

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

J.-F. Monin   J. Courant

VERIMAG  
Grenoble, France

Trends in Functional Programming, Nottingham, 2006

## 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

## Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

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

### Issues

Lifting

Alternation

Fake incl

Fixpoints

Conv rule

### Conclusion

# Formal models of cryptographic systems

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

## Conclusion

# Formal models of cryptographic systems

- ▶ 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

Analysis of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratification

## Issues

Lifting

Alternation

Fix incl

Fixpoints

Conv rule

## Conclusion

# Formal models of cryptographic systems

- ▶ Protocols
- ▶ Security APIs

Xor is ubiquitous

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

Conclusion

# Formal models of cryptographic systems

- ▶ 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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Formal models of cryptographic systems

- ▶ 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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Outline

## Motivation

The case of cryptographic systems

**State of the art**

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

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

### Issues

Lifting

Alternation

Fake incl

Fixpoints

Conv rule

### Conclusion

# General setting: quotiented first order-terms

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $\text{Set} :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

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

Conclusion

# General setting: quotiented first order-terms

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $Set :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

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

## 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

## Conclusion

# General setting: quotiented first order-terms

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $\text{Set} :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

- ▶ A congruence  $\simeq : \mathcal{T} \rightarrow \mathcal{T} \rightarrow \text{Prop}$

- ▶ For each constructor  $C_k$

$\forall a, \dots, x_1, y_1, b, \dots, x_2, y_2, \dots, c,$

$x_1 \simeq y_1 \rightarrow x_2 \simeq y_2 \rightarrow$

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

## Motivation

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

## Solution (intuitive)

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# General setting: quotiented first order-terms

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $\text{Set} :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

- ▶ A congruence  $\simeq : \mathcal{T} \rightarrow \mathcal{T} \rightarrow \text{Prop}$

- ▶ 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.  $\forall xy, C_2 x C_1 y \simeq C_2 y x$

## Motivation

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

## Solution (intuitive)

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# General setting: quotiented first order-terms

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $\text{Set} :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

- ▶ A congruence  $\simeq : \mathcal{T} \rightarrow \mathcal{T} \rightarrow \text{Prop}$

- ▶ 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.  $\forall xy, C_2 x C_1 y \simeq C_2 y x$

## Motivation

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

## Solution (intuitive)

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# General setting: quotiented first order-terms

We are given

- ▶ A type of terms  $\mathcal{T}$  with constructors  $C_k$ :

Inductive  $\mathcal{T}$ :  $\text{Set} :=$

|  $C_1 : \mathcal{T}$

⋮

|  $C_k : \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T} \dots \rightarrow \mathcal{T}$

⋮

- ▶ A congruence  $\simeq : \mathcal{T} \rightarrow \mathcal{T} \rightarrow \text{Prop}$

- ▶ 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.  $\forall xy, C_2 x C_1 y \simeq C_2 y x$

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

## Motivation

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

## Solution (intuitive)

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# 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

Analysis of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratification

## Issues

Lifting

Alternation

Fake incl

Fixpoints

Conv rule

## Conclusion

# 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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Already well-known examples

- ▶ finite bags represented by finite lists
- ▶ algebra of formal arithmetic expressions
- ▶ (mobile) process calculi, chemical abstract machines

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

## Conclusion

# Already well-known examples

- ▶ finite bags represented by finite lists
- ▶ algebra of formal arithmetic expressions
  - + is associative, commutative, 0 is neutral
  - $\times$  is associative, commutative, 1 is neutral
  - $\times$  distributes over +
- ▶ (mobile) process calculi, chemical abstract machines
  - parallel composition and choice operators are AC

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

Conclusion

# 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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Quotients in type theory

- ▶ High level approach : setoids
  
- ▶ Explicit approach :
  - ▶ 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Quotients in type theory

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

- ▶ High level approach : setoids
  
- ▶ Explicit approach :
  - ▶ Define a normalization function  $N$  on  $\mathcal{T}$
  - ▶ Compare terms using syntactic equality on their norms :  
 $x \simeq y$  iff  $Nx = Ny$

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

Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

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

### Issues

Lifting

Alternation

Fake incl

Fixpoints

Conv rule

### Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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

## 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

## Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

- ▶ a relation  $\preceq$  for occurrence:  
if  $x$ ,  $y$  and  $z$  are **different** 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

- ▶ a relation  $\preceq$  for occurrence:  
if  $x$ ,  $y$  and  $z$  are **different** terms,
  - ▶  $y$  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  
Analyse of  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

