Collaboration Opportunities with the School of Computer Science at St Andrews

Ian Miguel (ijm@st-Andrews.ac.uk)

View of St Andews Castle Ruins – Image from Wikipedia

St Andrews

- Best known as the “home of golf”,
- and as being the home of the University of St Andrews.

St Andrews, Charles Connections

- Connections:
  - Educational contact between Charles and St Andrews was first established in the **Middle Ages**, when the writings of Laurence of Lindores were widely read in Prague.
  - In the early twentieth century, St Andrews received a quincentenary scroll from Charles University, and research contact was ongoing.
  - In the 1930s and 1940s, both universities hosted astronomer Erwin Finlay-Freundlich when he was forced to leave Germany.
  - Assisted by her husband, St Andrews graduate Willa Muir also made the first English translations of the work of Charles University graduate Franz Kafka.
  - Building on friendly relations and research links across all faculties, Charles University and St Andrews signed a **strategic partnership agreement** at the end of 2019.

Charles Visit to St Andrews, March 2025

Collaborative Opportunities

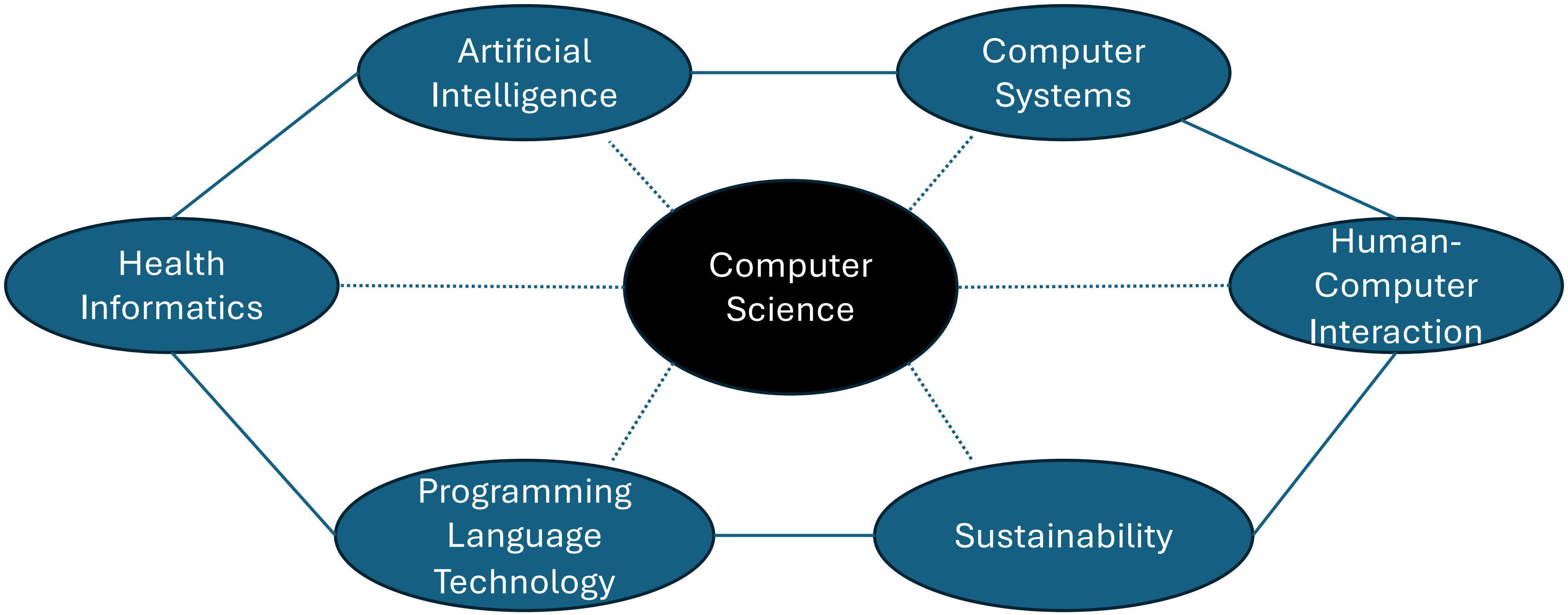
- There are undergraduate and postgraduate student exchanges.
  - At present limited to International Relations, Comparative Literature and History.
- There is a joint **seed funding** scheme.
  - 12-month projects, up to 5K.
  - Annual call, deadline in May.
- St Andrews has a **Global Fellowship** scheme, which we can use to host you.
  - Visits of up to 4 weeks.
  - Flights, accommodation, stipend.
- The UK has associated to **Horizon**.
- My visit here paid for by a Global Partnerships Travel Grant.

