - \alpha_1 \to_d C_1 - \alpha_2 \to_d \cdots - \alpha_m \to_d C_m\}$$
End A.4

We prove that all these mappings are well-defined using property A.3.

Property A.5
The mappings $\mathcal{O}_d^m$ are well-defined.
Proof
We prove this property with induction on m.

1 Let $m = 1$.
$\mathcal{O}_d^1(C)$
$=$
$\{\alpha_1 \mid C - \alpha_1 \to_d [E, \theta']\} \cup \{\alpha_1 \mid C - \alpha_1 \to_d C_1\}$
By inspection of the labelled transition system we can derive the non-emptiness of this set. The compactness of this set follows immediately from the compactly branching property of the labelled transition system. The continuity of $\mathcal{O}_d^1$ follows from the continuity property of the labelled transition system.

2 Let $m > 1$.
$\mathcal{O}_d^m(C)$
$=$
$\{<\alpha_1, \ldots <\alpha_{n-1}, \alpha_n> \ldots >\mid C - \alpha_1 \to_d C_1 - \alpha_2 \to_d \cdots - \alpha_n \to_d [E, \theta'] \wedge n \leq m\} \cup$
$\{<\alpha_1, \ldots <\alpha_{m-1}, \alpha_m> \ldots >\mid C - \alpha_1 \to_d C_1 - \alpha_2 \to_d \cdots - \alpha_m \to_d C_m\}$
$=$
$\{\alpha_1 \mid C - \alpha_1 \to_d [E, \theta']\} \cup$
$\{<\alpha_1, \ldots <\alpha_{n-1}, \alpha_n> \ldots >\mid C - \alpha_1 \to_d C_1 \wedge C_1 - \alpha_2 \to_d \cdots - \alpha_n \to_d [E, \theta'] \wedge n \leq m\} \cup$
$\{<\alpha_1, \ldots <\alpha_{m-1}, \alpha_m> \ldots >\mid C - \alpha_1 \to_d C_1 \wedge C_1 - \alpha_2 \to_d \cdots - \alpha_m \to_d C_m\}$
$=$
$\{\alpha_1 \mid C - \alpha_1 \to_d [E, \theta']\} \cup \{<\alpha_1, \rho>\mid C - \alpha_1 \to_d C_1 \wedge \rho \in \mathcal{O}_d^{m-1}(C_1)\}$
By inspection of the labelled transition system we can derive the non-emptiness of this set. The set $\{\alpha_1 \mid C - \alpha_1 \to_d [E, \theta']\}$ is compact. Also the set $\{<\alpha_1, \rho>\mid \rho \in \mathcal{O}_d^{m-1}(C_1)\}$ is compact. Because $\mathcal{O}_d^{m-1}$ is continuous and the labelled transition system is compactly branching, the set $\{\{<\alpha_1, \rho>\mid \rho \in \mathcal{O}_d^{m-1}(C_1)\} \mid C - \alpha_1 \to_d C_1\}$ is compact. Michael's theorem gives us that the set $\{<\alpha_1, \rho>\mid C - \alpha_1 \to_d C_1 \wedge \rho \in \mathcal{O}_d^{m-1}(C_1)\}$ is compact. The continuity of $\mathcal{O}_d^m$ follows immediately from the the continuity of $\mathcal{O}_d^{m-1}$ and the fact that the labelled transition system is continuous.

End A.5

Because the mappings $\mathcal{O}_d^m$ are well-defined, we can conclude that $\mathcal{O}_d^*$ is well-defined.

Lemma A.6
The mapping $\mathcal{O}_d^*$ is well-defined.

Proof
The sequence $\{\mathcal{O}_d^m\}_m$ is a Cauchy sequence : $\forall N \in I\!N : \forall m > N : \forall n > N : d_{\mathcal{P}_{nc}(T)}(\mathcal{O}_d^m, \mathcal{O}_d^n) \leq 2^{-N}$. Furthermore, we have that $\lim_{m\to\infty} \mathcal{O}_d^m = \mathcal{O}_d^*$.
End A.6

We conclude this appendix with the well-definedness proof of the higher-order mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*}$. Also in this proof we will use property A.3.

