EQUATIONS ARE in several places!

From Old fashioned Mathematics to Modern Computing

Daniele Nantes – UnB

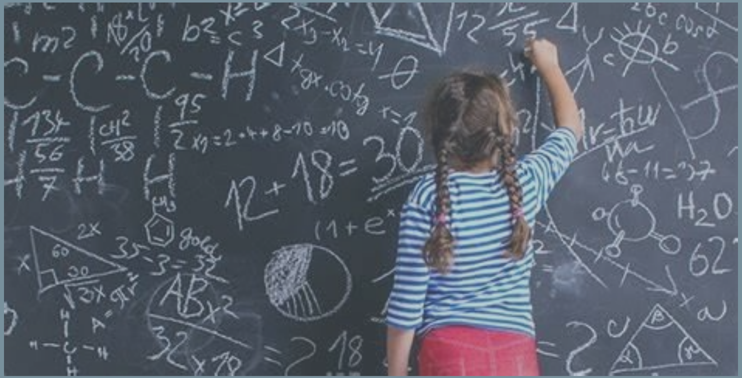
Logic Mentoring Workshop – CSL 2022

FIND THE VALUE OF X

$X - 25 = -10 - 2x \rightarrow X + 2x = -10 + 25$

$\rightarrow 3x = +15$

$\rightarrow x = 5$



Find the value of x

Quadratic Formula

Before substituting values for a, b, and c, rearrange your equation into the form $ax^2+bx+c=0$.

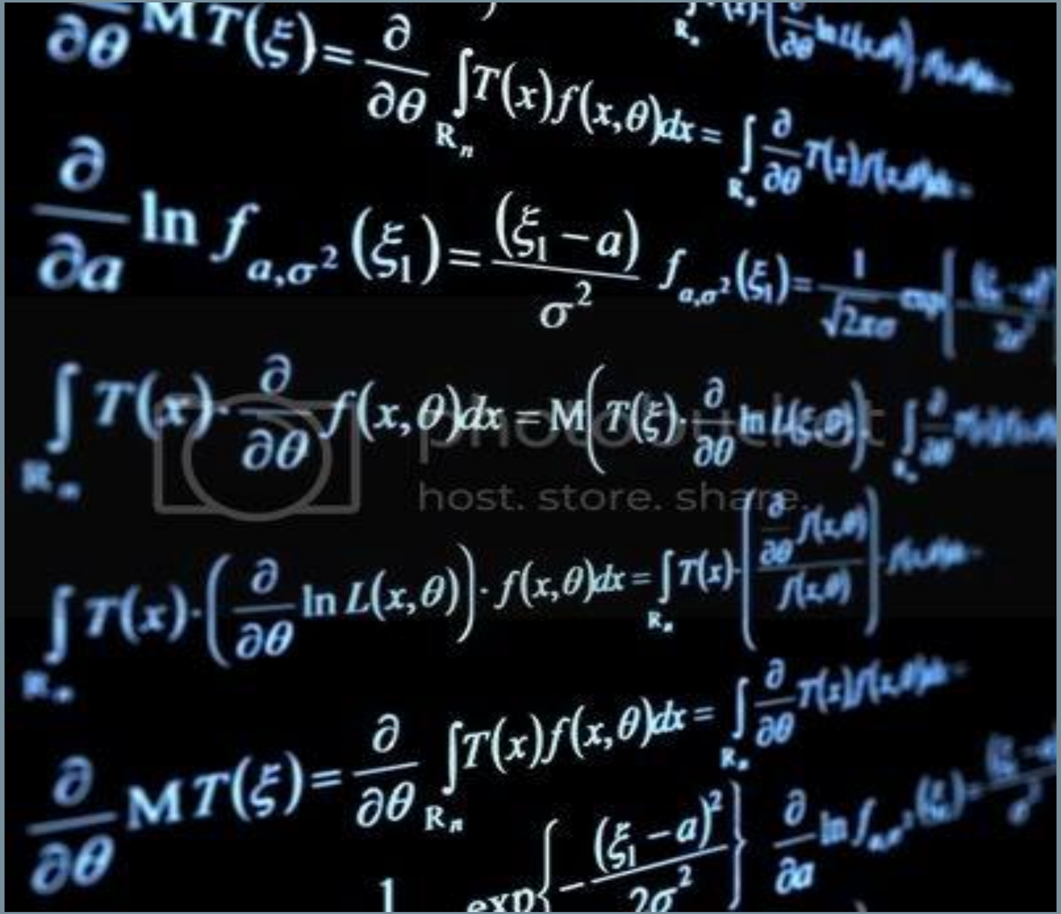
$$\boxed{a}x^2 + \boxed{b}x + \boxed{c} = 0$$

$$x = \frac{-\boxed{b} \pm \sqrt{\boxed{b}^2 - 4\boxed{a}\boxed{c}}}{2\boxed{a}}$$