The School of Computer Science at St Andrews

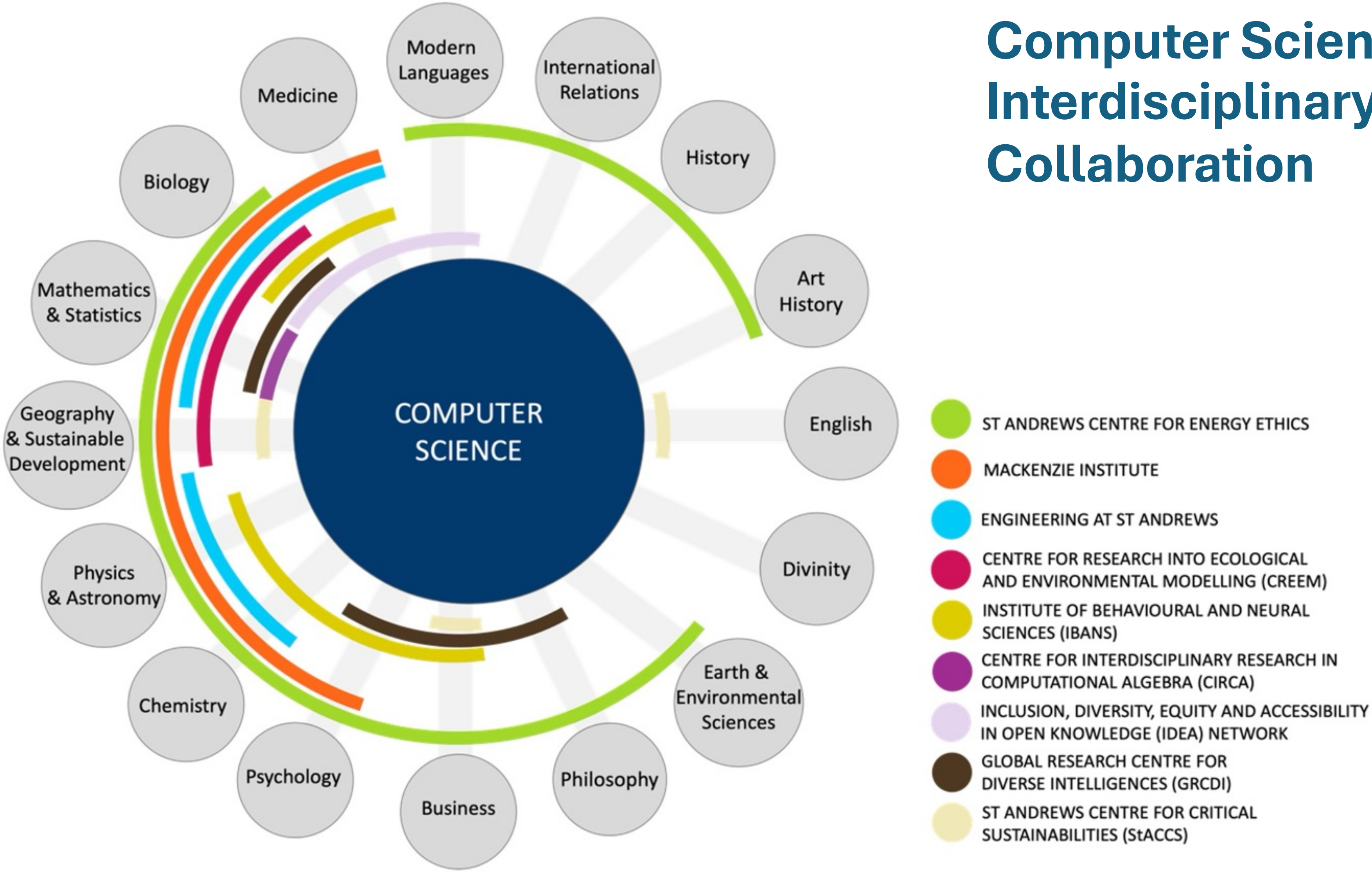
- ~70 academic, research and professional services staff.
- ~55 PGR students.
- ~140 Physical PGT students.
- ~100 Online PGT students.
- ~700 Undergraduate students.

