Interactive verification of Lustre programs in Vélus

Timothy Bourke Paul Jeanmaire Marc Pouzet

ENS, Inria Parkas team

Synchron'21, November 26



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

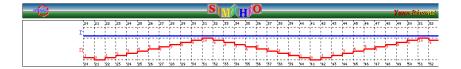
Goal: prove properties of streams in a program

```
node f (r : bool) returns (n : int)
var up : bool;
let
    up = true fby ((up and n < 9) or (not up and n <= 1));
    n = 0 fby (if up then n + 1 else n - 1);
tel</pre>
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

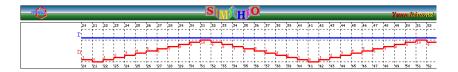
Goal: prove properties of streams in a program

```
node f (r : bool) returns (n : int)
var up : bool;
let
    up = true fby ((up and n < 9) or (not up and n <= 1));
    n = 0 fby (if up then n + 1 else n - 1);
tel</pre>
```



```
Goal: prove properties of streams in a program
node f (r : bool) returns (n : int)
var up : bool;
let
    up = true fby ((up and n < 9) or (not up and n <= 1));
    n = 0 fby (if up then n + 1 else n - 1);
tel</pre>
```

Show that values of stream n stay between 0 and 10 $\,$



▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

```
Goal: prove properties of streams in a program
node f (r : bool) returns (n : int)
var up : bool;
let
    up = true fby ((up and n < 9) or (not up and n <= 1));
    n = 0 fby (if up then n + 1 else n - 1);</pre>
```

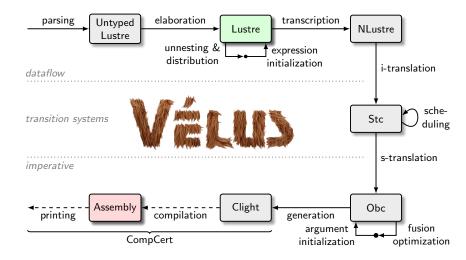
tel

Show that values of stream n stay between 0 and 10 $\,$

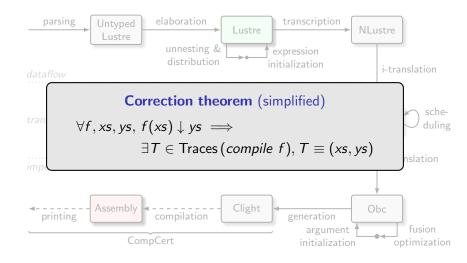
Interactive approach:

- load definitions in the proof assistant (ITP)
- use reasoning techniques to manipulate the goal/hypotheses
- obtain a mathematical proof of the result

Interactive verification in Vélus

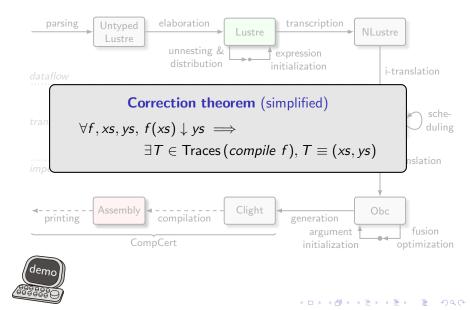


Interactive verification in Vélus



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Interactive verification in Vélus



Cécile Dumas Canovas Deductive methods for proof of Lustre programs

PhD thesis, 2000

UNIVERSITE JOSEPH FOURIER - GRENOBLE 1 SCIENCES & GEOGRAPHIE Dur obtenir le grade de DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER Discipline : « Informatique : Systèmes et Communication » Présentée et soutenne publiquement par Cécile Dunnas Canovas

Le 17 novembre 2000

Méthodes déductives pour la preuve de programmes LUSTRE

Directeur de thèse : Paul Caspi

COMPOSITION DU JURY

M. Farid Ouabdesselam. Mine Véronique Viguié Donzeau-Gouge, Mine Christine Paulin-Mohring, M. Paul Caspi, Mine Catherine Parent-Vigouroux, Mine Catherine Parent-Vigouroux, Mine Marie-Laure Potet,

Président Rapporteur Rapporteur Directeur de thèse Co-encadrante Examinateur

900

Cécile Dumas Canovas Deductive methods for proof of Lustre programs

PhD thesis, 2000

Contents

- Example of program verification (train)
- Translation from Lustre to Atelier B
- Proof principles for Lustre
- From Lustre to PVS
- Thoughts on refinement

UNIVERSITE JOSEPH FOURIER - GRENOBLE 1 SCIENCES & GEOGRAPHIE

THESE

pour obtenir le grade de DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER

Discipline : « Informatique : Systèmes et Communication » Présentée et souteme publiquement

par

Cécile Dumas Canovas Le 17 novembre 2000

Méthodes déductives pour la preuve de programmes LUSTRE

Directeur de thèse : Paul Caspi

COMPOSITION DU JURY

M. Farid Ouabdesselam, Mme Véronique Vignié Donzeau-Gouge, Mme Christine Paulin-Mohring, M. Paul Caspi, Mme Catherine Parent-Vigouroux, Mme Marie-Laure Potet,

Président Rapporteur Rapporteur Directeur de thèse Co-encadrante Examinateur

Cécile Dumas Canovas Deductive methods for proof of Lustre programs

PhD thesis, 2000

Contents

- Example of program verification (train)
- Translation from Lustre to Atelier B
- Proof principles for Lustre
- From Lustre to PVS
- Thoughts on refinement

UNIVERSITE JOSEPH FOURIER - GRENOBLE 1 SCIENCES & GEOGRAPHIE

THESE

pour obtenir le grade de DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER

Discipline : « Informatique : Systèmes et Communication » Présentée et souteme publiquement

par

Cécile Dumas Canovas Le 17 novembre 2000

Méthodes déductives pour la preuve de programmes LUSTRE

Directeur de thèse : Paul Caspi

COMPOSITION DU JURY

M. Farid Ouabdesselam, Mme Véronique Vignié Donzeau-Gouge, Mme Christine Paulin-Mohring, M. Paul Caspi, Mme Catherine Parent-Vigouroux, Mme Marie-Laure Potet,

Président Rapporteur Rapporteur Directeur de thèse Co-encadrante Examinateur Three different encodings of streams

1. Temporal & consecutive representations: $\mathbb{N} \to \mathsf{Val}$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

Three different encodings of streams

- 1. Temporal & consecutive representations: $\mathbb{N} \to \mathsf{Val}$
- 2. Co-induction with infinite proof principle (\approx Vélus)

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ □臣 ○のへ⊙

Three different encodings of streams