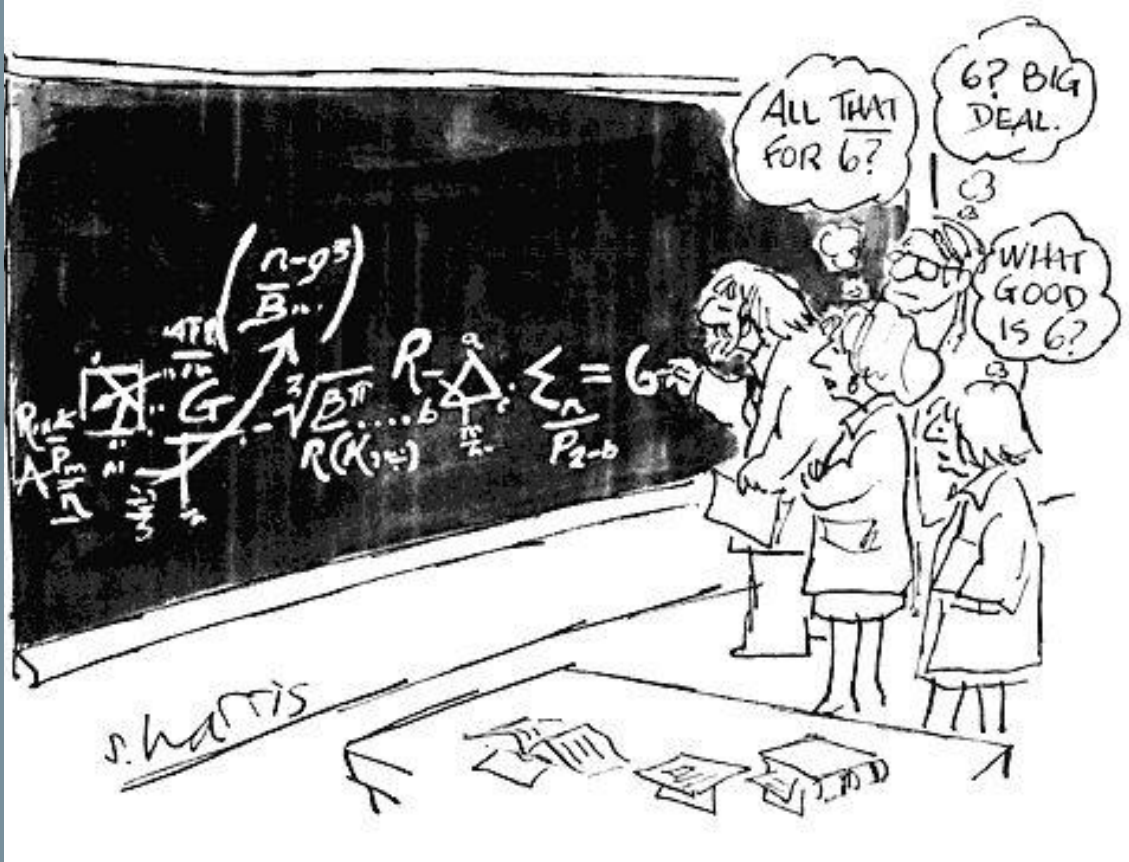
When we talk about equations we think like this

It can be like this..

- Calculus
- Differential Geometry
- Theoretical Physics
- N-dimensional Analysis
- etc..



