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

AR-based Interaction Techniques for Remote Assistance

Troels A. Rasmussen – UbiComp Group

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

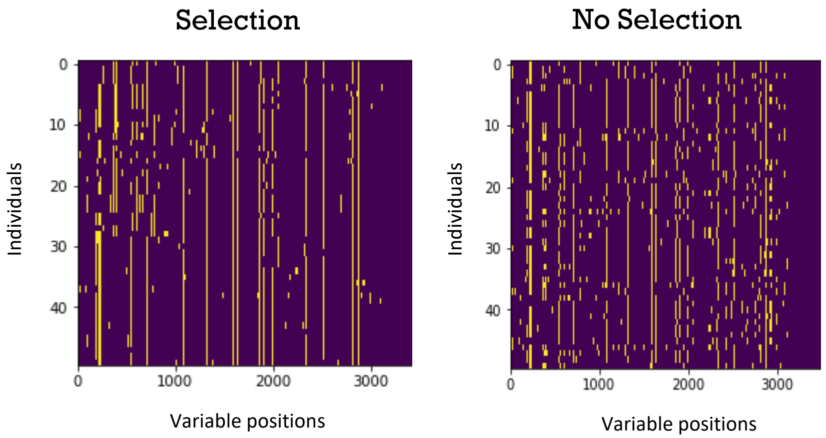
Introduction

- 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

Verifiable cryptographic software

The goal of my PhD research is to develop a formally verified library of cryptographic software.

Focus will be on elliptic curve cryptography including

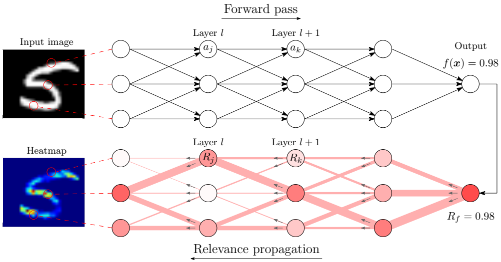
- Implementing concrete curves and their group operation.
- Provide a general library for elliptic curve cryptography.
- Pairing-based elliptic curve cryptography including implementations of pairing algorithms and pairing-friendly curves.

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

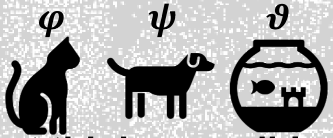
Anonymous Information Flow

Motivating example: Simple auction with three users

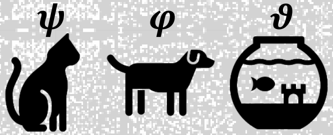
- Bidding according to secret strategies
  - *Cat* follows strategy φ
  - *Dog* follows strategy ψ
  - *Fish* follows strategy ϑ
- Can the highest bid be publicly disclosed without disclosing who won?
  - Does depend on secrets so violates Non-Interference
  - …yet, information disclosed is 'symmetric' in the users

Capturing the symmetry:

- Poirot thinks possible



- $\Longrightarrow$ Poirot thinks possible

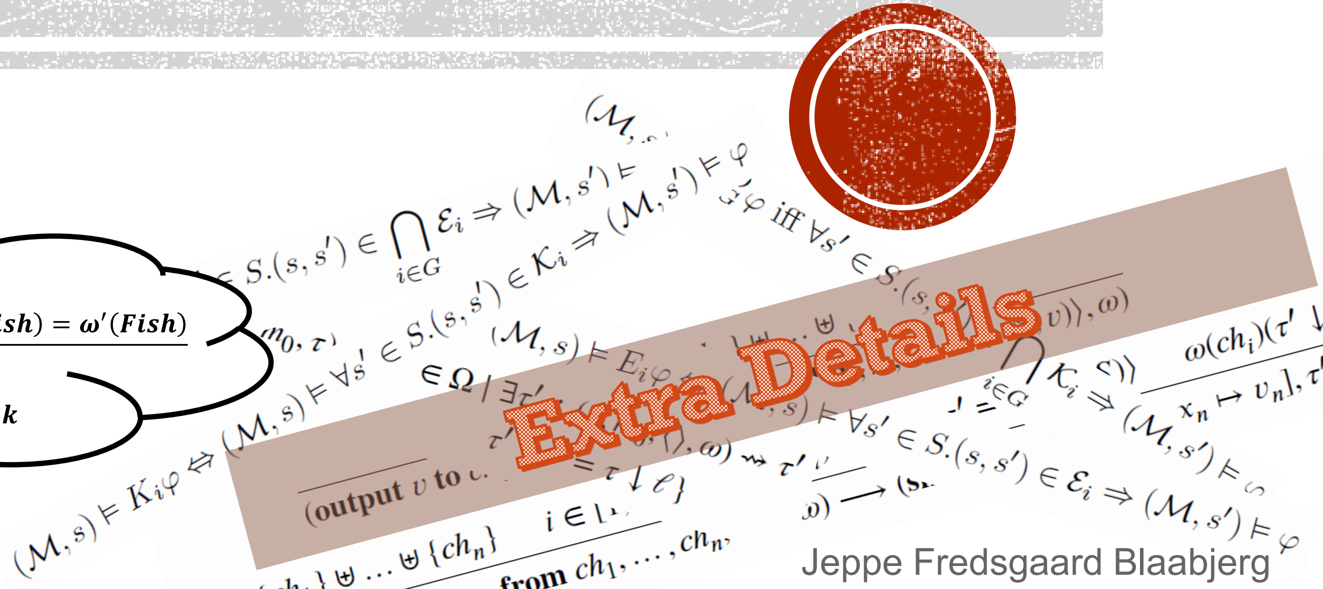


- Gives bound on Poirot's knowledge s.t. composition preserves symmetry

Who used φ?
I shall use my
little grey cells!

$$\frac{\omega(Cat) = \omega'(Dog) \quad \omega(Dog) = \omega'(Cat) \quad \omega(Fish) = \omega'(Fish)}{\omega \sim^{Cat,Dog} \omega'}$$

$$\forall \omega, \omega' : \omega \sim^{Cat,Dog} \omega' \wedge \omega \in k \Rightarrow \omega' \in k$$



"Basic" Introduction to Johnson-Lindenstrauss Transforms (JLTs)

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

δ: The JLT is allowed to fail with probability $1 - \delta$

ε: The ℓ_2 norm of the embedded datum is within $(1 \pm \varepsilon)$ of the original ℓ_2 norm

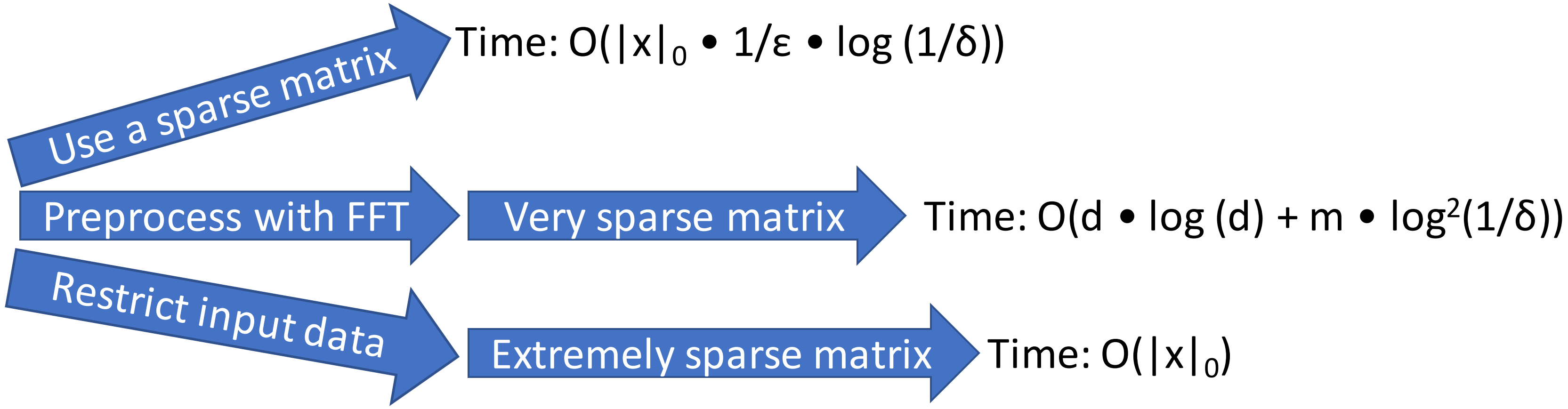
m does not depend on d!

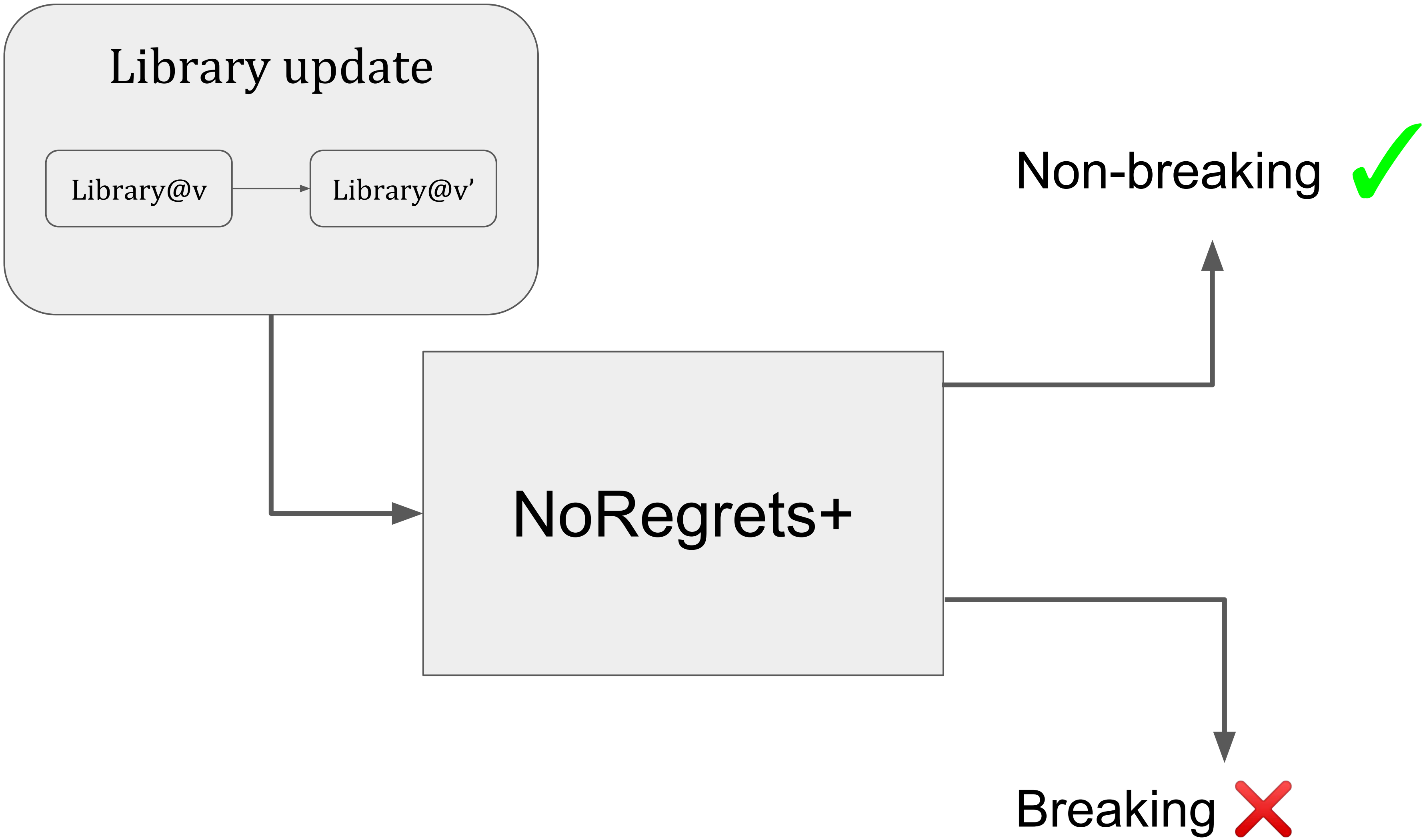
An example JLT:

An $m \times d$ matrix, where each entry is i.i.d. sampled from ± 1.

