J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution intuitive)

Basic idea
Analyse of T
Decomposing T

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

onclusion

Proving termination using dependent types: the case of xor-terms

J.-F. Monin J. Courant

VERIMAG Grenoble, France

Trends in Functional Programming, Nottingham, 2006

Basic idea Analyse of T Decomposing TStratification

Lifting Alternation Fake incl. Fixpoints Conv rule

Motivation

The case of cryptographic systems

Back to cryptographic systems

Solving strategies

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst.

State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

- Protocols
- Security APIs

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

- Protocols
- Security APIs

Xor is ubiquitous

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivatio

Crypto syst.

State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

- ► Protocols
- ► Security APIs

Xor is ubiquitous

Examples from a security API called CCA (Common Cryptographic Architecture):

$$x, y, \{z\}_{x \oplus KP \oplus KM} \mapsto \{z \oplus y\}_{x \oplus KP \oplus KM}$$
$$x, y, \{z\}_{x \oplus KP \oplus KM} \mapsto \{z \oplus y\}_{x \oplus KM}$$

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting Alternation Fake incl Fixpoints

`an aluaian

- Protocols
- ► Security APIs

Xor is ubiquitous

Examples from a security API called CCA (Common Cryptographic Architecture):

$$x, y, \{z\}_{x \oplus KP \oplus KM} \mapsto \{z \oplus y\}_{x \oplus KP \oplus KM}$$
$$x, y, \{z\}_{x \oplus KP \oplus KM} \mapsto \{z \oplus y\}_{x \oplus KM}$$

Reasoning involves:

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$

Involutivity: $x \oplus x \simeq 0$

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto syst

State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssu es Lifting

Alternation Fake incl Fixpoints



Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Conclusion

Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of 2

Decomposing 7

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion

Fixpoint

Conversion rule

We are given

▶ A type of terms \mathcal{T} with constructors C_k : Inductive \mathcal{T} : Set := $\mid C_1 : \mathcal{T}$

 $\mid C_k : \ldots \to \mathcal{T} \ldots \to \mathcal{T} \ldots \to \mathcal{T}$

:

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

We are given

▶ A type of terms \mathcal{T} with constructors C_k : Inductive \mathcal{T} : Set :=

```
 \begin{array}{c} \mid \ C_1 : \mathcal{T} \\ \vdots \\ \mid \ C_k : \ldots \to \mathcal{T} \ldots \to \mathcal{T} \end{array}
```

▶ A congruence \simeq : $\mathcal{T} \to \mathcal{T} \to \textit{Prop}$

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing TStratification

Lifting Alternation Fake incl Fixpoints Conv. rule

We are given

 \triangleright A type of terms \mathcal{T} with constructors \mathcal{C}_k : Inductive \mathcal{T} : Set :=

$$| C_1 : \mathcal{T}$$

$$\vdots$$

$$| C_k : \ldots \to \mathcal{T} \ldots \to \mathcal{T} \ldots \to \mathcal{T}$$

- ▶ A congruence \simeq : $\mathcal{T} \to \mathcal{T} \to Prop$
 - For each constructor C_ν $\forall a, \ldots x_1, y_1, b, \ldots x_2, y_2, \ldots c,$ $x_1 \simeq y_1 \rightarrow x_2 \simeq y_2 \rightarrow$ $C_{\nu} a \dots x_1 b \dots v_1 c \simeq C_{\nu} a \dots x_2 b \dots v_2 c$

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing T Stratification

Lifting Alternation Fake incl Fixpoints Conv. rule

We are given

 \triangleright A type of terms \mathcal{T} with constructors \mathcal{C}_k : Inductive \mathcal{T} : Set :=

$$\mid C_1 : \mathcal{T} \mid C_k : \ldots \to \mathcal{T} \ldots \to \mathcal{T} \ldots \to \mathcal{T}$$

- ▶ A congruence \simeq : $\mathcal{T} \to \mathcal{T} \to Prop$
 - For each constructor C_ν $\forall a, \ldots x_1, y_1, b, \ldots x_2, y_2, \ldots c,$ $x_1 \simeq y_1 \rightarrow x_2 \simeq y_2 \rightarrow$ $C_k a \dots x_1 b \dots y_1 c \simeq C_k a \dots x_2 b \dots y_2 c$
 - ▶ specific laws, e.g. $\forall xy$, $C_2 \times C_1 y \simeq C_2 y \times C_2 x = 0$

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing T Stratification

Lifting Alternation Fake incl Fixpoints Conv. rule

We are given

 \triangleright A type of terms \mathcal{T} with constructors \mathcal{C}_k :

Inductive \mathcal{T} : Set := $\mid C_1 : T$ $C_{\nu}:\ldots\to\mathcal{T}\ldots\to\mathcal{T}$