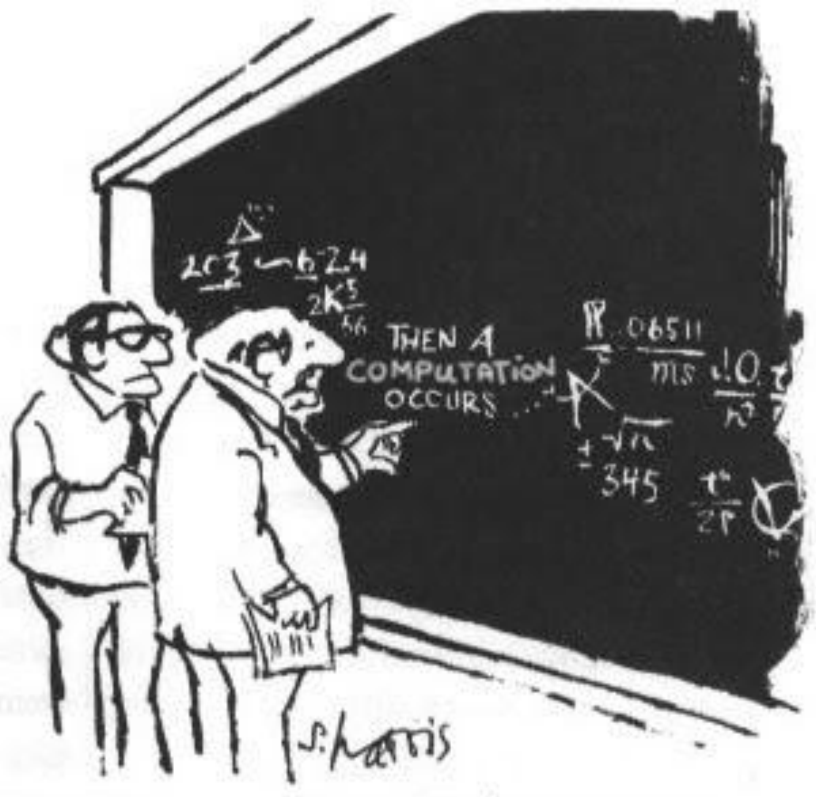
Or like this ...



(... If you get lucky)