Research Themes at Computer Science St Andrews

- Our research themes cut across the major areas of the discipline and underpin interdisciplinary work.



Computer Science Interdisciplinary Collaboration



University-wide interdisciplinary initiatives and Schools with which the School of Computer Science closely collaborates

Artificial Intelligence

Theme Lead: Ruth Hoffmann (rh347@st-andrews.ac.uk)

Artificial Intelligence: Overview

- Both **symbolic** and **sub-symbolic** AI represented:
  - Constraint Programming.
  - AI Planning.
  - Algorithm Selection.
  - Argumentation.
  - Natural Language Processing.
  - Computational Algebra.
  - Machine Learning.
  - Machine Vision.
- Good deal of overlap with **Health Informatics** theme described later.

Constraint Programming

- See Part II of this talk.

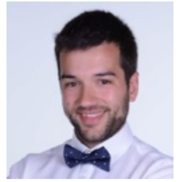
Nguyen Dang
nttd@st-andrews.ac.uk

Ozgur Akgun
Ozgur.akgun@st-andrews.ac.uk

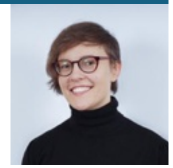
Ian Miguel
ijm@st-andrews.ac.uk

Andras Salamon
Andras.Salamon@st-andrews.ac.uk

Ian Gent
Ian.Gent@st-andrews.ac.uk



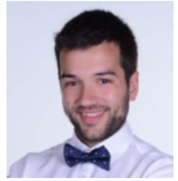
Joan Espasa
jea20@st-andrews.ac.uk



Ruth Hoffmann
rh347@st-andrews.ac.uk

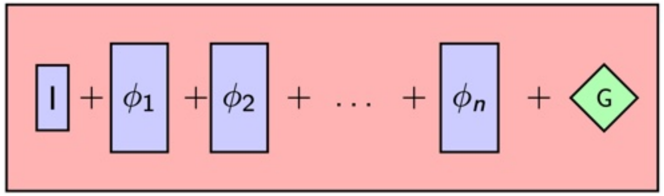
Chris Jefferson
caj21@st-andrews.ac.uk

Planning & Scheduling via SAT/SMT/CP



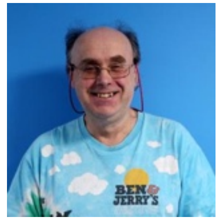
Joan Espasa
jea20@st-andrews.ac.uk

- AI Planning
  - Understanding **modelling** challenges and extending language capabilities
  - Much work on improving scalability of **planning as satisfiability**
  - **Solution sets** for robustness/diversity
  - **Tailoring** off-the-shelf combinatorial solvers for planning



