Andreas Lynge
Verified computer algebra in homotopy type theory

- Build a homotopy type theoretic algebra library
- Apply novelties of homotopy type theory:
  - Univalence axiom
  - Higher inductive types
- Computer verify correctness of algorithms in algebra
- Develop techniques to obtain efficient algorithms
I conduct empirical studies of current practices in industry. Cases include LEGO and Vestas.

I develop AR-prototypes for remote assistance.

I study the effects of different interaction techniques on the performance, process and satisfaction of remote assistance. In the lab and in the field.
DETECTING SELECTION USING DEEP LEARNING

Bakhtawar Noor

Introduction

- Selection leaves patterns of genetic variation in the DNA
- **Goal**: Detect these patterns of genetic variation a.k.a selection given some input matrix

Method

- **Tool**: Convolution neural network
  - Why? They are good at handling high dimensional

Data

Input:
- Input is a 2D matrix filled with 0s’ and 1s’.
- **Rows** in the matrix are individuals
- **Columns** are variable positions among individuals
The goal of my PhD research is to develop a formally verified library of cryptographic software. Focus will be on elliptic curve cryptography including
- Implementing concrete curves and their group operation.
- Provide a general library for elliptic curve cryptography.
- Pairing-based elliptic curve cryptography including implementations of pairing algorithms and pairing-friendly curves.

Challenges include
- Implementations must be timing-attack resistant.
- Achieving performance which is comparable to unverified implementations.
- Achieving both good performance and security.
Fast, effective and interpretable Deep Learning

▶ What input features were salient for a given prediction?
▶ How should I change a given input to make the network “choose” another class?

Current approach: Invertible neural networks
Motivating example: Simple auction with three users

- Bidding according to secret strategies
  - Cat follows strategy $\varphi$
  - Dog follows strategy $\psi$
  - Fish follows strategy $\theta$
- Can the highest bid be publicly disclosed without disclosing who won?
  - Does depend on secrets so violates Non-Interference
  - ...yet, information disclosed is 'symmetric' in the users

Capturing the symmetry:

- Poirot thinks possible

  $\varphi \quad \psi \quad \theta$

  $\Rightarrow$ Poirot thinks possible

- Gives bound on Poirot's knowledge s.t. composition preserves symmetry

Who used $\varphi$?
I shall use my little grey cells!

$\omega(Cat) = \omega'(Dog)$  $\omega(Dog) = \omega'(Cat)$  $\omega(Fish) = \omega'(Fish)$

$\forall \omega, \omega', \omega \sim Cat \rightarrow \omega \in k \Rightarrow \omega' \in k$
"Basic" Introduction to Johnson-Lindenstrauss Transforms (JLTs)

High dimensional data: $\mathbb{R}^d$  
JLT  
Low dimensional data: $\mathbb{R}^m$

$m = O(1/\varepsilon^2 \cdot \log (1/\delta))$

$\delta$: The JLT is allowed to fail with probability $1 - \delta$
$\varepsilon$: The $\ell_2$ norm of the embedded datum is within $(1 \pm \varepsilon)$ of the original $\ell_2$ norm
$m$ does not depend on $d$!

An example JLT:
An $m \times d$ matrix, where each entry is i.i.d. sampled from $\pm 1$.
Time to apply this JLT: $O(d \cdot m)$ or $O(|x|_0 \cdot m)$

Time: $O(|x|_0 \cdot 1/\varepsilon \cdot \log (1/\delta))$

Are there faster JLTs?
Preprocess with FFT  
Very sparse matrix  
Time: $O(d \cdot \log (d) + m \cdot \log^2(1/\delta))$

Use a sparse matrix
Restrict input data

Extremely sparse matrix  
Time: $O(|x|_0)$
Reasoning about capability machines in Iris, a higher-order concurrent separation logic framework

Capability: unforgeable token of authority

A capability with read-write-local-execute permissions, over the range \([b,e]\), currently pointing at \(a\)

Formalized proofs in Iris

Goal: a compiler that provably preserves the security properties of a high level source language
Substantial Taxi Service
**SECURE MULTI-PARTY COMPUTATION (MPC)**

- Parties with **private** inputs

- **Goal**: Compute **joint** function $f(x_1, x_2, x_3, x_4)$

- **Mutual distrust**
  - Adversary corrupts $t$ out of $n$ parties

- **Two important requirements**
  - Privacy
  - Correctness

- MPC emulates TTP
- TTP – Trusted Third Party
FieldAI

\[ x \in \{ \text{multispectral, SAR, height, weather, ...} \} \]
\[ y \in \text{???} \]
How to design **programming languages** that protects secret data from being inadvertently disclosed?