- ▶ a relation  $\preceq$  for occurrence:  
if  $x$ ,  $y$  and  $z$  are **different** terms,
  - ▶  $y$  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

Solution  
(intuitive)

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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

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

Solution  
(intuitive)

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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

- ▶ a relation  $\preceq$  for occurrence:  
if  $x$ ,  $y$  and  $z$  are **different** terms,

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

$$x \preceq y \quad \text{if } x \simeq y$$

$$x \preceq t \quad \text{if } t \simeq x \oplus y_0 \dots \oplus y_n$$

$$\text{and } x \not\preceq y_i \text{ for all } i, 0 \leq i \leq n$$

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Motivation

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

Solution  
(intuitive)

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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Cryptographic systems need more

Reasoning on such systems involves

- ▶ 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:  $x \oplus x \simeq 0$

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

$$x \preceq y \quad \text{if } x \simeq y$$

$$x \preceq t \quad \text{if } t \simeq x \oplus y_0 \dots \oplus y_n$$

$$\text{and } x \not\preceq y_i \text{ for all } i, 0 \leq i \leq n$$

→ normalization is needed!

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  
Analysis of  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

**Solving strategies**

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

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

## Issues

Lifting

Alternation

Fake incl

Fixpoints

Conv rule

## Conclusion

First attempt: rewrite, rewrite, rewrite...

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

Conclusion

First attempt: rewrite, rewrite, rewrite...

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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# First attempt: rewrite, rewrite, rewrite...

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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find a suitable well ordering on terms

Functional programming approach:

- ▶ Not very difficult – use **general recursion**

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

Conclusion

First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find a 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find a 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find a 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

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

Conclusion

# First attempt: rewrite, rewrite, rewrite...

Replace equations with rewrite rules

Commutativity: find a 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?

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

Conclusion

# 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

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

# (Dependent) type theoretic 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# (Dependent) type theoretic approach

## 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**

### Solution (intuitive)

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

### Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

### Conclusion

# (Dependent) type theoretic approach

## 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**

### Solution (intuitive)

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

### Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

### Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## Step 1

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

### 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

### Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## 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  $t$

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

Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## 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  $t$

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

Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## 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.

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

Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## 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  $\mathcal{T}' \rightarrow \mathcal{T}$ .

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

Conclusion

# (Dependent) type theoretic approach

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

## 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  $\mathcal{T}' \rightarrow \mathcal{T}$ .

Interesting part =  $\mathcal{T} \rightarrow \mathcal{T}'$

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

Conclusion

# Outline

## Motivation

- The case of cryptographic systems
- State of the art
- Back to cryptographic systems
- Solving strategies

## Solution (intuitive)

### Basic idea

- Analyse of  $\mathcal{T}$
- Decomposing  $\mathcal{T}$
- Stratifying and normalizing a term

## Issues

- Lifting
- Alternation
- Forbid fake inclusions
- 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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# 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**  
Analysis of  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Lasagnas reveal the truth



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**  
Analysis of  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lasagnas reveal the truth



- ▶ 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**  
Analysis of  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lasagnas reveal the truth



- ▶ 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Lasagnas reveal the truth



- ▶ 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion





# Outline

## Motivation

- The case of cryptographic systems
- State of the art
- Back to cryptographic systems
- Solving strategies

## Solution (intuitive)

- Basic idea
- Analyse of  $\mathcal{T}$**
- Decomposing  $\mathcal{T}$
- Stratifying and normalizing a term

## Issues

- Lifting
- Alternation
- Forbid fake inclusions
- 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

## Issues

- Lifting
- Alternation
- Fake incl
- Fixpoints
- Conv rule

## Conclusion

# $\mathcal{T}$ as a lasagna

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

## Conclusion

# $\mathcal{T}$ as a lasagna

Inductive  $\mathcal{T}$ : Set :=

| *Zero*:  $\mathcal{T}$

| *PC*: *public\_const*  $\rightarrow \mathcal{T}$       | *SC*: *secret\_const*  $\rightarrow \mathcal{T}$

| *E*:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| *Xor*:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| *Hash*:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$ .

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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# $\mathcal{T}$ as a lasagna

Inductive  $\mathcal{T}$ : Set :=

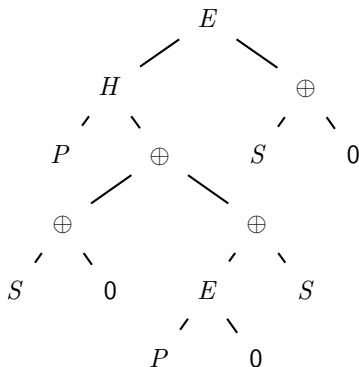
| Zero:  $\mathcal{T}$

| PC:  $public\_const \rightarrow \mathcal{T}$       | SC:  $secret\_const \rightarrow \mathcal{T}$

| E:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| XOR:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| Hash:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$ .