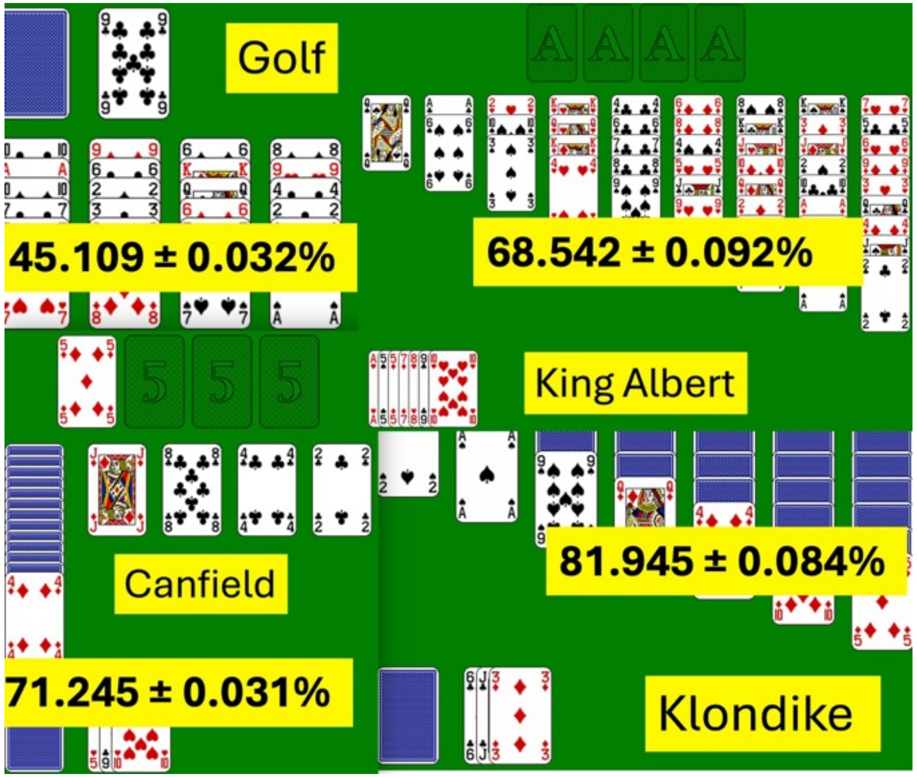
- Planning & Games:
  - Player support systems
  - General Modelling & solving

Games & Puzzles



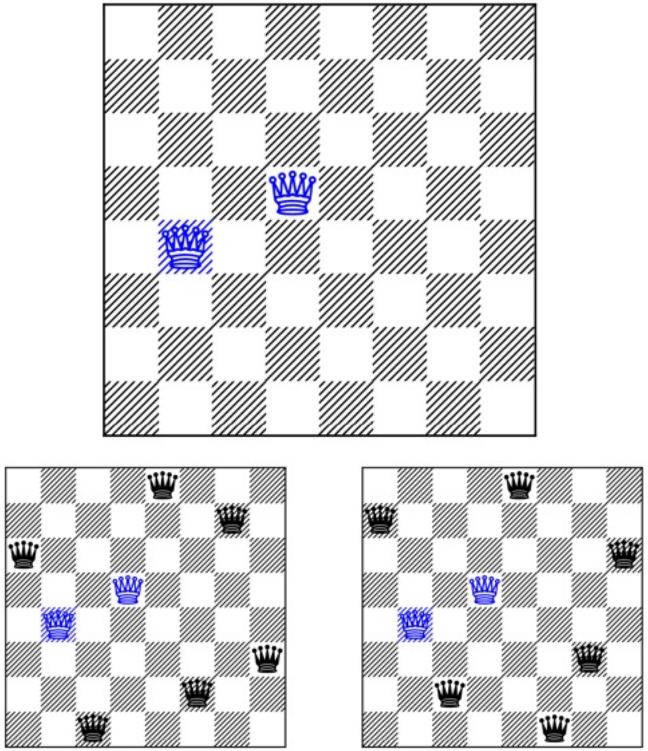
Ian Gent
Ian.Gent@st-andrews.ac.uk

Solving Solitaire Games



- We wrote the World's best program for solving games of Solitaire/Patience.
- E.g. Klondike which is the game in Windows Solitaire which is still played 100 million times every day.
- We improved knowledge of how much it can be won by a factor of 30 – and with a program that also solves lots of other games like those on the screen.

