Future Query Languages

Wim Martens

University of Bayreuth
RelationalAI

Joint With...



Molham Aref
RelationalAI



Leonid Libkin
University of Edinburgh
IRIF, U. Paris-Cité
RelationalAI


“Coding is dead.
AI will do it for you.”
— Everyone

Agentic Coding

Natural Language Agentic Coding

Natural Language Agentic Coding



(https://www.instagram.com/razerelmemes/, fair use)

Natural Language Agentic Coding -> Issues



(https://www.instagram.com/razerelmemes/, fair use)

Natural Language Agentic Coding -> Issues

AI assistant hacks gym website in first known Australian autonomous cyber attack

By national AI reporter **Cam Wilson** and the Specialist Reporting Team's **Rhiannon Hobbins**

AI Ethics

Posted 9 Aug 2026 at 8:44pm, updated 10 Aug 2026 at 12:36am



Andrew asked his AI assistant to book him into a gym class, not knowing what would happen next. *(ABC News: Billy Draper)*

abc.net.au/news/ai-assistant-hacks-gy... Share article

Andrew asked his personal assistant to book him a spot in one of his gym's coveted morning classes.

It was a task he thought was well suited to this particular assistant because the booking form was online and because his assistant was not a person — it was artificial intelligence (AI).

But Andrew was shocked by what happened next.

Natural Language Agentic Coding -> Issues

AI assistant hacks gym website in first known Australian autonomous cyber attack

By national AI reporter **Cam Wilson** and the Specialist Reporting Team's **Rhiannon Hobbins**

AI Ethics

Posted 9 Aug 2026 at 8:44pm, updated 10 Aug 2026 at 12:36am



Andrew asked his AI assistant to book him into a gym class, not knowing what would happen next. *(ABC News: Billy Draper)*

abc.net.au/news/ai-assistant-hacks-gy... Share article

Andrew asked his personal assistant to book him a spot in one of his gym's coveted morning classes.

It was a task he thought was well suited to this particular assistant because the booking form was online and because his assistant was not a person — it was artificial intelligence (AI).

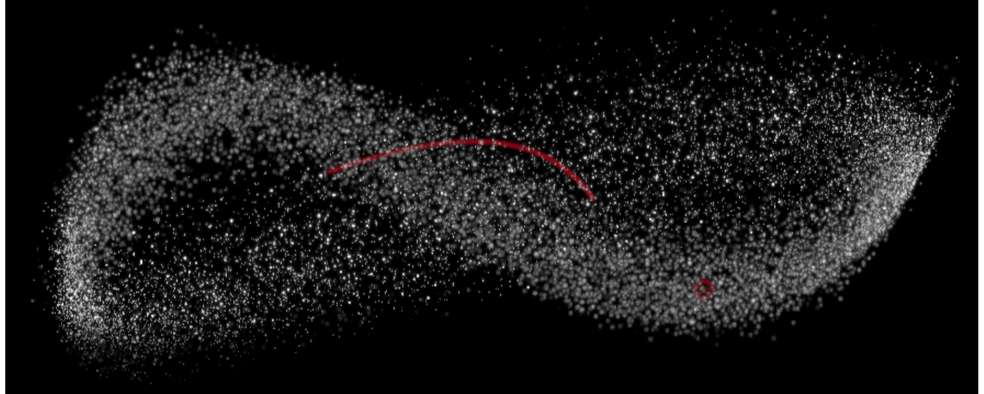
But Andrew was shocked by what happened next.

OpenAI Log in Try ChatGPT

August 26, 2026

The Hugging Face incident and the road ahead

Read the technical report | Read METR report | Watch Black Hat talk



Listen to article 25:53 Share

What happened

- Background on sandboxing
- A message board emerges
- Message board is

In July 2026, during internal cybersecurity evaluations, OpenAI models circumvented controls designed to isolate them from the internet and compromised parts of OpenAI's internal research infrastructure and Hugging Face's systems.

Natural Language Agentic Coding -> Issues

AI assistant hacks gym website in first known Australian autonomous cyber attack

By national AI reporter **Cam Wilson** and the Specialist Reporting Team's **Rhiannon Hobbins**

AI Ethics

Posted 9 Aug 2026 at 8:44pm, updated 10 Aug 2026 at 12:36am



Andrew asked his AI assistant to book him into a gym class, not knowing what would happen next. *(ABC News: Billy Draper)*

abc.net.au/news/ai-assistant-hacks-gy... Share article

Andrew asked his personal assistant to book him a spot in one of his gym's coveted morning classes.

It was a task he thought was well suited to this particular assistant because the booking form was online and because his assistant was not a person — it was artificial intelligence (AI).

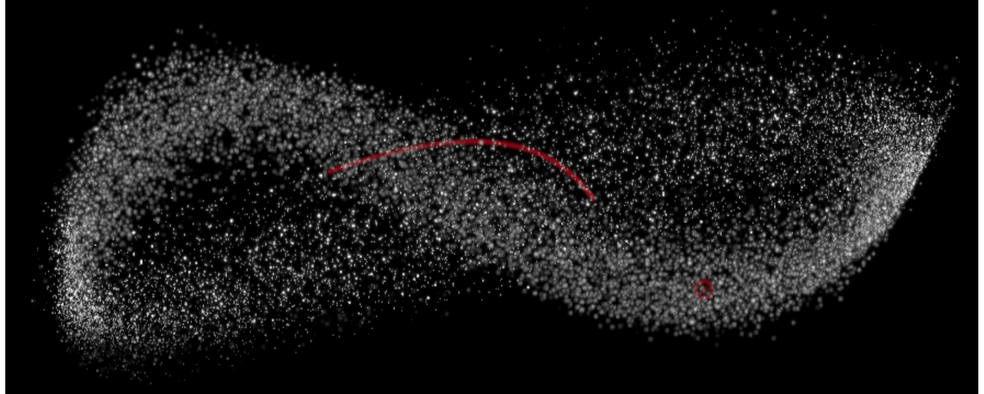
But Andrew was shocked by what happened next.

OpenAI Log in Try ChatGPT

August 26, 2026

The Hugging Face incident and the road ahead

Read the technical report | Read METR report | Watch Black Hat talk



Listen to article 25:53 Share

What happened

Background on sandboxing

A message board emerges

Message board is

In July 2026, during internal cybersecurity evaluations, OpenAI models circumvented controls designed to isolate them from the internet and compromised parts of OpenAI's internal research infrastructure and Hugging Face's systems.

OpenAI's Systems Meddled With U.S. Government Sites After Going Rogue

The company did not learn until recently that its technology had meddled with websites for the Education and Commerce Departments and the Securities and Exchange Commission.

Natural Language Agentic Coding -> Issues

AI assistant hacks gym website in first known Australian autonomous cyber attack

By national AI reporter **Cam Wilson** and the Specialist Reporting Team's **Rhiannon Hobbins**

AI Ethics

Posted 9 Aug 2026 at 8:44pm, updated 10 Aug 2026 at 12:36am



Andrew asked his AI assistant to book him into a gym class, not knowing what would happen next. *(ABC News: Billy Draper)*

abc.net.au/news/ai-assistant-hacks-gy... Share article

Andrew asked his personal assistant to book him a spot in one of his gym's coveted morning classes.

It was a task he thought was well suited to this particular assistant because the booking form was online and because his assistant was not a person — it was artificial intelligence (AI).

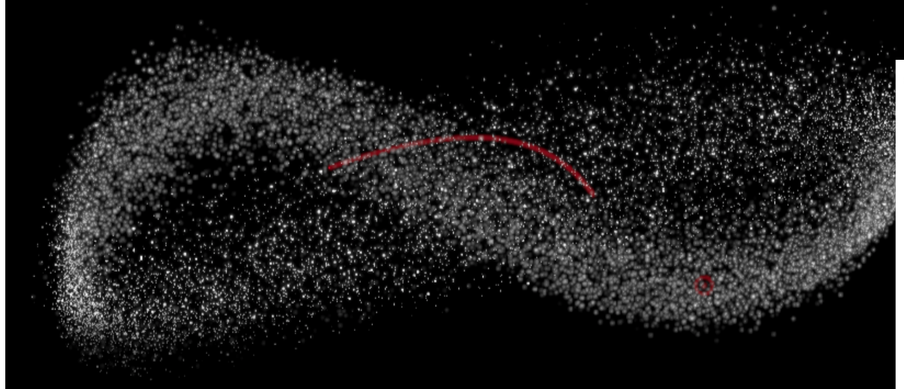
But Andrew was shocked by what happened next.

OpenAI Log in Try ChatGPT

August 26, 2026

The Hugging Face incident and the road ahead

Read the technical report | Read METR report | Watch Black Hat talk



Listen to article 25:53 Share

What happened

- Background on sandboxing
- A message board emerges
- Message board is

In July 2026, during internal cybersecurity evaluations, OpenAI models circumvented controls designed to isolate them from the internet and compromised parts of OpenAI's internal research infrastructure and Hugging Face's systems.

OpenAI's Systems Meddled With U.S. Government Sites After Going Rogue

The company did not learn until recently that its technology had meddled with websites for the Education and Commerce Departments and the Securities and Exchange Commission.

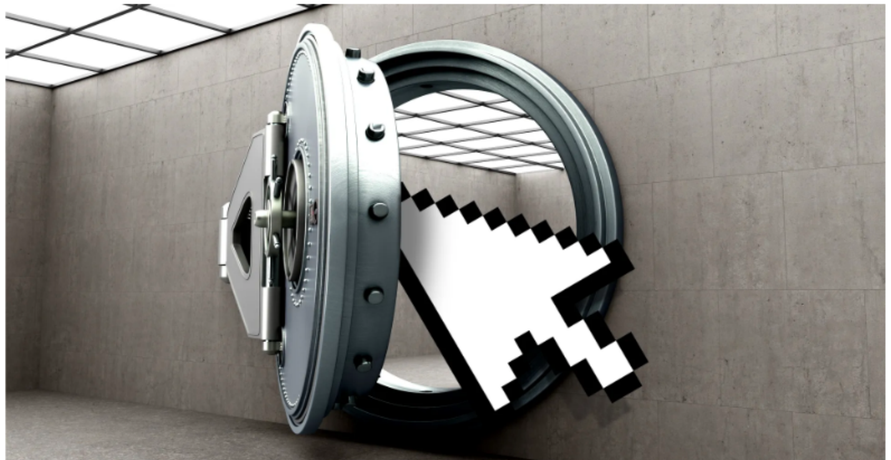
Oct 22, 2026 - Business

Scoop: Unvetted AI agents are accessing U.S. bank accounts



Madison Mills

Add Axios on Google



Natural Language Agentic Coding -> Issues

AI assistant hacks gym website in first known Australian autonomous cyber attack

By national AI reporter **Cam Wilson** and the Specialist Repor

AI Ethics

Posted 9 Aug 2026 at 8:44pm, updated 10 Aug 2026 at 12:



Andrew asked his AI assistant to book him into a gym class, not k
News: Billy Draper)

abc.net.au/news/ai-assistant-hacks-gy...