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  
**Analysis of  $\mathcal{T}$**   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# $\mathcal{T}$ as a lasagna

Inductive  $\mathcal{T}$ : Set :=

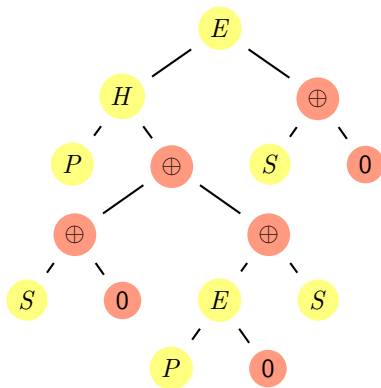
| Zero:  $\mathcal{T}$

| PC:  $public\_const \rightarrow \mathcal{T}$       | SC:  $secret\_const \rightarrow \mathcal{T}$

| E:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| XOR:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$

| Hash:  $\mathcal{T} \rightarrow \mathcal{T} \rightarrow \mathcal{T}$ .



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  
**Analysis of  $\mathcal{T}$**   
Decomposing  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Outline

## Motivation

- The case of cryptographic systems
- State of the art
- Back to cryptographic systems
- Solving strategies

## Solution (intuitive)

- Basic idea
- Analyse of  $\mathcal{T}$
- Decomposing  $\mathcal{T}$**
- Stratifying and normalizing a term

## Issues

- Lifting
- Alternation
- Forbid fake inclusions
- 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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Decomposing $\mathcal{T}$

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Inductive  $\mathcal{T}_x : \text{Set} :=$

|  $X\_Zero : \mathcal{T}_x$

|  $X\_Xor : \mathcal{T}_x \rightarrow \mathcal{T}_x \rightarrow \mathcal{T}_x$

Inductive  $\mathcal{T}_n : \text{Set} :=$

|  $NX\_PC : \text{public\_const} \rightarrow \mathcal{T}_n$

|  $NX\_SC : \text{secret\_const} \rightarrow \mathcal{T}_n$

|  $NX\_E : \mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$

|  $NX\_Hash : \mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$

## 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

## Conclusion

# Decomposing $\mathcal{T}$

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Variable  $A$  : Set.

Inductive  $\mathcal{T}_x$ :Set :=

- |  $X\_Zero$  :  $\mathcal{T}_x$
- |  $X\_Xor$  :  $\mathcal{T}_x \rightarrow \mathcal{T}_x \rightarrow \mathcal{T}_x$
- |  $X\_ns$  :  $A \rightarrow \mathcal{T}_x$

Inductive  $\mathcal{T}_n$ : Set :=

- |  $NX\_PC$  :  $public\_const \rightarrow \mathcal{T}_n$
- |  $NX\_SC$  :  $secret\_const \rightarrow \mathcal{T}_n$
- |  $NX\_E$  :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$
- |  $NX\_Hash$  :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$
- |  $NX\_sum$  :  $A \rightarrow \mathcal{T}_n$

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

Conclusion

# Outline

## Motivation

- The case of cryptographic systems
- State of the art
- Back to cryptographic systems
- Solving strategies

## Solution (intuitive)

- Basic idea
- Analyse of  $\mathcal{T}$
- Decomposing  $\mathcal{T}$
- Stratifying and normalizing a term

## Issues

- Lifting
- Alternation
- Forbid fake inclusions
- 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**

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Stratifying and normalizing 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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
**Stratification**

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Stratifying and normalizing a term

**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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
**Stratification**

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Stratifying and normalizing a term

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

**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

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

Conclusion

# Stratifying and normalizing a term

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

**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

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

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

Conclusion

# Stratifying and normalizing a term

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

**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

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

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

Conclusion

# Stratifying and normalizing a term

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

**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

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

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

Conclusion

# Stratifying and normalizing a term

**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

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

## 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

## Conclusion

# Stratifying and normalizing a term

**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

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

## 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

## Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

**Lifting**

Alternation

Forbid fake inclusions

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

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

$\mathcal{L}_x k \stackrel{\text{def}}{=} \underbrace{\mathcal{I}_x(\mathcal{I}_n(\mathcal{I}_x(\dots(\emptyset))))}_{k \text{ layers}}$  for  $k$  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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