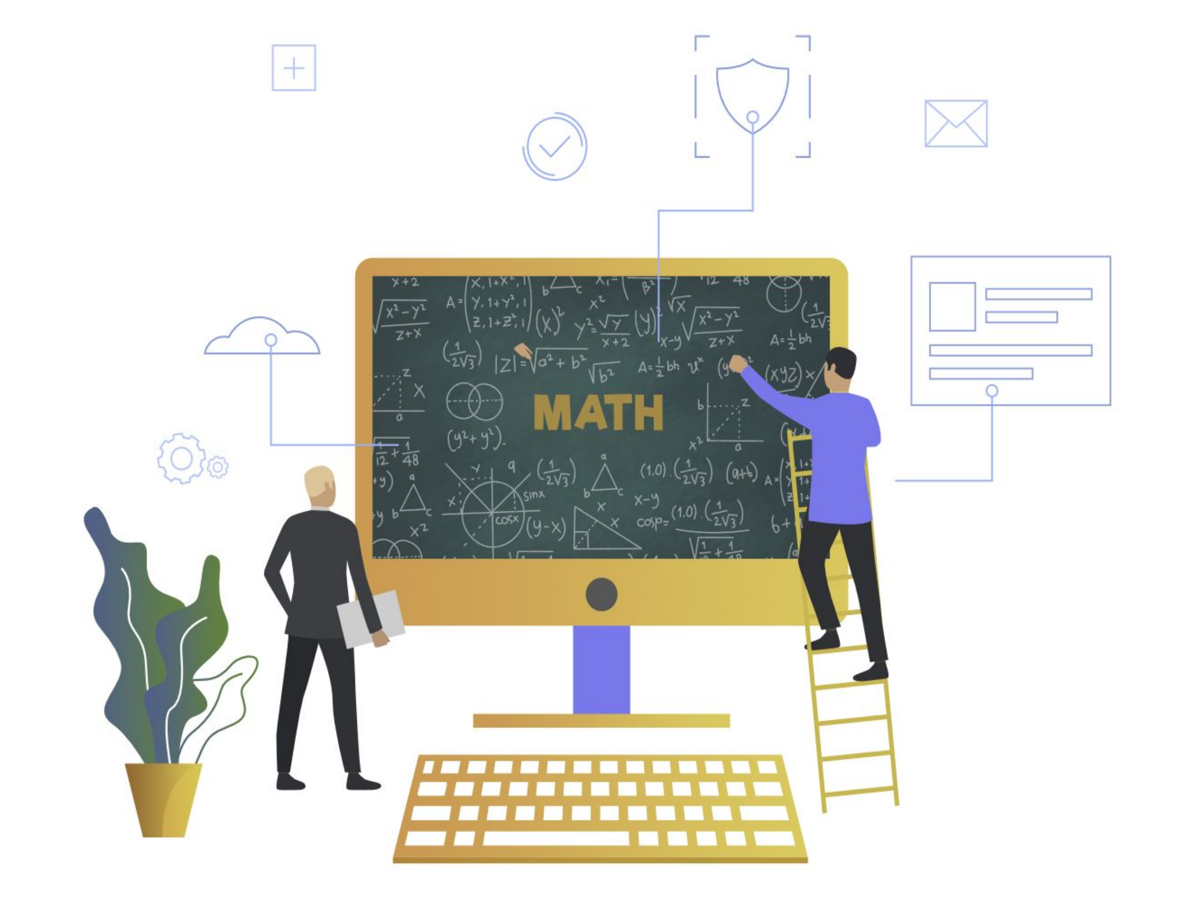
Life became more interesting

(for mathematicians at least)



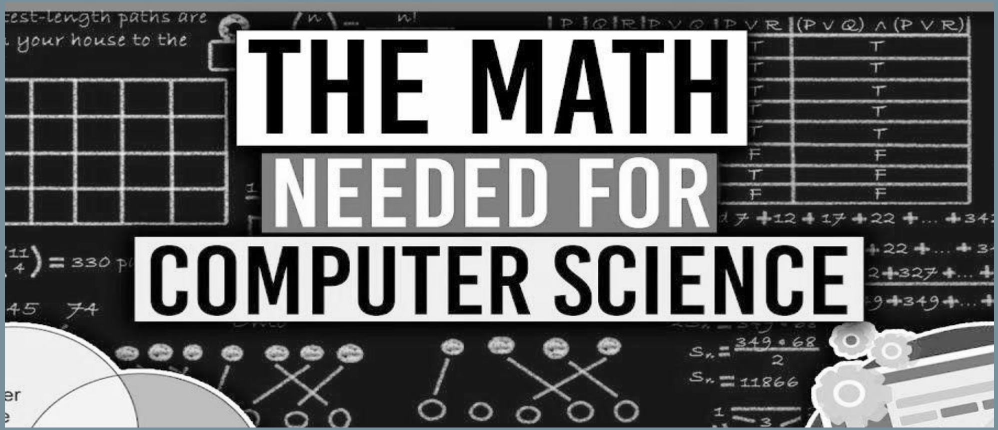
" good call using a computer here in step two."

Computers Vs. Mathematics



.. computers for helping with some math..

I WAS INTERESTED ON THE OTHER WAY AROUND..



UNIFICATION

Problem: Given two terms s and t over some signature Σ.

Find: a substitution σ such that σs=σt.

Notation: s=? t

Problem: Given two terms s and t over some signature Σ. Does s=?t?

Find: a substitution σ such that σs=σt.

$$\begin{tikzcd} swxyz \arrow[d, "\sigma"'] & =? & swex'y' \arrow[d, "\sigma"] \\ sweet & & sweet \end{tikzcd}$$

Example: Σ={a,b,c,...}

x,y,z,x',y':variables

swxyz =? swex'y'

- σ={x→e,y→e,z→t, x'→ e, y'→ t}
- σ'={x→e,y→a,z→r, x'→ a, y'→ r}
- σ''={x→e,y→x',z→y'}

Unification modulo Equational theories

Example: Σ={a,b,c,...}

x,y,z,x',y':variables

Idempotence: xxy=y (+ associativity)

swxyz =? swex'y'

- σ={x→o,y→r,z→d, x'→ eor, y'→ d}
- σ'={x'→ex, y'→yz}

swxyz =? swex'y'

