# Theory of Infinite Streams and Objects

[Extended Abstract]

Konstantin Chekin
Institute of Theoretical Computer Science
Dresden University of Technology
D-01062 Dresden
tchekine@tcs.inf.tu-dresden.de

This paper provides a theory of infinite streams and objects, which contains our point of view on the problem of formal modelling of behaviors of objects and their systems with big or infinite number of internal states.

The most closed analogue of our theory are the theory of finite automata, FOCUS theory of M.Broy and K.Stolen [4], and Reo calculus introduced by F.Arbab and J.Rutten [3] together with its alternative semantics given by constraint finite automata in [2].

In FOCUS objects are described in terms of their input/output behaviors given by sets of stream processing functions. In the coalgebraic semantics of Reo objects are modelled by means of relations on timed data streams. The second semantics of Reo uses constrain finite automata to describe objects.

Theory of infinite streams and objects differs cardinally: 1) from the automata theory, FOCUS theory and Reo calculus so, that we use infinite streams for modelling of internal variables as well as for inputs and outputs of objects; 2) from FOCUS and Reo so, that we work with all involved streams concurrently and synchronously. In our theory we give up the representation of an object through its states and switch to the representation through the sets of all admissible behaviors of interface streams and internal streams constituting the object. This provides our theory of infinite streams and objects greater expressive power than theory of finite automata and moreover than push-down automata. The use of streams gives us uniform way for modelling of inputs, outputs and internal variables of objects, simplifies object's definitions, and renders possible our theory to be not afraid of the object's dimensionality increase. A side benefit of the suggested theory is presence of the algorithm for automatic obtaining of the formal model of a system from the given models of objects constituting the system and their topology in the system.

We use the theory of universal coalgebras [5] to describe semantics of our theory of infinite streams and objects. This paper provides detailed description of the formal syntax of objects, of their composition into more complex ones, and outlines corresponding coalgebraic semantics.

## Keywords

Theory of objects and streams, systems of objects, set of admissible infinite behaviors, stateless representation

## 1. STREAMS

*Stream* is an infinite sequence of data (elements of the stream) of the certain type. Under notion *stream data type* we will understand a set $A$ of values, which can be taken on by elements of this stream. *The set $A$ may be finite or infinite.*

Infinite streams of elements from the fixed set $A$ are *modelled* by means of the final $F_A$-coalgebra of the functor $F_A : Set \to Set$ defined on sets $M$ by $F_A(M) = A \times M$:

$$(A^\omega; \langle head_{A^\omega}, tail_{A^\omega}\rangle)$$

where the function $\langle head_{A^\omega}, tail_{A^\omega}\rangle$ has form $A^\omega \to A^\omega$. The *carrier set* $A^\omega$ of the final $F_A$-coalgebra

$$A^\omega \overset{def}{=} \{\, s \mid s : \mathbb{N} \to A \,\}$$

models the set of all streams of this type. It is formally defined as the set of all functions $s : \mathbb{N} \to A$ mapping natural numbers $\mathbb{N}$ into set $A$ of elements of the stream.

Since elements of the set $A^\omega$ are functions, we can define for every infinite stream $s \in A^\omega$ the operations, which return the first element of the stream $head_{A^\omega} : A^\omega \to A$ and the tail of the stream after removing its first element $tail_{A^\omega} : A^\omega \to A^\omega$, in the following way:

$$head_{A^\omega}(s) = s(0) \qquad tail_{A^\omega}(s) = \lambda x \in \mathbb{N}.\, s(x+1)$$

THEOREM 1. *For arbitrary data types $A_1, ..., A_r$ the set of all $r$-tuples of infinite streams $A_1^\omega \times ... \times A_r^\omega$ with head and tail operations executing concurrently on all streams is isomorphic to the set of infinite streams $(A_1 \times ... \times A_r)^\omega$ of $r$-tuples of the corresponding elements of the streams $A_1^\omega, ..., A_r^\omega$.*