Time to apply this JLT: $O(d \bullet m)$ or $O(|x|_0 \bullet m)$

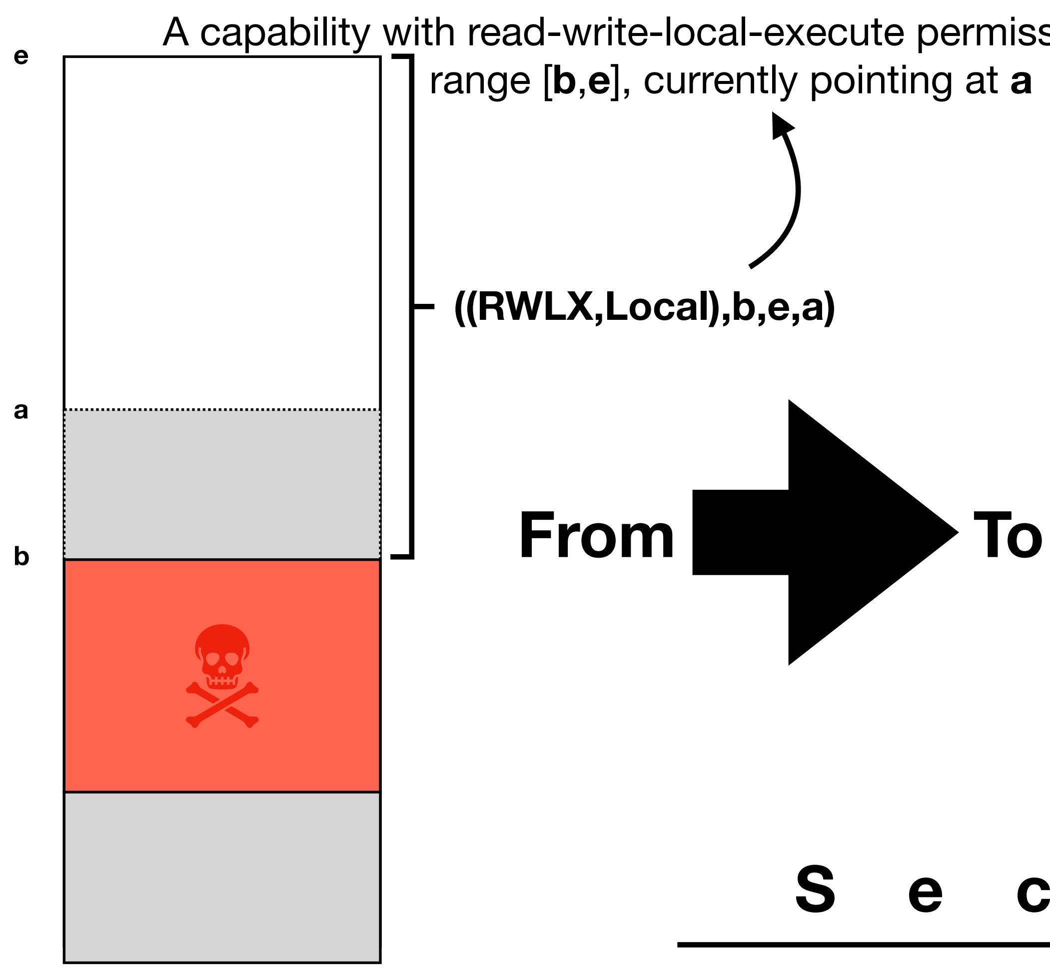
Are there faster JLTs?




Library update
Library@v
Library@v'
NoRegrets+
Non-breaking
Breaking

Reasoning about capability machines in Iris, a higher-order concurrent separation logic framework

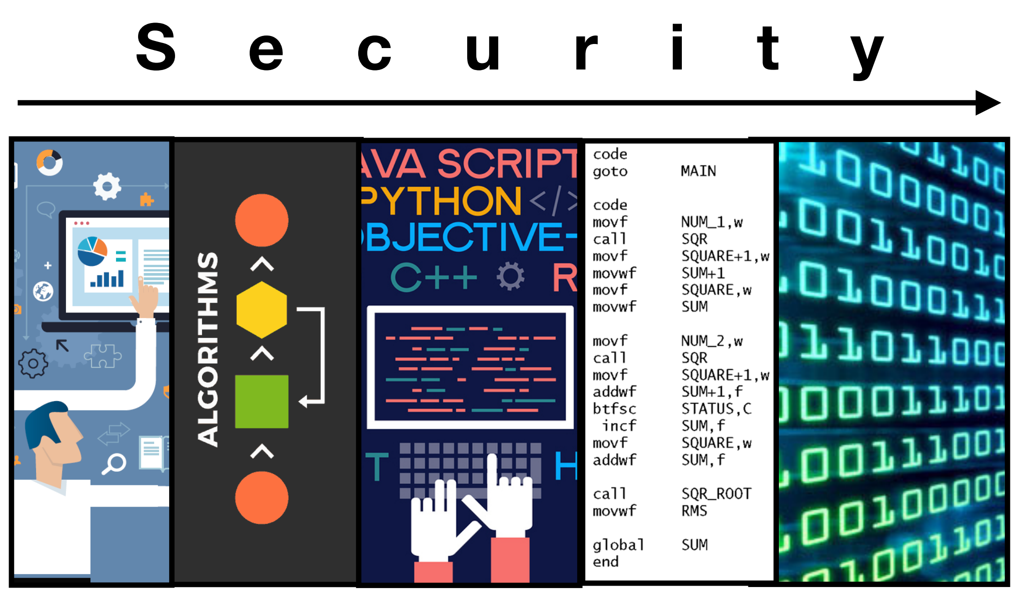
Capability: unforgeable token of authority



Capability Machine model

```
Theorem fundamental (perm : Perm) b e g (a : Addr) stsf E r :
  (⌜perm = RX⌝ ∧ (∃ ws, [∗ list] a;w ∈ (region_addrs b e);ws, na_inv logrel_nais (logN .@ a)
                                  (read_only_cond a w interp))) ∨
  (⌜perm = RWX⌝ ∧ ([∗ list] a ∈ (region_addrs b e), na_inv logrel_nais (logN .@ a)
                                  (read_write_cond a interp))) ∨
  (⌜perm = RWLX⌝ ∧ ([∗ list] a ∈ (region_addrs b e), na_inv logrel_nais (logN .@ a)
                                  (read_write_local_cond a interp))) -∗
  ⟦ inr ((perm,g),b,e,a) ⟧ₑ stsf E r.
Proof.
  iIntros "[#[-> Hinv] | [#[-> Hinv] | #[-> Hinv]]]".
  - iDestruct "Hinv" as (ws) "Hinv". by iApply fundamental_RX.
  - by iApply fundamental_RWX.
  - by iApply fundamental_RWLX.
Qed.
```

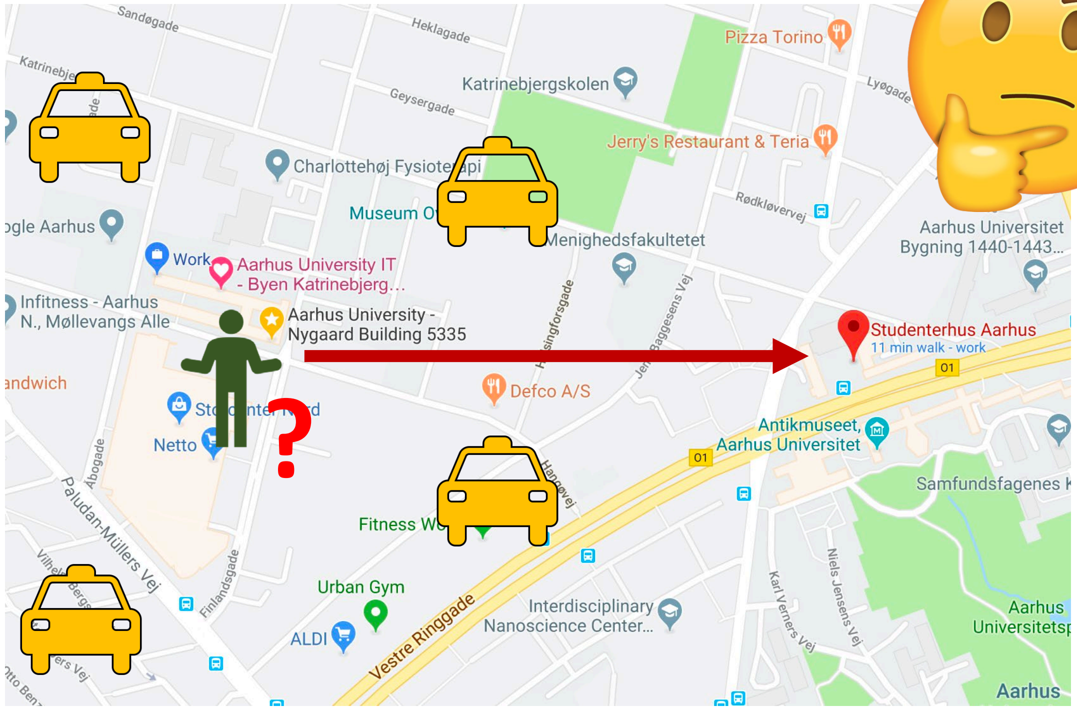
Formalized proofs in Iris

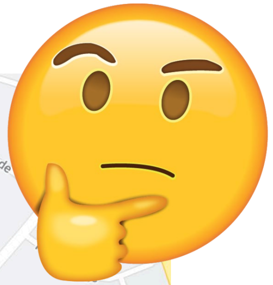


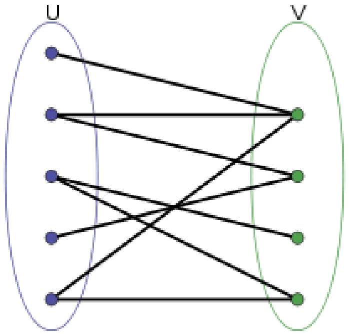
Abstraction Layers

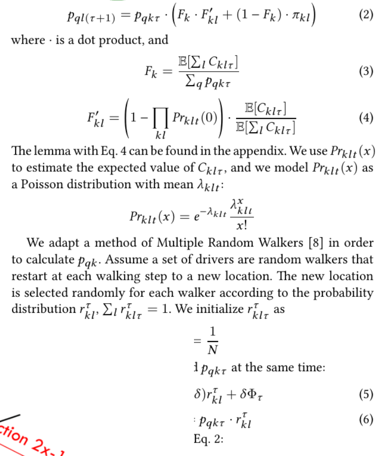
Goal: a compiler that *provably* preserves the security properties of a high level source language

