

# Models and analysis of security protocols

## 1st Semester 2009-2010

### Passive Intruder

### Lecture 7

**Pascal Lafourcade**

*Université Joseph Fourier, Verimag*

Master: October 12th 2009

## Last Time (I)

### Symmetric encryption and Protocols

- ▶ Logical Attacks
- ▶ Needham Schroeder
- ▶ DH
- ▶ Dolev Yao Intruder
- ▶ Undecidability in general

Remarks, questions, comments ?

## Outline of Today:

Notion of Locality

Passive Intruder: Intruder Deduction Problem

Unification Notions

- Terms and Messages

- Unification

Conclusion

# Outline

Notion of Locality

Passive Intruder: Intruder Deduction Problem

Unification Notions

Terms and Messages

Unification

Conclusion

## Syntactic Subterms

### Equivalent definition for Dolev Yao model

$S(t)$  is the smallest set such that:

- ▶  $t \in S(t)$
- ▶  $\langle u, v \rangle \in S(t) \Rightarrow u, v \in S(t)$
- ▶  $\{u\}_v \in S(t) \Rightarrow u, v \in S(t)$

Exercise:

- ▶ Let  $t = \{\langle a, \{b\}_{k_2} \rangle\}_{k_1}$

## Syntactic Subterms

Equivalent definition for Dolev Yao model

$S(t)$  is the smallest set such that:

- ▶  $t \in S(t)$
- ▶  $\langle u, v \rangle \in S(t) \Rightarrow u, v \in S(t)$
- ▶  $\{u\}_v \in S(t) \Rightarrow u, v \in S(t)$

Exercise:

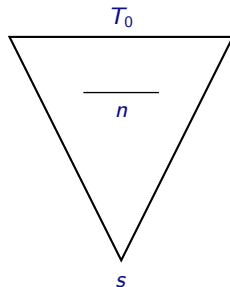
- ▶ Let  $t = \{\langle a, \{b\}_{k_2} \rangle\}_{k_1}$

$$S(t) = \{t, a, b, k_1, k_2, \{b\}_{k_2}, \langle a, \{b\}_{k_2} \rangle\}$$

## Definition of S-Locality

- ▶ A proof  $P$  of  $T_0 \vdash s$  is S-local :

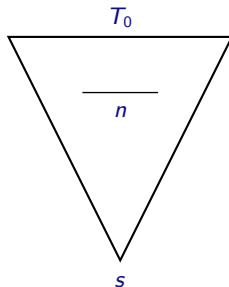
$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



## Definition of S-Locality

- ▶ A proof  $P$  of  $T_0 \vdash s$  is S-local :

$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



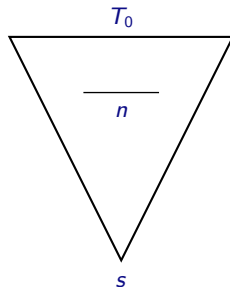
S-Local Proof:

A proof  $P$  of  $T \vdash w$  is **S-local** if all nodes are in  $S(T \cup \{w\})$ .

## Definition of S-Locality

- ▶ A proof  $P$  of  $T_0 \vdash s$  is S-local :

$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



### S-Local Proof:

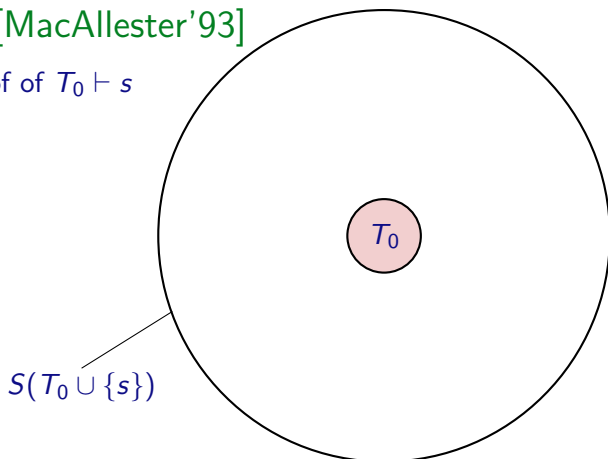
A proof  $P$  of  $T \vdash w$  is **S-local** if all nodes are in  $S(T \cup \{w\})$ .

### S-Locality :

A proof system is **S-local** if whenever there is a proof of  $T \vdash w$  then there is also a S-local proof of  $T \vdash w$ .