Understanding n-Queens Puzzles



- Theoretical research gives us new understanding of famous old puzzles.
- E.g. the n-queens completion puzzle which goes back to 1850.
- The picture shows a board with 2 queens placed and then the two ways it is possible to complete the layout with 8 queens and no two queens attacking each other.
- We showed this puzzle is NP-complete, a key complexity class

Human-Agent Argumentation and Deliberation

Alice Toniolo
a.toniolo@st-andrews.ac.uk

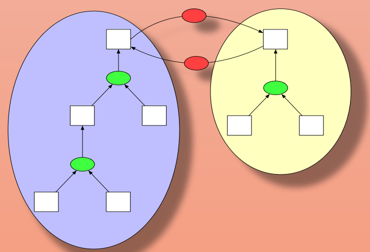
Problem

How can we help people make effective decisions in conflicting situations?

Approach

Computational Argumentation

- Define Pro/Con Arguments
- Analyse Conflicts



Aim

- Find Agreement
- Explain & Motivate
- Identify reasoning flaws

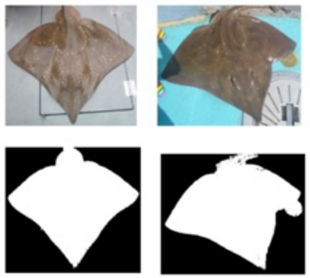
Argumentation in Human-Agent teams to support:

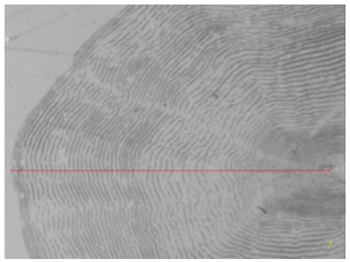
- Reasoning: in **Intelligence Analysis**, Analysis of Debates & Essays, Explanations
- Deliberation dialogue: Models of **Dialogue**, Multiagent Games

Computer Vision and Learning in Conservation

Kasim Terzic
kt54@st-andrews.ac.uk

- Counting seals in aerial surveys (350GB)
- Fish specimen re-identification in photographs
- Inferring history of salmon based on scale images (100k+)
- Classifying skin lesions in dolphin populations
- Classifying fishing activities based on GPS traces
- Monitoring activity in videos of captive quails
- Image-based sizing and sexing of lobsters and crabs

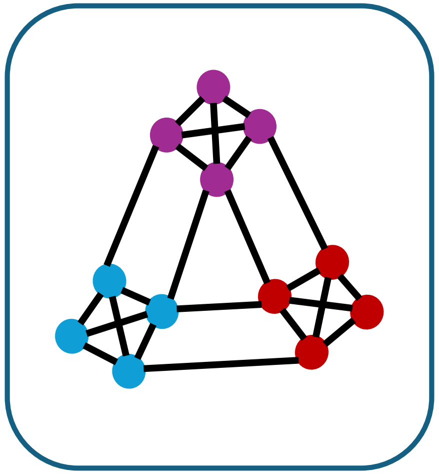




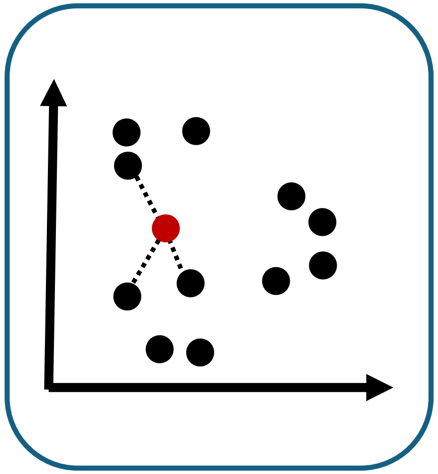
Graph Theory and Computational Geometry for Data Science

Peter Macgregor
prm4@st-andrews.ac.uk

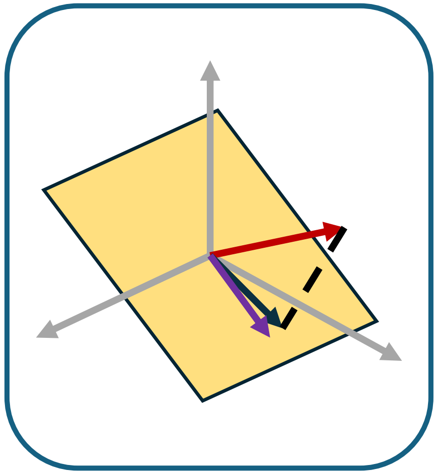
- Algorithms for processing large data sets. Applications in ML.