- 1. Temporal & consecutive representations: $\mathbb{N} \to \mathsf{Val}$
- 2. Co-induction with infinite proof principle (\approx Vélus)
- 3. Kahn streams with a principle of continuous induction
 - the set of streams is a CPO
 - \blacktriangleright with a smallest element ϵ
 - stream operations are defined as least fix-points

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

Christine Paulin-Mohring

A constructive denotational semantics for Kahn networks in Cog

From semantics to CS, 2007

A constructive denotational semantics for Kahn networks in Coq

Christine Paulin-Mohring INRIA Futurs, ProVal, Parc Orsay Université, F-91893 LRI, Univ Paris-Sud, CNRS, Orsay, F-91405

July 18, 2007

Semantics of programming languages and interactive environments for the development of proofs and Sommance or programming sequences and increasive environments sor the development of process and programs are two important aspects of Gilkes Kahn's scientific contributions. In his paper "The emandice of Jerograms as e two important sepects of view frame barrene concretence on the paper view semantics of a simple language for parallel programming. [11], he proposed an interpretation of (deterministic) parallel a sampa magnaga ne preama programming: 111, ne proposes an instruction or (structmanaster) instants programs (now called Kalm networks) as strong transformers based on the theory of complete partial orders. programs (now care) many networks) as strong randormy so and on the incory or complete partial orders (pos). A ratifiction of this language to synchronous programs is the basis of the data-flow Lutre language (pos).

int as users for the generalization of this seminal paper in the God proof assistant [4, 15]. For that purpose, we present a torgatization or tins seimou paper in the Goq provi assistant [9, 10]. For that purpose, we developed a general library for cpos. Our cpos are defined with an explicit function computing the least we uneverspine a generic a more source of elements with an explorat tunction computing the weak upper bound (lub) of an increasing sequence of elements. This is different from what G. Kahn developed for

oper control (not) is an introducing sequence to compose. This is no converting mass to mean or empty to the standard (eq likely where only the existence of luke (for arbitrary directed sets) is required, giving no In seamant on any service one the define a construction into the type of possibly infinite strongs. It is applicitly compute a fixpoint. We define a constructive for the type of possibly infinite strongs. It way to explicitly compute a faxpoint. We define a two structure for the type of possibly instructs structures, it is then possible to define formally what is a Kahn network and what is its semantics, achieving the goal of

is steer possion or usual available ways is a ratio arrange and area and a so well-subject advanced to grave or having a concept closed by composition and recursion. The library is illustrated by the example taken from the original paper as well as the Sieve of Erstosthenes, an example of a dynamic network.

1 Introduction

Semantics of programming languages and interactive environments for the development of proofs and programs commance of programming negroups and memory environments for the devised ment of process and programs are two important aspects of Gilles, Kahn's scientific contributions. In his paper "The semantics of a simple are row important aspects or Gauss Kann's scientific contributions. In no paper The semantics of a simple language for parallel programming." [11], he proposed an interpretation of (deterministic) parallel programming sangenge to persons programming [11], to proposed an interpretation of (preseministic) parame programs (now called Kaim networks) as streams transformers based on the theory of complete partial orders (cps). A tome constant technical in second in second constants constructed on second or use under the compare percent others (clear), in restriction of this language to synchronous programs is the basis of the data. Bow Lustre language [14, 10] which restriction or this singuage to synchronous programs is the basis or the units-forw closer anguage (1.4, 10) which is used now for the development of critical embedded systems. Because of the degance and generality of the as used now us the transmission of classification of systems. Second of the data-flow synchronous paradigm to model, Kahn networks are also a source of inspiration for extrainons of the data-flow synchronous paradigm to

2m-races constructions (7) or to more permassive modules or synchrony [8]. We present a formalization of this seminal paper in the Coq proof assistant [4, 15]. For that purpose,