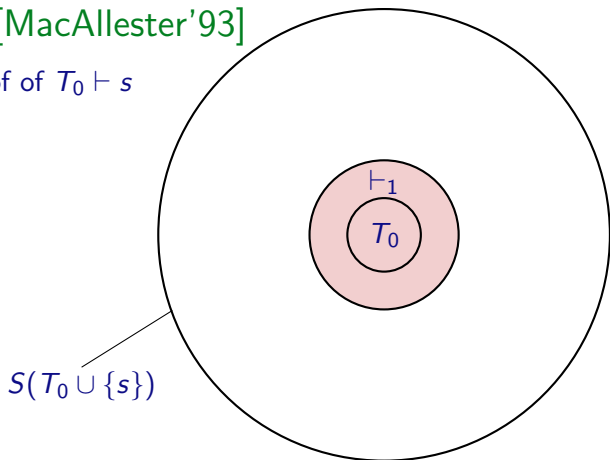
## Locality Idea [MacAllester'93]

P a S-local proof of  $T_0 \vdash s$



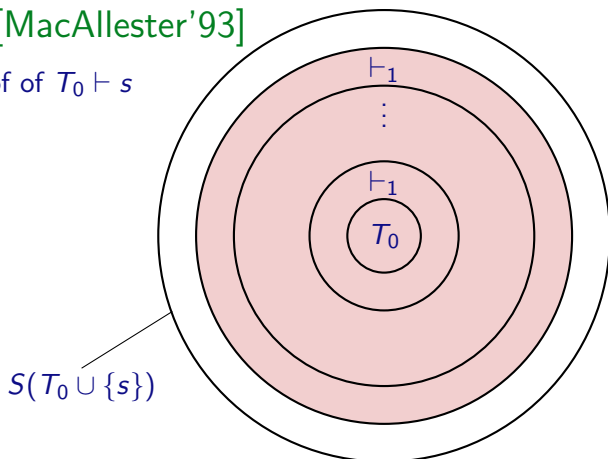
## Locality Idea [MacAllester'93]

P a S-local proof of  $T_0 \vdash s$



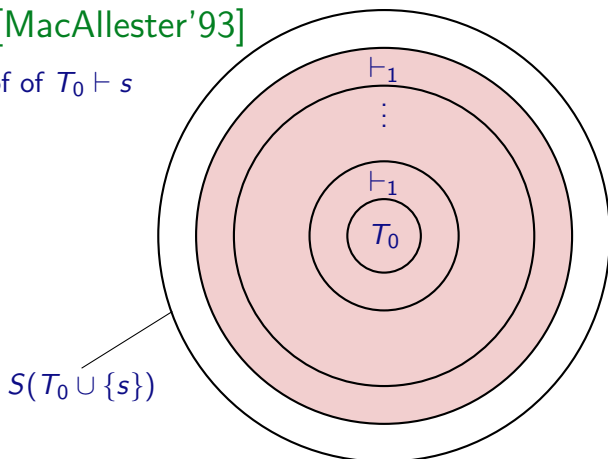
## Locality Idea [MacAllester'93]

P a  $S$ -local proof of  $T_0 \vdash s$



## Locality Idea [MacAllester'93]

P a S-local proof of  $T_0 \vdash s$



Intruder Deduction Problem :  $T_0 \vdash^? s$

- ▶ S-locality
- ▶ One-step deductibility

## Example: a local proof of $T_0 \vdash s$

### Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$  and  $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k} \\
 (D) \frac{}{T_0 \vdash b}
 \end{array}
 \end{array}$$

## Example: a local proof of $T_0 \vdash s$

### Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$  and  $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k} \\
 (D) \frac{}{T_0 \vdash b}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c}
 \end{array}
 \end{array}$$

$$S(T_0 \cup \{s\}) = T_0 \cup \{a, b, c, \{c\}_k\}$$

# Locality Theorem

## Theorem of Locality [McAllester 93]

If a proof system  $P$  is SyntacticSubterm-local then there is a  $P$ -time procedure to decide the deductibility in  $P$ .

# Locality Theorem

## Theorem of Locality [McAllester 93]

If a proof system  $P$  is SyntacticSubterm-local then there is a  $P$ -time procedure to decide the deductibility in  $P$ .