Lemma A.7
The mapping $\Psi_{\mathcal{O}^*\mathcal{D}^*}$ is well-defined.
Proof
By inspection of the labelled transition system we can immediately derive the non-emptiness of the set $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C)$.

Next we have to prove that the set $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C)$ is compact. Let $\{\sigma_i\}_i$ be a sequence in $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C)$. Then there exits a subsequence $\{\sigma_{f(i)}\}_i$ in one of the following sets.

1 $\{< \alpha, \rho >\mid C - \alpha \rightarrow_d C' \wedge \rho \in F(C')\}$

2 $\{\alpha \mid C - \alpha \rightarrow_d [E, \theta']\}$

In the first case we have that $\sigma_{f(i)} =< \alpha_{f(i)}, \rho_{f(i)} >$ where $C - \alpha_{f(i)} \rightarrow_d C'_{f(i)}$ and $\rho_{f(i)} \in F(C'_{f(i)})$. Because the labelled transition system is compactly branching, the sequence $\{(\alpha_{f(i)}, C'_{f(i)})\}_i$ has a converging subsequence $\{(\alpha_{f(g(i))}, C'_{f(g(i))})\}_i$, which converges to some (α, C') such that $C - \alpha \rightarrow_d C'$. Because F is continuous, $\{F(C'_{f(g(i))})\}_i$ converges to $F(C')$. For each $\rho_{f(g(i))} \in F(C'_{f(g(i))})$ we can find a $\rho'_i \in F(C')$ such that $d_{TS}(\rho_{f(g(i))}, \rho'_i) \leq 2 d_{\mathcal{P}_{nc}(TS)}(F(C'_{f(g(i))}), F(C'))$. Because $F(C')$ is compact, the sequence $\{\rho'_i\}_i$ has a converging subsequence $\{\rho'_{h(i)}\}_i$, which converges to some $\rho \in F(C')$. We have that the sequence $\{\rho_{f(g(h(i)))}\}_i$ also converges to ρ, because $d_{TS}(\rho_{f(g(h(i)))}, \rho) \leq d_{TS}(\rho_{f(g(h(i)))}, \rho'_{h(i)}) + d_{TS}(\rho'_{h(i)}, \rho) \leq 2 d_{\mathcal{P}_{nc}(TS)}(F(C'_{f(g(h(i)))}), F(C')) + d_{TS}(\rho'_{h(i)}, \rho)$. So $\{\sigma_{f(g(h(i)))}\}_i$ converges to $< \alpha, \rho >$, which is an element of $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C)$. In the second case, we have that $\sigma_{f(i)} = \alpha_{f(i)}$ where $C - \alpha_{f(i)} \rightarrow_d [E, \theta'_{f(i)}]$. Because the labelled transition system is compactly branching, the sequence $\{(\alpha_{f(i)}, [E, \theta'_{f(i)}])\}_i$ has a converging subsequence $\{(\alpha_{f(g(i))}, [E, \theta'_{f(g(i))}])\}_i$, which converges to $(\alpha, [E, \theta'])$ where $C - \alpha \rightarrow_d [E, \theta']$. So $\{\sigma_{f(g(i))}\}_i$ converges to α, which is an element of $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C)$.

Finally, we have to prove that $\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)$ is continuous. Assume that $\lim_{i\to\infty} C_i = C$, then we have

$\lim_{i\to\infty} \Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C_i)$
$=$
$\lim_{i\to\infty}\{< \alpha, \rho >\mid C_i - \alpha \rightarrow_d C' \wedge \rho \in F(C')\} \cup \{\alpha \mid C_i - \alpha \rightarrow_d [E, \theta']\}$
$=$ the labelled transition system is continuous
$\{< \alpha, \rho >\mid C - \alpha \rightarrow_d C' \wedge \rho \in F(C')\} \cup \{\alpha \mid C - \alpha \rightarrow_d [E, \theta']\}$
$=$
$\Psi_{\mathcal{O}^*\mathcal{D}^*}(F)(C_i)$
End A.7