## Issues

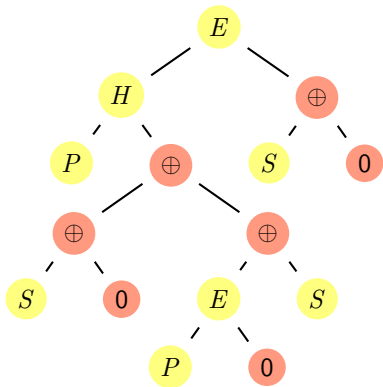
**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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

## Conclusion

# Lifting lasagna

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

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

Take the max

## Motivation

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

## Solution (intuitive)

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

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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:  $\{le\ n\ m\} + \{le\ m\ n\}$

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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

Issues

**Lifting**

Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Lifting lasagna

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

- ▶ 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

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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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  $n - m$

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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Lifting lasagna

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

- ▶ 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  $\mathcal{L}_x (\max\ n\ m)$

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  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Lifting lasagna

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

- ▶ 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:  $\max\ n\ m \stackrel{\text{def}}{=} m + (n - m)$

## Motivation

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

## Solution (intuitive)

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

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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:  $\max\ n\ m \stackrel{\text{def}}{=} m + (n - m)$ 
  - ▶  $lift_x : \mathcal{L}_x k \rightarrow \mathcal{L}_x(k + d)$ ,  $lift_n : \mathcal{L}_n k \rightarrow \mathcal{L}_n(k + d)$

## Motivation

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

## Solution (intuitive)

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

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Lifting lasagna

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

- ▶ 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:  $\max\ n\ m \stackrel{\text{def}}{=} m + (n - m)$ 
  - ▶  $lift_x : \mathcal{L}_x k \rightarrow \mathcal{L}_x (k + d)$ ,  $lift_n : \mathcal{L}_n k \rightarrow \mathcal{L}_n (k + d)$
  - ▶ No need to proof that **max** is the max.

## Motivation

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

## Solution (intuitive)

Basic idea  
Analysis of  $\mathcal{I}$   
Decomposing  $\mathcal{I}$   
Stratification

## Issues

**Lifting**  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

**Alternation**

Forbid fake inclusions

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

## Issues

Lifting  
**Alternation**  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Internalizing alternation

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

## Conclusion

# Internalizing alternation

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Well designed types help us to design programs

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

Conclusion

# Internalizing alternation

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

Issues

Lifting  
**Alternation**  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Internalizing alternation

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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
**Alternation**  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Internalizing alternation

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

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  $alt_{even}: Set :=$

|  $0_e: alt_{even}$

|  $S_{o \rightarrow e}: alt_{odd} \rightarrow alt_{even}$

with  $alt_{odd}: Set :=$

|  $S_{e \rightarrow o}: alt_{even} \rightarrow alt_{odd}$

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

Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

**Forbid fake inclusions**

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

## Issues

Lifting  
Alternation  
**Fake incl**  
Fixpoints  
Conv rule

## Conclusion

# 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

## Issues

Lifting  
Alternation  
**Fake incl**  
Fixpoints  
Conv rule

## Conclusion

# Forbid fake inclusions

Inductive  $\mathcal{T}_x$ : Set :=  
| X\_Zero :  $\mathcal{T}_x$   
| X\_ns :  $A \rightarrow \mathcal{T}_x$   
| X\_Xor :  $\mathcal{T}_x \rightarrow \mathcal{T}_x \rightarrow \mathcal{T}_x$

Inductive  $\mathcal{T}_n$ : Set :=  
| NX\_PC : *public\_const*  $\rightarrow \mathcal{T}_n$   
| NX\_SC : *secret\_const*  $\rightarrow \mathcal{T}_n$   
| NX\_sum :  $A \rightarrow \mathcal{T}_n$   
| NX\_E :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$   
| NX\_Hash :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$

## 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

## Conclusion

# Forbid fake inclusions

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Inductive  $\mathcal{T}_x$ : Set :=  
| X\_Zero :  $\mathcal{T}_x$   
| X\_ns :  $A \rightarrow \mathcal{T}_x$   
| X\_Xor :  $\mathcal{T}_x \rightarrow \mathcal{T}_x \rightarrow \mathcal{T}_x$

Inductive  $\mathcal{T}_n$ : Set :=  
| NX\_PC : *public\_const*  $\rightarrow \mathcal{T}_n$   
| NX\_SC : *secret\_const*  $\rightarrow \mathcal{T}_n$   
| NX\_sum :  $A \rightarrow \mathcal{T}_n$   
| NX\_E :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$   
| NX\_Hash :  $\mathcal{T}_n \rightarrow \mathcal{T}_n \rightarrow \mathcal{T}_n$