We illustrate this theorem by the following example. Let us take two infinite streams: one is the stream of names in the lexicographical order $\alpha \in Lex^\omega$, $\alpha = a_1, b_1, c_1, d_1, e_1, f_1, ...$ and the other one is the stream consisting of natural numbers in the increasing order $\beta \in \mathbb{N}^\omega$, $\beta = 0, 1, 2, 3, 4, 5, ...$. The head of the first stream is the element $a_1$, i.e. $head(\alpha) = a_1$, and its tail is the infinite stream $tail(\alpha) = b_1, c_1, d_1, e_1, f_1, ...$

The pair of two streams $\alpha$ and $\beta$ according to the theorem and corollary above is isomorphic to the stream of pairs $\gamma \in (Lex \times I\!N)^\omega$, i.e. $\alpha \times \beta \cong \gamma$. The stream $\gamma = (a_1, 0), (b_1, 1), (c_1, 2), (d_1, 3), (e_1, 4), (f_1, 5)...$ consists of the pairs of the corresponding elements of the streams $\alpha$ and $\beta$.

Later on we will proceed from the requirements, that *names of all used streams and names of all used objects are unique.*

*Semantics* $[\![z]\!]$ of the stream $z : Z^\omega$ in the absence of additional restrictions is the set of all different streams of this type, i.e $[\![z]\!] \overset{def}{=} Z^\omega$, where $Z^\omega$ is the type of the stream with name $z$ and simultaneously the carrier set of the final $F_Z$-coalgebra of infinite streams.

*Set* $[\![z]\!]$ of all *admissible behaviors* of the stream $z$ is the set defined by the semantics of this stream. *Instance (variant) of stream behavior* or *behavior* $\alpha_z$ of the stream $z$ is the element from the set $[\![z]\!]$ of admissible behaviors of this stream, i.e. $\alpha_z \in [\![z]\!]$.

We formulate our rule of stream manipulations by the following proposition. *With all involved streams in every model, which is created by our theory, we will work concurrently, i.e. operations head and tail will be executed concurrently on all streams.*

## 2. OBJECT

In our formal theory a fundamental notion is the notion of an object. An object can have several different inputs (input streams), outputs (output streams) as well as several internal streams. Streams describe chronological succession of occurrences of values on the modelled inputs, outputs and internal variables of an object.

*Object* $A = \langle\ X\ ;\ S\ ;\ Y\ ;\ I\ ;\ \Phi \rangle$ consists of the set of names of input streams $X$, the set of names of internal streams $S$, the set of names of output streams $Y$, restriction $I$ on initial values of streams, and invariant property $\Phi$, which describes relation between streams of the object and between current and next values of these streams. The set $X \overset{def}{=} \{x_1 : X_1^\omega, ..., x_n : X_n^\omega\}$ is the finite set of names of input streams $x_1, ..., x_n$ and corresponding types $X_1^\omega, ..., X_n^\omega$ of these streams; the sets $S \overset{def}{=} \{s_1 : S_1^\omega, ..., s_m : S_m^\omega\}$ and $Y \overset{def}{=} \{y_1 : Y_1^\omega, ..., y_k : Y_k^\omega\}$ are the finite sets of names, correspondingly, of internal streams and of output streams.

In case of when the object has no inputs, or no outputs, or no internal streams, we will use the keyword *empty* to denote the absence of streams in the corresponding field in the object's quintuple.

*Syntax of the restriction I on initial values* of streams of the object we define in the following way:

$$I ::= true \ |\ z(0) = val \ |\ \neg I \ |\ I_1 \wedge I_2 \ |\ I_1 \vee I_2$$

where *true* denotes the absence of any restrictions on initial values of streams; $z \in X \cup S \cup Y$ is one of the streams of the object, and *val* is a value from the set of values of elements of the stream $z$.