▶ A congruence \simeq : $\mathcal{T} \to \mathcal{T} \to Prop$

For each constructor C_ν $\forall a, \ldots x_1, y_1, b, \ldots x_2, y_2, \ldots c,$ $x_1 \simeq y_1 \rightarrow x_2 \simeq y_2 \rightarrow$

 $C_k a \dots x_1 b \dots y_1 c \simeq C_k a \dots x_2 b \dots y_2 c$

▶ specific laws, e.g. $\forall xy$, $C_2 \times C_1 y \simeq C_2 y \times C_2 x = 0$

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

lssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

We are given

▶ A type of terms \mathcal{T} with constructors C_k : Inductive \mathcal{T} : Set :=

 $\mid C_1 : \mathcal{T} \mid C_k : \ldots \to \mathcal{T} \ldots \to \mathcal{T} \ldots \to \mathcal{T}$

▶ A congruence \simeq : $\mathcal{T} \to \mathcal{T} \to \textit{Prop}$

► For each constructor C_k $\forall a, \dots x_1, y_1, b, \dots x_2, y_2, \dots c,$ $x_1 \simeq y_1 \rightarrow x_2 \simeq y_2 \rightarrow$ $C_k a \dots x_1 b \dots y_1 c \simeq C_k a \dots x_2 b \dots y_2 c$ ► specific laws, e.g. $\forall xy, C_2 \times C_1 y \simeq C_2 y \times c$

We want to reason on ${\mathcal T}$ up to \simeq

Already well-known examples

▶ finite bags represented by finite lists

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Issues

Lifting Alternation Fake incl Fixpoints Conv rule

Already well-known examples

▶ finite bags represented by finite lists

▶ algebra of formal arithmetic expressions

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Lifting
Alternation
Fake incl
Fixpoints

Conclusion

▶ finite bags represented by finite lists

algebra of formal arithmetic expressions

▶ (mobile) process calculi, chemical abstract machines

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

▶ finite bags represented by finite lists

▶ algebra of formal arithmetic expressions

+ is associative, commutative, ${\tt 0}$ is neutral

imes is associative, commutative, 1 is neutral

 \times distributes over +

 (mobile) process calculi, chemical abstract machines parallel composition and choice operators are AC

Quotients in type theory

► High level approach : setoids

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

Quotients in type theory

► High level approach : setoids

► Explicit approach :

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

.

Lifting Alternation Fake incl Fixpoints

Quotients in type theory

► High level approach : setoids

- Explicit approach :
 - ightharpoonup Define a normalization function N on \mathcal{T}

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting Alternation Fake incl Fixpoints

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

► High level approach : setoids

- Explicit approach :
 - ightharpoonup Define a normalization function N on T
 - ► Compare terms using syntactic equality on their norms : $x \simeq y$ iff Nx = Ny

Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Conclusion

Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of 2

Decomposing \mathcal{T}

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion

Fixpoint

Conversion rule

Cryptographic systems need more

Reasoning on such systems involves

ightharpoonup comparing terms up to AC + involutivity of \oplus :

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$

Involutivity: $x \oplus x \simeq 0$

Proving termination using dependent types

J.-F. Monin, J. Courant

iviotivation

Crypto. syst. State of the art

Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

 \triangleright comparing terms up to AC + involutivity of \oplus :

Commutativity: $x \oplus y \simeq y \oplus x$

 $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$ Associativity:

Neutral element: $x \oplus 0 \simeq x$ Involutivity: $x \oplus x \simeq 0$

if x, y and z are different terms,

Proving termination using dependent types

> J.-F. Monin, J. Courant

Crypto, syst. State of the art

Back to crypto Solving strategies

Basic idea Analyse of T

Decomposing TStratification

Lifting Alternation Fake incl. Fixpoints

Conv. rule

▶ comparing terms up to AC + involutivity of ⊕:

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$

Involutivity: $x \oplus x \simeq 0$

▶ a relation <u></u> for occurrence: if x, y and z are different terms,

ightharpoonup y occurs in $x \oplus y \oplus z$

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art

Back to crypto Solving strategies

Solution (intuitive) Basic idea

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$ Involutivity: $x \oplus x \simeq 0$

- ▶ a relation \leq for occurrence: if x, y and z are different terms,
 - ightharpoonup y occurs in $x \oplus y \oplus z$
 - ▶ but y does not occur in $x \oplus y \oplus z \oplus y$

Proving termination using dependent types

> J.F. Monin J. Courant

Motivation

Crypto. syst. State of the art

Back to crypto Solving strategies

(intuitive)

Basic idea

Analyse of T

Decomposing T

Stratification

Lifting
Alternation

Alternation Fake incl Fixpoints

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$ Involutivity: $x \oplus x \simeq 0$

- ▶ a relation \leq for occurrence: if x, y and z are different terms,
 - ightharpoonup y occurs in $x \oplus y \oplus z$
 - ▶ but y does not occur in $x \oplus y \oplus z \oplus y$