$X\_ns (NX\_sum ( X\_ns (NX\_sum (...))))$

## 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

## Conclusion

# Forbid fake inclusions

Inductive  $\mathcal{T}_x: \text{bool} \rightarrow \text{Set} :=$

|  $X\_Zero : \forall b, \mathcal{T}_x b$

|  $X\_ns : \forall b, \text{ls\_true } b \rightarrow A \rightarrow \mathcal{T}_x b$

|  $X\_Xor : \forall b, \mathcal{T}_x \text{ true} \rightarrow \mathcal{T}_x \text{ true} \rightarrow \mathcal{T}_x b$

Inductive  $\mathcal{T}_n: \text{bool} \rightarrow \text{Set} :=$

|  $NX\_PC : \forall b, \text{public\_const} \rightarrow \mathcal{T}_n b$

|  $NX\_SC : \forall b, \text{secret\_const} \rightarrow \mathcal{T}_n b$

|  $NX\_sum : \forall b, \text{ls\_true } b \rightarrow A \rightarrow \mathcal{T}_n b$

|  $NX\_E : \forall b, \mathcal{T}_n \text{ true} \rightarrow \mathcal{T}_n \text{ true} \rightarrow \mathcal{T}_n b$

|  $NX\_Hash : \forall b, \mathcal{T}_n \text{ true} \rightarrow \mathcal{T}_n \text{ true} \rightarrow \mathcal{T}_n b$

$X\_ns (NX\_sum ( X\_ns (NX\_sum (...))))$

## 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

## Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

**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

## Issues

Lifting  
Alternation  
Fake incl  
**Fixpoints**  
Conv rule

## Conclusion

# Mutual induction

- ▶ 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

## Issues

Lifting  
Alternation  
Fake incl  
**Fixpoints**  
Conv rule

## Conclusion

# Mutual induction

- ▶ 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

## Issues

Lifting  
Alternation  
Fake incl  
**Fixpoints**  
Conv rule

## Conclusion

# Mutual induction

- ▶ 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

## Issues

Lifting  
Alternation  
Fake incl  
**Fixpoints**  
Conv rule

## Conclusion

# Mutual induction

- ▶ 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

## Issues

Lifting  
Alternation  
Fake incl  
**Fixpoints**  
Conv rule

## Conclusion

# Mutual induction

- ▶ 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  $\mathcal{L}_x e_1 \rightarrow \mathcal{L}_x (e_1 + e_2)$  with  
|  $0_e \Rightarrow \text{fun } emp \Rightarrow \text{match } emp \text{ with end}$   
|  $S_{o \rightarrow e} o_1 \Rightarrow \text{map}_x (\text{lift\_lasagna\_n } o_1 e_2) \text{ false}$   
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  $\mathcal{L}_n o_1 \rightarrow \mathcal{L}_n (o_1 + e_2)$  with  
|  $S_{e \rightarrow o} e_1 \Rightarrow \text{map}_n (\text{lift\_lasagna\_x } e_1 e_2) \text{ false}$   
end.
```

## 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

## Conclusion

# Outline

## Motivation

The case of cryptographic systems

State of the art

Back to cryptographic systems

Solving strategies

## Solution (intuitive)

Basic idea

Analyse of  $\mathcal{T}$

Decomposing  $\mathcal{T}$

Stratifying and normalizing a term

## Issues

Lifting

Alternation

Forbid fake inclusions

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
**Conv rule**

## Conclusion

# 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  $\mathcal{T}$   
Stratification

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
**Conv rule**

## Conclusion

# Conversion rule

Used everywhere

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**

## Conclusion

# Conversion rule

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

Used everywhere

Definition *bin\_xor*

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

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**

Conclusion

# Conclusion

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

## Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

## Conclusion

# Conclusion

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

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Conclusion

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  $\mathcal{T}$   
Decomposing  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Conclusion

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  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Conclusion

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  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Conclusion

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  $\mathcal{T}$   
Stratification

Issues

Lifting  
Alternation  
Fake incl  
Fixpoints  
Conv rule

Conclusion

# Conclusion

Type theory is flexible

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

Future work

- ▶ Study variants, compare and choose the best

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

Conclusion

# Conclusion

Proving  
termination  
using dependent  
types

J.-F. Monin,  
J. Courant

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

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

Conclusion