Substantial Taxi Service









$$p_{ql(\tau+1)} = p_{qk\tau} \cdot \left(F_k \cdot F'_{kl} + (1 - F_k) \cdot \pi_{kl}\right) \quad (2)$$

where · is a dot product, and

$$F_k = \frac{\mathbb{E}[\sum_l C_{kl\tau}]}{\sum_q p_{qk\tau}} \quad (3)$$

$$F'_{kl} = \left(1 - \prod_{kl} Pr_{klt}(0)\right) \cdot \frac{\mathbb{E}[C_{kl\tau}]}{\mathbb{E}[\sum_l C_{kl\tau}]} \quad (4)$$

The lemma with Eq. 4 can be found in the appendix. We use $Pr_{klt}(x)$ to estimate the expected value of $C_{kl\tau}$, and we model $Pr_{klt}(x)$ as a Poisson distribution with mean λ_{klt}:

$$Pr_{klt}(x) = e^{-\lambda_{klt}} \frac{\lambda_{klt}^x}{x!}$$

We adapt a method of Multiple Random Walkers [8] in order to calculate p_{qk}. Assume a set of drivers are random walkers that restart at each walking step to a new location. The new location is selected randomly for each walker according to the probability distribution r_{kl}^{τ}, $\sum_l r_{kl\tau}^{\tau} = 1$. We initialize $r_{kl\tau}^{\tau}$ as

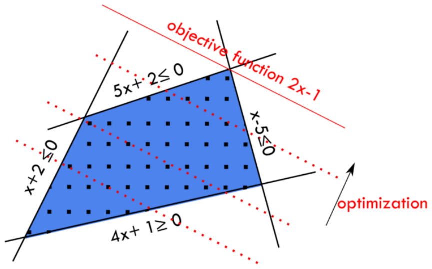
$$= \frac{1}{N}$$

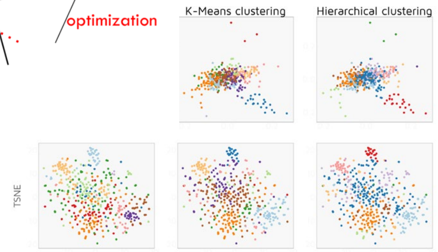
$p_{qk\tau}$ at the same time:

$$\delta) r_{kl}^{\tau} + \delta \Phi_{\tau} \quad (5)$$

$$p_{qk\tau} \cdot r_{kl}^{\tau} \quad (6)$$

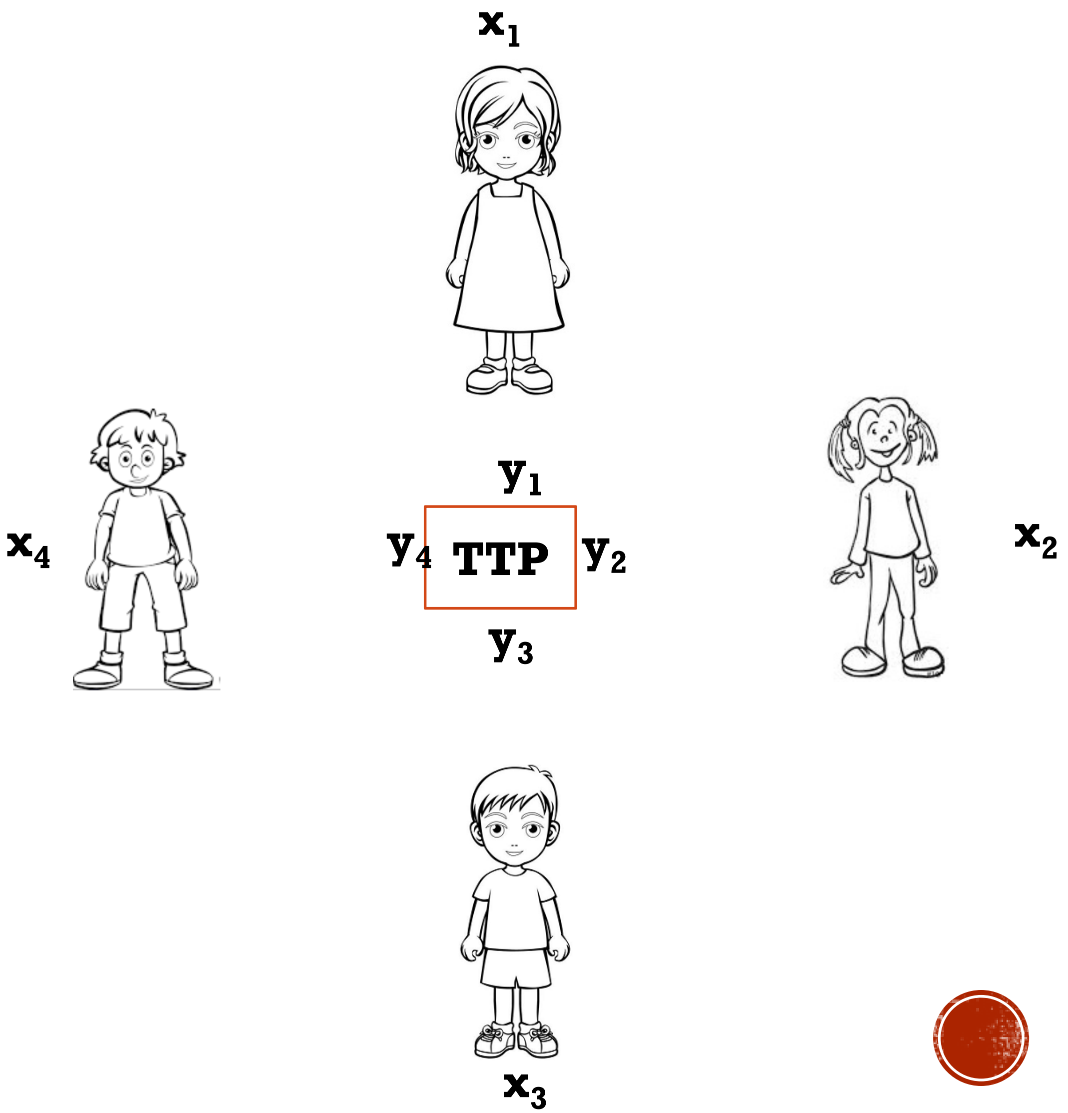
Eq. 2:



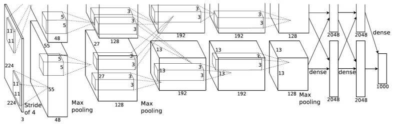


SECURE MULTI-PARTY COMPUTATION (MPC)

- 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



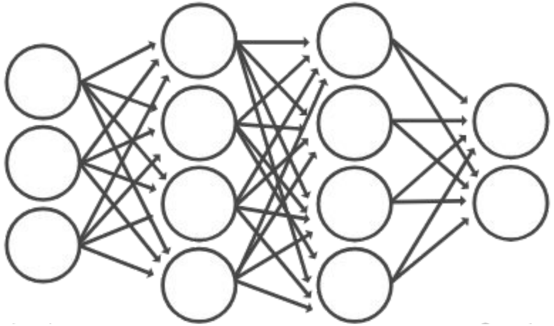
FieldAI



?

deep learning

+



=

$x \in$ { multispectral, SAR, height, weather, …}

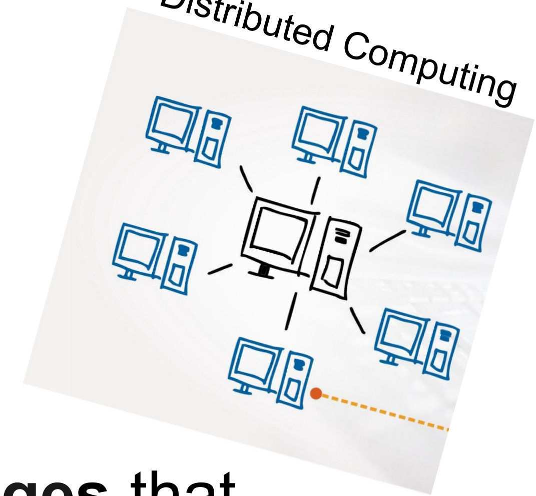
$y \in$???

???

FIELDSENSE

Coq

Distributed Computing



How to design **programming languages** that **protects secret data** from being inadvertently disclosed?

$$\text{S-ASSIGN}\quad \frac{\langle e, m\rangle \Downarrow \langle v; \ell_e\rangle \qquad pc \sqcup \ell_e \sqsubseteq lev(x)}{\langle x = e, m, pc\rangle \longrightarrow_{a(x,v)} \langle \mathsf{stop}, m[x \mapsto v], pc\rangle}$$