Proving termination using dependent types

> J.F. Monin J. Courant

Motivation

Crypto. syst. State of the art

Back to crypto Solving strategies

(intuitive)

Basic idea

Analyse of T

Decomposing T

Stratification

Lifting
Alternation

Alternation Fake incl Fixpoints

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$ Involutivity: $x \oplus x \simeq 0$

▶ a relation \leq for occurrence: if x, y and z are different terms,

- \triangleright y occurs in $x \oplus y \oplus z$
- ▶ but y does not occur in $x \oplus y \oplus z \oplus y$

$$x \leq y$$
 if $x \simeq y$
 $x \leq t$ if $t \simeq x \oplus y_0 \dots \oplus y_n$
and $x \not\simeq y_i$ for all $i, 0 \leq i \leq n$

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art

Back to crypto Solving strategies

(intuitive)

Basic idea

Analyse of T

Decomposing T

Stratification

Lifting

Alternation
Fake incl
Fixpoints
Conv. rule

Commutativity: $x \oplus y \simeq y \oplus x$

Associativity: $(x \oplus y) \oplus z \simeq x \oplus (y \oplus z)$

Neutral element: $x \oplus 0 \simeq x$

Involutivity: $x \oplus x \simeq 0$

▶ a relation \leq for occurrence: if x, y and z are different terms,

- ightharpoonup y occurs in $x \oplus y \oplus z$
- ▶ but y does not occur in $x \oplus y \oplus z \oplus y$

$$x \leq y$$
 if $x \simeq y$
 $x \leq t$ if $t \simeq x \oplus y_0 \dots \oplus y_n$
and $x \not\simeq y_i$ for all $i, 0 \leq i \leq n$

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst. State of the art Back to crypto

Solving strategies

(intuitive)

Basic idea

Analyse of T

Decomposing T

Stratification

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Outline

Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of 7

Decomposing 7

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion:

Fixpoint

Conversion rule

Conclusion

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

leeu ee

Lifting Alternation Fake incl Fixpoints Conv rule

Replace equations with rewrite rules

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst State of the art Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

► Not very difficult — use general recursion

Proving termination using dependent types

J.-F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of TDecomposing TStratification

Lifting Alternation Fake incl Fixpoints Conv. rule

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

- ► Not very difficult use general recursion
- ► Just boring

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T

cellec

Lifting Alternation Fake incl Fixpoints

First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

- ► Not very difficult use general recursion
- ► Just boring

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T

ssues

Lifting Alternation Fake incl Fixpoints

Jution

(intuitive)

Basic idea

Analyse of \mathcal{T} Decomposing \mathcal{T}

Stratification Issues Lifting Alternation

Fake incl Fixpoints Conv rule

Conclusion

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

- ► Not very difficult use general recursion
- ► Just boring

In a type theoretic framework, termination proof mandatory and non-trivial:

combination of polynomial and lexicographic ordering

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

- ► Not very difficult use general recursion
- ► Just boring

In a type theoretic framework, termination proof mandatory and non-trivial:

- combination of polynomial and lexicographic ordering
- other approaches (lpo, rpo,...): overkill?

Conclusion

Replace equations with rewrite rules

Commutativity: find an suitable well ordering on terms

Functional programming approach:

- ► Not very difficult use general recursion
- ► Just boring

In a type theoretic framework, termination proof mandatory and non-trivial:

- combination of polynomial and lexicographic ordering
- other approaches (lpo, rpo,...): overkill?
- ► AC matching: a non trivial matter

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst.

State of the art

Back to crypto

Solving strategies

Solution

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

Step 1

Consider a more structured version of t

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

lution

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

Step 1

Consider a more structured version of t

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

lution

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

Step 1

- Consider a more structured version of t
 - = provide an accurate and informative typing to t

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst. State of the art Back to crypto

Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Step 1

- Consider a more structured version of t
 - $=\,$ provide an accurate and informative typing to t

Step 2

Normalize by structural induction on the newly typed version of t Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing T

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Step 1

- Consider a more structured version of t
 - $=\,$ provide an accurate and informative typing to t

Step 2

Normalize by structural induction on the newly typed version of t Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing T

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Step 1

- Consider a more structured version of t
 - = provide an accurate and informative typing to t

Step 2

Normalize by structural induction on the newly typed version of t

Step 1 makes step 2 easy.

Proving termination using dependent types

J.F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution

Basic idea
Analyse of TDecomposing T

ssues

Lifting Alternation Fake incl Fixpoints

Step 1

- Consider a more structured version of t
 provide an accurate and informative typing to t
- Step 2
 - Normalize by structural induction on the newly typed version of t

Step 1 makes step 2 easy.