Johan Bay
SINGLE CELL RNA-SEQ OF TESTIS SAMPLES

PROBLEM
• Around 10-15% of couples are infertile
• Testicles include many cell types

SOLUTION
• Using single cell RNA-Seq we can identify cell types
• By comparing the expression of healthy and non-healthy cell types we can extract markers for infertility
• If we have markers, we can better understand infertility and

CURE IT
HYPOTHESIS
The X and Y chromosomes have been fighting a war during the evolution of our species

SCIENTIFIC DESIGN’S MAIN IDEAS
• The battle ground is spermatogenesis
• After meiosis, the battle begins
• Using single cell sequencing, we can target cell types
• By analyzing the expression in the haploid phase, we can detect signs of past fights
• We can better understand the complex evolutionary history of our sex chromosomes
Generation time differences between Europeans and Asians inferred by Neanderthal fragment length

Moisès Coll Macià

- 2% of our DNA comes from Neanderthals
- Each generation DNA fragments shorten
- The more generations, the shorter the fragments are
- We find:
  - Europeans have short fragments
  - Asians have long fragments
- Thus, Europeans must have had younger parents than Asians during the last 50,000 years
Data Science on the Desktop
- Data Mining on Modern Hardware

Exploit multi-core CPUs and GPUs to speed up Data Mining tasks, such as
- Clustering
- Outlier Detection
- Trend Detection
- ...

Identify suitable task for GPU and CPU
Chickens, bugs and compilers
Data Science on the Desktop
- Data Mining on Modern Hardware

Exploding multi-core CPUs and GPUs to speed up data mining algorithms, for tasks such as:

- Clustering
- Outlier detection
- Trend detection
- ...

Identify suitable tasks for CPU and GPU

Balance throughput and data transfer
Range Searching: Given a set of n points in d dimensional space, we want to build a data structure such that given a query range (a subspace), we can count or report the points in the query range efficiently.

Examples in 2D:

- Orthogonal Range Searching
- Simplex Range Searching
- Semialgebraic Range Searching
many people engaged in many activities mediated by many artifacts. discuss.

welcome to modern life
welcome to artifact ecologies
(Peter Lyle, CMA)
Progressive Visual Analytics

Map-Like Visualization
João Belo
Ubiquitous Computing and Interaction Group

Context Aware User Interfaces for Augmented Reality

- Computational Interaction
  "Smart" user interfaces, optimized depending on user needs/behaviour
- Deep Learning (ConvNets)
  Understand the user and his/her environment

We make cool stuff! Collaborate with us!
Privacy-Preserving Machine Learning (PPML)

- Infeasible for consumers and companies to train complex machine learning models independently

- So machine learning as a service (MLaaS) is offered by companies such as Amazon, Google and Microsoft

- Concerns:
  - Privacy about the queries being made to the servers
  - Loss of competitive advantage due to sharing data with Amazon when performing collaborative machine learning
Data Structures and Algorithms

Binary Tree

Stack

Matrix

Unbalanced Tree

Array

Heap

Rebalanced Tree

Linked List

Sparse Matrix
A Component Model for Ubiquitous Analytics

Integrating Data-Driven Reporting in Collaborative Visual Analytics
\[
Z(n)(z) = \text{Some from } m = (\text{from}, \text{to}, \text{msg}, \text{SENT}) \quad m_id \notin \text{dom}(M) \quad M' = M[m_id \mapsto m] \\
\langle n; \text{sendto } z \text{ msg to}, (H, Z, L, P, M) \rangle \rightarrow_h \langle n; \text{length msg}, (H, Z, L, P, M') \rangle
\]

\[
Z(n)(z) = \text{None } (ip, p) \notin \text{dom}(L) \quad p \notin P(ip) \\
Z'(z) = Z[n \rightarrow S[z \rightarrow \text{Some } a]] \\
L' = L[a \rightarrow n] \\
P' = P[a \rightarrow P(a) \cup \{p\}] \\
\langle n; \text{socketbind } z(a), (H, Z, L, P, M) \rangle \rightarrow_h \langle n; 0, (H, Z', L', P', M) \rangle
\]

\[
\phi_{req}(p) \triangleq \lambda m, \exists ps, r, sp. \text{parts}(p) \cup ps) \ast \text{is_req(body)(m, r + 1)} \ast \\
\text{from(m)} \Rightarrow^\text{prot} \phi_{coord} * p \xrightarrow{c} (r + 1, \text{WAIT}) * p \xrightarrow{\frac{1}{3}} (r, \text{INIT sp}) * P(\text{body}(m), p)
\]

\[
\phi_{glob}(p) \triangleq \lambda m, \exists ga, ms, ps, r, sc, sp. \{\text{from(m)} \in \text{ms}\} * ps * \text{is_global(body)(m, r)} * \\
\text{parts}(ps) * \text{from(m)} \Rightarrow^\text{prot} \phi_{coord} * p \xrightarrow{c} (r, sc) * p \xrightarrow{\frac{1}{3}} (r, sp) * \\
\left(\forall m_id, \pi. \text{is_vote(body)(ms, r)} * m_id \xrightarrow{m} \pi \right) * \\
(ga = \emptyset \land \text{is_abort(body)(m, r)} \land sc = \text{COMM} \lor \\
(ga \neq \emptyset \land \text{is_abort(body)(m, r))} \land sc = \text{ABORT)
\]