### Restrictions:

- ▶ Deduction system must be finite
- ▶ Use just syntactic subterms

## Adapted McAllester Results

### McAllester's Algorithm

Input :  $T_0, w$

$T \leftarrow T_0;$

while  $(\exists s \in S(T_0, w)$  such that  $T \vdash^{\leq 1} s$  and  $s \notin T)$

$T \leftarrow T \cup \{s\};$

Output :  $w \in T$

### Theorem

Let be  $P$  a proof system, if:

- ▶ the size of  $S(T)$  is polynomial in the size of  $T$ ,
- ▶  $P$  is S-local,
- ▶ one-step deducibility is P-time decidable,

then provability in the proof system  $P$  is P-time decidable.

# Outline

Notion of Locality

Passive Intruder: Intruder Deduction Problem

Unification Notions

Terms and Messages

Unification

Conclusion

# Locality Theorem

## Theorem of Locality [McAllester 93]

If a proof system  $P$  is SyntacticSubterm-local then there is a  $P$ -time procedure to decide the deductibility in  $P$ .

# Locality Theorem

## Theorem of Locality [McAllester 93]

If a proof system  $P$  is SyntacticSubterm-local then there is a  $P$ -time procedure to decide the deductibility in  $P$ .

## Result:

Dolev Yao deduction system is S-local.

## Example of necessity of $S(T \cup \{s\})$

### Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$  and  $s = \langle b, k \rangle$

## Example of necessity of $S(T \cup \{s\})$

### Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$  and  $s = \langle b, k \rangle$

$$\begin{array}{c}
 \frac{(A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} (D) \frac{(UR) \frac{(A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} (A) \frac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash c}}{T_0 \vdash b} \\
 (P) \frac{}{T_0 \vdash \langle b, k \rangle} \quad (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}$$

## Example of necessity of $S(T \cup \{s\})$

### Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$  and  $s = \langle b, k \rangle$

$$\begin{array}{c}
 \text{(A)} \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \quad \text{(A)} \frac{k \in T_0}{T_0 \vdash k} \\
 \text{(UR)} \frac{\quad}{T_0 \vdash \{c\}_k} \\
 \text{(D)} \frac{\text{(A)} \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \quad \text{(D)} \frac{\quad}{T_0 \vdash c}}{T_0 \vdash b} \quad \text{(A)} \frac{k \in T_0}{T_0 \vdash k} \\
 \text{(P)} \frac{\quad}{T_0 \vdash \langle b, k \rangle}
 \end{array}$$

$S(T_0) = T_0 \cup \{s, a, b, c, k, \{b\}_k, \{c\}_k\}$  but  $\langle b, k \rangle \notin S(T_0)$

It is Not enough

Notice that  $\langle b, k \rangle \in S(T_0 \cup \{s\})$

## Example non minimal proof si not $S$ -local

GOAL: Find a good  $S$ .

### Example

$T_0 = \{k, \{c\}_k\}$  and  $s = c$

## Example non minimal proof si not $S$ -local

GOAL: Find a good  $S$ .

### Example

$T_0 = \{k, \{c\}_k\}$  and  $s = c$

$$\begin{array}{c}
 \frac{(A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k} (A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}}{(P) \frac{}{T_0 \vdash \langle \{c\}_k, \{c\}_k \rangle}} \\
 \frac{(UL) \frac{}{T_0 \vdash \{c\}_k}}{(D) \frac{}{c}} \quad (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}$$

## Example non minimal proof si not $S$ -local

GOAL: Find a good  $S$ .

### Example

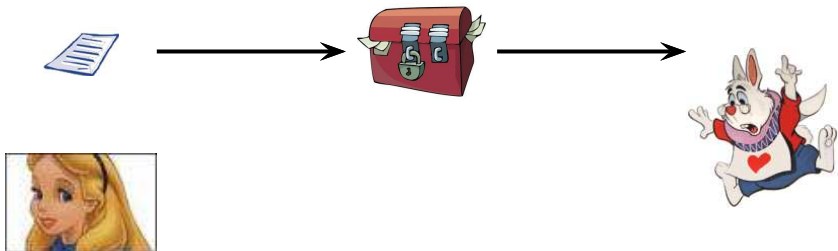
$T_0 = \{k, \{c\}_k\}$  and  $s = c$

$$\begin{array}{c}
 \frac{(A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k} (A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}}{(P) \frac{}{T_0 \vdash \langle \{c\}_k, \{c\}_k \rangle}} \\
 \frac{(UL) \frac{}{T_0 \vdash \{c\}_k}}{(D) \frac{}{c}} \quad \frac{(A) \frac{k \in T_0}{T_0 \vdash k}}{}
 \end{array}$$