Better formulation: $t: \mathcal{T}$ transformed into $t': \mathcal{T}'$ \mathcal{T}' enriched version of \mathcal{T} , trivial forgetful morphism $\mathcal{T}' \to \mathcal{T}$. Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Lifting
Alternation
Fake incl
Fixpoints

Proving termination

Lifting Alternation Fake incl. Fixpoints Conv. rule

Step 1

Consider a more structured version of t = provide an accurate and informative typing to t

Step 2

Normalize by structural induction on the newly typed version of t

Step 1 makes step 2 easy.

Better formulation: $t: \mathcal{T}$ transformed into $t': \mathcal{T}'$ T' enriched version of T. trivial forgetful morphism $\mathcal{T}' \to \mathcal{T}$.

Interesting part = $T \rightarrow T'$

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Conclusion

Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of \mathcal{I}

Decomposing T

Stratifying and normalizing a terr

Issues

Lifting

Alternation

Forbid fake inclusion

Fixpoints

Conversion rule

Lunch time!



Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ı

Lifting Alternation Fake incl Fixpoints Conv rule



Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of T Decomposing T Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule



► layering a term

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea Analyse of T Decomposing T Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule



- ► layering a term
- ▶ layers do not communicate: each layer possesses its own normalization function

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule



- layering a term
- layers do not communicate: each layer possesses its own normalization function
- ▶ in our case: need 2 layers, pasta and sauce

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting Alternation Fake incl Fixpoints Conv rule



- layering a term
- layers do not communicate: each layer possesses its own normalization function
- ▶ in our case: need 2 layers, pasta and sauce
- normalizing pasta = identity

Proving termination using dependent types

> J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule



- ► layering a term
- layers do not communicate: each layer possesses its own normalization function
- ▶ in our case: need 2 layers, pasta and sauce
- ► normalizing pasta = identity
- ▶ normalizing sauce = rearranging + removing duplicates

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Analyse of \mathcal{T}

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of $\mathcal T$ Decomposing TStratification

Lifting Alternation Fake incl. Fixpoints Conv. rule

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

Issue

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

Inductive T: Set :=

Zero: T

 $PC: public_const o \mathcal{T}$

 $E\colon \mathcal{T} \to \mathcal{T} \to \mathcal{T}$

 $\textit{Xor} \colon\thinspace \mathcal{T} \to \mathcal{T} \to \mathcal{T}$

Hash: $\mathcal{T} o \mathcal{T} o \mathcal{T}$

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of TDecomposing TStratification

Lifting Alternation Fake incl Fixpoints Conv. rule

 $\mid SC \colon secret_const \to \mathcal{T}$

Inductive \mathcal{T} : Set :=

| Zero: T

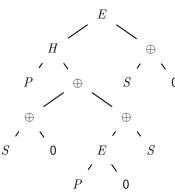
 $PC: public_const \rightarrow \mathcal{T}$

 \mid SC: secret_const \rightarrow T

 $E\colon\thinspace \mathcal{T}\to\mathcal{T}\to\mathcal{T}$

Xor: $\mathcal{T} o \mathcal{T} o \mathcal{T}$

Hash: $\mathcal{T} o \mathcal{T} o \mathcal{T}$



Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

Inductive T: Set :=

| Zero: T

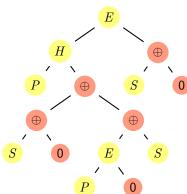
 $PC\colon \mathsf{public_const} o \mathcal{T}$

 \mid SC: secret_const \rightarrow T

 $E \colon \mathcal{T} \to \mathcal{T} \to \mathcal{T}$

Xor: $\mathcal{T} o \mathcal{T} o \mathcal{T}$

Hash: $\mathcal{T} o \mathcal{T} o \mathcal{T}$



Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

(intuitive) Basic idea

Analyse of TDecomposing TStratification

ssue

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of 7

Decomposing ${\mathcal T}$

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion:

Fixpoints

Conversion rule

Conclusion

Proving termination using dependent types

J.F. Monin J. Courant

Motivation

Crypto. syst.

State of the art

Back to crypto

Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

ssues

Lifting Alternation Fake incl Fixpoints Conv rule

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

on clusion

Inductive T_x : Set := $|X_Z|$ Zero : T_x

 X_X or: $T_X \to T_X \to T_X$

Inductive T_n : Set :=

 $\mid \mathit{NX_PC} : \mathit{public_const}
ightarrow \mathcal{T}_{\mathit{n}} \ \mid \mathit{NX_SC} : \mathit{secret_const}
ightarrow \mathcal{T}_{\mathit{n}}$

 $NX_-E:\mathcal{T}_n o \mathcal{T}_n o \mathcal{T}_n$

 $\mathsf{NX} extstyle\mathsf{-Hash}:\,\mathcal{T}_\mathsf{n} o\mathcal{T}_\mathsf{n} o\mathcal{T}_\mathsf{n}$

Solution (intuitive) Basic idea

Analyse of T

Decomposing T

Stratification

Lifting Alternati

Alternation Fake incl Fixpoints

Conclusion

Variable A : Set.