The formal *syntax of invariant property* we define as $\forall i \in I\!N.\ \Phi$, where $\Phi$ is a notion of invariant property in its *short form* with implicit universal quantifier. The expression $\forall i \in I\!N.\ \Phi$ represents a complete form of invariant property, which holds infinite number of times, i.e. on all elements of all involved streams. In the further at specification of invariant property we will leave out universal quantifier $\forall i \in I\!N$, nevertheless we will have its presence permanently in mind. By using of the short form of invariant property under notion $\Phi$ we have in mind the notion $\forall i \in I\!N.\ \Phi$.

*Syntax* of the *short form* $\Phi$ of *invariant property* with implicit universal quantifier we define as:

$$\Phi ::= true \ |\ z_{1(i)} = f(z_{1(i)}, ..., z_{p(i)}) \ |\ \mathbf{X}z_{(i)} = g(z_{1(i)}, ..., z_{r(i)})$$
$$|\ \neg \Phi \ |\ \Phi_1 \wedge \Phi_2 \ |\ \Phi_1 \vee \Phi_2$$

where *true* denotes the absence of any relations between streams in the object; $z, z_1, ..., z_p, ..., z_r$ are streams of the object; $f$ and $g$ are total computable functions, which return values of the same type as the elements of the stream $z$.

In the short form of invariant property with implicit universal quantifier an element $z(i)$ of stream $z$ with index $i$ we call *current element* of the stream, and its value — *current value* of the stream; an element $z(i+1)$ of stream $z$ with index $i+1$ we call *next element* of the stream, and its value — *next value* of the stream. Moreover by the reasons of visual clearness increasing we will denote current element of stream $z$ by $z_{(i)}$ instead of $z(i)$, and next element of the stream by $\mathbf{X}z_{(i)} \overset{def}{=} z(i+1)$.

For the given object $A$ the two sets: the set $X$ of the names of input streams and the set $Y$ of the names of output streams form together the set of names of *interface* streams of the object: $Interface(A) \overset{def}{=} X \cup Y = \{x_1, ..., x_n, y_1, ..., y_k\}$.

## 3. PARALLEL COMPOSITION AND CONNECTION OF INTERFACE STREAMS

In order to guarantee that our formal theory can be applicable for modelling of system with arbitrary topology and consisting of big number of objects, we have to introduce a formal mechanism of step-by-step successive construction of those systems from their constituting subsystems and objects. In our theory the mechanism of step-by-step successive construction of system of objects will be based on the following proposition. *A result of composition of objects is a system of these objects, which in one's turn is considered in our theory as a bigger object, and with which we syntactically and semantically operate in the same way as with any other object, i.e. we will use it as a constituting part for building of even bigger objects.*

We lay operations of composition of objects under a number of practical-oriented requirements. Operations of composition must have adequate expressive power in order to create big practical-oriented systems by using them. Moreover, operations of composition have to provide an ability of all-around automatization of generating process of formal models of systems from models of objects, which constitute these systems, and from their topology.

In order to satisfy the requirements above we define two operations of composition: the first one is called *parallel composition* and serves for for putting objects in parallel; the second one is called *connection* and serves for connecting of inputs and outputs of an object.

For the first time analogous operations of composition were introduced in the papers of Henzinger on component and interface theories [1] and of Lee on block-diagram languages [6]. Operations of composition, which are used in our theory, simplify the representation and understanding of different kinds of object compositions, and, moreover, they provide us ability to create objects of arbitrary topology and dimension.

For two arbitrary objects $A$ and $B$:

$$A = \langle\, X_A \,;\, S_A \,;\, Y_A \,;\, I_A \,;\, \Phi_A \rangle \quad B = \langle\, X_B \,;\, S_B \,;\, Y_B \,;\, I_B \,;\, \Phi_B \rangle$$

*parallel composition* transforms these two objects into new object $A||B$ in the following way:

$$A||B \overset{def}{=} \langle X_A \cup X_B \,;\, S_A \cup S_B \,;\, Y_A \cup Y_B \,;\, I_A \wedge I_B \,;\, \Phi_A \wedge \Phi_B \rangle$$

Sets of names of input, internal and output streams of composition $A||B$ are set-theoretical unions of the corresponding sets of stream names of objects $A$ and $B$: $X_{A||B} = X_A \cup X_B$, $S_{A||B} = S_A \cup S_B$ and $Y_{A||B} = Y_A \cup Y_B$. Consequently, we define the set of names of interface streams of the new object as $\mathit{Interface}(A||B) = \mathit{Inrface}(A) \cup \mathit{Interface}(B)$.