Clustering



Similarity Search

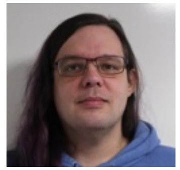


Dimensionality Reduction

- Developing graph-based clustering algorithms with improved complexity, approximation guarantees,
- Studying the geometry of high-dimensional data for applications in similarity search,
- Applying algorithmic techniques such as dimensionality reduction to improve the performance of ML algorithms.

At the Interface with Discrete Mathematics

Ruth Hoffmann
rh347@st-andrews.ac.uk

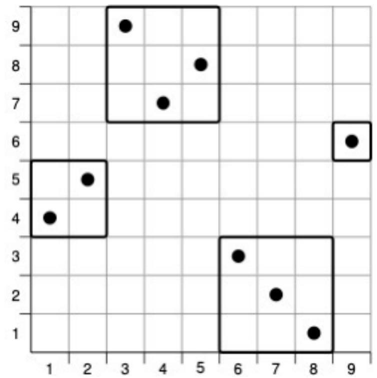


Chris Jefferson
caj21@st-andrews.ac.uk

Steve Linton
sl4@st-andrews.ac.uk

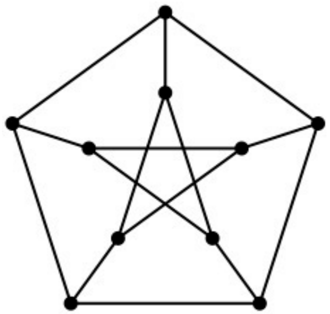
Olexandr Konovalov
alexander.konovalov@st-andrews.ac.uk

- Applications in and of:
- Graph theory,
- Combinatorics,
- Formal languages
- Computational group theory.
- Example: **symmetry breaking** in combinatorial search.



Representation of a permutation.
Sub-permutations that have consecutive indices and contiguous values indicated

CENTRE FOR INTERDISCIPLINARY RESEARCH IN COMPUTATIONAL ALGEBRA

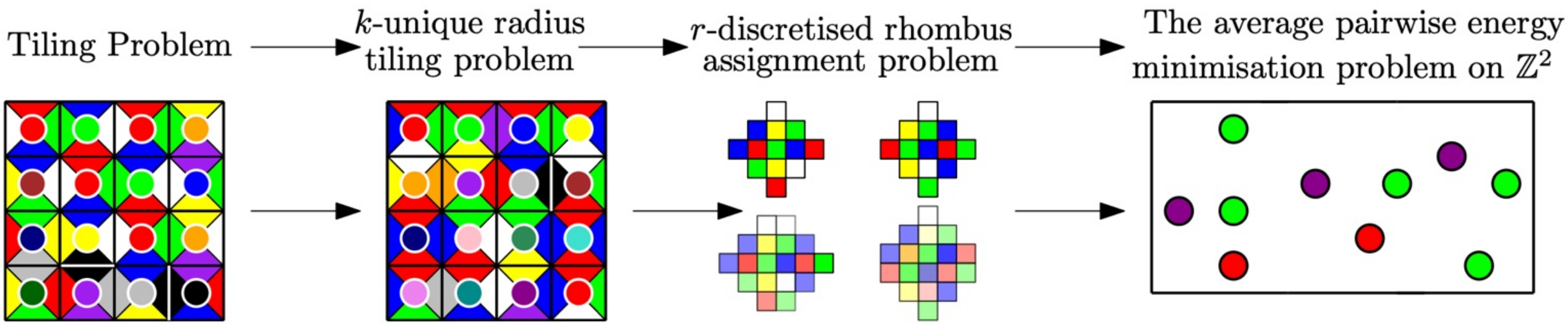


The Petersen graph: a small graph that is often a counter-example for new graph conjectures (Don Knuth)

Computational Tools For Chemistry Problems

Duncan Adamson
Duncan.Adamson@st-andrews.ac.uk

- Predicting structure of a crystalline material from chemical formula.
- Taking computational problems from chemistry and applying tools from computer science to solve them.
- Main results:
  - This problem is hard - indeed it is undecidable- in general,
  - Exact solutions for restricted instances - Novel combinatorial algorithms for solving problems in chemistry.
- Overview of reduction:
  - starting with the Wang Tiling problem, and ending with the chemistry problem.
  - Interesting as we take a problem that relies strongly on operating in oriented, discrete space, and convert it into a problem in non-oriented, continuous space.