we present a nermanostron or this seminar paper in the code provi assistant (s. 15). For this purpose, we developed a general library for epos. Our cose are defined with an explicit function computing the loss we unvestigated a gategrate interary out cpose. Our cpose are unique write an organicit interaction computing the issue upper bound (lub) of a monotomic sequence of elements. This is different from what G. Kohn developed for upper country (unit) on a monotone sequences or sequences. Anne a universe from work or norm university or the standard Coq ibraries where only the existence of labs is required, giving no way to explicitly compute a the singular out unames where only use canceuse or most a required, giving to way to expandit computer field, in the paper interaction of the pape happane. However, mann's meany was meened as the background of a computer seminassion of the paper "Concrete Domains" by G. Kahn and G. Plotkin [13] and it covers general cpos with the existence of a hib "Concrete tocamage: 10 G. Kann and G. Pottam [16] and it covers generat the wath the concerns us a nu-fee arbitrary directed sets while our work only considers s-cpos with hise on monotonic sequences which is a

meetin rangework nor mooreing baam networks. We define a cpo structure for the type of possibly infinite streams. This is done using a coinductive type in

we evanue a cro structure not the type of position minute streams. This is none using a constructive type in Coq with two constructors, one for adding an element in front of a stream, the second constructor add a silent Use what we constructions one or sound an example an evaluant in iron or a stream, into second conservation and a second step Eqs. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the inter that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on the muc. However, we anostry the empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our wata type covers noti mure any munity arvants. We using the period ware on two uses type new un-corresponding equality. We also develop useful basic functions: the functions for head, tail and append used

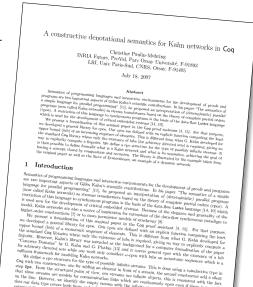
Christine Paulin-Mohring

A constructive denotational semantics for Kahn networks in Cog

From semantics to CS, 2007

Contents

General library for CPOs



step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the inter that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on use mue, However, we anostry use empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our wata type covers noti mure any munity arvants. We using the period ware on two uses type new un-corresponding equality. We also develop useful basic functions: the functions for head, tail and append used

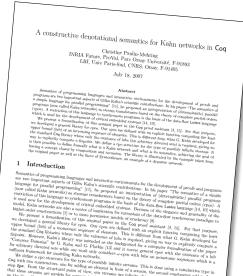
Christine Paulin-Mohring

A constructive denotational semantics for Kahn networks in Coq

From semantics to CS. 2007

Contents

- General library for CPOs
- Encoding of streams s := Eps s | Cons a s



we evanue a cro structure not the type of position minute streams. This is none using a constructive type in Coq with two constructors, one for adding an element in front of a stream, the second constructor add a silent Use what also constructions one for sound as compute in iterating the second constructor and a sound stop gas. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the inter that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on use mue, However, we anostry use empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our wata type covers noti mure any munity arvants. We using the period ware on two uses type new un-corresponding equality. We also develop useful basic functions: the functions for head, tail and append used

Christine Paulin-Mohring

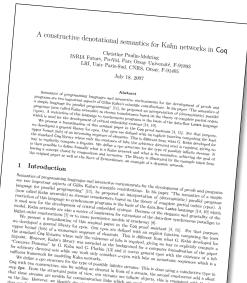
A constructive denotational semantics for Kahn networks in Coq

From semantics to CS. 2007

Contents

- General library for CPOs
- Encoding of streams s := Eps s | Cons a s

 $\perp := \mathsf{Eps} \perp$



Use what also constructions one for sound as compute in iterating the second constructor and a sound stop gas. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the inter that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on use mue, However, we anostry use empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our wata type covers noti mure any munity arvants. We using the period ware on two uses type new un-corresponding equality. We also develop useful basic functions: the functions for head, tail and append used

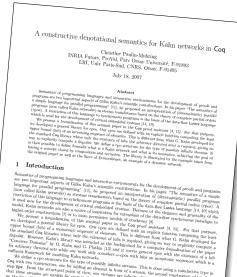
Christine Paulin-Mohring

A constructive denotational semantics for Kahn networks in Coq

From semantics to CS. 2007

Contents

- General library for CPOs
- Encoding of streams s := Eps s | Cons a s
 - $\perp := \mathsf{Eps} \perp$
- Primitive stream functions