In the next step we introduce the operation of connection, which transforms an arbitrary object $A$ and a well-formed interconnect $\varphi_A$ into new object $A \cdot \varphi_A$.

*Interconnect* $\varphi$ defined on an object $A$ is a finite list (with length $l$)

$$\varphi = [(z_{S1}, z_{T1}), (z_{S2}, z_{T2}), ..., (z_{Sl}, z_{Tl})]$$

of pairs of stream names of the object $A$ such that in every pair $(z_{Si}, z_{Ti})$, where $0 \le i \le l$, both streams are interface streams $z_{Si}, z_{Ti} \in \mathit{Interface}(A)$, and moreover the types of these streams must be the same. In every pair $(z_{Si}, z_{Ti})$ the first component $z_{Si}$ we call *source* and the second one $z_{Ti}$ — *target*.

A special case of interconnect is an empty interconnect. *Empty interconnect* is interconnect containing no pairs of stream names, and we will denote it by the empty list [ ]. The empty interconnect is an identity element for the connection operation. Therefore, by the connection of an arbitrary object $A$ with the empty interconnect we obtain the previous object $A$, i.e. $A \cdot [\,] = A$.

The other special case of interconnect is an atomic interconnect, which will be needed in the inductive definition of the connection operation. *Atomic interconnect* is an interconnect containing only one pair of stream names.

In the operation of connection we produce pairwise connection of interface streams of an object according to list of pairs of their names, which is contained in interconnect. There are a number of interconnects admissible from the point of view of the definition above, but which can not be used in the operation of connection in the form, in which it is defined in this paper. Therefore we introduce a notion of "well-formed" interconnect, and later on in the operation of connection we will use only well-formed interconnects. Interconnect $\varphi_A$ defined on an object $A$ is called *well-formed*, if it satisfy the following conditions: 1) the target and source in every pair are different, i.e. in the interconnect $\varphi_A$ there is no pair of the form $(x, x)$; 2) all targets of the interconnect $\varphi_A$ are pairwise different; 3) when the same name of stream occurs in the interconnect $\varphi_A$ in different pairs in source position as well as in target position, then it must appear as target only after all its occurrences as source; 4) in every pair either both stream names denote input streams, or both denote output streams, or the source is a name of output stream and the target is a name of input stream.

In the cases, when at least one of these four conditions is not satisfied, we will consider that such interconnects are not well-formed, and such interconnects we will not use in the operations of connection.

For an arbitrary object $A$ and a well-formed on it *atomic interconnect* $[(z_S, z_T)]$ the operation of *connection* transforms the object $A$ into new object $A \cdot [(z_S, z_T)]$ according to the pair of names of the connected streams, which is contained in the atomic interconnect. Depending on the kind of atomic interconnect there are three possible cases of connection:

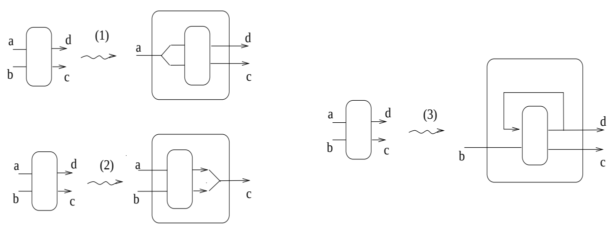

**Figure 1: Three different cases of connection of an object and a well-formed interconnect.**

1. in the case where both elements of the pair $[(z_S, z_T)]$ are names of input streams $z_S, z_T \in X_A$, we remove from the set $X_A$ of names of input streams of the object $A$ the target of the atomic interconnect, i.e. the name of the stream $z_T$: $A \cdot [(z_S, z_T)] \overset{def}{=}$

   $$\langle X_A \setminus \{z_T\} \;;\; S_A \;;\; Y_A \;;\; I_A\ \{z_T/z_S\} \;;\; \Phi_A\ \{z_T/z_S\}\rangle$$