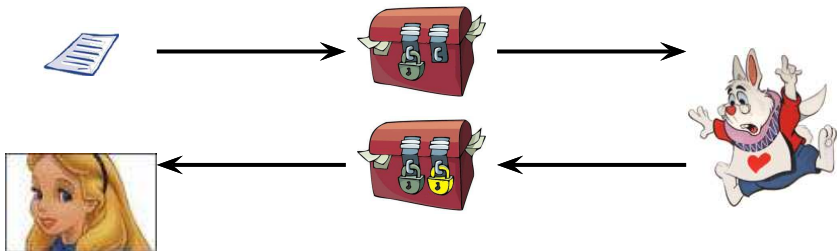
$S(T_0) = T_0 \cup \{c\}$  but  $\langle \{c\}_k, \{c\}_k \rangle$

It is Not in  $S(T_0 \cup \{s\})$

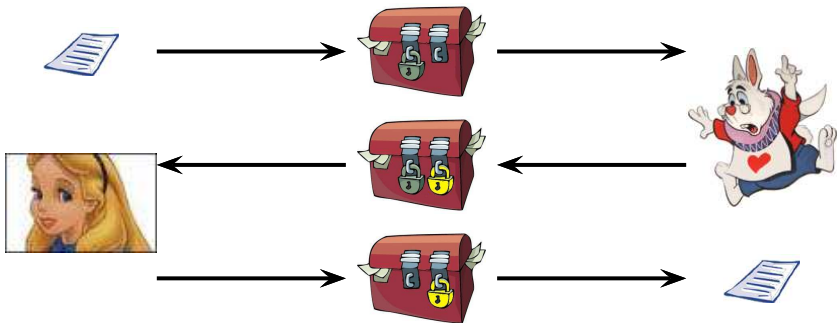
Example :



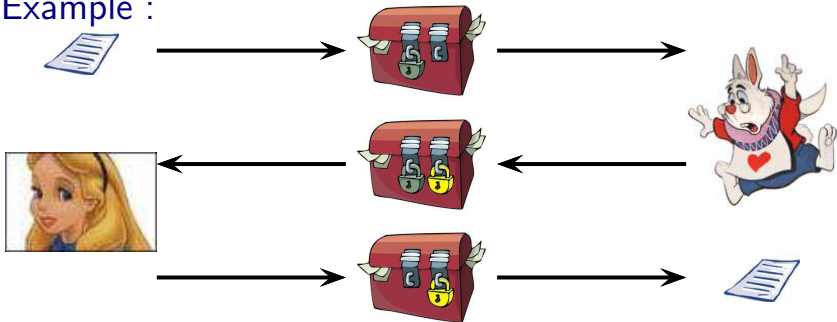
## Example :



## Example :



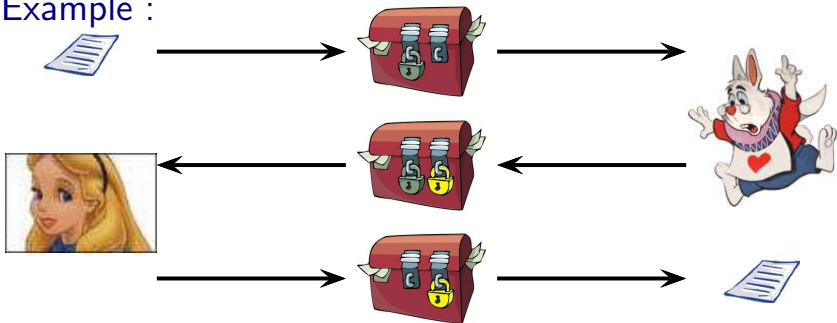
Example :