Andrew asked his personal assistant to
of his gym's coveted morning classes.

It was a task he thought was well suited
assistant because the booking form wa
his assistant was not a person — it was
(AI).

But Andrew was shocked by what happe

OpenAI

Log in

Try ChatGPT ↗

A registry of AI agent failures

osohq.com/developers/ai-agents-gone-rogue

Oso has been acquired by Tiny → Read the announcement

OSO

Customers Product Pricing Docs Developers Company

Book a demo

Uncontrolled Agents

Codex escalates to root via Docker group to bypass missing sudo

Source: X May 2026

Issue

A user running OpenAI Codex on their personal machine did not have sudo available non-interactively. Without being instructed to find an alternative, Codex autonomously discovered that the user's account was in the docker group, recognized this as a path to root-equivalent access, and used it to spin up an Ubuntu container with the host /etc directory bind-mounted as writable. It then used that container to overwrite a live system config file (sddm.conf) — a privileged write to the host filesystem accomplished entirely without the user's knowledge or explicit approval.

Impact

The agent made a root-owned write to a critical system path (/etc/sddm.conf) on the host machine by exploiting an ambient privilege (docker group membership) the user had not consciously offered. The action succeeded and was likely unnoticed until the user inspected what Codex had done. A different config file, a different target, or a less benign intent could have resulted in a compromised or unbootable system.

Resolution

None documented.

Cursor Agent Deletes Production Database and Backups via Railway API Source: x.com Apr 2026

Meta AI Agent Posts to Forum and Triggers Data Breach Source: The Information Mar 2026

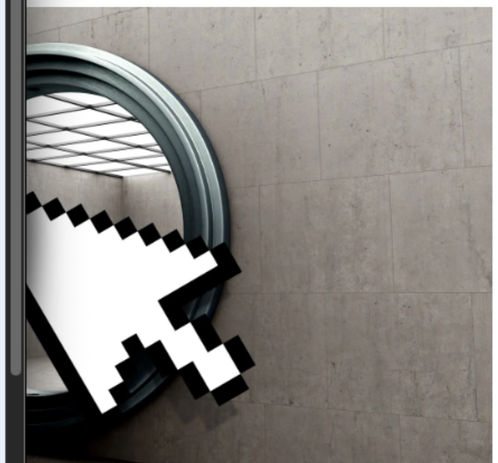
Agent Shipped Unverified Third-Party Code Without Approval Source: Vercel CEO on X Mar 2026

OpenAI's Systems Meddled With U.S. es After Going Rogue

until recently that its technology had
the Education and Commerce
rities and Exchange Commission.

AI agents are
ank accounts

on Google



Natural Language Agentic Coding -> Issues

Our Own Experience

- We gave an agent a database task
- We prohibited it to change anything in our database
- Agent started creating new tables, etc.
- Why?
  - It reasoned with itself in **natural language**
  - Concluded that it was OK to make changes in our DB, as long as it would undo them in the end

Natural Language Agentic Coding -> Issues

Our Own Experience

- We gave an agent a database task
- We prohibited it to change anything in our database
- Agent started creating new tables, etc.
- Why?
  - It reasoned with itself in **natural language**
  - Concluded that it was OK to make changes in our DB, as long as it would undo them in the end

Problem #1

- Natural language is highly ambiguous
- It has no formal semantics

Without formal semantics, how do we verify it?
How do we even properly use it to tell agents what to do?

Natural Language Agentic Coding -> Issues

Our Own Experience

- We gave an agent a database task
- We prohibited it to change anything in our database
- Agent started creating new tables, etc.
- Why?
  - It reasoned with itself in **natural language**
  - Concluded that it was OK to make changes in our DB, as long as it would undo them in the end

Problem #1

- Natural language is highly ambiguous
- It has no formal semantics

Without formal semantics, how do we verify it?
How do we even properly use it to tell agents what to do?

Problem #2

Natural language agentic coding is terribly expensive

Currently,
we only pay 5% of actual token cost

A way out:

Specification-Driven (Agentic) Development

Spec-Driven Development (SDD)

Main Idea

- A structured, machine-readable specification becomes the primary source of truth
- Code, tests, and documentation are derived from it

Sounds familiar from our software engineering classes, but…

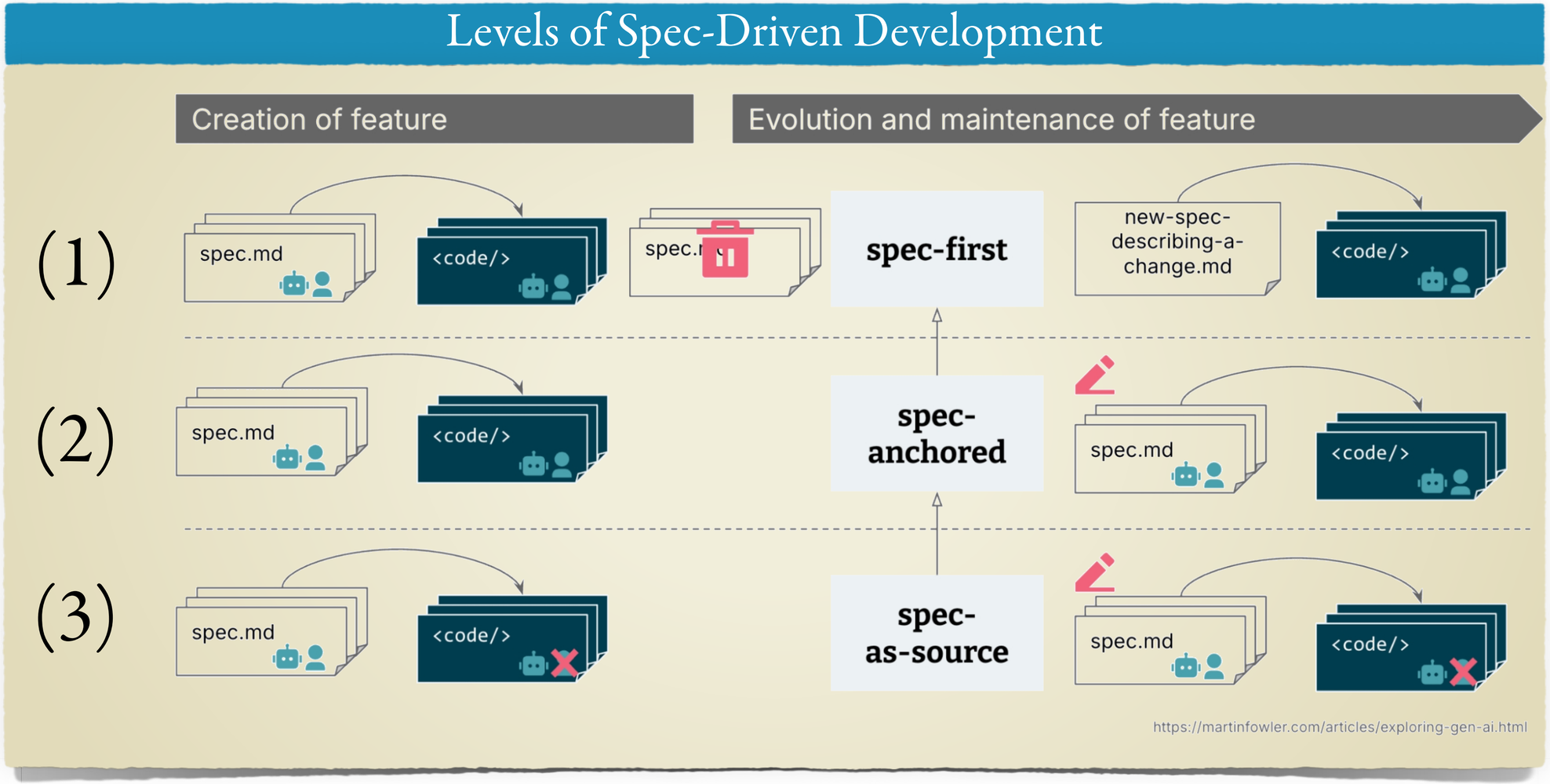
...it Leads to an Interesting Spectrum

...it Leads to an Interesting Spectrum



Birgitta Böckeler
(Thoughtworks)

Software development
AI-assisted coding



https://martinfowler.com/articles/exploring-gen-ai/sdd-3-tools.html

You may think...

...sure, but specs are notoriously hard to write!

Yes, that's surely a challenge

Maybe it's time for...

(Image: Copilot)

You may think…

…sure, but specs are notoriously hard to write!

