Execution of Extracted Programs

Luís Cruz-Filipe¹ Pierre Letouzey²

¹Center for Logic and Computation Lisbon, Portugal

2

Mathematisches Institut Ludwig-Maximilians-Universitt Mnchen Mnchen, Germany

CALCULEMUS workshop July 18th 2005

Motivation

• Coq extraction

- C-CoRN library
- work on extracting programs from C-CoRN (with Bas Spitters)

・ロト ・日ト ・ヨト ・ヨト

Motivation

- Coq extraction
- C-CoRN library
- work on extracting programs from C-CoRN (with Bas Spitters)

・ロト ・日ト ・ヨト ・ヨト

Motivation

- Coq extraction
- C-CoRN library
- work on extracting programs from C-CoRN (with Bas Spitters)

イロト イヨト イヨト イヨト

Practical problems

• extracted code unreadable

- compilation requires manual editing
- compiled code does not terminate

<ロ> (四) (四) (三) (三) (三)

Practical problems

- extracted code unreadable
- compilation requires manual editing
- compiled code does not terminate

Practical problems

- extracted code unreadable
- compilation requires manual editing
- compiled code does not terminate

<ロ> (四) (四) (三) (三) (三)

Compiling extracted code

2 Computing e

3 Computing $\sqrt{2}$

4 Conclusions

L. Cruz-Filipe, P. Letouzey Execution of Extracted Programs

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ









L. Cruz-Filipe, P. Letouzey Execution of Extracted Programs









L. Cruz-Filipe, P. Letouzey Execution of Extracted Programs

(日) (문) (문) (문) (문)









(日) (문) (문) (문) (문)

• Coq type system too powerful:

- types may depend on terms
- records may include types
- possibility of using unsafe coercions
- correctness still guaranteed.

• Coq type system too powerful:

- types may depend on terms
- records may include types
- possibility of using unsafe coercions
- correctness still guaranteed

(日) (四) (王) (王) (王)

• Coq type system too powerful:

- types may depend on terms
- records may include types
- possibility of using unsafe coercions
- correctness still guaranteed

・ロト ・日ト ・ヨト ・ヨト

- Coq type system too powerful:
 - types may depend on terms
 - records may include types
- possibility of using unsafe coercions
- correctness still guaranteed

・ロト ・日ト ・ヨト ・ヨト

- Coq type system too powerful:
 - types may depend on terms
 - records may include types
- possibility of using unsafe coercions
- correctness still guaranteed

・ロト ・日ト ・ヨト ・ヨト

Definitions

e is defined as the sum of the series

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

• first non-trivial example of a "real" real number

precise representation depends on proof terms

<ロ> (四) (四) (三) (三) (三)

Definitions

e is defined as the sum of the series

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

• first non-trivial example of a "real" real number

• precise representation depends on proof terms

イロト イヨト イヨト イヨト

르

Definitions

e is defined as the sum of the series

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

- first non-trivial example of a "real" real number
- precise representation depends on proof terms

Immediate problems & solutions

natural numbers in unary notation changed to binary notation

proofs by induction

L. Cruz-Filipe, P. Letouzey Execution of Extracted Programs

<ロ> (四) (四) (三) (三) (三)

Immediate problems & solutions

- natural numbers in unary notation changed to binary notation
- proofs by induction changed to more structured proofs

(日) (四) (三) (三)

Immediate problems & solutions

- natural numbers in unary notation changed to binary notation
- proofs by induction changed to more structured proofs

(日) (四) (三) (三)

크

Immediate problems & solutions

- natural numbers in unary notation changed to binary notation
- proofs by induction changed to more structured proofs

(日) (四) (三) (三) (三)

Immediate problems & solutions

- natural numbers in unary notation changed to binary notation
- proofs by induction changed to more structured proofs

Less trivial problems

• no advantage is taken of the concrete model

- solution: parameterize on proof terms
- efficient program!

<ロ> (四) (四) (三) (三) (三)

Less trivial problems

- no advantage is taken of the concrete model
- solution: parameterize on proof terms
- efficient program!

<ロ> (四) (四) (三) (三) (三)

Less trivial problems

- no advantage is taken of the concrete model
- solution: parameterize on proof terms
- efficient program!