### Shamir 3-Pass Protocol

$$1 \quad A \rightarrow B : \{m\}_{K_A}$$

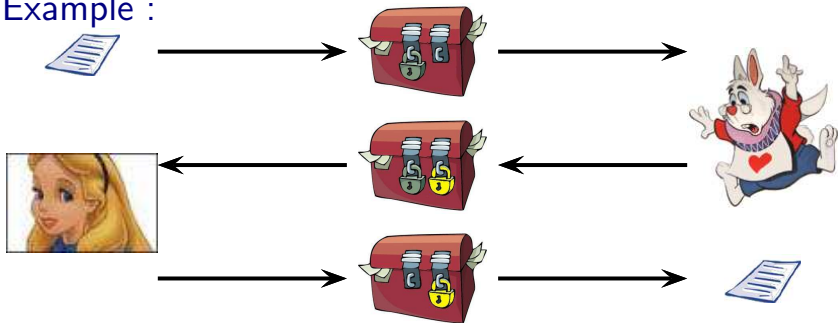
Example :



### Shamir 3-Pass Protocol

- 1  $A \rightarrow B : \{m\}_{K_A}$
- 2  $B \rightarrow A : \{\{\{m\}_{K_A}\}_{K_B}\}$

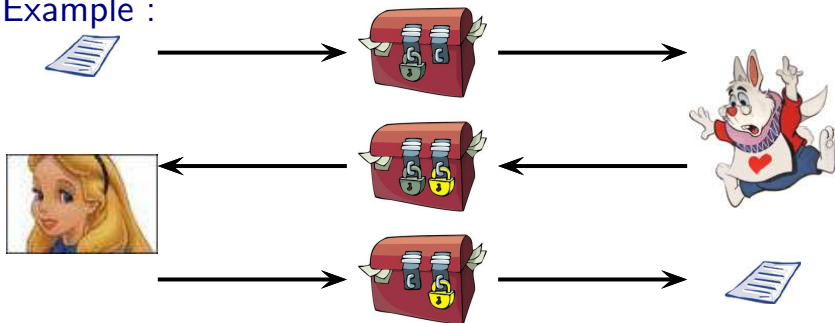
Example :



### Shamir 3-Pass Protocol

- 1  $A \rightarrow B : \{m\}_{K_A}$
  - 2  $B \rightarrow A : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$
- Commutative  
Encryption

Example :



### Shamir 3-Pass Protocol

1  $A \rightarrow B : \{m\}_{K_A}$

2  $B \rightarrow A : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$

3  $A \rightarrow B : \{m\}_{K_B}$

Commutative  
Encryption

# Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

XOR Properties (ACUN)

▶  $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

**A**ssociativity

▶  $x \oplus y = y \oplus x$

**C**ommutativity

▶  $x \oplus 0 = x$

**U**nity

▶  $x \oplus x = 0$

**N**ilpotency

# Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

XOR Properties (ACUN)

▶  $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

**A**ssociativity

▶  $x \oplus y = y \oplus x$

**C**ommutativity

▶  $x \oplus 0 = x$

**U**nity

▶  $x \oplus x = 0$

**N**ilpotency

Vernam encryption is a **commutative encryption** :

$$\{\{m\}_{K_A}\}_{K_I} = (m \oplus K_A) \oplus K_I = (m \oplus K_I) \oplus K_A = \{\{m\}_{K_I}\}_{K_A}$$

# Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

Shamir 3-Pass Protocol



- 1  $A \rightarrow B : m \oplus K_A$
- 2  $B \rightarrow A : (m \oplus K_A) \oplus K_B$
- 3  $A \rightarrow B : m \oplus K_B$



Passive attacker :

$$m \oplus K_A \quad m \oplus K_B \oplus K_A \quad m \oplus K_B$$



# Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

Shamir 3-Pass Protocol



- 1  $A \rightarrow B : m \oplus K_A$
- 2  $B \rightarrow A : (m \oplus K_A) \oplus K_B$
- 3  $A \rightarrow B : m \oplus K_B$



Passive attacker :

$$m \oplus K_A \oplus m \oplus K_B \oplus K_A \oplus m \oplus K_B = m$$



# Outline

Notion of Locality

Passive Intruder: Intruder Deduction Problem

Unification Notions

- Terms and Messages

- Unification

Conclusion

# Arity

## Definition

- ▶  $\mathcal{F}$  is a finite set
  - ▶ *Arity* is a mapping from  $\mathcal{F}$  into  $\mathbb{N}$
  - ▶  $(\mathcal{F}, \text{Arity})$  is a **ranked alphabet** or **signature** denoted  $\Sigma$
- 
- ▶ The **arity** of a symbol  $f \in \mathcal{F}$  is  $\text{Arity}(f)$
  - ▶ The set of symbols of arity  $p$  is denoted by  $\mathcal{F}_p$ .
  - ▶ Elements of arity 0, 1,  $\dots$   $p$  are respectively called constants, unary,  $\dots$   $p$ -ary symbols.