Inductive T_x : Set :=

 $\mid X_{-}Zero : \mathcal{T}_{_{X}}$

 X_X or: $T_X \to T_X \to T_X$

 $|X_ns:A\to T_x$

Inductive T_n : Set :=

 $\mid NX_PC: public_const
ightarrow \mathcal{T}_n$

 NX_SC : $secret_const o \mathcal{T}_n$

 $NX_-E: \mathcal{T}_n \to \mathcal{T}_n \to \mathcal{T}_n$

 $NX_{-}Hash: \mathcal{T}_{n}
ightarrow \mathcal{T}_{n}
ightarrow \mathcal{T}_{n}$

 $NX_sum:A o \mathcal{T}_n$

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Stratifying and normalizing a term

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing T Stratification

Lifting Alternation Fake incl. Fixpoints Conv. rule

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

Issues

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero$, $Xor \mapsto X_{-}Xor$, $PC \mapsto NX_{-}PC$, etc.

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T

.

Lifting Alternation Fake incl Fixpoints Conv rule

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero$, $Xor \mapsto X_{-}Xor$, $PC \mapsto NX_{-}PC$, etc.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing T

Lifting Alternation

Alternation Fake incl Fixpoints

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero$, $Xor \mapsto X_{-}Xor$, $PC \mapsto NX_{-}PC$, etc.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

 $ightharpoonup \mathcal{T}_n(A)$ is sortable as well;

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

olution intuitive)

Basic idea
Analyse of TDecomposing T

Lifting

Alternation
Fake incl
Fixpoints
Conv. rule

C - - - | - - | - - -

Basic idea Analyse of TDecomposing T Stratification

Lifting

Alternation Fake incl. Fixpoints Conv. rule

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero, Xor \mapsto X_{-}Xor, PC \mapsto NX_{-}PC$, etc.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

- $ightharpoonup T_n(A)$ is sortable as well;
- ▶ the multiset of A-leaves of a $T_x(A)$ -term can be sorted (and removed when possible) into a list;

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

onclusion

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_Z Zero$, $Xor \mapsto X_Z Xor$, $PC \mapsto NX_Z PC$, etc.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

- $ightharpoonup \mathcal{T}_n(A)$ is sortable as well;
- ▶ the multiset of A-leaves of a $T_x(A)$ -term can be sorted (and removed when possible) into a list;
- ▶ list(A) is sortable.

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting Alternation Fake incl Fixpoints

Conclusion

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero$, $Xor \mapsto X_{-}Xor$, $PC \mapsto NX_{-}PC$, etc.

The typing of t' is $\underbrace{\mathcal{T}_{x}(\mathcal{T}_{n}(\mathcal{T}_{x}(\dots(\emptyset))))}_{k \text{ layers}}$ for k large enough.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

- $ightharpoonup \mathcal{T}_n(A)$ is sortable as well;
- the multiset of A-leaves of a T_x(A)-term can be sorted (and removed when possible) into a list;
- ▶ list(A) is sortable.

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting Alternation Fake incl Fixpoints

Conclusion

Step 1 Translate a term t into t' according to the mapping $0 \mapsto X_{-}Zero$, $Xor \mapsto X_{-}Xor$, $PC \mapsto NX_{-}PC$, etc.

The typing of t' is $\underbrace{\mathcal{T}_{x}(\mathcal{T}_{n}(\mathcal{T}_{x}(\dots(\emptyset))))}_{k \text{ layers}}$ for k large enough.

Step 2 A type is sortable if it is equipped with a decidable equality and a decidable total ordering. If A is sortable, then

- $ightharpoonup \mathcal{T}_n(A)$ is sortable as well;
- the multiset of A-leaves of a T_x(A)-term can be sorted (and removed when possible) into a list;
- ▶ list(A) is sortable.

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of \mathcal{I}

Decomposing \mathcal{T}

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion

Fixpoints

Conversion rule

Conclusion

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

sues

Lifting

Alternation Fake incl Fixpoints Conv rule

Lifting lasagna

$$\mathcal{L}_X k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_X(\mathcal{T}_n(\mathcal{T}_X(\dots(\emptyset))))}_{k \text{ layers}} \text{ for } k \text{ large enough.}$$

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

Issue

Lifting

Alternation Fake incl Fixpoints Conv rule

Lifting lasagna

$$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ layers}}$$
 for k large enough.

 \blacktriangleright What is k?