Algorithms For Dynamic And Distributed Networks

Duncan Adamson
Duncan.Adamson@st-andrews.ac.uk

- Tools for exploring **temporal graphs**.
- Local symmetry breaking in networks based on hypergraphs.
  - Via colourings or independent sets
- Exploring
  - Determining am**dynamically changing networks** exploration schedule for an agent so that they can visit every node in the network.
- Use case: a **robot chemist**.
  - These robots move around lab spaces, which may be modelled as a dynamic network, with nodes representing positions at which the robot has to complete a task, and connections representing corridors in which the robot can move.
  - Over time, these corridors may become blocked (i.e., by a person moving though it) or become available.

How can Technology Help Effective Regulation of AI?

Tristan Henderson
tnhh@st-andrews.ac.uk

- Understanding and monitoring collection of training data:
  - Disconnect between technical solutions (robots.txt) and legal solutions (contract, chattel)
  - Need to gather empirical data on what is happening and what will work (ongoing work)
- Tools for efficient AI impact assessments:
  - Impact assessments are a common *a priori* tool for mitigating harm
  - But there are lots of proposed assessments for different scenarios, and little enforcement
  - How can technology help?
  - See BILETA 2025 (forthcoming)

Advancing NLP: Efficiency & Emergent Communication

Phong Le
pl200@st-andrews.ac.uk

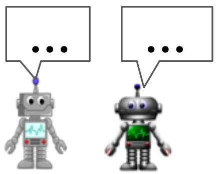
Research Focus 1: Making NLP more cost-effective and accessible

Key Methods:

See "low power" theme later.

- Employing weakly supervised learning to minimize annotation costs
- Developing noise-tolerant Machine Learning algorithms for unreliable data

Impact and Applications: Enabling low-cost, scalable, reliable NLP applications



Research Focus 2: Understanding Inductive Biases via Emergent Communication

Key Methods:

- Training deep-learning agents in language games
- Exploring inductive biases in agent architectures
- Evaluating agent behaviors using game-theoretic and linguistic metrics

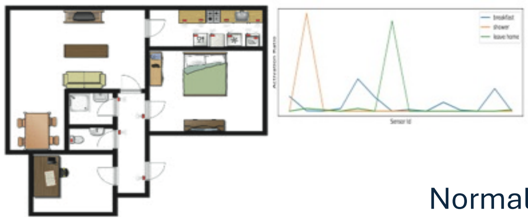
Impact and Applications: Insights into **language evolution**, cognitive science, and NLP

Sensor Data Analysis and Human Activity Recognition

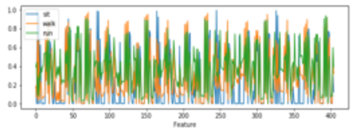
Juan Ye
jy31@st-andrews.ac.uk

- Work with different types of sensors to understand human behaviors, gestures, environmental events