2. in the case where both elements of the pair $[(z_S, z_T)]$ are names of output streams $z_S, z_T \in Y_A$, we remove from the set $Y_A$ of names of output streams of the object $A$ the target of the atomic interconnect, i.e. the name of the stream $z_T$: $A \cdot [(z_S, z_T)] \overset{def}{=}$

   $$\langle X_A \;;\; S_A \;;\; Y_A \setminus \{z_T\} \;;\; I_A\ \{z_T/z_S\} \;;\; \Phi_A\ \{z_T/z_S\}\rangle$$

3. in the case where in the atomic interconnect $[(z_S, z_T)]$ the source is a name of output stream $z_S \in Y_A$ and the target is a name of input stream $z_T \in X_A$, we remove from the set $X_A$ of names of input streams of the object $A$ the target of the atomic interconnect, i.e. the name of the stream $z_T$: $A \cdot [(z_S, z_T)] \overset{def}{=}$

   $$\langle X_A \setminus \{z_T\} \;;\; S_A \;;\; Y_A \;;\; I_A\ \{z_T/z_S\} \;;\; \Phi_A\ \{z_T/z_S\}\rangle$$

In all three cases restriction on initial values $I_A\{z_T/z_S\}$ and invariant property $\Phi_A\{z_T/z_S\}$ of the new object $A \cdot [(z_S, z_T)]$ are obtained, correspondingly, from restriction on initial values $I_A$ and invariant property $\Phi_A$ of the original object $A$ by means of substitution of all occurrences of the name $z_T$ by name $z_S$.

Hypothetic fourth case, where the source is a name of input stream and the target is a name of output stream, is syntactically impossible, since it is filtered out by the requirements of the well-formedness of atomic interconnect.

We define operation of *connection*, which transforms an arbitrary object $A$ and a well-formed on it interconnect $\varphi_A = [(z_{S1}, z_{T1}), (z_{S2}, z_{T2}), ..., (z_{Sl}, z_{Tl})]$ into new object $A \cdot \varphi_A$, in the following inductive way:

$$A \;\cdot\; [\,] \overset{def}{=} A$$

$$A \;\cdot\; [(z_{S1}, z_{T1}), (z_{S2}, z_{T2}), ..., (z_{Sl}, z_{Tl})] \overset{def}{=} (A \cdot [(z_{S1}, z_{T1})]) \;\cdot\; [(z_{S2}, z_{T2}), ..., (z_{Sl}, z_{Tl})]$$

The sets of names of input and output streams of the connection $A \cdot \varphi$ of an object $A$ and a well-formed interconnect $\varphi = [(z_{S1}, z_{T1}), ..., (z_{Sl}, z_{Tl})]$ represent them self set-theoretical difference of the corresponding sets of stream names of the original object and the sets of targets of the interconnects: $X_{A \cdot \varphi} = X_A \setminus \{z_{Ti} \mid \forall i \in I\!N.\ 1 \leq i \leq l \wedge z_{Ti} \in X_A\}$ and $Y_{A \cdot \varphi} = Y_A \setminus \{z_{Ti} \mid \forall i \in I\!N.\ 1 \leq i \leq l \wedge z_{Ti} \in Y_A\}$. The set of names of internal streams of the connection $A \cdot \varphi$ is the set of names of internal streams of the original object $A$, since internal streams of the object are not influenced by operations of connection, i.e. $S_{A \cdot \varphi} = S_A$. Therefore we define the set of names of interface streams of the new object $A \cdot \varphi$ as a set-theoretical difference of the set of names of interface streams of the original object $A$ and the set of targets of the interconnect $\varphi$, i.e. $\mathit{Interface}(A \cdot \varphi) = \mathit{Interface}(A) \setminus \{z_{Ti} \mid \forall i \in I\!N.\ 1 \leq i \leq l\}$.