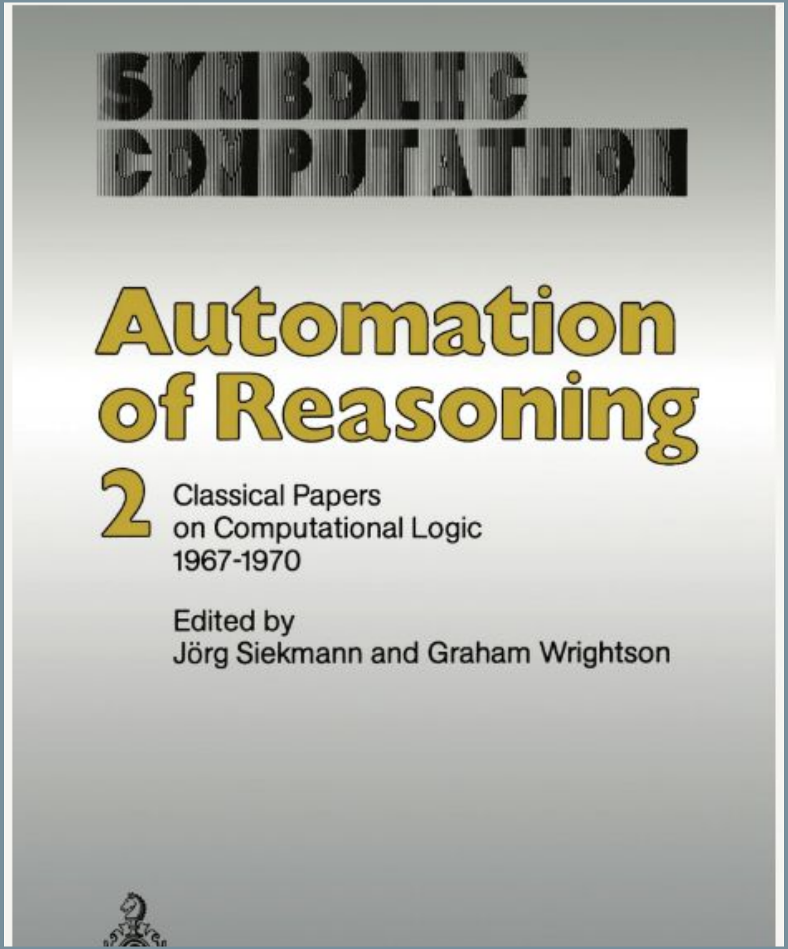
σ σ

sword sweeord=sword

Not Really Modern..

Works on resolution and automatic theorem proving dating back from the 60's

- N.G. De Bruijn
- J.R. Slagle
- D. Prawitz
- D.E. Knuth…



Unification for Theorem Proving

Journal of Automated Reasoning 1 (1985) 327–332. 0168–7433/85.15.

Problem Corner:

The Lion and the Unicorn

H. J. OHLBACH and M. SCHMIDT-SCHAUSS
University of Kaiserslautern, West Germany

(Received: 5 January 1985)

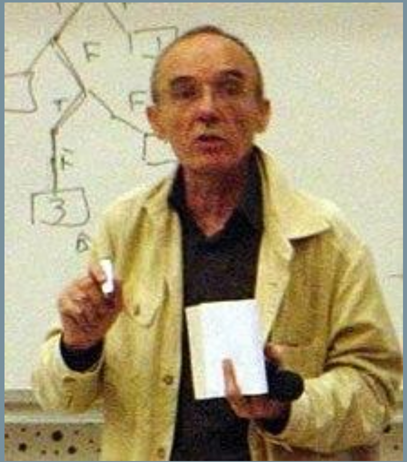
Key words. Example for automated theorem proving, many-sorted logic, refinements of resolution.

Raymond Smullyan's logic puzzles, published in *What is the name of this book?* [6] seem to be a goldmine for theorem proving examples. During a course on automated theorem proving in the last semester, our students had to translate these puzzles into first order predicate logic and to solve them with our theorem prover (Markgraf Karl Refutation Procedure [2]). One of these problems reads as follows:

When Alice entered the forest of forgetfulness, she did not forget everything, only certain things. She often forgot her name, and the most likely thing for her to forget was the day of the week. Now, the lion and the unicorn were frequent visitors to this forest. These two are strange creatures. The lion lies on Mondays, Tuesdays and Wednesdays and tells the truth on the other days of the week. The unicorn, on the other hand, lies on Thursdays, Fridays and Saturdays, but tells the truth on the other days of the week.