<ロ> (四) (四) (三) (三) (三)

Main strategy

Try to apply the same techniques that worked so well for e:

- identify and attack potential bottlenecks
- factor common proof steps
- change proofs in an intelligent way

Main strategy

Try to apply the same techniques that worked so well for e:

- identify and attack potential bottlenecks
- factor common proof steps
- change proofs in an intelligent way

(日) (部) (注) (注)

Main strategy

Try to apply the same techniques that worked so well for e:

- identify and attack potential bottlenecks
- factor common proof steps
- change proofs in an intelligent way

(日) (部) (注) (注)

Main strategy

Try to apply the same techniques that worked so well for e:

- identify and attack potential bottlenecks
- factor common proof steps
- change proofs in an intelligent way

Classical IVT

Theorem

Let f be a function defined on [a, b] with f(a) < y < f(b). Then there is $x \in [a, b]$ such that f(x) = y.

A constructive variant

Theorem

Let f be a locally non-constant function...

Definition

A function f is locally non-constant if, on every interval [a, b], it satisfies

 $\forall y \exists x. f(x) \neq y.$

_emma

Polynomials are locally non-constant.

(日) (四) (三) (三) (三)

A constructive variant

Theorem

Let f be a locally non-constant function...

Definition

A function f is locally non-constant if, on every interval [a, b], it satisfies

$$\forall y \exists x. f(x) \neq y.$$

Lemma

Polynomials are locally non-constant.

<ロ> (四) (四) (三) (三) (三)

A constructive variant

Theorem

Let f be a locally non-constant function...

Definition

A function f is locally non-constant if, on every interval [a, b], it satisfies

$$\forall y \exists x. f(x) \neq y.$$

Lemma

Polynomials are locally non-constant.

(□) (□) (□) (□) (□)

크

Another constructive variant

Theorem

Let f be a strictly increasing function...

_emma

 $\lambda x.x^n - c$ is strictly increasing on [0, c + 1].

・ロト ・ 日 ・ ・ ヨ ・ ・ モ ト ・

Another constructive variant

Theorem

Let f be a strictly increasing function...

Lemma

 $\lambda x.x^n - c$ is strictly increasing on [0, c+1].

・ロト ・ 日 ・ ・ ヨ ・ ・ モ ト ・

크

The bad news

After several rounds of optimizations... the extracted program is still $\ensuremath{\mathsf{VERY}}$ slow.

• exponential complexity (factor pprox 3)

• no obvious improvements left...

(日) (部) (注) (注)

The bad news

After several rounds of optimizations... the extracted program is still VERY slow.

- exponential complexity (factor \approx 3)
- no obvious improvements left...

The bad news

After several rounds of optimizations... the extracted program is still VERY slow.

• exponential complexity (factor \approx 3)

• no obvious improvements left...

The bad news

After several rounds of optimizations... the extracted program is still VERY slow.

- exponential complexity (factor pprox 3)
- no obvious improvements left...

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

- reduced fractions
- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

(日) (四) (王) (王) (王)

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

reduced fractions

- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

・ロト ・ 日 ・ ・ ヨ ・ ・ モ ト ・

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

- reduced fractions
- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

イロト イヨト イヨト イヨト

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

- reduced fractions
- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

- reduced fractions
- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

(日) (同) (日) (日)

An alternative approach

Formalization of the FTA proof in the concrete model, with extraction in mind.

- reduced fractions
- explicit bounds for Cauchy sequences
- functions as limit of rational-valued functions

Much better results!

Unfortunately, not portable to C-CoRN

(D) (A) (A)

extraction is not a magic button

- extraction probably will never be a magic button
- proving and computing seem to be essentially different things
- no notion of "good" proof
- will "good" proofs yield good programs?

- extraction is not a magic button
- extraction probably will never be a magic button
- proving and computing seem to be essentially different things
- no notion of "good" proof
- will "good" proofs yield good programs?

(日) (四) (王) (王) (王)

- extraction is not a magic button
- extraction probably will never be a magic button
- proving and computing seem to be essentially different things
- no notion of "good" proof
- will "good" proofs yield good programs?

<ロ> (四) (四) (三) (三) (三)

크

- extraction is not a magic button
- extraction probably will never be a magic button
- proving and computing seem to be essentially different things
- no notion of "good" proof
- will "good" proofs yield good programs?

(日) (四) (王) (王) (王)

크

- extraction is not a magic button
- extraction probably will never be a magic button
- proving and computing seem to be essentially different things
- no notion of "good" proof
- will "good" proofs yield good programs?

イロト イポト イヨト イヨト