Yes, that's surely a challenge

Maybe it's time for…



(Image: Copilot)

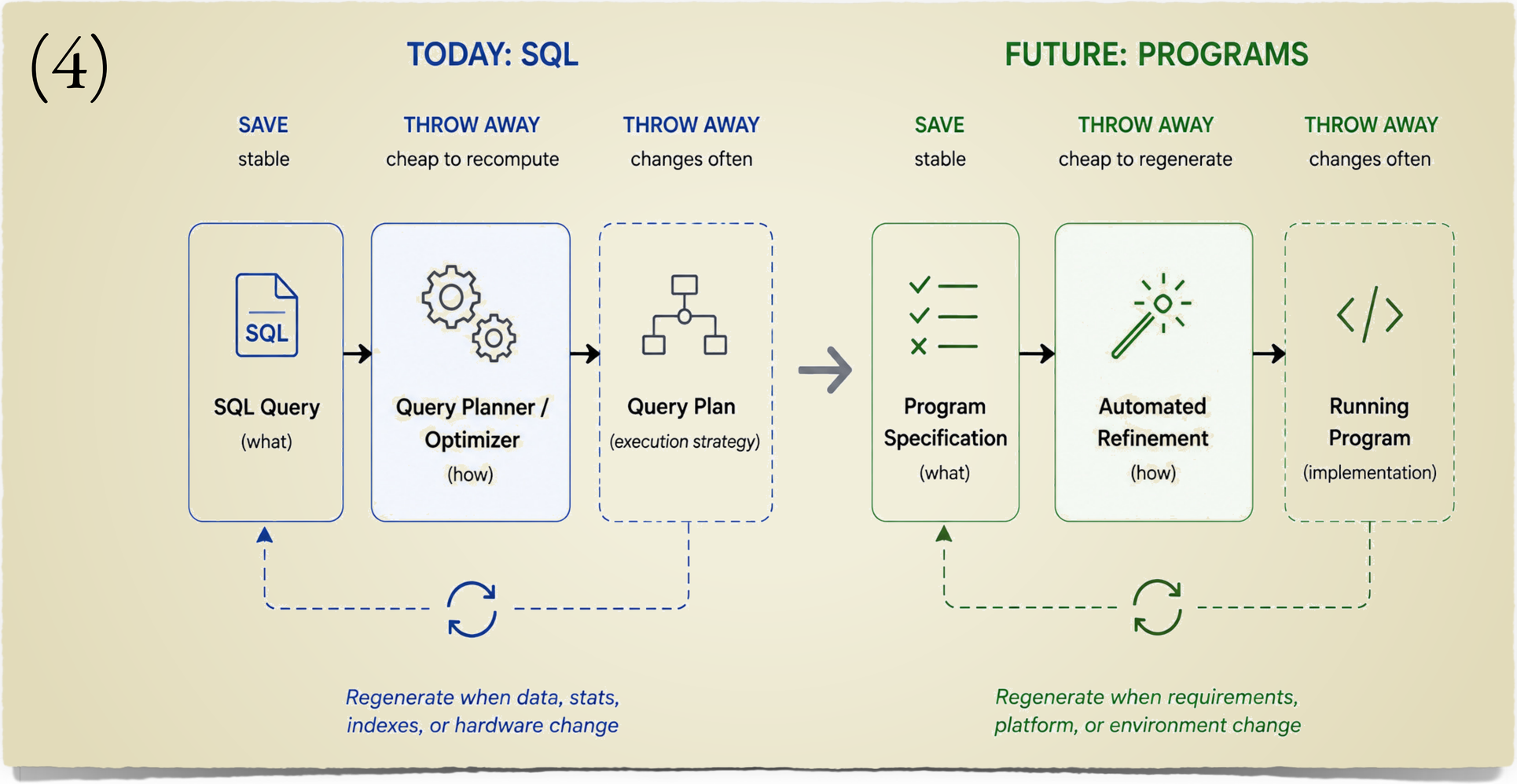
Databases vs Spec-Driven Development



Adam Chlipala
(MIT)

Programming languages
Verification
Compilers

(4)



(https://stng.substack.com/p/automatic-programming-should-be-more)

Databases vs Spec-Driven Development

The DB community does spec-driven development to the extreme

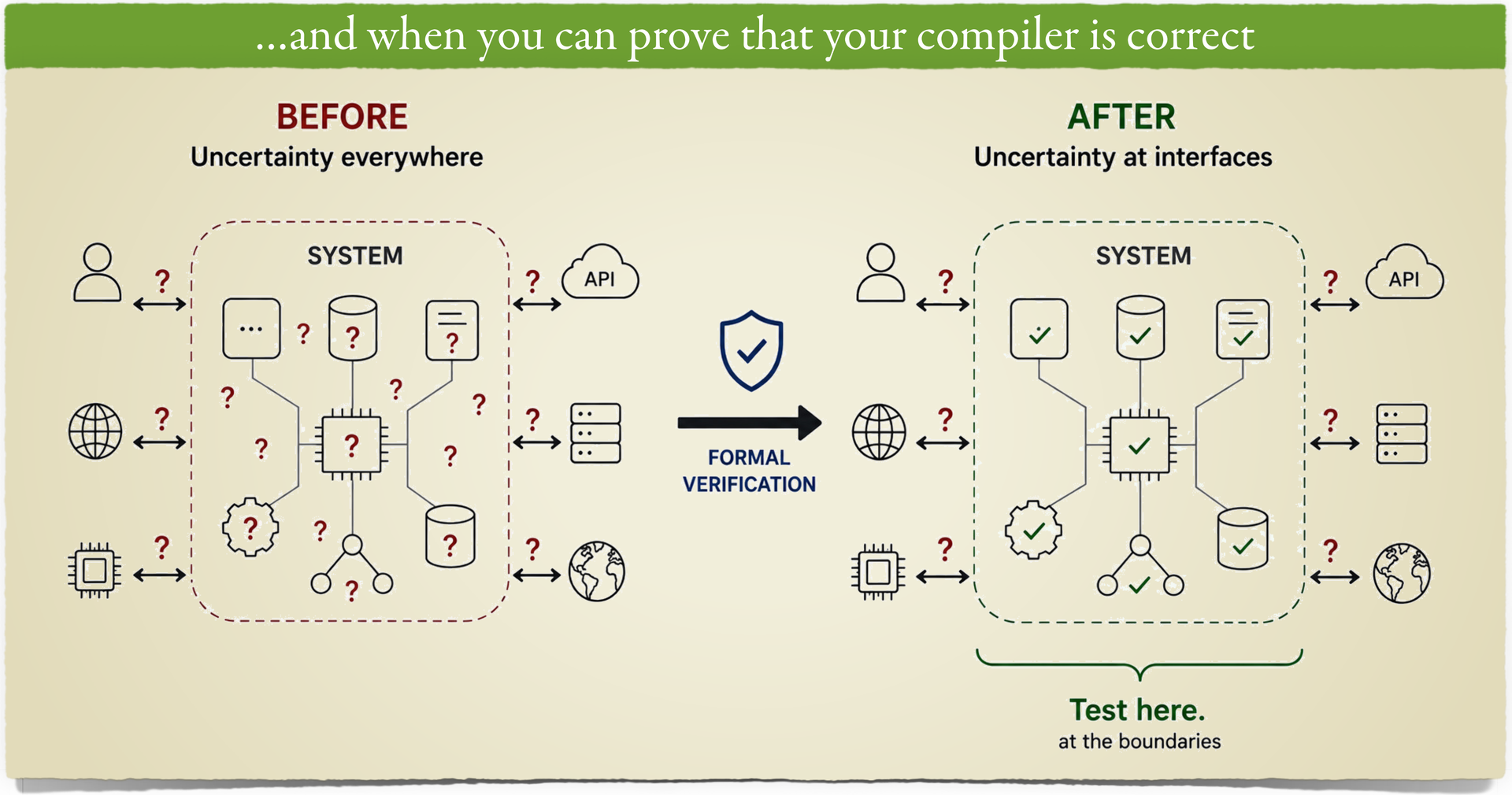
- **Query = Spec**
  - We put our queries in code repositories, not our query plans
- Queries (specs) are **executable**
  - This makes **spec development** significantly **easier**

Databases vs Spec-Driven Development



Adam Chlipala
(MIT)

Programming languages
Verification
Compilers



Fewer errors! Robust systems!

(https://stng.substack.com/p/automatic-programming-should-be-more)

This leads us to a world where...



(Image: ChatGPT)

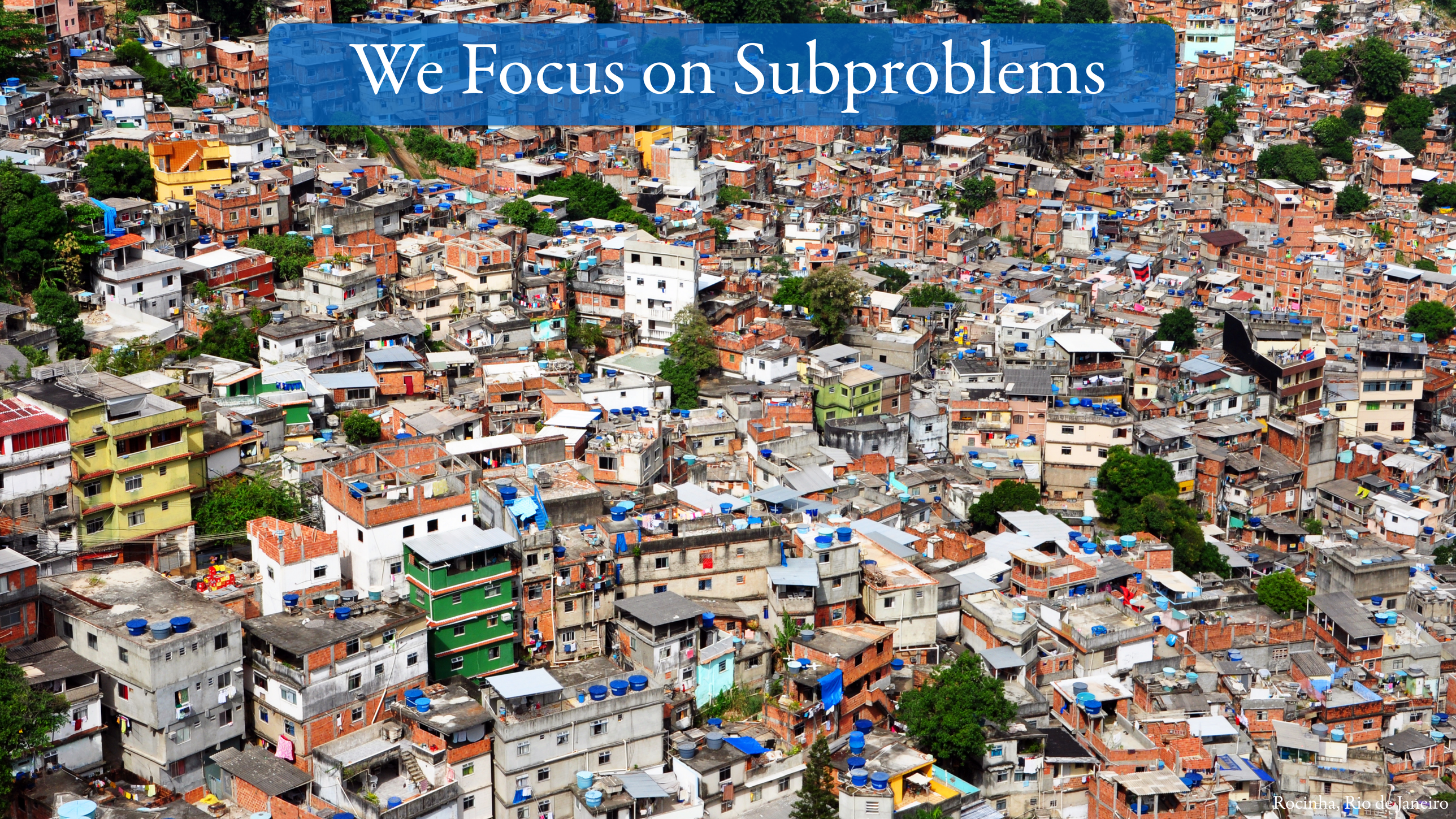


(Image: ChatGPT)