One day Alice met the lion and the unicorn resting under a tree. They made the

Unification for Equational Reasoning



J. Symbolic Computation (1989) **8**, 449–477

Unification in Boolean Rings and Abelian Groups

ALEXANDRE BOUDET JEAN-PIERRE JOUANNAUD
MANFRED SCHMIDT-SCHAUSS†

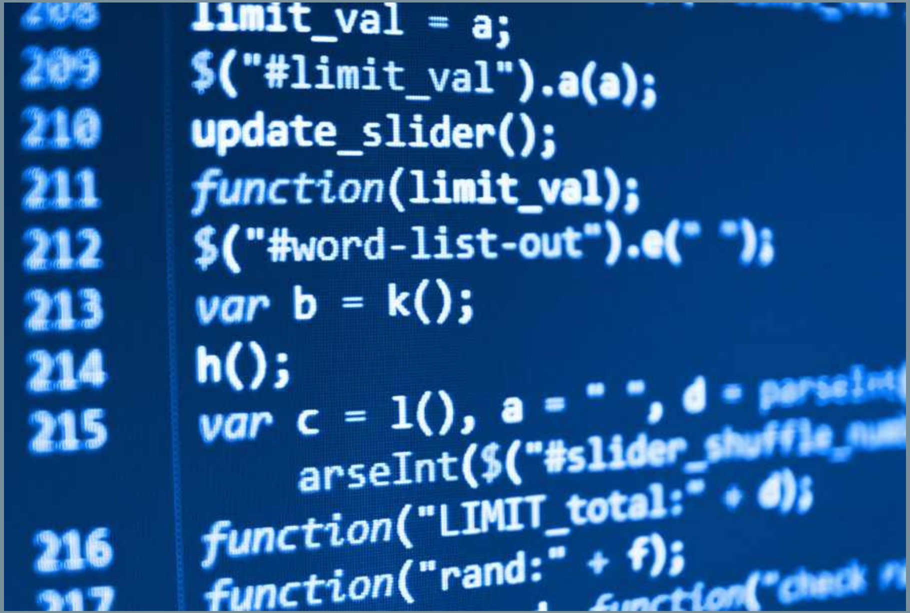
LRI, Université Paris-Sud, Bât 490
91405 ORSAY Cedex, France
†Fachbereich Informatik, Postfach 3049, Universität Kaiserslautern
6750 Kaiserslautern, West Germany

A complete unification algorithm is presented for the combination of two theories E in T(F,X) and E' in T(F',X) where F and F' denote two disjoint sets of function symbols. E and E' are arbitrary equational theories for which are given, for E: a complete unification algorithm for terms in $T(F \cup C, X)$, where C is a set of free constants and a complete constant elimination algorithm for eliminating a constant c from a term s; for E': a complete unification algorithm. E' is supposed to be cycle free, i.e., equations $x = t$ where x is a variable occurring in t have no E'-solution. The method adapts to unification of infinite trees. It is applied to two well-known open problems, when E is the theory of Boolean Rings or the theory of Abelian Groups, and E' is the free theory. Our interest to Boolean Rings originates in VLSI verification.

Equations between logical formulas. .

	propositional logic	calculus of classes	Boolean algebra
1.	$\varphi \vee \varphi \sim \varphi$	$X \cup X \approx X$	$x \vee x \approx x$
2.	$\varphi \wedge \varphi \sim \varphi$	$X \cap X \approx X$	$x \wedge x \approx x$
3.	$\varphi \vee \psi \sim \psi \vee \varphi$	$X \cup Y \approx Y \cup X$	$x \vee y \approx y \vee x$
4.	$\varphi \wedge \psi \sim \psi \wedge \varphi$	$X \cap Y \approx Y \cap X$	$x \wedge y \approx y \wedge x$
5.	$\varphi \vee (\psi \vee \chi) \sim (\varphi \vee \psi) \vee \chi$	$X \cup (Y \cup Z) \approx (X \cup Y) \cup Z$	$x \vee (y \vee z) \approx (x \vee y)$
6.	$\varphi \wedge (\psi \wedge \chi) \sim (\varphi \wedge \psi) \wedge \chi$	$X \cap (Y \cap Z) \approx (X \cap Y) \cap Z$	$x \wedge (y \wedge z) \approx (x \wedge y)$
7.	$\varphi \vee (\varphi \wedge \psi) \sim \varphi$	$X \cup (X \cap Y) \approx X$	$x \vee (x \wedge y) \approx x$
8.	$\varphi \wedge (\varphi \vee \psi) \sim \varphi$	$X \cap (X \cup Y) \approx X$	$x \wedge (x \vee y) \approx x$
9.	$\varphi \vee (\psi \wedge \chi) \sim (\varphi \vee \psi) \wedge (\varphi \vee \chi)$	$X \cup (Y \cap Z) \approx (X \cup Y) \cap (X \cup Z)$	$x \vee (y \wedge z) \approx (x \vee y)$
10.	$\varphi \wedge (\psi \vee \chi) \sim (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$	$X \cap (Y \cup Z) \approx (X \cap Y) \cup (X \cap Z)$	$x \wedge (y \wedge z) \approx (x \wedge y)$
11.	$\varphi \vee \neg\varphi \sim 1$	$X \cup X' \approx 1$	$x \vee x' \approx 1$
12.	$\varphi \wedge \neg\varphi \sim 0$	$X \cap X' \approx 0$	$x \wedge x' \approx 0$
13.	$\neg\neg\varphi \sim \varphi$	$X'' \approx X$	$x'' \approx x$
14.	$\varphi \vee 1 \sim 1$	$X \cup 1 \approx 1$	$x \vee 1 \approx 1$
15.	$\varphi \wedge 0 \sim 0$	$X \cap \emptyset \approx \emptyset$	$x \wedge 0 \approx 0$
16.	$\neg(\varphi \vee \psi) \sim \neg\varphi \wedge \neg\psi$	$(X \cup Y)' \approx X' \cap Y'$	$(x \vee y)' \approx x' \wedge y'$