Johan Bay

SINGLE CELL RNA-SEQ OF TESTIS SAMPLES

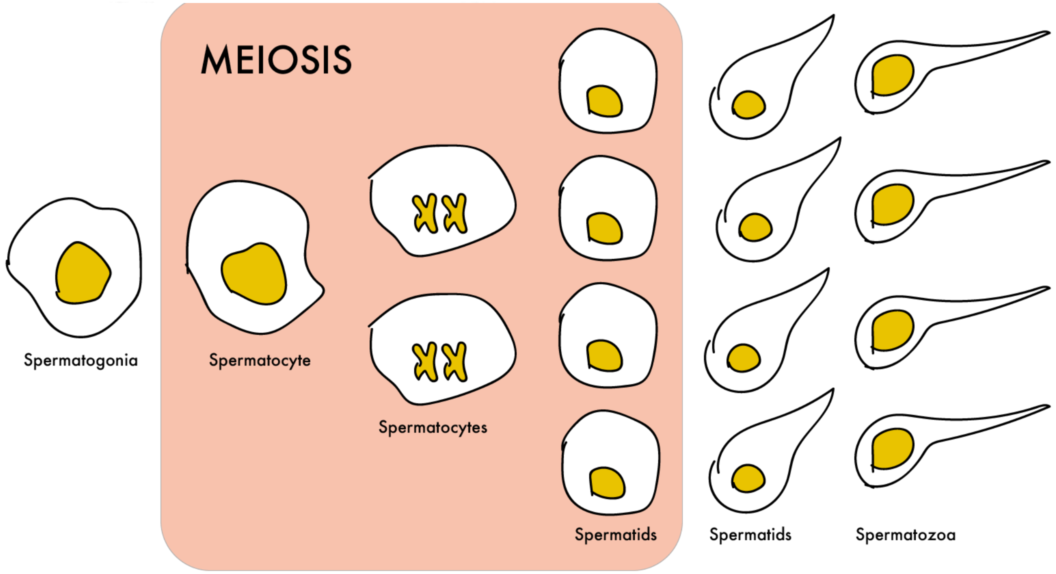
PROBLEM

- 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



Meritxell Riera Bellés

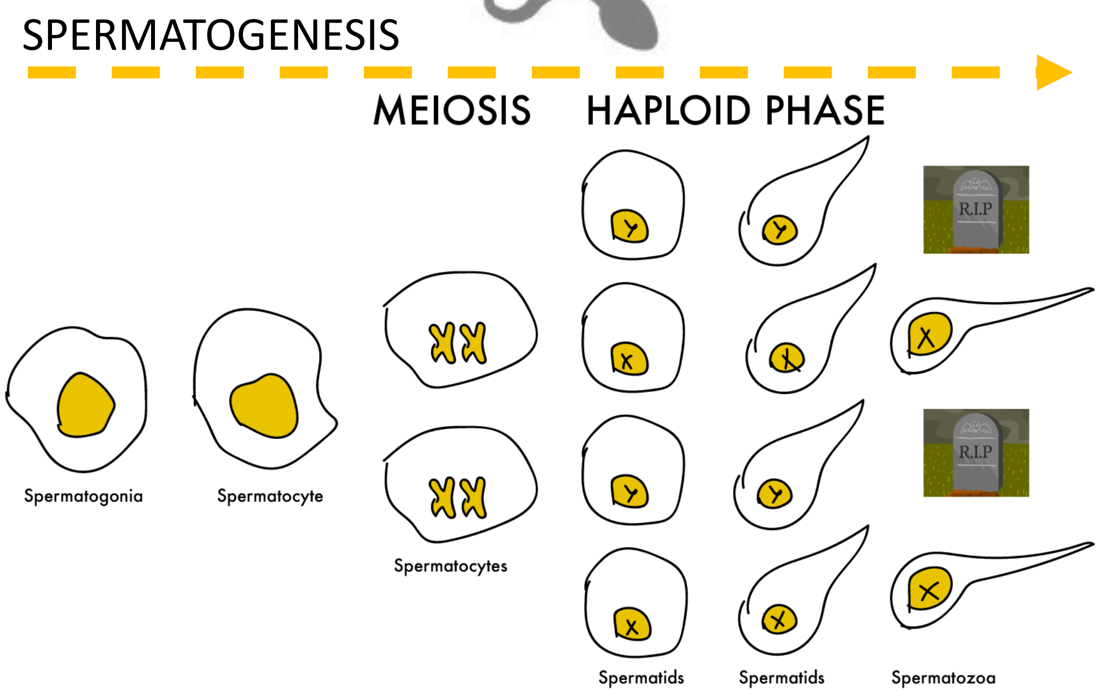
SINGLE CELL RNA-SEQ OF TESTIS SAMPLES

HYPOTHESIS

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

SCIENTIFIC DESIGN'S MAIN IDEAS

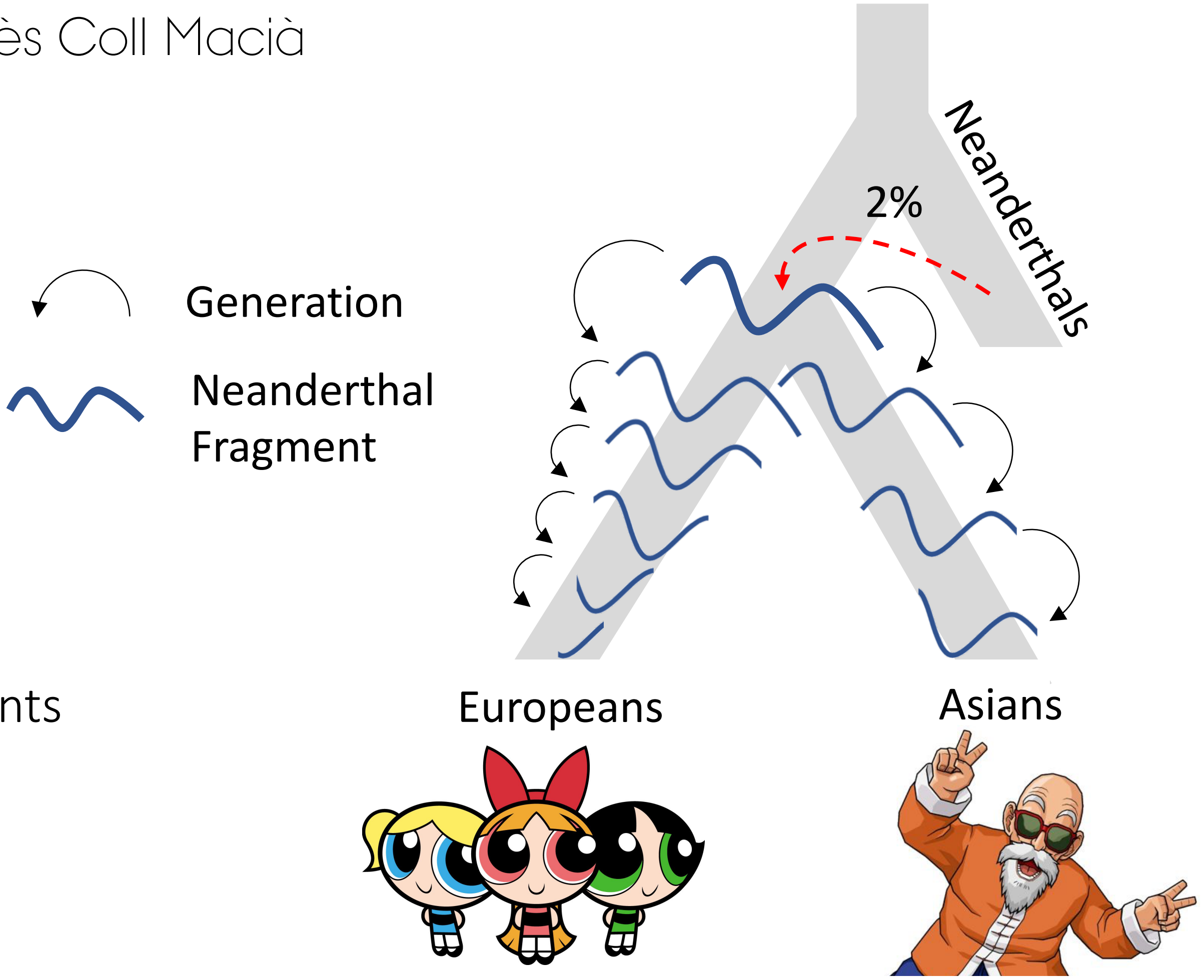
- The battle ground is spermatogenesis
- After meiosis, the battle begins
- Using single cell sequencing, we can target cell types
- By analyzing the expression in the haploid phase, we can detect signs of past fights
- We can better understand the complex evolutionary history of our sex chromosomes



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

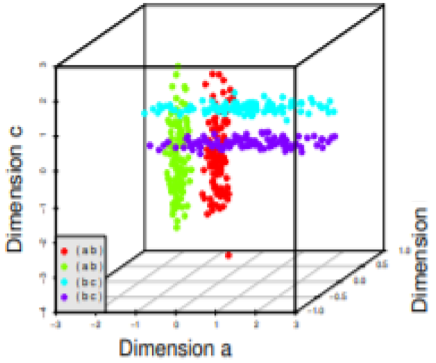


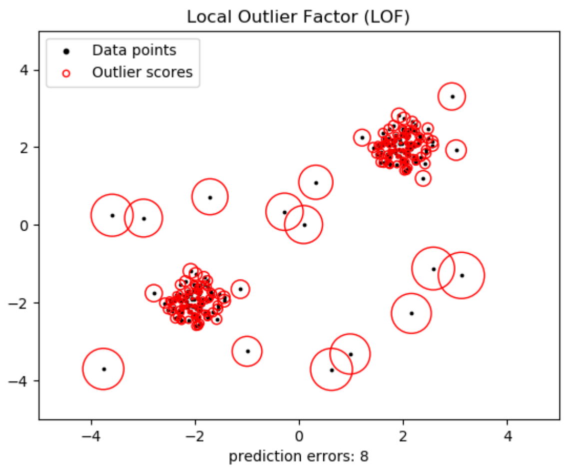
Data Science on the Desktop

- Data Mining on Modern Hardware

Exploit multi-core CPUs and GPUs to speed up Data Mining tasks, such as

- Clustering
- Outlier Detection
- Trend Detection
- …



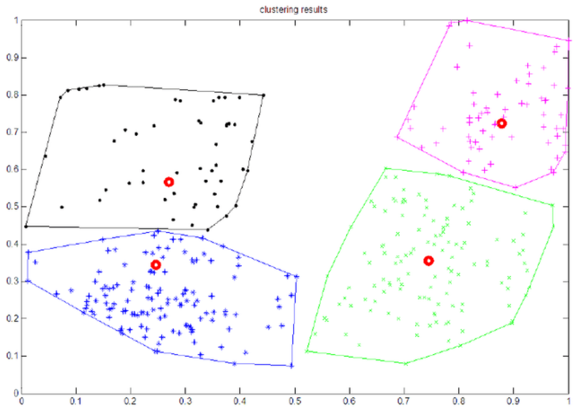


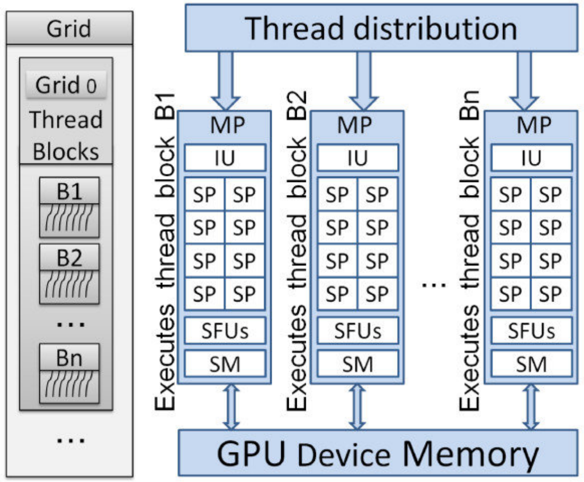
Standard C Code

```c
void saxpy(int n, float a,
           float *x, float *y)
{
  for (int i = 0; i < n; ++i)
    y[i] = a*x[i] + y[i];
}

int N = 1<<20;

// Perform SAXPY on 1M elements
saxpy(N, 2.0, x, y);
```