\[
\phi_{part}(p) \triangleq \lambda m, \phi_{req}(p)(m) \lor \phi_{glob}(p)(m)
\]

\[
P \triangleq \lambda p, m. \exists \log, s. m = "$\text{REQUEST }$" \ast s * p \xrightarrow{l} \left\{ \frac{1}{2} \right\} \log * p \xrightarrow{w} \left\{ \frac{1}{4} \right\} \log, s
\]

\[
Q \triangleq \lambda p, n. \exists \log, s. p \xrightarrow{l} \left\{ \frac{1}{2} \right\} \log @ s * p \xrightarrow{w} \left\{ \frac{1}{4} \right\} \log, s
\]

\[
\forall z. z \xrightarrow{s} \text{None } \rightarrow \text{wp } \langle n; z \rangle \{\Phi\} \Rightarrow ^\text{IsNode}(n)
\]

\[
\text{wp } \langle n; \text{socketbind } z(a, p) \rangle \{\Phi\}
\]

\[
\forall g. z \xrightarrow{s} \text{Some } (ip, p) \xrightarrow{\text{proj}} g * (ip, p) \Rightarrow ^\text{prot} \phi \rightarrow \text{wp } \langle n; 0 \rangle \{\Phi\}
\]

\[
\text{wp } \langle n; \text{socketbind } z(a, p) \rangle \{\Phi\}
\]

\[
\forall g. z \xrightarrow{s} \text{Some } (ip, p) \xrightarrow{\text{proj}} g * (ip, p) \Rightarrow ^\text{prot} \phi \rightarrow \text{wp } \langle n; 0 \rangle \{\Phi\}
\]

\[
f \xrightarrow{A} \quad (ip, \notin A) \quad \phi \quad z \xrightarrow{s} \text{None }
\]

\[
\text{wp } \langle n; \text{socketbind } z(a, p) \rangle \{\Phi\}
\]

\[
\forall m_id, M. m\text{Soup(M)} * m_id \xrightarrow{s} (a, d, s) * P \Rightarrow m\text{Soup(M)} * \phi(a, d, s) * Q
\]

\[
z \xrightarrow{s} \text{Some } a * Q \rightarrow \text{wp } \langle n; \text{length}(s) \rangle \{\Phi\}
\]

\[
\text{wp } \langle n; \text{sendto } z s d \rangle \{\Phi\}
\]

\[
\forall p, m_id, M. m\text{Soup(M)} * m_id \xrightarrow{s} (a, d, s) * P \Rightarrow m\text{Soup(M)} * \phi(a, d, s) * Q
\]

\[
z \xrightarrow{s} \text{Some } a * Q \rightarrow \text{wp } \langle n; \text{length}(s) \rangle \{\Phi\}
\]

\[
\text{wp } \langle n; \text{sendto } z s d \rangle \{\Phi\}
\]
ML problems are underdetermined (like all problems requiring induction).

• There are multiple (often infinite) allowed solutions (models) some of which might not align with the intention of the programmer.

Examples:
• Training images divided into wolf and husky categories → to the model they might as well be divided into snow-in-background and not-snow-in-background images because of insufficient data.
• Asthma correlates with lower risk of death from pneumonia but only through a missing confounder (treatment).

Incompleteness can be reduced but not removed.
Social and Practical Functioning during Collaborative Writing

Ida Larsen-Ledet
Ph.D. student
The Computer Mediated Activity group

"[You] respect each other's sections so you don't go and edit them."

(Group R15)
Node.js Application

Instrumented Application

Several callbacks scheduled to run

- Dynamic analysis
- Happens-before relations
- Controlled execution

Race Bug
My research involves building better tools to reason about programming languages:

- Modern PLs are a zoo (concurrency, mutable variables, exceptions, oh my!)
- To reason about complicated programs, we need complicated tools.
- To reason about those tools, we use.... well moderately less complex math.\(^1\)

Example: for concurrent code and no GC, we built a logic to prove nothing leaks.

\(^1\)I also am interested in building tools for reasoning about that math. That’s a different slide though
THE INTERPLAY BETWEEN
FORCE FEEDBACK AND
SHAPE CHANGE
Optimising Functional-/Stream-based programming in Java

```java
public int sumEvenSquares() {
    return IntStream.range(0, 100000000)
        .filter(x -> x % 2 == 0)
        .map(x -> x * x)
        .sum();
}
```

**Declarative:**
Decouples behaviour from implementation

Requires invocations of virtual functions and lambdas

**Compiler optimisations**

```java
public int sumEvenSquares() {
    int sum = 0;
    for(int x = 0; x < 100000000; x++)
        if(x % 2 == 0)
            sum += x * x;
    return sum;
}
```

**Fast**

No function invocations required
VIBROTACTILE CUES FOR PROVIDING GUIDANCE IN INTERACTIVE DATA VISUALIZATION

Visual cues are often used in data visualization but can be ineffective when multiple visual cues used at the same time.