Equations between programs???



Higher Order Unification

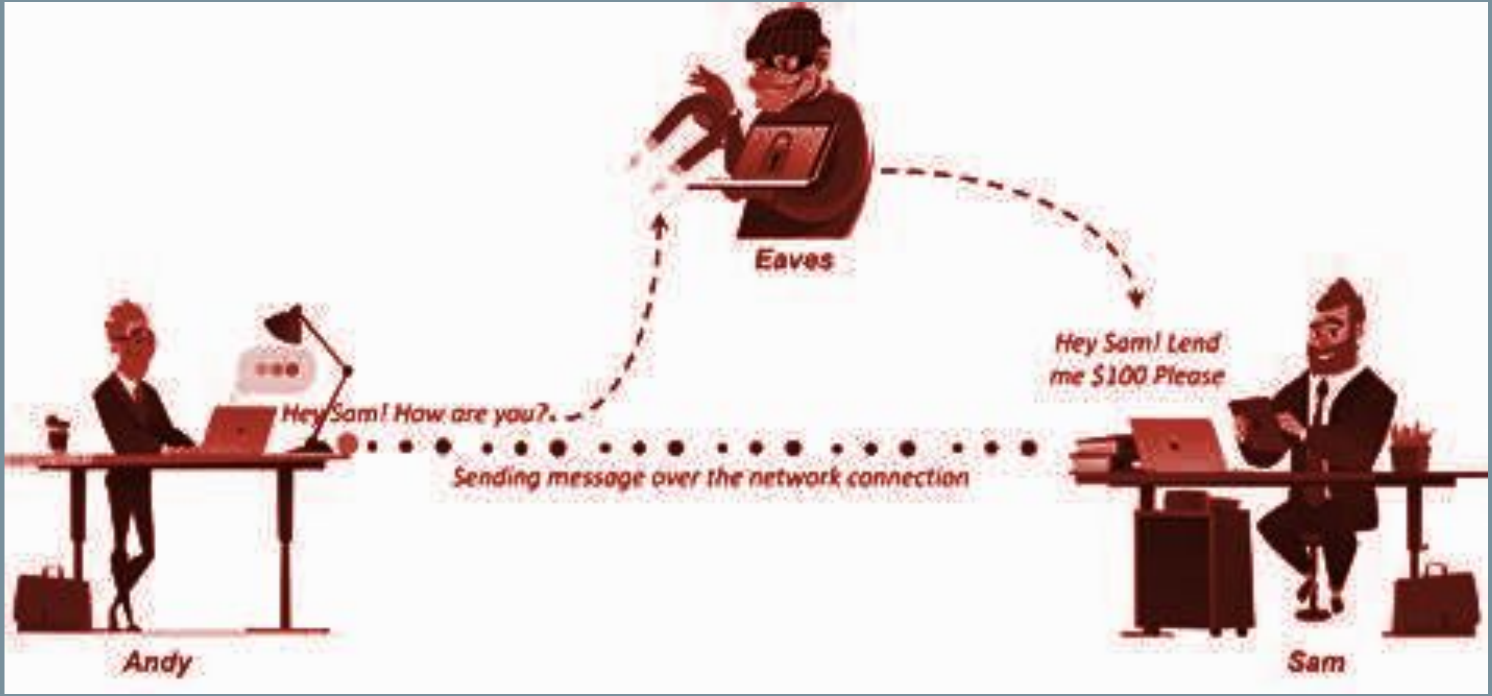
Unification of Simply Typed Lambda-Terms as Logic Programming[1]