# Example

## Example

Let  $\mathcal{F} = \{\mathbf{enc}, \mathbf{pair}, \mathbf{k}_1, \mathbf{k}_2, \mathbf{0}, \mathbf{1}\}$

$Ariety(\mathbf{enc}) = Ariety(\mathbf{pair}) = 2$

$Ariety(\mathbf{k}_1) = Ariety(\mathbf{k}_2) = Ariety(\mathbf{0}) = Ariety(\mathbf{1}) = 0$

We also denote  $\mathcal{F} = \{\mathbf{enc}/2, \mathbf{pair}/2, \mathbf{k}_1/0, \mathbf{k}_2/0, \mathbf{0}/0, \mathbf{1}/0\}$

# Terms

Let  $\mathcal{X}$  be a set of symbols called **variables**.

## Definition

The set  $\mathcal{T}(\mathcal{F}, \mathcal{X})$  of **terms** over the ranked alphabet  $\mathcal{F}$  and the set of variables  $\mathcal{X}$  is the smallest set defined by:

- $\mathcal{F}_0 \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$
- if  $p \geq 1$ ,  $f \in \mathcal{F}_p$  and  $t_1, \dots, t_p \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ , then  $f(t_1, \dots, t_p) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ .

- ▶ If  $\mathcal{X} = \emptyset$  then  $\mathcal{T}(\mathcal{F}, \mathcal{X})$  is also written  $\mathcal{T}(\mathcal{F})$ . Terms in  $\mathcal{T}(\mathcal{F})$  are called **ground terms**.
- ▶ A term in  $\mathcal{T}(\mathcal{F}, \mathcal{X})$  is **linear** if each variable occurs at most once in  $t$ .

# Example

## Example

Let  $\mathcal{F} = \{\mathbf{enc}/2, \mathbf{pair}/2, \mathbf{k}_1/0, \mathbf{k}_2/0, \mathbf{0}/0, \mathbf{1}/0\}$  and  $\mathcal{X} = \{x, y, z\}$   
 $\mathbf{pair}(x, \mathbf{1})$ ,  $\mathbf{enc}(\mathbf{pair}(y, z), \mathbf{k}_1)$  and  $\mathbf{enc}(\mathbf{0}, \mathbf{k}_1)$  are terms in  $\mathcal{T}(\mathcal{F}, \mathcal{X})$   
 $\mathbf{pair}(\mathbf{0}, \mathbf{1})$ ,  $\mathbf{enc}(\mathbf{0}, \mathbf{k}_1)$  are terms in  $\mathcal{T}(\mathcal{F})$ , i.e., ground terms

We also denote  $\{-\}_-$  for  $\mathbf{enc}(-, -)$  and  $\langle -, - \rangle$  for  $\mathbf{pair}(-, -)$ .

# Substitution

## Definition

- ▶ A **substitution** (respectively a **ground substitution**)  $\sigma$  is a mapping from  $\mathcal{X}$  into  $\mathcal{T}(\mathcal{F}, \mathcal{X})$  (respectively into  $\mathcal{T}(\mathcal{F})$ ) where there are only finitely many variables not mapped to themselves.
- ▶ Substitutions can be extended to  $\mathcal{T}(\mathcal{F}, \mathcal{X})$  in such a way that  $\forall f \in \mathcal{F}_n, \forall t_1, \dots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ :

$$\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n)).$$

The **domain** of a substitution  $\sigma$  is the subset of variables  $x \in \mathcal{X}$  such that  $\sigma(x) \neq x$ .

## Example:

Let  $\sigma = \{x \leftarrow N_A, y \leftarrow \{\langle N_A, N_B \rangle\}_{k_B}\}$  and  $t = \langle x, \langle y, \langle x, x \rangle \rangle \rangle$ .