C with CUDA extensions

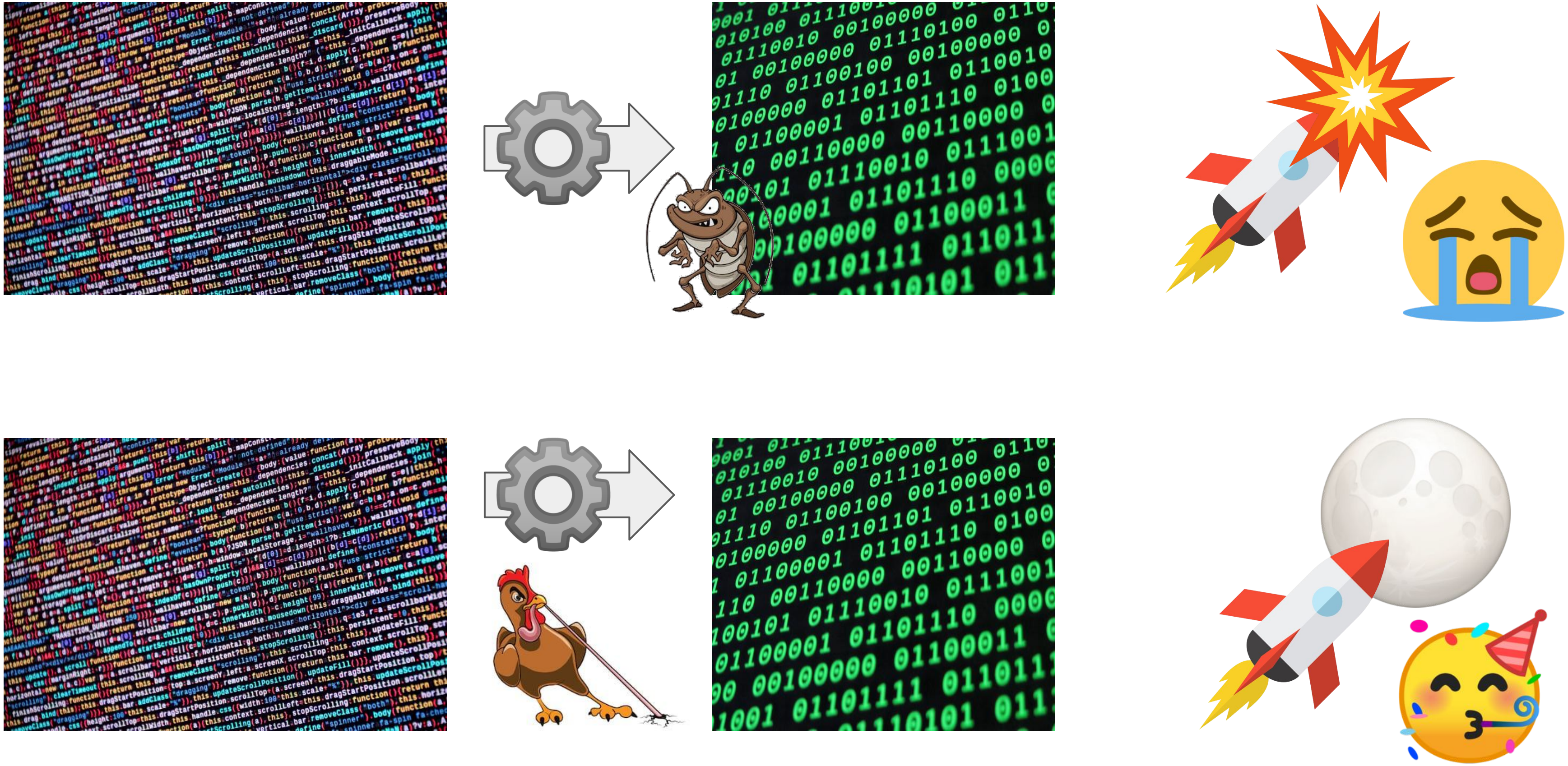
```c
__global__
void saxpy(int n, float a,
           float *x, float *y)
{
  int i = blockIdx.x*blockDim.x + threadIdx.x;
  if (i < n) y[i] = a*x[i] + y[i];
}

int N = 1<<20;
cudaMemcpy(x, d_x, N, cudaMemcpyHostToDevice);
cudaMemcpy(y, d_y, N, cudaMemcpyHostToDevice);

// Perform SAXPY on 1M elements
saxpy<<<4096,256>>>(N, 2.0, x, y);

cudaMemcpy(d_y, y, N, cudaMemcpyDeviceToHost);
```





Identify suitable task for GPU and CPU

Chickens, bugs and compilers

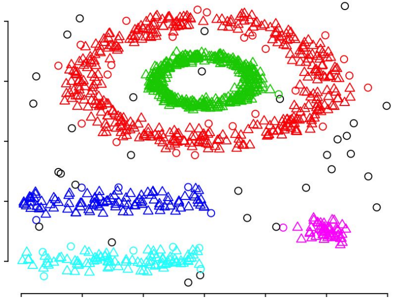


Data Science on the Desktop

- Data Mining on Modern Hardware

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

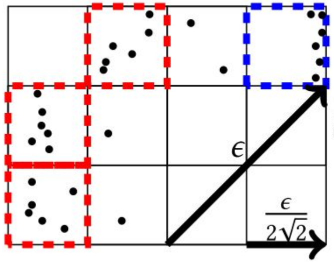
- Clustering
- Outlier detection
- Trend detection
- …



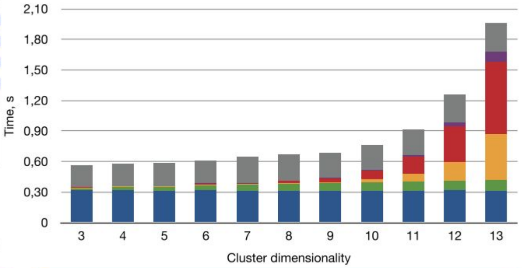
Identify suitable tasks for CPU and GPU

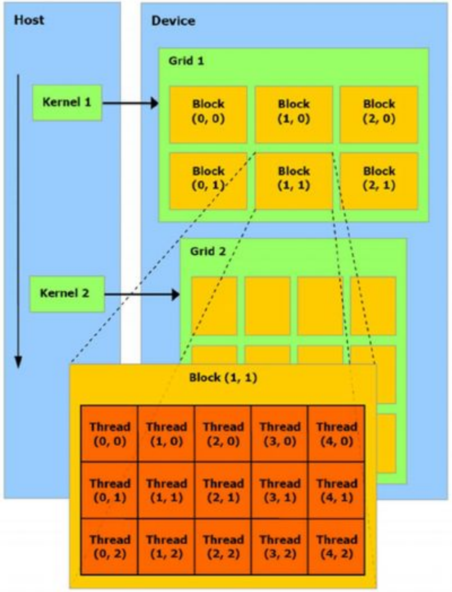
Balance throughput and data transfer

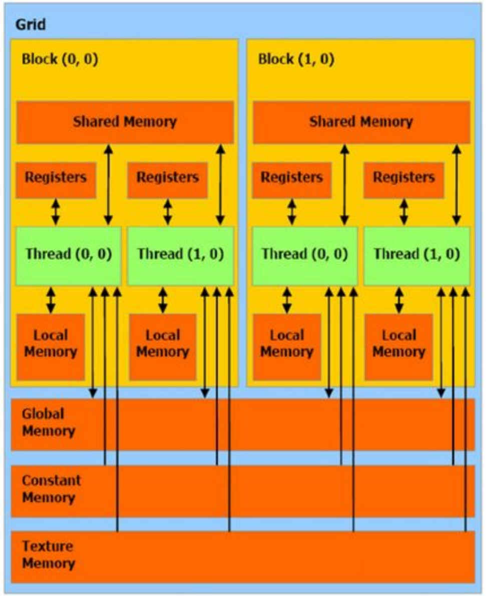
Partition



Analize

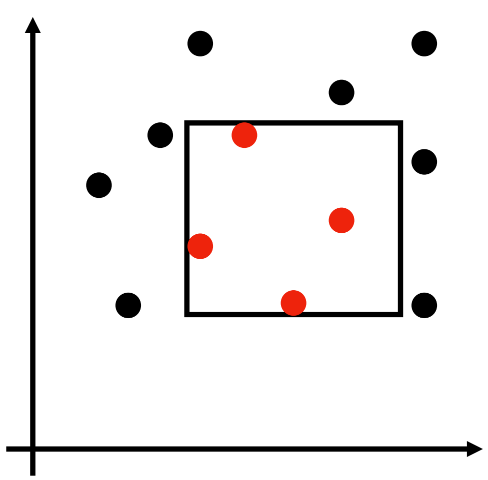




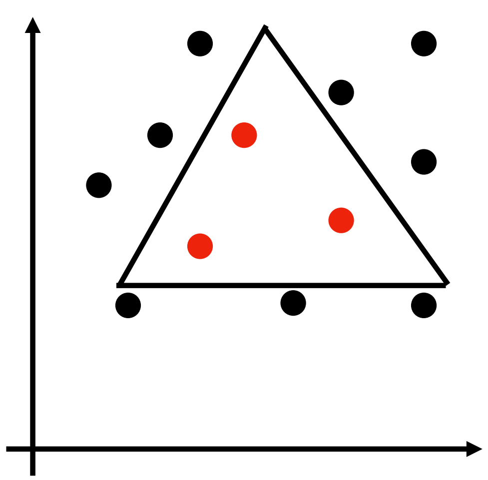


Range Searching: Given a set of n points in d dimensional space, we want to build a data structure such that given a query range (a subspace), we can count or report the points in the query range efficiently.