Dale Miller
Laboratory for the Foundation of Computer Science
University of Edinburgh, and
Computer Science Department
University of Pennsylvania

Abstract

The unification of simply typed λ-terms modulo the rules of β- and η-conversions is often called "higher-order" unification because of the possible presence of variables of functional type. This kind of unification is undecidable in general and if unifiers exist, most general unifiers may not exist. In this paper, we show that such unification problems can be coded as a query of the logic programming language L_λ in a natural and clear fashion. In a sense, the translation only involves explicitly axiomatizing in L_λ the notions of equality and substitution of the simply typed λ-calculus: the rest of the unification process can be viewed as simply an interpreter of L_λ searching for proofs using those axioms.

Unification in cryptography



The Intruder Deduction Problem

Problem: Given a set of messages S={m1,m2,...,mk} observed by an intruder, and a secret s.

Question: Is there a combination of the messages in S that entails s?

Is there a context C such that C[m1,...mk]=?s

(V. Cortier, S. Delaune, H. Comon)

The Nominal Intruder Deduction Problem

Problem: Given a set of messages S={m1,m2,...,mk} observed by an intruder, a secret s, and private names a1,...,ar.

Question:

Δ ⊩subst([z]X, m1,...mk)=?s

(Ayala, Fernández and Nantes. FSCD 2016.)

Nominal Intruder Deduction Problem

A nominal DOLEV-YAO INTRUDER

$$
\begin{array}{llcrll}
(1) & \emptyset & \vdash & \pi_i(\langle X_1, X_2 \rangle) & \rightarrow X_i & (i \in \{1,2\}) \\
(2) & \emptyset & \vdash & d(\{X\}_Y, Y^{-1}) & \rightarrow X & \\
(3) & \emptyset & \vdash & d(\{X\}_{Y^{-1}}, Y) & \rightarrow X & \\
(4) & \emptyset & \vdash & (X^{-1})^{-1} & \rightarrow X & \\
(5) & \emptyset & \vdash & subst_j^n([\vec{z}]z_k, \vec{X}) & \rightarrow X_k & (1 \leq k \leq j) \\
(6) & z\#Y & \vdash & subst_1^n([z]Y, X) & \rightarrow Y & \\
(7) & z_k\#Y & \vdash & subst_j^n([\vec{z}]Y, \vec{X}) & \rightarrow subst_{j-1}^n([\overrightarrow{z'}]Y, \overrightarrow{X'}) & (1 \leq k \leq j, j > 1) \\
(8) & \emptyset & \vdash & subst_j^n([\vec{z}]f(\vec{W}), \vec{X}) & \rightarrow \overrightarrow{f(subst_j^n([\vec{z}]W, \vec{X}))} &
\end{array}
$$

Nominal Intruder Deduction Problem

mple 20. Consider $\Gamma = \{\underbrace{\{\{m\}_b\}_c}_{t_1}, \underbrace{\{b^{-1}\}_k}_{t_2}, \underbrace{\{c^{-1}\}_r}_{t_3}, \underbrace{k^{-1}}_{t_4}, \underbrace{r^{-1}}_{t_5}\}$ and a secret m (a constant). Taking into account the theory DYT, this IDP can be stated as $X\{\overrightarrow{z \mapsto t}\} \overset{\mathsf{DYT}}{\underset{?}{\approx}} m$, where $\{\overrightarrow{z \mapsto t}\}$ denotes the substitution of t_i for z_i, $i = 1, \ldots, 5$. Figure 6 shows part of the first level of the narrowing tree for this problem.

```
d(t3,t5)=c^{-1}
d(t2,t4)= b^{-1}
d(t1,c^{-1})={m}b
d({m}b,b^{-1})=m
```

```
X=d(d(t1,d(t3,t5)), d(t2,t4))
```

(Some) Systems using equational reasoning

- Maude - model-checker
- Isabelle/HOL - theorem prover
- Abella - theorem prover
- Tamarin Prover- security
- Proverif - security
- CiME - rewriting tool

(UNIF) Unification Community- Since 1987

Some links:

- https://www.mat.unb.br/dnantes/
- http://nominal.cic.unb.br/
- **Survey:** Franz Baader, Wayne Snyder: **Unification Theory.** Handbook of Automated Reasoning 2001: 445-532

Thank you!