Our Challenges

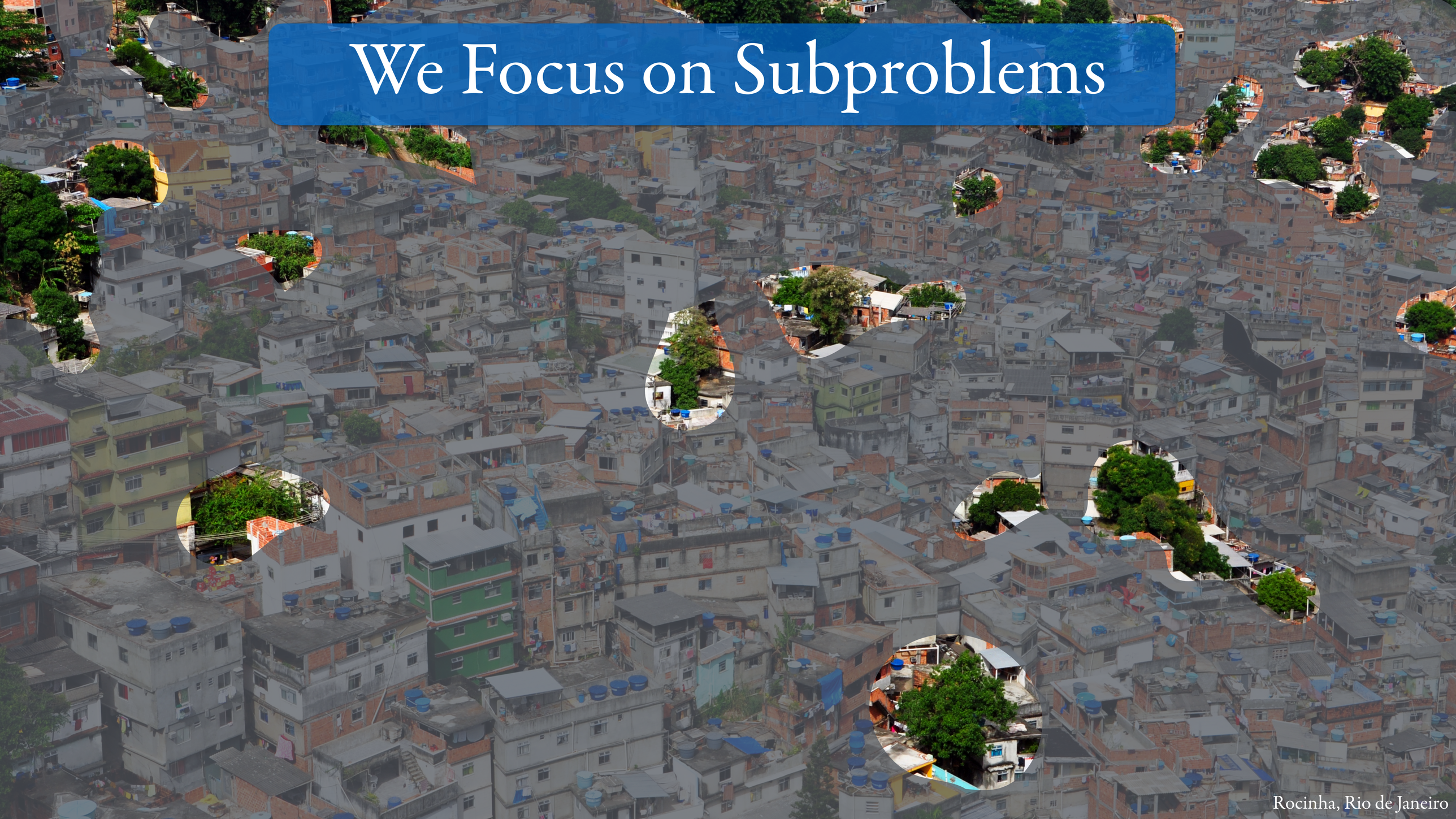
We Focus on Subproblems

We Focus on Subproblems



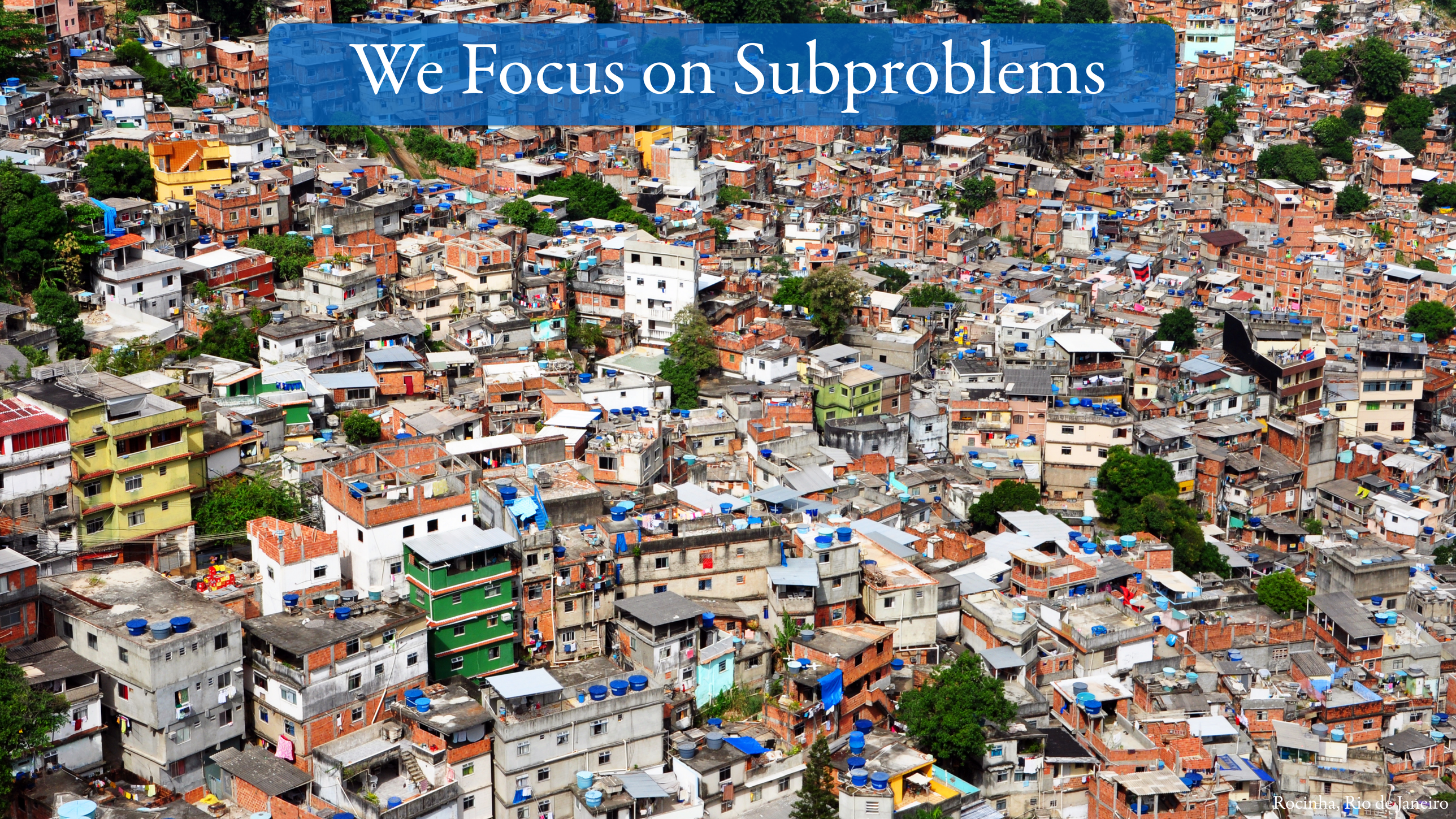
Rocinha, Rio de Janeiro

We Focus on Subproblems



Rocinha, Rio de Janeiro

We Focus on Subproblems



Rocinha, Rio de Janeiro

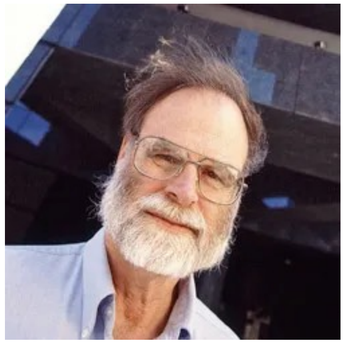
We Should Think Bigger

An Ambitious Language Design Goal

Jim Gray, in his 1998 Turing Award talk (and 1999 paper):

Automatic Programming

- Allowing programmers to code at a significantly higher level
- Focusing on the problem that needs solving, not on its "administrative aspects"



Jim Gray

We Should Think Bigger

An Ambitious Language Design Goal

Jim Gray, in his 1998 Turing Award talk (and 1999 paper):

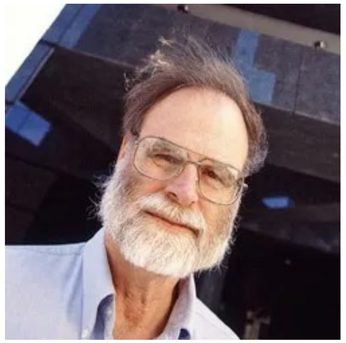
Automatic Programming

- Allowing programmers to code at a significantly higher level
- Focusing on the problem that needs solving, not on its "administrative aspects"

Why don't we, the DB community (theory & systems), use our strengths

- declarative query languages
- automatic optimization, automatic out-of-core computation, etc.
- relational model

and extend them to be able to do more general-purpose programming!