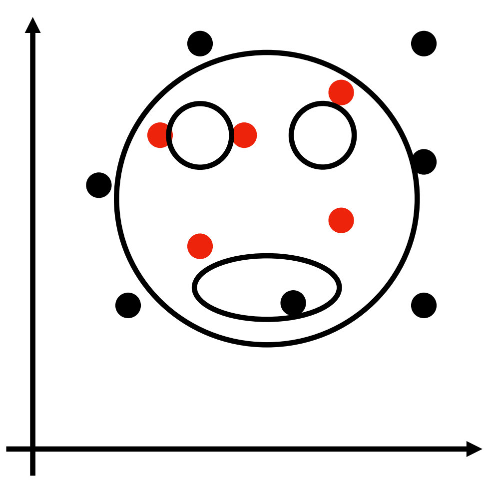
Examples in 2D:



Othogonal
Range Searching



Simplex
Range Searching



Semialgebraic
Range Searching

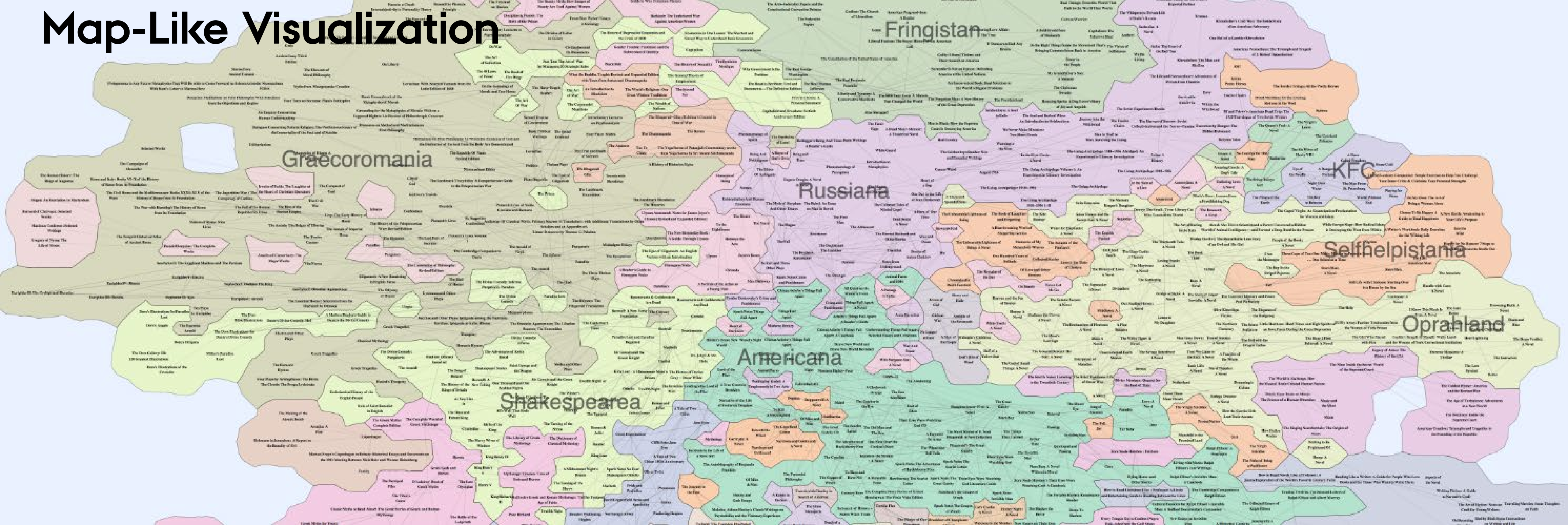
many people engaged in many activities mediated by many artifacts. discuss.

welcome to modern life

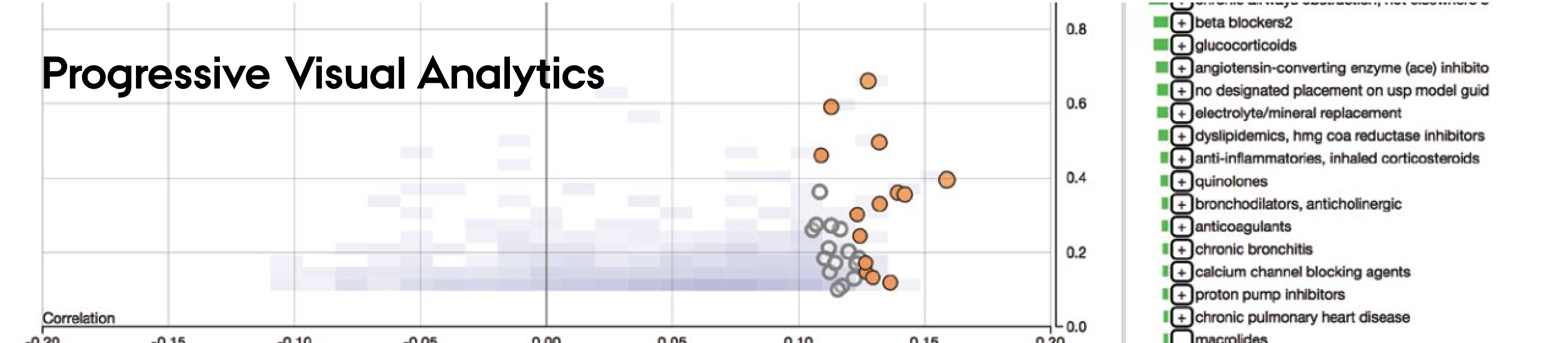
welcome to artifact ecologies

(Peter Lyle, CMA)

Map-Like Visualization



Progressive Visual Analytics



YOU PROBABLY CAN'T PRONOUNCE THIS

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

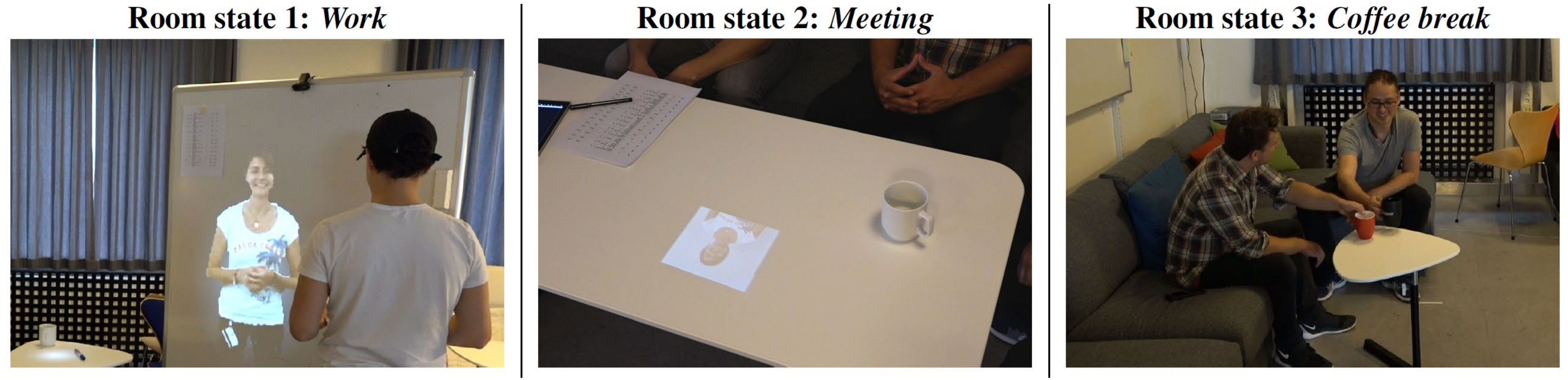
- Deep Learning (ConvNets)

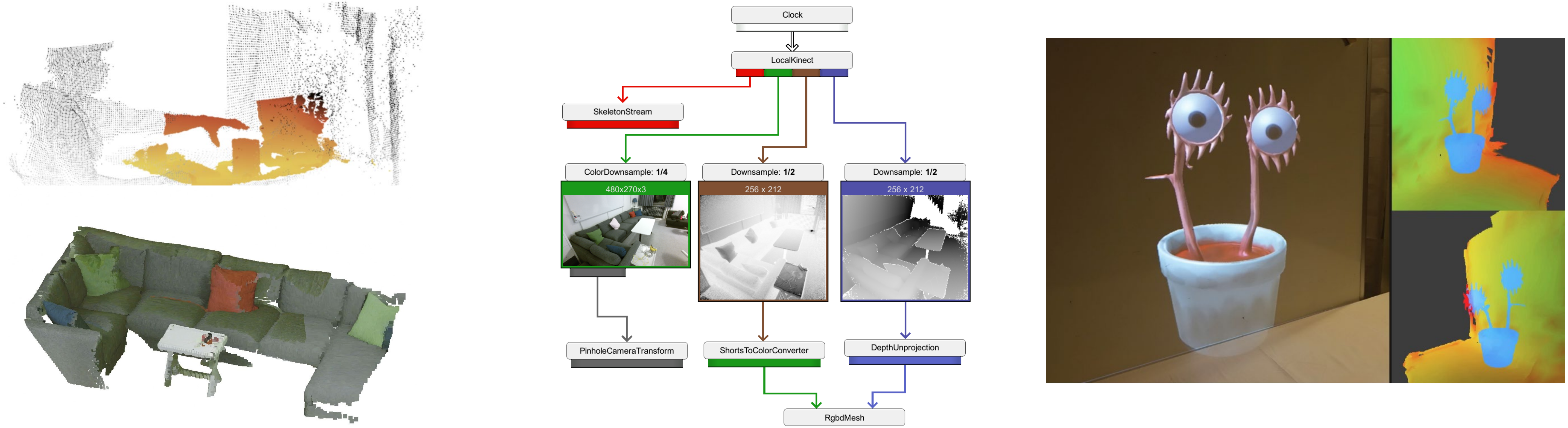
UNDERSTAND THE USER AND HIS / HER ENVIRONMENT

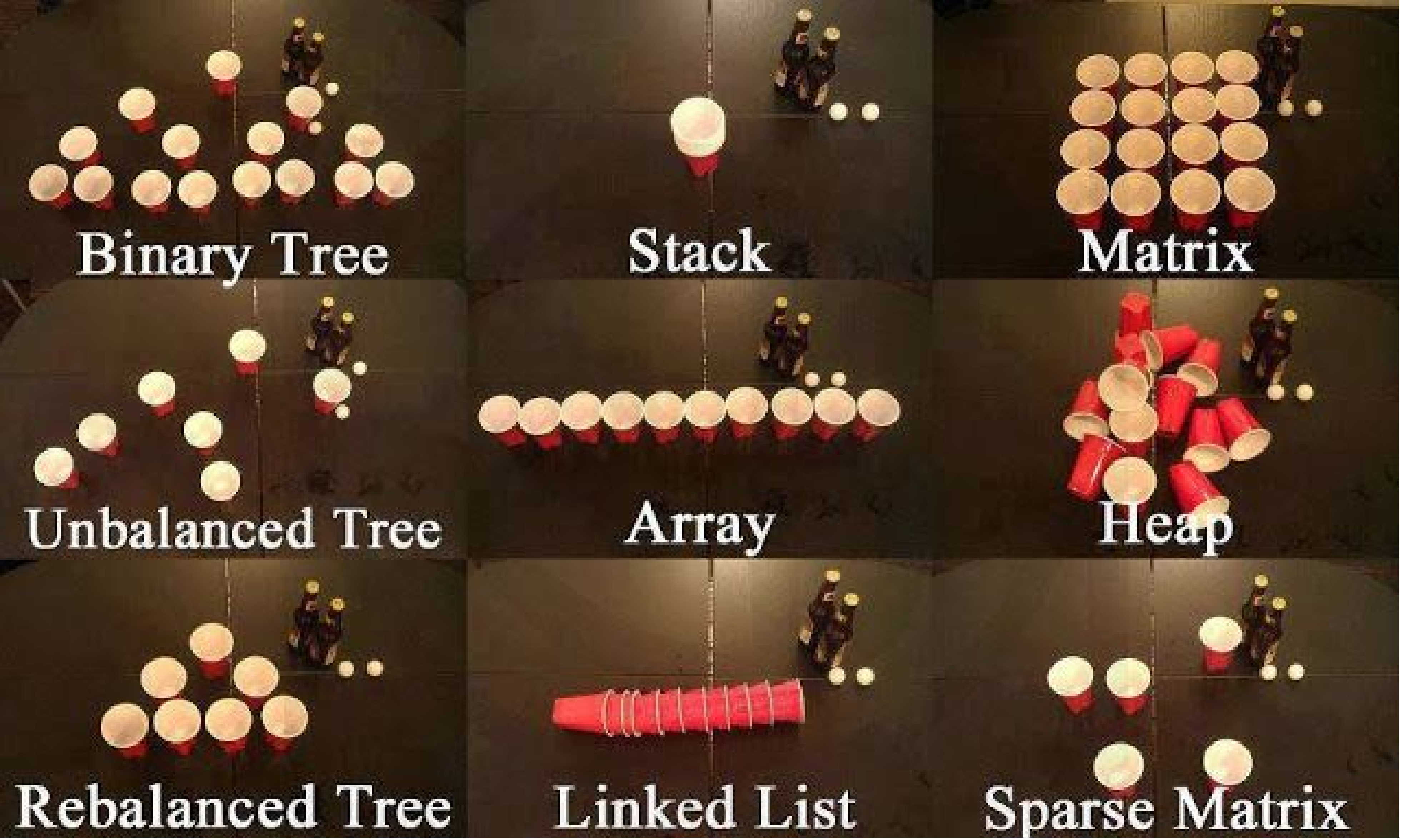
WE MAKE COOL STUFF! COLLABORATE WITH US!