Let us illustrate the three cases of connection described above by the following example (see on the Fig. 1). We take the object $A$ with two input $a, b$ and two output $c, d$ streams of the same type:
$A = \langle a, b; \mathit{empty}; c, d; b(0)=0; c_{(\mathrm{i})} = a_{(\mathrm{i})} + 5 \wedge d_{(\mathrm{i})} = 2 * b_{(\mathrm{i})} \rangle$

We connect in the first case two input streams $a$ and $b$ by means of the interconnect $\varphi_1 = [(a, b)]$, in the second case — two output streams by $\varphi_2 = [(c, d)]$, and in the third case — one input stream with one output $\varphi_3 = [(d, a)]$. As a result we obtain three new objects: first object with one input stream $a$ and two output streams $c, d$:
$A \cdot [(a, b)] = \langle a; \mathit{empty}; c, d; a(0)=0; c_{(\mathrm{i})} = a_{(\mathrm{i})} + 5 \wedge d_{(\mathrm{i})} = 2 * a_{(\mathrm{i})} \rangle$,
second object with two input $a, b$ and one output $c$ streams:
$A \cdot [(c, d)] = \langle a, b; \mathit{empty}; c; b(0)=0; c_{(\mathrm{i})} = a_{(\mathrm{i})} + 5 \wedge c_{(\mathrm{i})} = 2 * b_{(\mathrm{i})} \rangle$,
third object with one input $b$ and two output $c, d$ streams:
$A \cdot [(d, a)] = \langle b; \mathit{empty}; c, d; b(0)=0; c_{(\mathrm{i})} = d_{(\mathrm{i})} + 5 \wedge d_{(\mathrm{i})} = 2 * b_{(\mathrm{i})} \rangle$.

## 4. SEMANTICS

According to the theorem in the first section, a tuple of stream is isomorphic to the stream of tuples of the corresponding elements of these streams. Let a stream of type $Z^\omega$ be isomorphic to all streams, which constitute a structure of an object $A$. We call this stream *generalized* stream of the object. Then semantics $[\![A]\!]$ of the object $A$ is a set of all admissible behaviors of generalized object's stream, which satisfy all requirements imposed on the object. Moreover, every element from the set $[\![A]\!]$ is a concrete instance of behavior of the generalized stream, or in other words, a tuple of concrete instances of behaviors of all streams constituting the object.

Formal *semantics* $[\![A]\!]$ of an object $A = \langle\, X\ ;\ S\ ;\ Y\ ;\ I\ ;\ \Phi \rangle$ is given by:

$$[\![A]\!] \overset{def}{=} \{\![I]\!\} \cap \{\![\Phi]\!\}$$

where $\{\![I]\!\} \subseteq Z^\omega$ is the set of generalized streams, which satisfy the restriction on initial values $I$; the invariant $\{\![\Phi]\!\} \subseteq Z^\omega$ is the set of generalized streams, on which the invariant property $\Phi$ is satisfied infinitely long on every element of the stream. Formally $\{\![\Phi]\!\}$ is the carrier set of the subcoalgebra of the invariant constructed by the invariant property $\Phi$.

## 5. COMPARISON WITH AUTOMATA THEORY

Theorem 2. *Theory of infinite streams and objects has strictly greater expressive power than the theory of push-down automata.*

In our theory we can encode arbitrary push-down automata. We know that the language $\{a^n b^n c^n *^\omega \mid n \in I\!N\}$ can not be recognized by push-down automata . In contrast, in our theory we can create object, which recognizes this language. This object is given by:
$A = \langle x\ ;\ k, l, m\ ;\ \mathit{empty}\ ;\ k(0)=0 \wedge l(0)=0 \wedge m(0)=0\ ;\ \Phi_A \rangle$
with the invariant property $\Phi_A \overset{def}{=}$