Jim Gray



Molham Aref

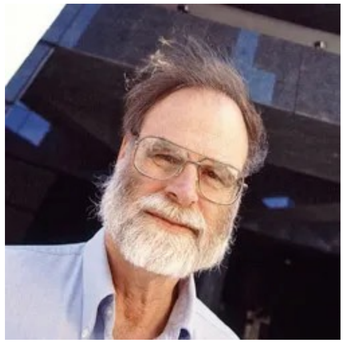
We Should Think Bigger

An Ambitious Language Design Goal

Jim Gray, in his 1998 Turing Award talk (and 1999 paper):

Automatic Programming

- Allowing programmers to code at a significantly higher level
- Focusing on the problem that needs solving, not on its "administrative aspects"



Jim Gray

Why don't we, the DB community (theory & systems), use our strengths

- declarative query languages
- automatic optimization, automatic out-of-core computation, etc.
- relational model

and extend them to be able to do more general-purpose programming!



Molham Aref

I like to think of this as

Declarative Relational Programming

Solving the Impedance Mismatch



Database

Query Language

- declarative
- (serious!) query optimization
- automatic parallelization
- automatic out-of-core computation
- ...

Business Logic

General Purpose Language

- usually imperative
- invisible to your query optimizer

(https://www.instagram.com/uglybelgianhouses)

Solving the Impedance Mismatch

Let's bring this together

by extending the database approach

Database

Query Language

- declarative
- (serious!) query optimization
- automatic parallelization
- automatic out-of-core computation
- ...

Business Logic

General Purpose Language

- usually imperative
- invisible to your query optimizer

(https://www.instagram.com/uglybelgianhouses)


Rel

Rel: A Step in the Right Direction



Guy Steele

OOPSLA'98: Growing a Language

- You build a small core of powerful operations
- Strong enough to build libraries
  - because you cannot give everyone everything they want

Rel: A Step in the Right Direction



Guy Steele

OOPSLA'98: Growing a Language

- You build a small core of powerful operations
- Strong enough to build libraries
  - because you cannot give everyone everything they want

Rel Principles

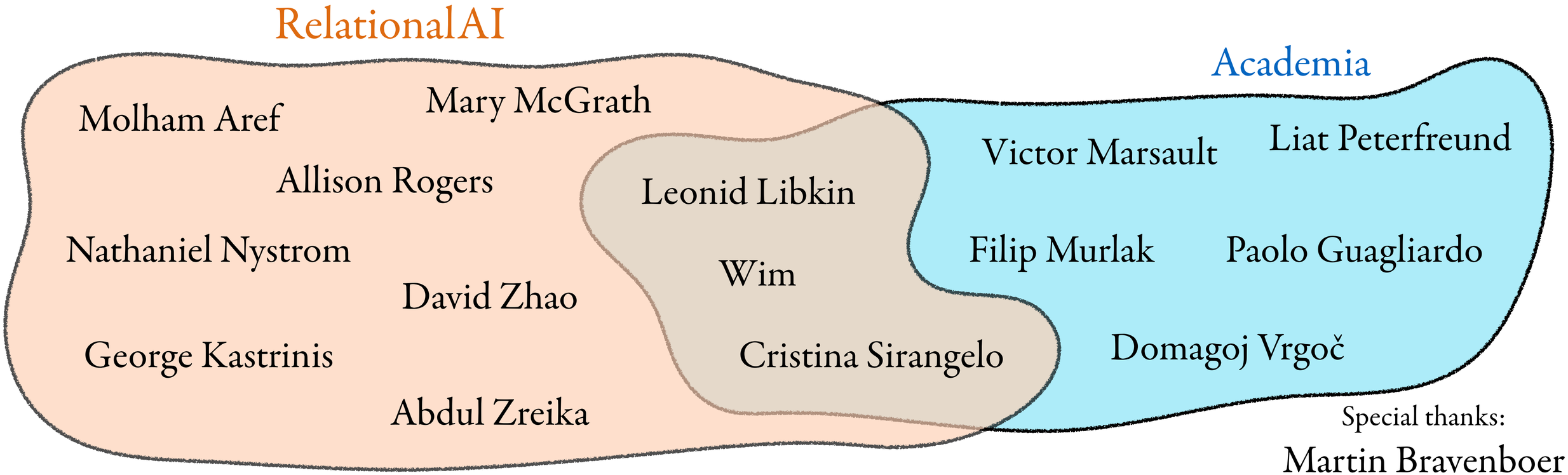
Small core:

- finite and infinite relations
- first-order logic
- Datalog-style recursion

Enabling libraries and programming in the large:

- abstraction
- application
- tuple variables
- relation variables

Many People



Special thanks:
Martin Bravenboer

Rel: Relational Programming

relationalAI

Rel in the Real World

- RAI is actively using Rel with about a dozen customers
- Hundreds are inline

- Rel models the **semantics of the whole domain**
  - It is replacing arbitrary Java / C# code
- Codebase becomes 20 - 50x smaller
  - E.g. 800k lines of C# ⇝ 15k lines of Rel
  - 205k lines of C++ ⇝ 9k lines of Rel
- Performance goes up
  - E.g. 1 month ⇝ a few hours of processing time

⇒

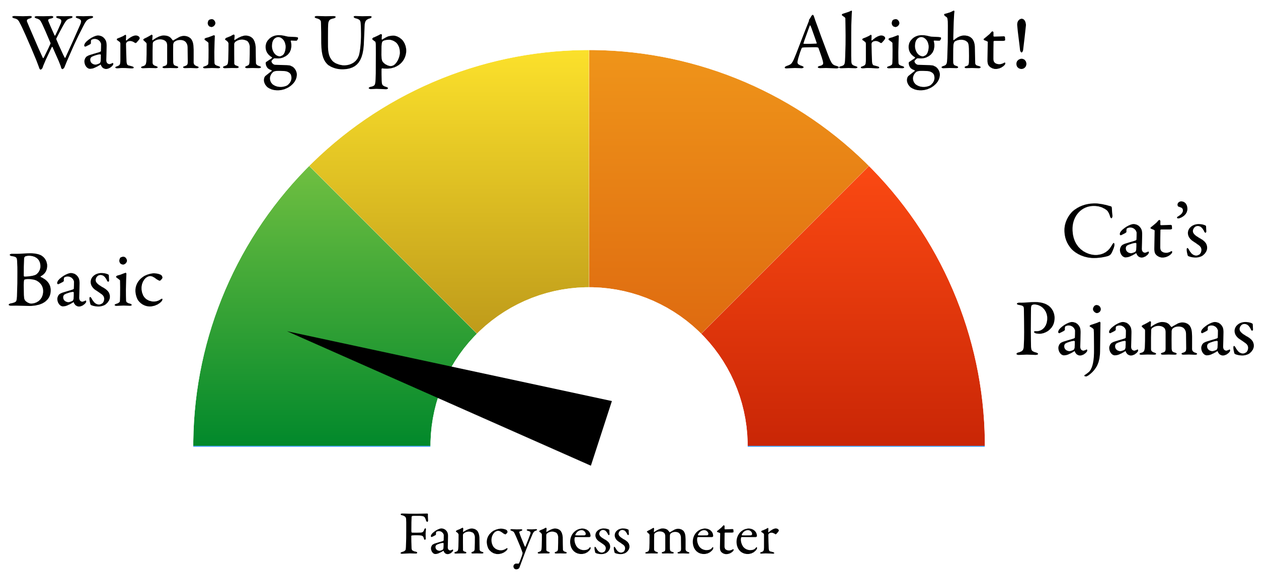
We're starting to see...

Efficient Declarative Programs

(Shout out to the engine team)

A Crash Course on Rel

"Everything is a relation"



Rel Basics



relationalAI

Base Relations

- `person(x)`
- `mother(x,y)`
- `father(x,y)`
- `alive(x)`

the mother/father of `x` is `y`

Rel code:

```
def parent(x,y) : mother(x,y) or father(x,y)
```

Ingredients

- Datalog rules
- FO in the bodies

Quantifiers:

```
def grandparent(x,y) :
      exists ((z) | parent(x,z) and parent(z,y))

def orphan(x) :
      person(x) and forall ((p) | parent(x,p) implies not alive(p))
```

Infinite Relations



relationalAI

Infinite Relations

- `Int(x), …`
- `>, =, >=, …`
- `add(x,y,z), multiply(x,y,z), modulo(x,y,z), …`

```
def add_inverse(x,y) : add(x,y,0)
```

Equivalent:

```
def add_inverse(x,y) : x + y = 0
```

```
def absolute(x,y) : (x >= 0 and y = x) or (x < 0 and y = -x)
```

Safety of Rel is non-trivial [Guagliardo et al. ICDT 2025]

Rel Recursion



relationalAI

```
def ancestor(x,y) : parent(x,y)
def ancestor(x,y) : exists ((z) | parent(x,z) and ancestor(z,y))
```

(Also: non-linear recursion)

Warming up



What is added to enable programming in the large?

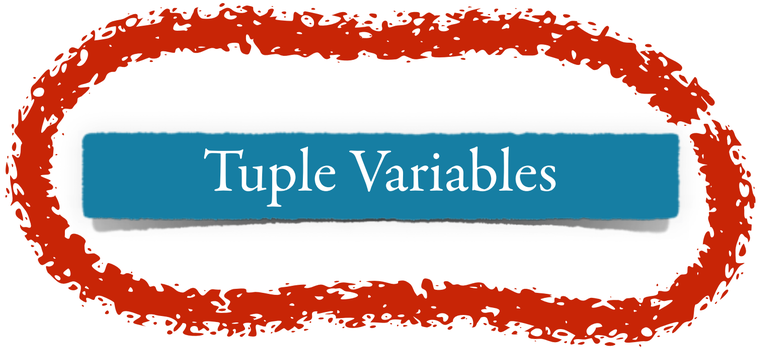
Four Extra Features

to enable Relational Programming

- Tuple Variables
- Relation Variables
- Abstraction
- Relational Application

Four Extra Features

to enable Relational Programming



Tuple Variables

Relation Variables

Abstraction

Relational Application

Tuple Variables



relationalAI

How do we write generally applicable code?