Privacy-Preserving Machine Learning (PPML)

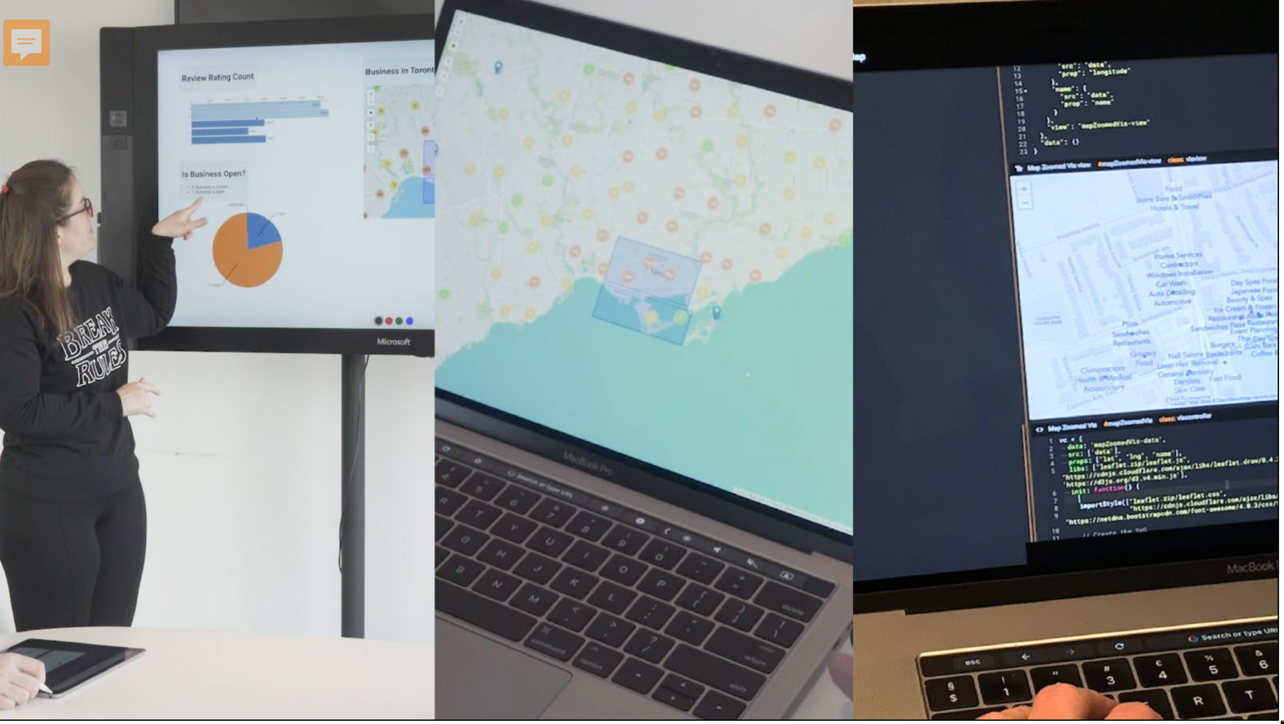
- Infeasible for consumers and companies to train complex machine learning models independently
- So machine learning as a service (MLaaS) is offered by companies such as Amazon, Google and Microsoft
- Concerns:
  - Privacy about the queries being made to the servers
  - Loss of competitive advantage due to sharing data with Amazon when performing collaborative machine learning


Room state 1: Work
Room state 2: Meeting
Room state 3: Coffee break


Clock
LocalKinect
SkeletonStream
ColorDownsample: 1/4
480x270x3
Downsample: 1/2
256 x 212
Downsample: 1/2
256 x 212
PinholeCameraTransform
ShortsToColorConverter
DepthUnprojection
RgbdMesh



Data Structures and Algorithms



A Component Model for Ubiquitous Analytics

Integrating Data-Driven Reporting in Collaborative Visual Analytics

$$\frac{(e,h,S) \to_{n,h} (e',h',S') \qquad Z' = Z[n \mapsto S'] \qquad H' = H[n \mapsto h']}{\langle n; e\rangle, (H[n\mapsto h], Z[n\mapsto S], L, P, M) \to_h \langle n; e'\rangle, (H', Z', L, P, M)}$$

$$\frac{\begin{array}{c} Z(n) = S \qquad S(z) = \mathsf{None} \qquad (ip,p) \notin \mathrm{dom}(L) \qquad p \notin P(ip) \\ Z' = Z[n \mapsto S[z \mapsto \mathsf{Some}\ a]] \qquad L' = L[a \mapsto n] \qquad P' = P[a \mapsto P(a) \cup \{p\}] \end{array}}{\langle n; \mathsf{socketbind}\ z\ a\rangle, (H,Z,L,P,M) \to_h \langle n; 0\rangle, (H,Z',L',P',M)}$$

$$\frac{Z(n)(z) = \mathsf{Some}\ \mathit{from} \qquad m = (\mathit{from}, \mathit{to}, \mathit{msg}, \textsc{Sent}) \qquad m_{id} \notin \mathit{dom}(M) \qquad M' = M[m_{id} \mapsto m]}{\langle n; \mathsf{sendto}\ z\ \mathit{msg}\ \mathit{to}\rangle, (H,Z,L,P,M) \to_h \langle n; \mathsf{length}\ \mathit{msg}\rangle, (H,Z,L,P,M')}$$

$$\frac{\begin{array}{c} Z(n)(z) = \mathsf{None} \qquad (ip,p) \notin \mathit{dom}(L) \qquad p \notin P(ip) \qquad m_{id} \notin \mathit{dom}(M) \\ m = ((ip,p), \mathit{to}, \mathit{msg}, \textsc{Sent}) \qquad M' = M[m_{id} \mapsto m] \end{array}}{\langle n; \mathsf{sendto}\ z\ \mathit{msg}\ \mathit{to}\rangle, (H,Z,L,P,M) \to_h \langle n; \mathsf{length}\ \mathit{msg}\rangle, (H,Z,L,P,M')}$$

$$\frac{\begin{array}{c} Z(n)(z) = \mathsf{Some}\ a \qquad m_{id} \mapsto m \in M \qquad \mathit{from}(m) = f \qquad \mathit{to}(m) = a \\ \mathit{msg}(m) = b \qquad \mathit{state}(m) = \textsc{Sent} \qquad m' = (f, a, b, \textsc{Received}) \qquad M' = M[m_{id} \mapsto m'] \end{array}}{\langle n; \mathsf{receivefrom}\ z\rangle, (H,Z,L,P,M) \to_h \langle n; \mathsf{Some}\ (b,f)\rangle, (H,Z,L,P,M')}$$

$$\frac{Z(n)(z) = \mathsf{Some}\ a \qquad \emptyset = \{m_{id} \mid m_{id} \mapsto (-, a, -, \textsc{Sent}) \in M\}}{\langle n; \mathsf{receivefrom}\ z\rangle, (H,Z,L,P,M) \to_h \langle n; \mathsf{None}\rangle, (H,Z,L,P,M)}$$

$$\begin{aligned}
\phi_{req}(p) &\triangleq \lambda m, \exists ps, r, sp.\ \mathit{parts}(\{p\} \cup ps) * \mathit{is_req}(\mathit{body}(m), r+1) * \\
&\qquad \mathit{from}(m) \Mapsto^{\mathsf{prot}} \phi_{coord} * p \overset{c}{\mapsto} (r+1, \textsc{wait}) * p \overset{p}{\mapsto}\{\tfrac{3}{4}\}\ (r, \textsc{init}\ sp) * P(\mathit{body}(m), p) \\
\phi_{glob}(p) &\triangleq \lambda m, \exists ga, ms, ps, r, sc, sp.\ \{\mathit{from}(m) | m \in ms\} = ps * \\
&\qquad \mathit{is_global}(\mathit{body}(m), r) * ga = \{m | m \in ms \wedge \mathit{is_abort}(m, r)\} * \\
&\qquad \mathit{parts}(ps) * \mathit{from}(m) \Mapsto^{\mathsf{prot}} \phi_{coord} * p \overset{c}{\mapsto} (r, sc) * p \overset{p}{\mapsto}\{\tfrac{3}{4}\}\ (r, sp) * \\
&\qquad \Big(\mathop{\Large *}_{m \in ms} \exists m_{id}, \pi.\ \mathit{is_vote}(\mathit{body}(ms), r) * m_{id} \overset{m}{\mapsto}\{\pi\} m \Big) * \\
&\qquad (ga = \emptyset \wedge \neg \mathit{is_abort}(\mathit{body}(m), r) \wedge sc = \textsc{commit} \vee \\
&\qquad (ga \neq \emptyset \wedge \mathit{is_abort}(\mathit{body}(m), r) \wedge sc = \textsc{abort}) \\
\phi_{part}(p) &\triangleq \lambda m, \phi_{req}(p)(m) \vee \phi_{glob}(p)(m)
\end{aligned}$$

$$P \triangleq \lambda p, m.\ \exists log, s.\ m = \text{"REQUEST\_"} @ s * p \overset{l}{\mapsto}\{\tfrac{1}{2}\}\ log * p \overset{w}{\mapsto}\{\tfrac{1}{4}\}\ log, s$$

$$Q \triangleq \lambda p, n.\ \exists log, s.\ p \overset{l}{\mapsto}\{\tfrac{1}{2}\}\ log@s * p \overset{w}{\mapsto}\{\tfrac{1}{4}\}\ log, s$$

WP-SOCKET

$$\frac{\forall z.\ z \overset{s\,[n]}{\mapsto} \mathsf{None} \mathrel{-\!\!*} \mathsf{wp}\ \langle n; z\rangle\ \{\Phi\} \qquad \triangleright \mathit{IsNode}(n)}{\mathsf{wp}\ \langle n; \mathsf{socket}\rangle\ \{\Phi\}}$$