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting

Alternation Fake incl Fixpoints Conv rule

- \triangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

Proving termination using dependent types

> J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

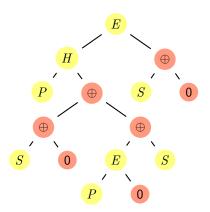
ssues

Lifting

Alternation Fake incl Fixpoints

$$\mathcal{L}_X k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_X(\mathcal{T}_n(\mathcal{T}_X(\dots(\emptyset))))}_{k \text{ lavers}} \text{ for } k \text{ large enough.}$$

- \blacktriangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.



Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

issues

Lifting Alternation

Fake incl
Fixpoints
Conv rule

Lifting lasagna

$$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_n(\mathcal{T}_x(\dots(\emptyset))))}_{k \text{ lavers}}$$
 for k large enough.

- \triangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

Take the max

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting

Alternation Fake incl Fixpoints

- ▶ What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

► Standard solution: {le n m} + {le m n}

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

ssues

Lifting

Alternation Fake incl Fixpoints

$$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ lavers}} \text{ for } k \text{ large enough.}$$

- \triangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

- ► Standard solution: {le n m} + {le m n}
 - interactive definition, large proof term

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

ssues

Lifting

Alternation Fake incl Fixpoints

$$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ lavers}} \text{ for } k \text{ large enough.}$$

- \triangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

- ► Standard solution: {le n m} + {le m n}
 - ▶ interactive definition, large proof term
 - ▶ heavy encoding of m n or n m

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

ssues

Lifting

Alternation
Fake incl
Fixpoints
Conv. rule

$$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ lavers}} \text{ for } k \text{ large enough.}$$

- ▶ What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

- ► Standard solution: {le n m} + {le m n}
 - ▶ interactive definition, large proof term
 - ▶ heavy encoding of m n or n m
 - ▶ need to lift \mathcal{L}_{x} n and \mathcal{L}_{x} m to \mathcal{L}_{x} (max n m)

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting

Alternation Fake incl Fixpoints

k lavers

- \triangleright What is k?
- ► The number of layers on the left subterm and on the right subterm are different in general.

Take the max

- ► Standard solution: {le n m} + {le m n}
 - ▶ interactive definition, large proof term
 - ▶ heavy encoding of m n or n m
 - ▶ need to lift $\mathcal{L}_{\times} n$ and $\mathcal{L}_{\times} m$ to $\mathcal{L}_{\times} (\max n m)$
- ▶ Lightweight approach: max $n m \stackrel{\text{def}}{=} m + (n m)$

Proving termination using dependent types

> J.-F. Monin J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting

Alternation
Fake incl
Fixpoints
Conv. rule

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

Issues

Lifting Alternation Fake incl

Fake incl Fixpoints Conv rule

Conclusion

- $\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ lavers}} \text{ for } k \text{ large enough.}$
 - \triangleright What is k?
 - ► The number of layers on the left subterm and on the right subterm are different in general.

Take the max

- ► Standard solution: {le n m} + {le m n}
 - interactive definition, large proof term
 - ▶ heavy encoding of m n or n m
 - ▶ need to lift $\mathcal{L}_x n$ and $\mathcal{L}_x m$ to \mathcal{L}_x (max n m)
- ▶ Lightweight approach: max $n m \stackrel{\text{def}}{=} m + (n m)$
 - $\qquad \qquad \textbf{lift}_{x}: \mathcal{L}_{x} \ k \rightarrow \mathcal{L}_{x} \ (k+d), \ \textbf{lift}_{n}: \mathcal{L}_{n} \ k \rightarrow \mathcal{L}_{n} \ (k+d)$

$$\mathcal{L}_x k \stackrel{\mathrm{def}}{=} \underbrace{\mathcal{T}_x(\mathcal{T}_x(\mathcal{L}_x(\dots(\emptyset))))}_{k \text{ layers}} \text{ for } k \text{ large enough.}$$

- \triangleright What is k?
- ▶ The number of layers on the left subterm and on the right subterm are different in general.

- Standard solution: {le n m} + {le m n}
 - interactive definition, large proof term
 - ▶ heavy encoding of m-n or n-m
 - ▶ need to lift $\mathcal{L}_x n$ and $\mathcal{L}_x m$ to \mathcal{L}_x (max n m)
- ▶ Lightweight approach: max $n m \stackrel{\text{def}}{=} m + (n m)$
 - $lift_x : \mathcal{L}_x k \to \mathcal{L}_x (k+d)$, $lift_n : \mathcal{L}_n k \to \mathcal{L}_n (k+d)$
 - No need to proof that max is the max.