$$\begin{aligned}
&x_{(\mathrm{i})} = a \implies (\mathbf{X}k_{(\mathrm{i})} = k_{(\mathrm{i})} + 1 \wedge \mathbf{X}l_{(\mathrm{i})} = l_{(\mathrm{i})} \wedge \mathbf{X}m_{(\mathrm{i})} = m_{(\mathrm{i})} \\
&\qquad\qquad \wedge (\mathbf{X}x_{(\mathrm{i})} = a \vee \mathbf{X}x_{(\mathrm{i})} = b)) \\
&\qquad\qquad \wedge \\
&x_{(\mathrm{i})} = b \implies (\mathbf{X}k_{(\mathrm{i})} = k_{(\mathrm{i})} \wedge \mathbf{X}l_{(\mathrm{i})} = l_{(\mathrm{i})} + 1 \wedge \mathbf{X}m_{(\mathrm{i})} = m_{(\mathrm{i})} \\
&\wedge (k_{(\mathrm{i})} = l_{(\mathrm{i})} + 1 \Rightarrow \mathbf{X}x_{(\mathrm{i})} = c) \wedge (k_{(\mathrm{i})} \neq l_{(\mathrm{i})} + 1 \Rightarrow \mathbf{X}x_{(\mathrm{i})} = b) \\
&\qquad\qquad \wedge \\
&x_{(\mathrm{i})} = c \implies (\mathbf{X}k_{(\mathrm{i})} = k_{(\mathrm{i})} \wedge \mathbf{X}l_{(\mathrm{i})} = l_{(\mathrm{i})} \wedge \mathbf{X}m_{(\mathrm{i})} = m_{(\mathrm{i})} + 1 \\
&\wedge (l_{(\mathrm{i})} = m_{(\mathrm{i})} + 1 \Rightarrow \mathbf{X}x_{(\mathrm{i})} = *) \wedge (l_{(\mathrm{i})} \neq m_{(\mathrm{i})} + 1 \Rightarrow \mathbf{X}x_{(\mathrm{i})} = c) \\
&\wedge \quad x_{(\mathrm{i})} = * \implies \mathbf{X}x_{(\mathrm{i})} = *.
\end{aligned}$$

## 6. ACKNOWLEDGMENT

I would like to thank my supervisor Prof. Horst Reichel and my colleague Dmitri Schamschurko. Without their help, useful comments and criticism this work would not be what it is today.

## 7. CONCLUSIONS

This paper provides formal theory of infinite streams and objects, which contains our point of view on the problem of formal modelling of behaviors of objects and their systems with big or infinite number of internal states. In our theory we give up the representation of an object through its states and switch to the representation through the sets of all admissible behaviors of interface streams and internal streams constituting the object. The synchronous use of all involved streams provides us uniform way for modelling of inputs, outputs and internal variables of objects, simplifies object's definitions, renders possible our theory to be not afraid of the object's dimensionality increase, and gives us enough expressive power. We have shown in this paper that our theory has strictly greater expressive power than push-down automata.

## 8. REFERENCES

[1] L. de Alfaro and T. Henzinger, Interface Theories for Component-based Design, in Proceedings of Workshop on Embedded Software EMSOFT, 2001.

[2] F. Arbab, C. Baier, J. Rutten and M. Sirjani, Modeling Component Connectors in Reo by Constraint Automata, in Proceedings of Workshop on Foundations of Coordination Languages and Software Architectures (FOCLASA), 2003.

[3] F. Arbab and J. Rutten, A Coinductive Calculus of Component Connectors, in Proceedings of WADT 2002, Lecture Notes in Computer Science, vol. 2755, Springer-Verlag, 2003, pp. 35-56.

[4] M. Broy, K. Stolen, Specification and development of interactive systems, Monographs in Computer Science, vol. 62, Springer, 2001.

[5] B. Jacobs and J. Rutten, A Tutorial on (Co)Algebras and (Co)Induction, EATCS Bulletin 62, p.222-259, 1997.

[6] A. Lee, Overview of the Ptolemy Project, Technical Memorandum UCB/ERL-M01/11, University of California, Berkeley, 2001.