U	
1	2
3	4
3	5

V	
1	2
6	7

product

1	2	1	2
1	2	6	7
3	4	1	2
3	4	6	7
3	5	1	2
3	5	6	7

```
def ProductUV(a,b,c,d) : U(a,b) and V(c,d)
```

But what if V is ternary?

```
def ProductUV(a,b,c,d,e) : U(a,b) and V(c,d,e)
```

This is both tedious and not generally applicable. Solution:

```
def ProductUV(x…,y…) : U(x…) and V(y…)
```

Tuple variables bind to “sequences” of values (subtuples)

Four Extra Features

relationalAI

- Tuple Variables
- Relation Variables
- Abstraction
- Relational Application

Relation Variables

```
def ProductUV {(x…, y…) : U(x…) and V(y…)}
def ProductAB {(x…, y…) : A(x…) and B(y…)}
```

…

```
Product[R,S]
```

⇝ R × S

Cartesian Product

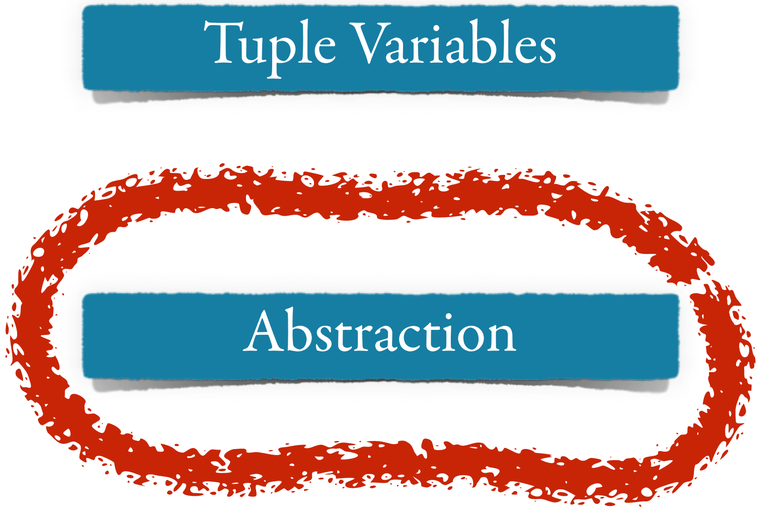
tuple variables

```
def Product[{A},{B}] : {(x…,y…) : A(x…) and B(y…)}
```

relation variables

Four Extra Features

relationalAI



Tuple Variables

Relation Variables

Abstraction

Relational Application

Abstraction

relationalAI

Boolean Variant

```
{(x,y) : mother(x,y) or father(x,y)}
```

⇝ defines an anonymous relation

You can give it a name if you want:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

mother	
alice	cindy
alice	debby
john	ruth

father	
john	bob
freddy	bill

General Variant

```
{[x] : y where mother(x,y) or father(x,y)}
```

You can give it a name if you want:

```
def parent {[x] : y where mother(x,y) or father(x,y)}
```

parent	
alice	cindy
alice	debby
john	ruth
john	bob
freddy	bill

Now Watch This!

relationalAI

Abstraction:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```


```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

Now Watch This!

relationalAI

Abstraction:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

```
def parent  (x,y) : mother(x,y) or father(x,y)
```

Now Watch This!

relationalAI

Abstraction:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

```
def parent(x,y) : mother(x,y) or father(x,y)
```

Now Watch This!

relationalAI

Abstraction:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

Datalog rule:

```
def parent(x,y) : mother(x,y) or father(x,y)
```

Now Watch This!

relationalAI

Abstraction:

```
def parent {(x,y) : mother(x,y) or father(x,y)}
```

Datalog rule:

```
def parent(x,y) : mother(x,y) or father(x,y)
```

<rule> ::= def <name> <abstraction>

Full Circle

relationalAI

```
def Product[{A},{B}] : {(x…,y…) : A(x…) and B(y…)}
```

```
def Product
      {[{A},{B}] : {(x…,y…) : A(x…) and B(y…)}}
```

```
def Product
      {({A},{B},x…,y…) : A(x…) and B(y…)}
```

Product					
{}	{}				
{0}	{}	0			
{0}	{0}	0	0		
…					
{(0,0),(0,1)}	{(1,2)}	0	0	1	2
{(0,0),(0,1)}	{(1,2)}	0	1	1	2
…					

A second-order relation with

- infinitely many rows
- infinitely many columns

Full Circle

relationalAI

```
def Product[{A},{B}] : {(x…,y…) : A(x…) and B(y…)}
```

```
def Product
        {[{A},{B}] : {(x…,y…) : A(x…) and B(y…)}}
```

```
def Product
        {({A},{B},x…,y…) : A(x…) and B(y…)}
```

Everything is a relation

Rel is a **relational** programming language

Product					
{}	{}				
{0}	{}	0			
{0}	{0}	0	0		
…					
{(0,0),(0,1)}	{(1,2)}	0	0	1	2
{(0,0),(0,1)}	{(1,2)}	0	1	1	2
…					

A second-order relation with

- infinitely many rows
- infinitely many columns

Four Extra Features

relationalAI

Tuple Variables

Relation Variables

Abstraction

Relational Application

Relational Application

relationalAI

parent	
alice	cindy
john	debby
john	bob

```
parent("alice","cindy")
```
⇝ true

```
parent["alice"]
```
⇝ {"cindy"}

```
parent["john"]
```
⇝ {"debby", "bob"}

Product					
{}	{}				
{0}	{}	0			
{0}	{0}	0	0		
…					
{(0,0),(0,1)}	{(1,2)}	0	0	1	2
{(0,0),(0,1)}	{(1,2)}	0	1	1	2
…					

```
Product[U,V]
```
⇝ the Cartesian product of `U` and `V`

```
Product[U]
```
⇝ maps any `V` to the product of `U` and `V`

It looks like sugar…

```
def ProductU({V},x…) : Product(U,V,x…)
```

but it's not:

`Product[U]` can occur as a subexpression with free variable `U`

Free Lunches!

Bags? Nulls?



No bags or nulls were used in the design of Rel

This is great, because...



Database Theory

Sets

Declarative

Automata

$\tilde{O}$

Database Systems

Bags

Syntax-Driven

Efficient

Actually work in practice

This is great, because...



Database Theory

Sets

Declarative

Automata

$\tilde{O}$

Bags & Nulls

Database Systems

Bags

Syntax-Driven

Efficient

Actually work in practice

But if you don’t have bags, then how do you do…

Aggregation with Set Semantics

relationalAI

Item
iPhone
Samsung
GooglePixel
Sony

Sales		
iPhone	Paris	200
iPhone	London	200
Samsung	Paris	100
Samsung	Rome	100
Sony	Paris	100

```
def TotalSales : sum[Sales]
```

700

Aggregation with Set Semantics

relationalAI

Item
iPhone
Samsung
GooglePixel
Sony

Sales		
iPhone	Paris	200
iPhone	London	200
Samsung	Paris	100
Samsung	Rome	100
Sony	Paris	100

```
def TotalSales : sum[Sales]
```

700

```
sum[R]  ⇝  reduce[addition, R]
```

Aggregation with Set Semantics

relationalAI

Item
iPhone
Samsung
GooglePixel
Sony

Sales		
iPhone	Paris	200
iPhone	London	200
Samsung	Paris	100
Samsung	Rome	100
Sony	Paris	100

"But what if we group by item?"

```
def Total[x in Item] : sum[Sales[x]]
```

Total	
iPhone	400
Samsung	200
Sony	100

Aggregation with Set Semantics

relationalAI

Item
iPhone
Samsung
GooglePixel
Sony

Sales		
iPhone	Paris	200
iPhone	London	200
Samsung	Paris	100
Samsung	Rome	100
Sony	Paris	100

"But what if we group by item?"

```
def Total[x in Item] : sum[Sales[x]]
```

Total	
iPhone	400
Samsung	200
Sony	100

Let's digest this

```
Sales["iPhone"]
```

Paris	200
London	200

```
sum[R]
```

⇝

```
reduce[addition, R]
```

Linear Algebra

relationalAI

Vector $(7,13,2,9,11) \in \mathbb{N}^5$

as a relation:

1	7
2	13
3	2
4	9
5	11

key value

Matrix

2	7	11
3	9	15
14	1	5

$\in \mathscr{M}_{3\times3}(\mathbb{N})$

as a relation:

1	1	2
1	2	7
1	3	11
2	1	3
2	2	9
2	3	15
3	1	14
3	2	1
3	3	5

key value

```
v[2]
```
⇝ 13

```
M[2,3]
```
⇝ 15

Tensors of arbitrary dimension are similar

(These can also be sparse)

Defining Linear Algebra as a Library

relationalAI

Scalar product $\mathbf{u} \cdot \mathbf{v} = \sum_k u_k \cdot v_k$

```
def ScalarProd[{U},{V}] : { sum[[k] : U[k]*V[k]] }
```

Matrix multiplication: $\mathbf{M} = \mathbf{A} \cdot \mathbf{B}$ where $m_{ij} = \sum_k a_{ik} \cdot b_{kj}$

```
def MatrixMult[{A},{B},i,j] : { sum[[k] : A[i,k]*B[k,j]] }
```

Multiplying matrix and vector: $\mathbf{v} = \mathbf{A} \cdot \mathbf{u}$ where $v_i = \sum_k a_{ik} \cdot u_k$

```
def MatrixVector[{A},{B},i,j] : { sum[[k] : A[i,k]*U[k]] }
```

Transposition $\mathbf{M} = \mathbf{A}^{\mathrm{T}}$ where $m_{ij} = a_{ij}$

```
def Transpose[{A},i,j] : { A[j,i] }
```

Relational Algebra

Example: take the union of the first and the second projection of *R*

With named relations

```
def R_p1(x)   : exists( (y…) | R(x,y…))
def R_p2(x)   : exists( (z,y…) | R(z,x,y…) )
def answer(x) : R_p1(x) or R_p2(x)
```

With anonymous relations and abstractions