Proving termination using dependent types

J.-F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing T Stratification

Lifting

Alternation Fake incl Fixpoints Conv. rule

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

Solution (intuitive)

Basic idea

Analyse of 7

Decomposing \mathcal{T}

Stratifying and normalizing a term

Issues

Lifting

Alternation

Forbid fake inclusion

Fixpoint

Conversion rule

Conclusion

Proving termination using dependent types

> J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

sues

Lifting Alternation Fake incl

Fake incl Fixpoints

Conclusion

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivatio

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of ${\mathcal T}$ Decomposing ${\mathcal T}$ Stratification

SSILES

Lifting Alternation Fake incl

Fixpoints Conv rule

Well designed types help us to design programs

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting Alternation Fake incl

Fake incl Fixpoints Conv rule

Well designed types help us to design programs

Many functions are defined by mutual induction, e.g. $lift_x$ and $lift_n$

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting Alternation Fake incl

Fixpoints

Well designed types help us to design programs

Many functions are defined by mutual induction, e.g. $lift_x$ and $lift_n$

Control them using alternating natural numbers

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Well designed types help us to design programs

Many functions are defined by mutual induction, e.g. $lift_x$ and $lift_n$

Control them using alternating natural numbers

```
Inductive alt_{even}: Set :=  \mid O_e: alt_{even} \mid S_{o \rightarrow e}: alt_{odd} \rightarrow alt_{even} with alt_{odd}: Set :=  \mid S_{e \rightarrow e}: alt_{even} \rightarrow alt_{odd}
```

Proving termination using dependent types

J.-F. Monin J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting Alternation Fake incl

Fake incl Fixpoints

Back to cryptographic systems

Solving strategies

Issues

Forbid fake inclusions

Proving termination using dependent types

> J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing TStratification

Lifting Alternation Eake incl Fixpoints

Conv. rule

Forbid fake inclusions

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

J.-F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of TDecomposing TStratification

Lifting Alternation Fake incl

Fixpoints Conv. rule

Inductive T_x : Set :=

 $X_Zero: T_x$

 $X_ns: A \to T_x$

 X_X

Inductive T_n : Set :=

 $|NX_PC: public_const \rightarrow \mathcal{T}_n$

 NX_SC : secret_const $\rightarrow T_n$

 $NX_{-}sum: A \rightarrow \mathcal{T}_{n}$

 $NX_{-}E: \mathcal{T}_{n} \to \mathcal{T}_{n} \to \mathcal{T}_{n}$

 $\mid NX_{Hash}: \mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

Lifting

Alternation
Fake incl
Fixpoints
Conv. rule

convitue

Conclusion

Inductive T_x : Set :=

 $\mid X_{-}Zero : \mathcal{T}_{x}$

 $X_ns: A \to T_x$

 $\mid X_{-}Xor: \mathcal{T}_{x} \rightarrow \mathcal{T}_{x} \rightarrow \mathcal{T}_{x}$

Inductive T_n : Set :=

 $\mid NX_PC : public_const \rightarrow \mathcal{T}_n$

 NX_SC : $secret_const o \mathcal{T}_n$

 $NX_{-}sum: A \rightarrow \mathcal{T}_{n}$

 $\mid \mathit{NX}_{-}\mathit{E} : \; \mathcal{T}_{n} \;
ightarrow \mathcal{T}_{n} \;
ightarrow \mathcal{T}_{n}$

 $\mid \mathit{NX}_\mathit{Hash}: \ \mathcal{T}_\mathsf{n} \ o \mathcal{T}_\mathsf{n} \ o \mathcal{T}_\mathsf{n}$

 X_{ns} (NX_sum (X_{ns} (NX_sum (...))))

olution intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

sues

Alternation Fake incl

Fake incl Fixpoints

```
Inductive T_x: bool \rightarrow Set :=
    | X_Zero : \forall b, T_x b |
     X_{-}ns: \forall b, I_{S_{-}}true b \rightarrow A \rightarrow T_{x} b
    | X_X \times X or | \forall b, T_X \text{ true} \rightarrow T_X \text{ true} \rightarrow T_X \text{ b}
Inductive T_n: bool \rightarrow Set :=
     NX_{-}PC: \forall b, public\_const \rightarrow \mathcal{T}_{n} b
      NX\_SC: \forall b, secret\_const \rightarrow T_n b
     NX_{sum}: \forall b. ls_{true} b \rightarrow A \rightarrow T_n b
     NX_{-}E: \forall b, T_n \text{ true} \rightarrow T_n \text{ true} \rightarrow T_n b
     NX_{-}Hash: \forall b, T_n true \rightarrow T_n true \rightarrow T_n b
X_{ns} (NX_sum ( X_{ns} (NX_sum (...))))
```

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing TStratification

Lifting Alternation Fake incl. Fixpoints

Conv rule

Back to cryptographic systems

Solving strategies

Stratifying and normalizing a term

Issues

Fixpoints

▶ Prefer fixpoints: built-in computation, no inversion

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv rule

100

- ▶ Prefer fixpoints: built-in computation, no inversion
- ► Use map combinators

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

100

- ▶ Prefer fixpoints: built-in computation, no inversion
- ► Use map combinators

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

100

- Prefer fixpoints: built-in computation, no inversion
- ► Use map combinators

Many 10 lines definitions, almost no theorem

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Basic idea
Analyse of T
Decomposing T
Stratification