we evanue a cro structure not the type of position minute streams. This is none using a constructive type in Coq with two constructors, one for adding an element in front of a stream, the second constructor add a silent Use what also constructions one for sound as compute in iterating the second constructor and a sound stop gas. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the incr that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on use mue, However, we anostry use empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our wata type covers noti mure any munity arvants. We using the period ware on two uses type new un-corresponding equality. We also develop useful basic functions: the functions for head, tail and append used

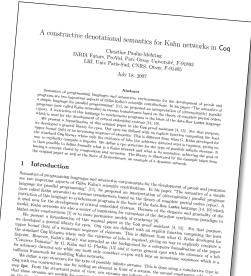
Christine Paulin-Mohring

A constructive denotational semantics for Kahn networks in Coq

From semantics to CS, 2007

Contents

- General library for CPOs
- Encoding of streams s := Eps s | Cons a s
 - $\perp := \mathsf{Eps} \perp$
- Primitive stream functions
- Example: modeling a (recursive) Kahn network



we evanue a cro structure not the type of possing influence streams. This is none using a constructive type in Coq with two constructors, one for adding an element in front of a stream, the second constructor add a silent Use what also constructions one for sound as compute in iterating the second constructor and a sound stop gas. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the incr that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on the muc. However, we anostry the empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our usuan type covers notin mute any munity areans. We usuae the process where ou take our systems can usua type next take or corresponding, equality: We also develop medial basic functions: the functions for head, tail and append used

Christine Paulin-Mohring

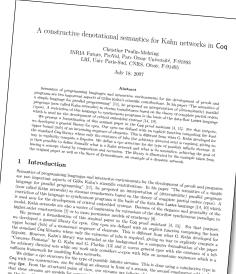
A constructive denotational semantics for Kahn networks in Coq

From semantics to CS, 2007

Contents

- General library for CPOs
- Encoding of streams s := Eps s | Cons a s
 - $\perp := \mathsf{Eps} \perp$
- Primitive stream functions
- Example: modeling a (recursive) Kahn network





we evanue a cro structure not the type of possing influence streams. This is none using a constructive type in Coq with two constructors, one for adding an element in front of a stream, the second constructor add a silent Use what also constructions one for sound as compute in iterating the second constructor and a sound stop gas. From the structural point of view, our streams are infinite objects, this is consistent with the fact step gas. From the structure point or view, our streams are manifed outpets, this is consistent with the incr that these streams are models for communication links which are continuously open even if there is no traffic may trave streams are movers for communication may when are continuously open even it there is no traile on the line. However, we identify the empty stream with the infinite stream with only Eqs constructors such on the mue. However, we another use empty stream with the infinite stream with only the constructors such that our data type covers both finite and infinite streams. We define the prefix order on this data type and the that our usuan type covers notin mute any munity areans. We usuae the process where ou take our systems can usua type next take or corresponding, equality: We also develop medial basic functions: the functions for head, tail and append used

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

Done

Done

Reproducing old results in the context of Vélus/Coq ITP

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

Verify some small programs with clocks

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language
- Verify some small programs with clocks

To do

Verify more (parameterized) programs, new proof techniques

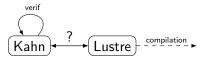
▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language
- Verify some small programs with clocks

To do

- Verify more (parameterized) programs, new proof techniques
- Link with the semantic model of Vélus



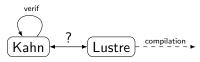
▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language
- Verify some small programs with clocks

To do

- Verify more (parameterized) programs, new proof techniques
- Link with the semantic model of Vélus



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

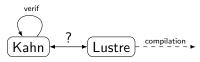
Is the Kahn semantics suitable for some compilation steps?

Done

- Reproducing old results in the context of Vélus/Coq ITP
- Identify best methods and use it to give a natural semantics to the full language
- Verify some small programs with clocks

To do

- Verify more (parameterized) programs, new proof techniques
- Link with the semantic model of Vélus



- Is the Kahn semantics suitable for some compilation steps?
- What about the *existence* of a semantics?