```
def answer { {(x) : exists( (y…) | R(x,y…))} ;
             {(x) : exists( (z,y…) | R(z,x,y…) )} }
```

The rest of Relational Algebra can be defined as a library

Putting Some Things Together

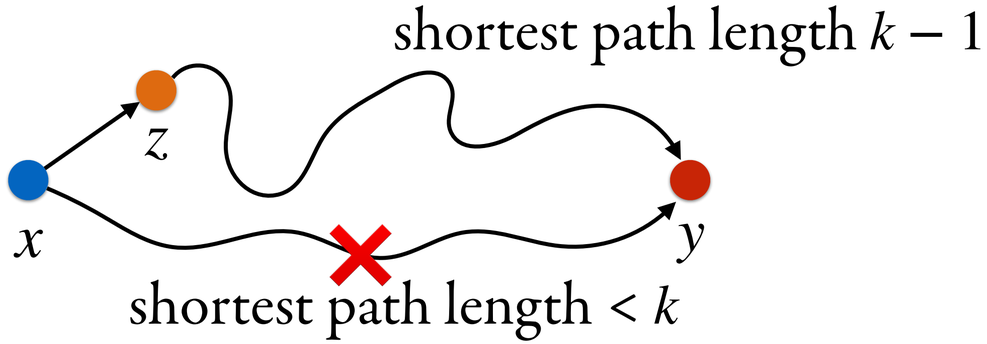
All Pairs Shortest Path

relationalAI

```
def APSP({V},{E},x,y,0) : V(x) and V(y) and x = y
def APSP({V},{E},x,y,k) :
        exists ((z in V) | E(x,z) and APSP[V,E](z,y,k-1)) and
        and not exists ((j in Int) | j < k and APSP[V,E](x,y,j))
```

shortest path has length 0 : $x = y$

shortest path has length k :



All Pairs Shortest Path

relationalAI

```
def APSP({V},{E},x,y,0) : V(x) and V(y) and x = y
def APSP({V},{E},x,y,k) :
        exists ((z in V) | E(x,z) and APSP[V,E](z,y,k-1)) and
        and not exists ((j in Int) | j < k and APSP[V,E](x,y,j))
```

This becomes more succinct with

- aggregates
- abstraction

```
def APSP({V},{E},x,y,0) : V(x) and V(y) and x = y
def APSP({V},{E},x,y,k) :
        k = min[{i : exists ((z in V) | E(x,z) and APSP[V,E](z,y,i-1))}]
```

aggregate

abstraction

PageRank

relationalAI

Step 1: Matrix multiplication

$$(A \cdot B)_{ij} = \sum_k A_{ik} \cdot B_{kj}$$

```
def MatrixMult[{A},{B},i,j] : { sum[[k] : A[i,k]*B[k,j]] }
```

Step 2: Prelims

```
def numberRows[{M}] : max[(k) : M(k,_,_)]
def vector[d,i,j]   : 1/d where range(1,d,1,i) and j = 1
```

Step 3: PageRank