Lifting
Alternation
Fake incl
Fixpoints

Conclusion

▶ Prefer fixpoints: built-in computation, no inversion

► Use map combinators

Many 10 lines definitions, almost no theorem

```
Fixpoint lift_lasagna_x e_1 e_2 \{struct e_1\}:
    \mathcal{L}_{x} e_{1} \rightarrow \mathcal{L}_{x} (e_{1} + e_{2}) :=
    match e_1 return \mathcal{L}_x e_1 	o \mathcal{L}_x (e_1 + e_2) with
    \mid \theta_e \Rightarrow \text{fun } emp \Rightarrow \text{match } emp \text{ with end}
    |S_{o\rightarrow e} o_1 \Rightarrow map_{\nu} (lift_lasagna_n o_1 e_2) false
    end
with lift_lasagna_n o_1 e_2 \{struct o_1\}:
   \mathcal{L}_n \ o_1 \rightarrow \mathcal{L}_n \ (o_1 + e_2) :=
    match o_1 return \mathcal{L}_n o_1 \to \mathcal{L}_n (o_1 + e_2) with
    |S_{e\rightarrow o}|e_1 \Rightarrow map_n (lift_lasagna_x e_1 e_2) false
    end.
```

Back to cryptographic systems

Solving strategies

Stratifying and normalizing a term

Issues

Conversion rule

Proving termination using dependent types

J.F. Monin, J. Courant

Crypto, syst. State of the art Back to crypto Solving strategies

Basic idea Analyse of T Decomposing TStratification

Lifting Alternation Fake incl. Fixpoints Conv. rule

4□ → 4問 → 4 = → 4 = → 9 Q P

Conversion rule

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivatio

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

..........

Conversion rule

Used everywhere

Proving termination using dependent types

J-F Monin, J. Courant

State of the art Back to crypto Solving strategies

Basic idea Analyse of TDecomposing TStratification

Lifting Alternation Fake incl Fixpoints Conv rule

Solution (intuitive) Basic idea

Basic idea Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

Lifting Alternation

Alternation Fake incl Fixpoints

Conv rule

Conclusion

Used everywhere

```
Definition bin\_xor
(bin: \forall A b, T_x A true \rightarrow T_x A true \rightarrow T_x A b) o_1 o_2 b
(l_1: lasagna\_cand\_x o_1 true)
(l_2: lasagna\_cand\_x o_2 true):
lasagna\_cand\_x (max\_oo o_1 o_2) b :=
bin (\mathcal{L}_n (max\_oo o_1 o_2)) b
(lift\_lasagna\_cand\_x true o_1 (o_2 - o_1) l_1)
(coerce\_max\_comm
(lift\_lasagna\_cand\_x true o_2 (o_1 - o_2) l_2)).
```

Type theory is flexible

Polymorphism

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst. State of the art Back to crypto Solving strategies

Solution (intuitive)

Basic idea
Analyse of $\mathcal T$ Decomposing $\mathcal T$ Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

Type theory is flexible

- ► Polymorphism
- ► Mutually inductive types

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T} Stratification

SSILES

Lifting Alternation Fake incl Fixpoints Conv rule

Type theory is flexible

- ► Polymorphism
- ► Mutually inductive types
- ▶ Dependent types

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of TDecomposing TStratification

.

Lifting Alternation Fake incl Fixpoints

Type theory is flexible

- ► Polymorphism
- ► Mutually inductive types
- Dependent types
- ► Conversion rule

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

.

Lifting Alternation Fake incl Fixpoints

Type theory is flexible

- ► Polymorphism
- Mutually inductive types
- ▶ Dependent types
- ► Conversion rule
- ► No JMEQ

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

.

Lifting Alternation Fake incl Fixpoints

Type theory is flexible

- ► Polymorphism
- Mutually inductive types
- ▶ Dependent types
- ► Conversion rule
- ► No JMEQ

Proving termination using dependent types

J.-F. Monin, J. Courant

Motivation

Crypto. syst.
State of the art
Back to crypto
Solving strategies

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

.

Lifting Alternation Fake incl Fixpoints

Solution (intuitive)

Basic idea
Analyse of T
Decomposing T
Stratification

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

Type theory is flexible

- ► Polymorphism
- ► Mutually inductive types
- Dependent types
- Conversion rule
- ► No JMEQ

Future work

Study variants, compare and choose the best

Solution (intuitive)

Basic idea
Analyse of \mathcal{T} Decomposing \mathcal{T}

ssues

Lifting
Alternation
Fake incl
Fixpoints
Conv. rule

Conclusion

Type theory is flexible

- ► Polymorphism
- ► Mutually inductive types
- Dependent types
- ► Conversion rule
- ► No JMEQ

Future work

- Study variants, compare and choose the best
- application to CCA