WP-SOCKET-BIND-DYN

$$\frac{\begin{array}{c} \forall g.\ z \overset{s\,[n]}{\mapsto} \mathsf{Some}\ (ip,p) * (ip,p) \overset{r}{\mapsto} g * (ip,p) \Mapsto^{\mathsf{prot}} \phi \mathrel{-\!\!*} \mathsf{wp}\ \langle n; 0\rangle\ \{\Phi\} \\ \mathit{freePorts}(ip, \{p\}) \qquad \overset{f}{\mapsto} A \qquad (ip, a) \notin A \qquad \phi \qquad z \overset{s\,[n]}{\mapsto} \mathsf{None} \end{array}}{\mathsf{wp}\ \langle n; \mathsf{socketbind}\ z\ (ip, a)\rangle\ \{\Phi\}}$$

WP-SOCKET-BIND-STAT

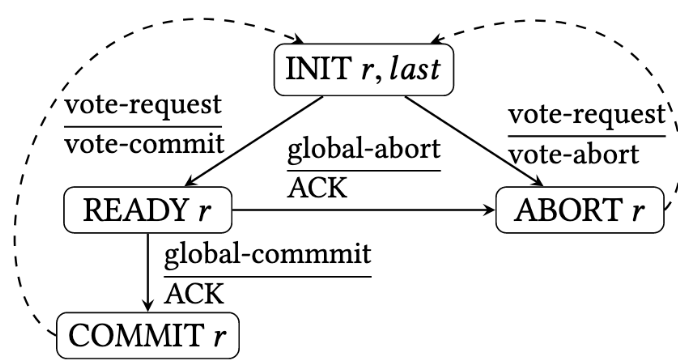
$$\frac{\begin{array}{c} \forall g.\ z \overset{s\,[n]}{\mapsto} \mathsf{Some}\ (ip,p) * (ip,p) \overset{r}{\mapsto} g* \mathrel{-\!\!*} \mathsf{wp}\ \langle n; 0\rangle\ \{\Phi\} \qquad \overset{f}{\mapsto} A \\ \mathit{freePorts}(ip, \{p\}) \qquad (ip,p) \in A \qquad (ip,p) \Mapsto^{\mathsf{prot}} \phi \qquad z \overset{s\,[n]}{\mapsto} \mathsf{None} \end{array}}{\mathsf{wp}\ \langle n; \mathsf{socketbind}\ z\ (ip, a)\rangle\ \{\Phi\}}$$

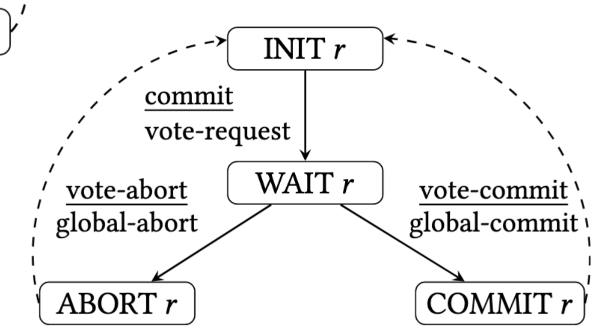
WP-SEND-TO-BOUND

$$\frac{\begin{array}{c} P \qquad z \overset{s\,[n]}{\mapsto} \mathsf{Some}\ a \qquad d \Mapsto^{\mathsf{prot}} \phi \\ \forall m_{id}, M.\ \mathit{mSoup}(M) * m_{id} \overset{st}{\mapsto} (a, d, s) * P \mathrel{\equiv\!\!*} \mathit{mSoup}(M) * \triangleright \phi(a, d, s) * Q \\ z \overset{s\,[n]}{\mapsto} \mathsf{Some}\ a * Q \mathrel{-\!\!*} \mathsf{wp}\ \langle n; \mathit{length}(s)\rangle\ \{\Phi\} \end{array}}{\mathsf{wp}\ \langle n; \mathsf{sendto}\ z\ s\ d\rangle\ \{\Phi\}}$$

WP-SEND-TO-UNBOUND

$$\frac{\begin{array}{c} P \qquad z \overset{s\,[n]}{\mapsto} \mathsf{None} \qquad d \Mapsto^{\mathsf{prot}} \phi \qquad a = (ip, p) \qquad \mathit{freePorts}(ip, \{p\}) \\ \forall p, m_{id}, M.\ \mathit{mSoup}(M) * m_{id} \overset{st}{\mapsto} (a, d, s) * P \mathrel{\equiv\!\!*} \mathit{mSoup}(M) * \triangleright \phi(a, d, s) * Q \\ z \overset{s\,[n]}{\mapsto} \mathsf{None} * Q \mathrel{-\!\!*} \mathsf{wp}\ \langle n; \mathit{length}(s)\rangle\ \{\Phi\} \end{array}}{\mathsf{wp}\ \langle n; \mathsf{sendto}\ z\ s\ d\rangle\ \{\Phi\}}$$





INCOMPLETENESS

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).

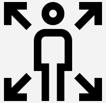
Incompleteness can be reduced but not removed.

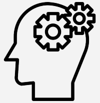
(a) Husky classified as wolf (b) Explanation

HasAsthma(x) → LowerRisk(x)

Rule predicting lower risk of death from pneumonia for asthmatics contrary to existing knowledge.

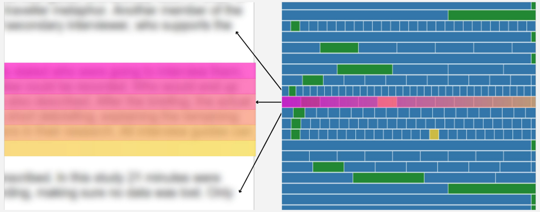
Phenomenon


Behaviors

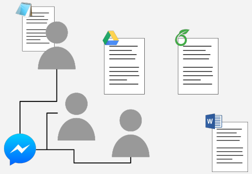

Cognitions


Sentiments

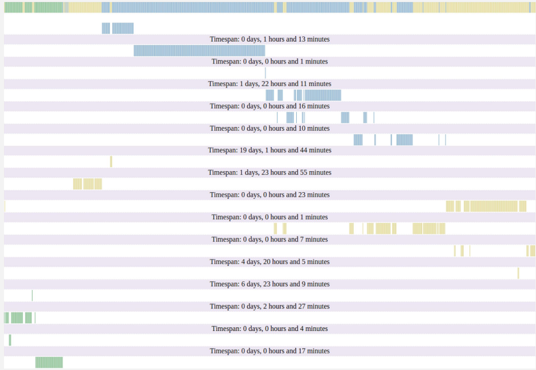
Methodology



Findings

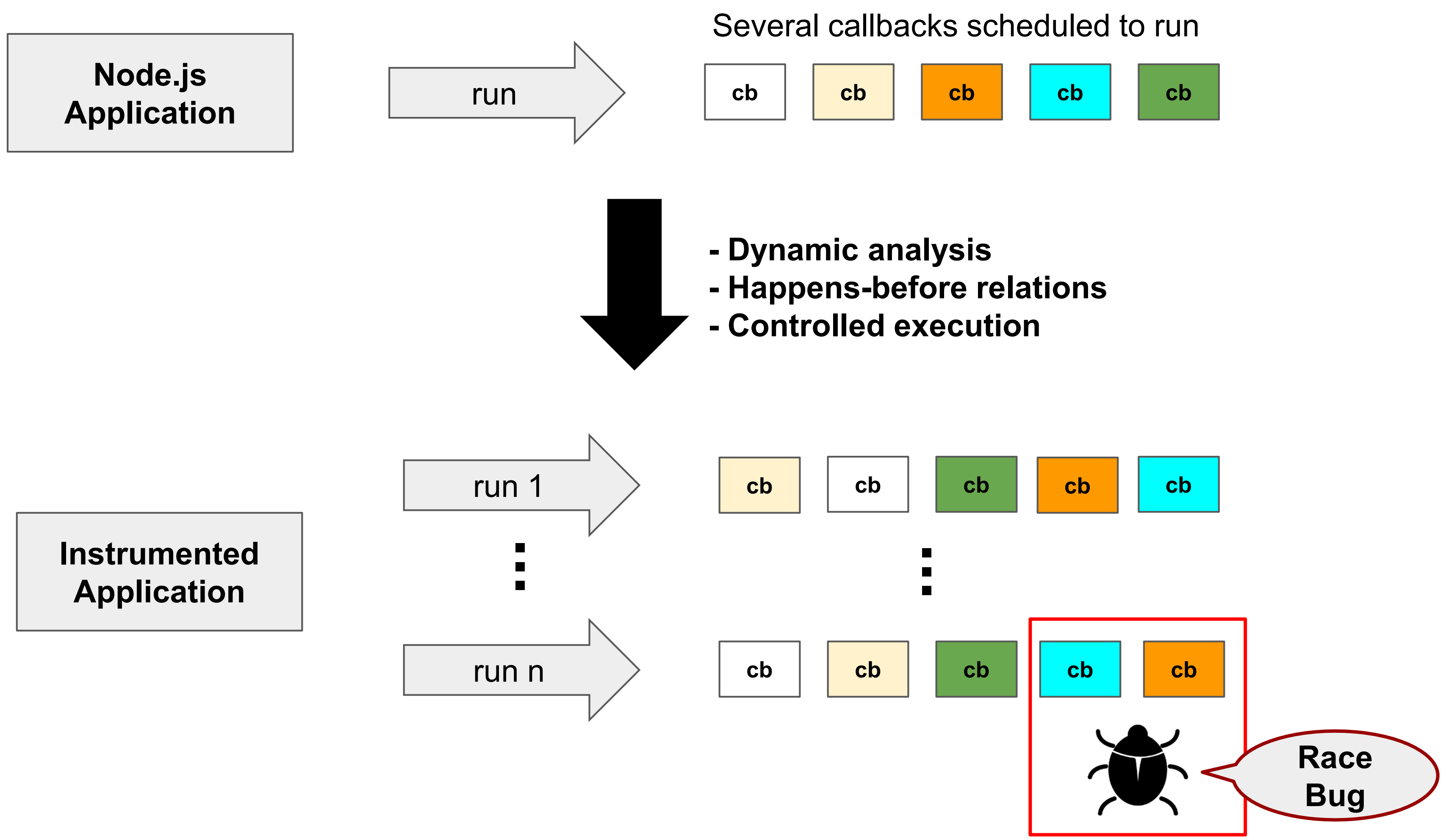


"[You] respect each other's sections so you don't go and edit them."
(Group R15)



Social and Practical Functioning during Collaborative Writing

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



Node.js Application
run
Several callbacks scheduled to run
cb
cb
cb
cb
cb
- Dynamic analysis
- Happens-before relations
- Controlled execution
Instrumented Application
run 1
run n
cb
cb
cb
cb
cb
cb
cb
cb
cb
cb
Race Bug

Daniel Gratzer — LogSem

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();
}
```

Static analyses

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;
}
```

Declarative:
Decouples behaviour from implementation

Requires invocations of virtual functions and lambdas

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.