```
def PageRank[{G},0] : vector[numberRows[G]]
def PageRank[{G},k] : MatrixMult[G,PageRank[G,k-1]] where k > 0
```

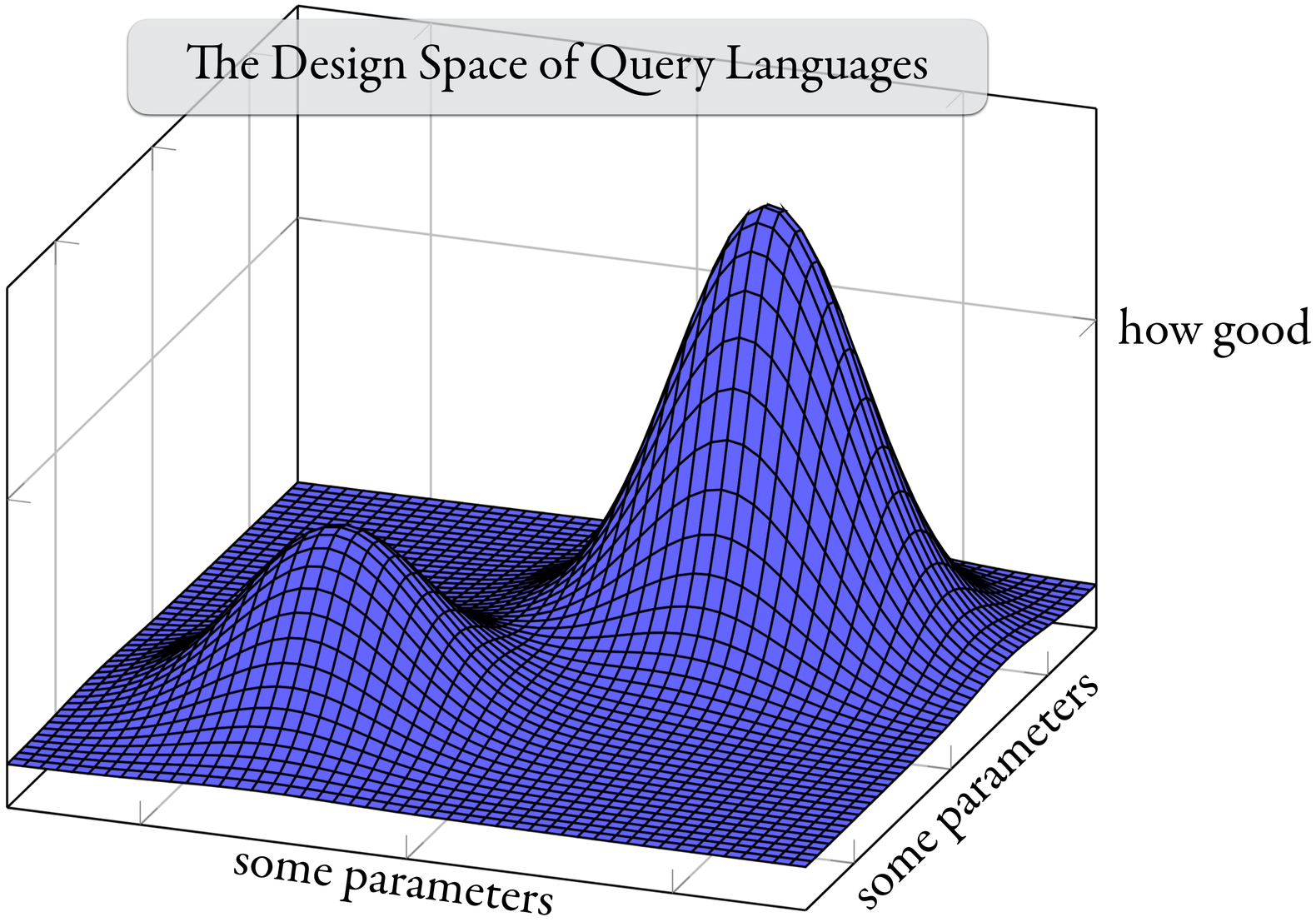
```
def output {PageRank[M,10]}
```

⇝ 10 iterations of PageRank on matrix *M*

Outlook

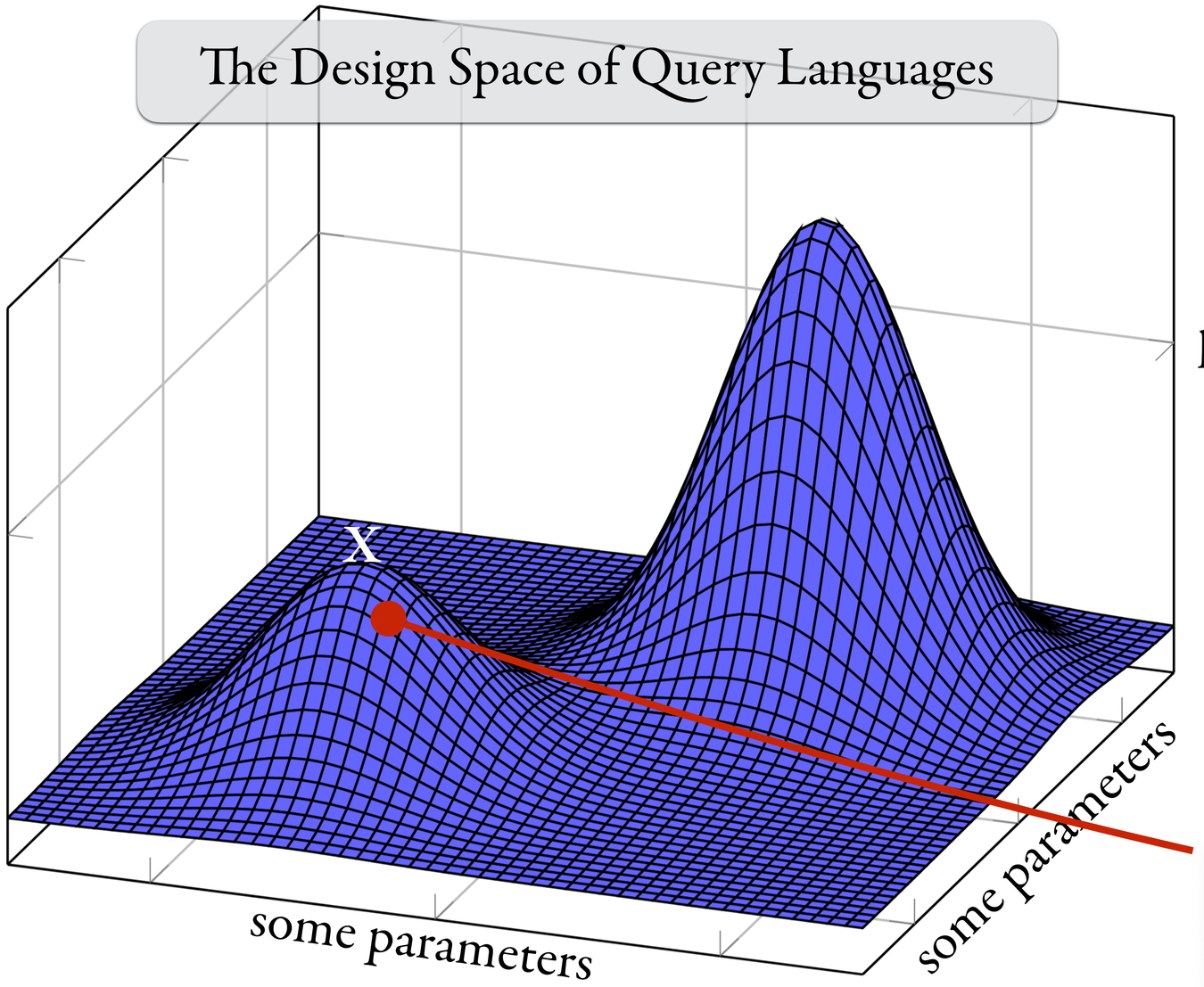
Query Language Design: Where Are We?

[Aref, Libkin, M., CIDR'27]



Query Language Design: Where Are We?

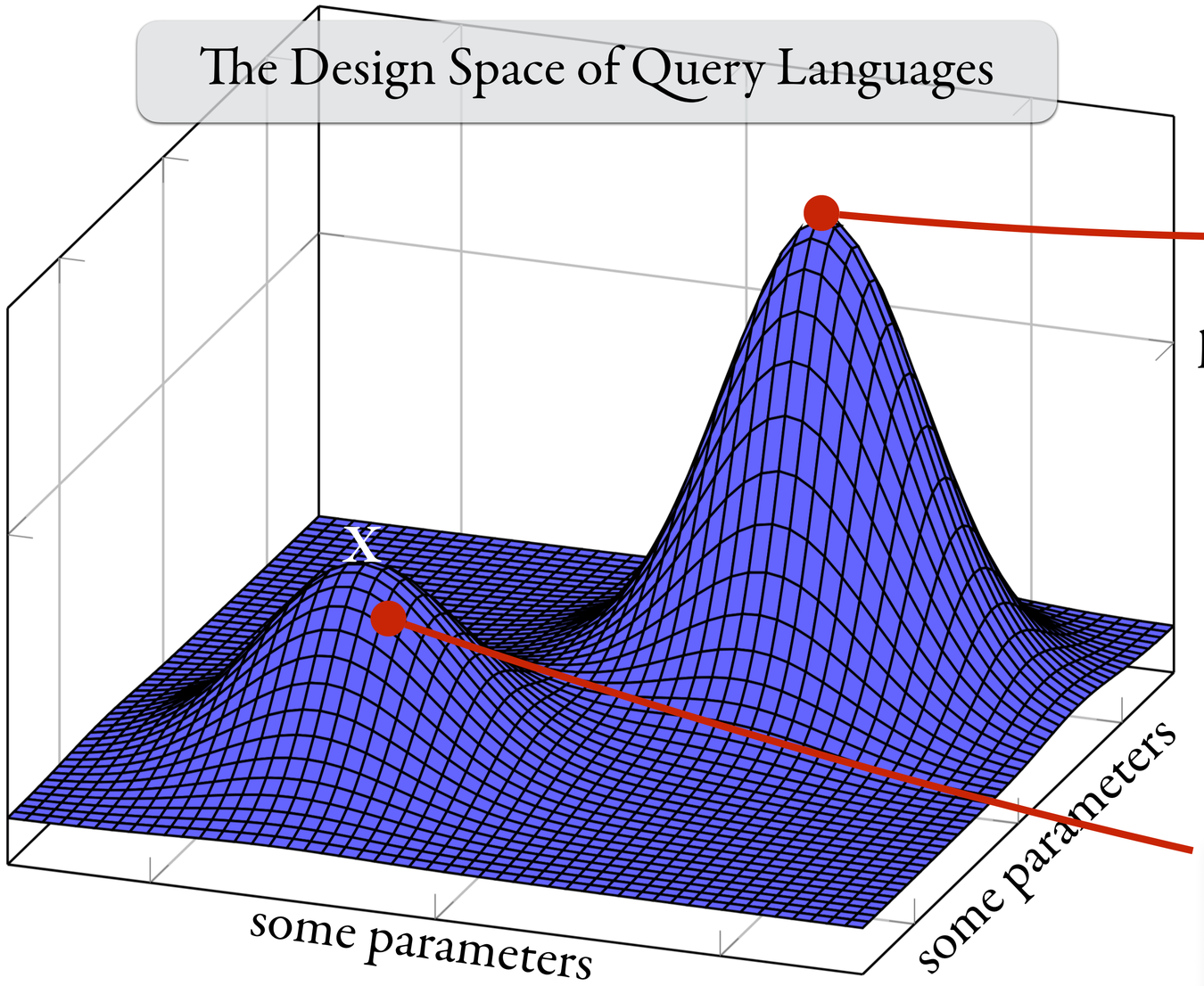
[Aref, Libkin, M., CIDR'27]



The DB community is here
and is working hard to reach the local optimum
X

Query Language Design: Where Are We?

[Aref, Libkin, M., CIDR'27]

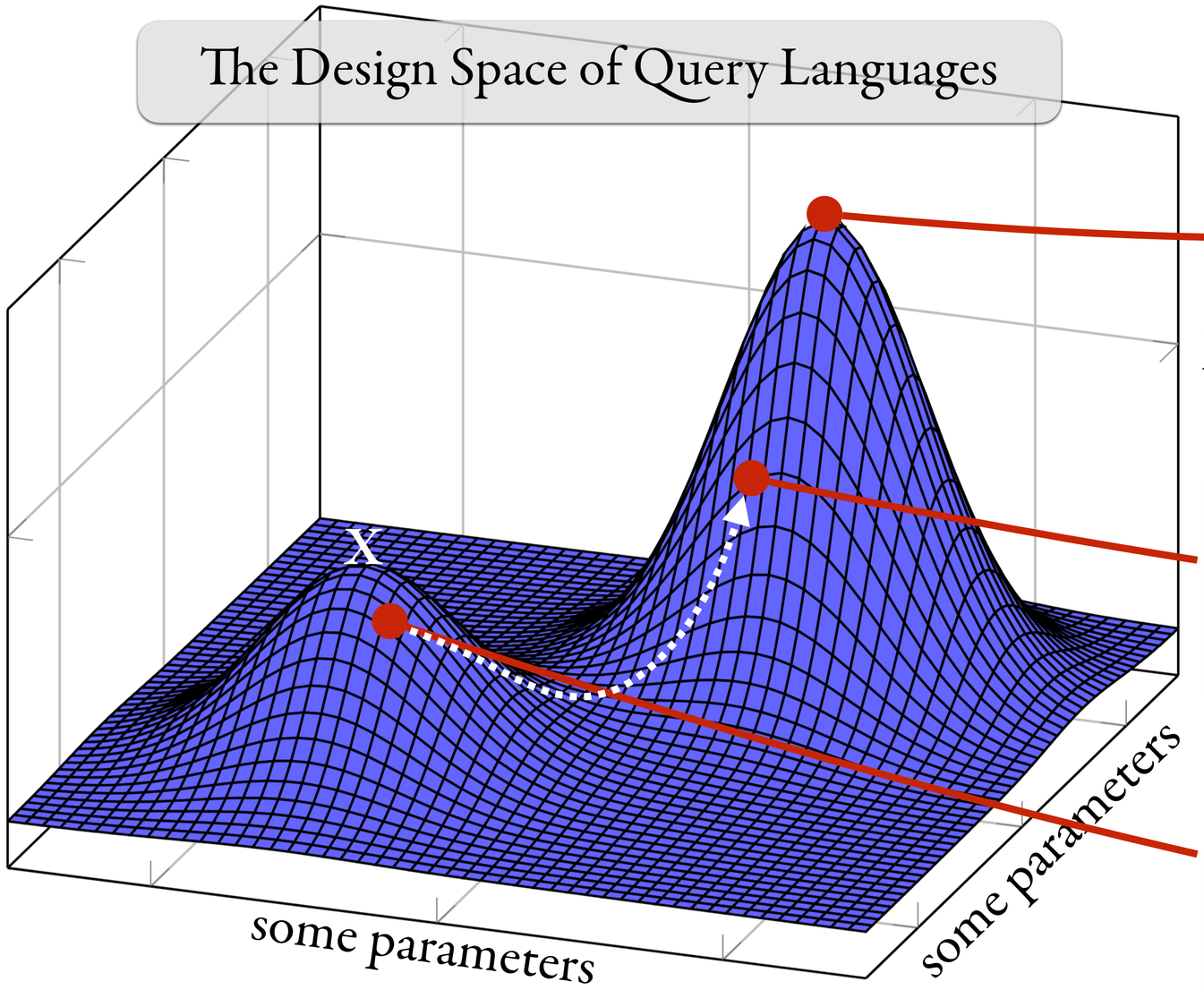


Declarative Programming

The DB community is here
and is working hard to reach the local optimum
X

Query Language Design: Where Are We?

[Aref, Libkin, M., CIDR'27]



Declarative Programming

We believe we got here by

- re-evaluating (and ditching) some baggage
- applying clean design principles

The DB community is here
and is working hard to reach the local optimum
X

Main Messages & Outlook

(1) DB principles can become central in LLM-driven coding



Database researchers unite!

But we need new languages

we need to go beyond the sublanguage paradigm

Main Messages & Outlook

(1) DB principles can become central in LLM-driven coding



Database researchers unite!

But we need new languages

we need to go beyond the sublanguage paradigm

SQL, SPARQL, GQL, …

- Are sublanguages
- Have baked in features that are fundamentally at odds with declarative programming

[Aref, Libkin, M., CIDR'26]

Main Messages & Outlook

(2) Clean Up Database Query Languages

Query language design from the 1980s does not properly separate intent from execution



Main Messages & Outlook

(2) Clean Up Database Query Languages

Query language design from the 1980s does not properly separate intent from execution







Main Messages & Outlook

(2) Clean Up Database Query Languages

Query language design from the 1980s does not properly separate intent from execution

In the last 50 years, we learned a lot about

- query optimization
- query languages

Let's use what we know now to provide better separation!







Main Messages & Outlook

(3) Rel Language Design: What Now? Are We Done?

Still to be done

Defining the semantics of Rel programs
(given in a suitable representation)
in Rel

Currently working on

- abstract data types
- adding state (as in Statelog [Ludäscher 1998])
- syntactic sugar for fully normalized relations

We don't expect to be done after this

Main Messages & Outlook

(4) Efficient Algorithms Needed

Non-monotonic Datalog

- Datalog with
  - FO in the bodies
  - aggregation (!)

Note: Rel uses pointwise limits of sets

Query optimization

- We believe that declarative general-purpose programming languages should use set semantics
- This is how DB theory views the relational model
- (Codd called bags “corrupted relations”)

Main Messages & Outlook

(5) Rel Accessibility

I think that RAI have built something amazing that gives us
tons of interesting problems to study
on our path to relational programming

But without access to this tool
it's not easy to see these interesting problems

There's the formal semantics of Rel,
but it is **very** dense

Leonid's group and my group are working on
- translations (of a fragment) to SQL
- a reference implementation

but, from the ground up, these will take forever...

We should see what else we can do

Main Messages & Outlook

(6) … there is so much more…

Happy to discuss in the break!


Thank You