Then,

$$\sigma(t) = \langle N_A, \{\langle N_A, N_B \rangle\}_{k_B}, \langle N_A, N_A \rangle \rangle$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X)$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

$$s = p(a, X) \quad t = p(Y, b)$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

$$s = p(a, X) \quad t = p(Y, b) \quad \sigma = \{X \leftarrow b; Y \leftarrow a\}$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

$$s = p(a, X) \quad t = p(Y, b) \quad \sigma = \{X \leftarrow b; Y \leftarrow a\}$$

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

$$s = p(a, X) \quad t = p(Y, b) \quad \sigma = \{X \leftarrow b; Y \leftarrow a\}$$

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

$$\sigma = \{X \leftarrow a; Y \leftarrow g(Z)\}$$

# Unification

## Definition

Two  $t$  and  $u$  are unifiable if there exists a substitution  $\sigma$  such that  $\sigma s = \sigma t$

Examples:

$$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$$

$$s = a \quad t = p(X) \quad \text{No unifier}$$

$$s = p(a, X) \quad t = p(Y, b) \quad \sigma = \{X \leftarrow b; Y \leftarrow a\}$$

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

$$\sigma = \{X \leftarrow a; Y \leftarrow g(Z)\} \text{ or } \sigma = \{X \leftarrow a; Y \leftarrow g(b); Z \leftarrow b\}$$

# Most General Unifier

## Definition

The most general unification between two terms  $s$  and  $t$ , denoted by  $mgu(s, t)$  if:  $\forall \sigma$  such that  $s\sigma = t\sigma, \exists \theta$  such that  $\sigma = mgu(s, t)\theta$

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

$$\sigma_1 = \{X \leftarrow a; Y \leftarrow g(Z)\} \quad \sigma_2 = \{X \leftarrow a; Y \leftarrow g(b); Z \leftarrow b\}$$

# Goal

Design an algorithm that for a given unification problem  $s =? t$

- ▶ returns an mgu of  $s$  and  $t$  if they are unifiable.
- ▶ reports failure otherwise.

# Naive Algorithm

Write down two terms and set markers at the beginning of the terms. Then:

1. Move the markers simultaneously, one symbol at a time, until both move off the end of the term (success), or until they point to two different symbols;
2. If the two symbols are both non-variables, then fail; otherwise, one is a variable (call it  $x$ ) and the other one is the first symbol of a subterm (call it  $t$ ):
  - ▶ If  $x$  occurs in  $t$ , then fail;
  - ▶ Otherwise, replace  $x$  everywhere by  $t$  (including in the solution), write down " $x \leftarrow t$ " as a part of the solution, and return to 1.

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

$\sigma = \{x \leftarrow g(y)\}$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(g(y), g(a), g(z))$

$f(g(y), g(y), g(g(g(y))))$

$\sigma = \{x \leftarrow g(y)\}$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(g(y), g(a), g(z))$

$f(g(y), g(y), g(g(y))))$

$\sigma = \{x \leftarrow g(y)\}$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$$f(g(a), g(a), g(z))$$

$$f(g(a), g(a), g(g(g(a))))$$

$$\sigma = \{x \leftarrow g(a), y \leftarrow a\}$$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(g(a), g(a), g(z))$

$f(g(a), g(a), g(g(g(a))))$

$\sigma = \{x \leftarrow g(a), y \leftarrow a\}$

Example:  $f(x, g(a), g(z)) \stackrel{?}{=} f(g(y), g(y), g(g(x)))$

$f(g(a), g(a), g(g(g(a))))$

$f(g(a), g(a), g(g(g(a))))$

$\sigma = \{x \leftarrow g(a), y \leftarrow a, z \leftarrow g(g(a))\}$

# Questions

1. Correctness:
  - ▶ Does the algorithm always terminate?
  - ▶ Does it always produce an mgu for two unifiable terms, and fail for non-unifiable terms?
  - ▶ Do these answers depend on the order of operations?
2. Complexity:
  - ▶ How much space does this take, and how much time?
3. Extension with equational theory, e.g.,  $ab = ba$ .

# Syntactic Unification is Unitary

## Theorem (Robinson)

*Without equational theory there exists a unique mgu for syntactic unification (modulo renaming). Unification is called unitary.*

Herbrand, Martelli, Montanari, Plotkin, Robinson, Huet, Knuth, Bendix, Siekman, Baader.

# Outline

Notion of Locality

Passive Intruder: Intruder Deduction Problem

Unification Notions

- Terms and Messages

- Unification

Conclusion

# Summary

## Today

- ▶ Locality
- ▶ Passive Intruder
- ▶ Unification

# Next Time

## Playing with Tools

- ▶ Active Intruder
- ▶ Bounded Number of Sessions
- ▶ NP-Hardness
- ▶ Tools

**Thank you for your attention.**

**Questions ?**