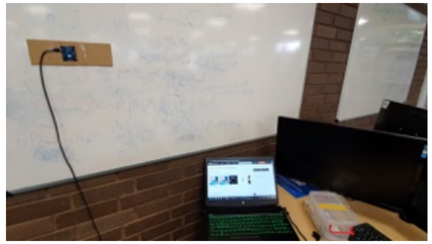
Ambient sensors



IMU sensors

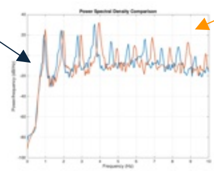


Radar sensors

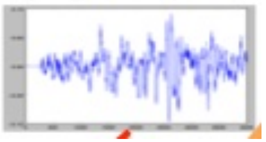


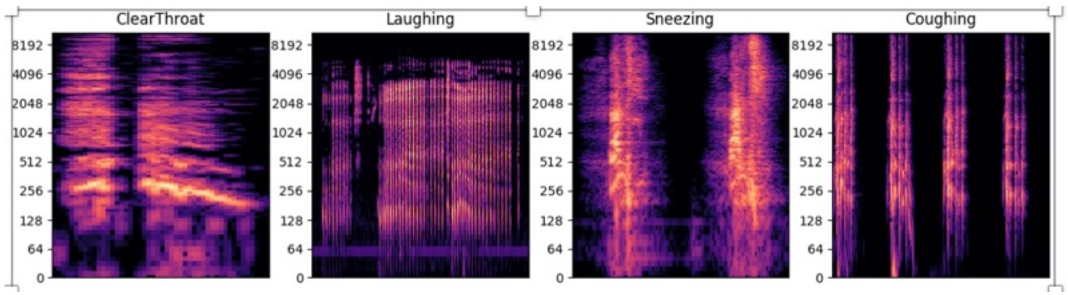
Normal walk

Happy walk



Microphone





Sensor Data Analysis and Human Activity Recognition

Juan Ye
jy31@st-andrews.ac.uk

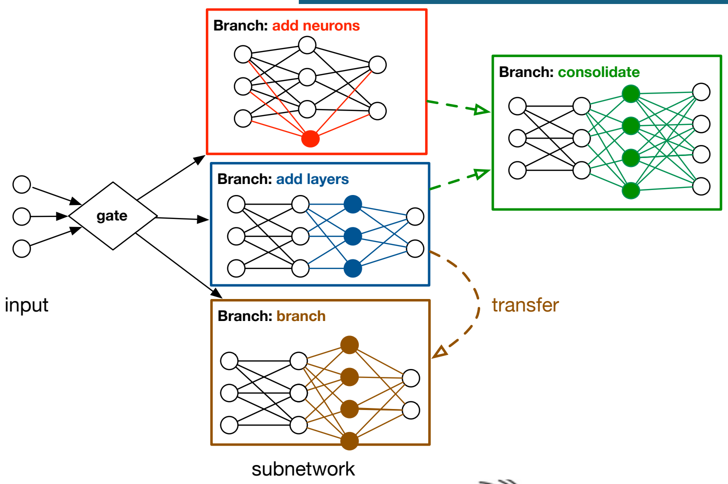
Energy Efficiency

Scalability

Systems in Society

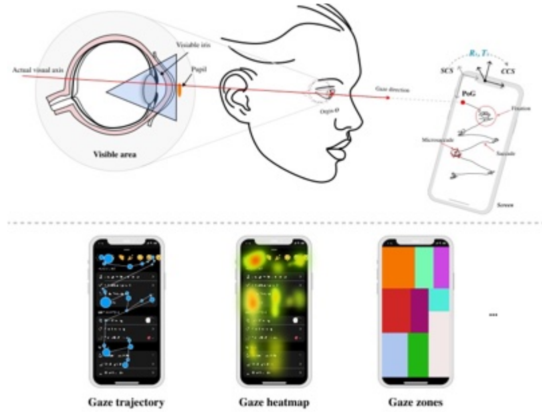
ElasticNet, a dynamic network architecture for continual learning

- Reduce parameter space and training time
- Keep the network small without making assumption of what or how many future classes will be
- Only grow the network when necessary



Appearance-based eye tracking

- Estimate gaze direction from facial images captured by camera
- Design gaze interface and gestures
- Enable user authentication and deepfake detection



Automated Algorithm Configuration and Selection

Nguyen Dang
nttd@st-andrews.ac.uk

- Almost every algorithm has its own **parameters** that can be tuned.
  - AI planning (choice of heuristics), deep learning (learning rates, number of hidden layers),...
- Automated **algorithm configuration**:
  - General purpose methods to automate the optimisation of algorithm parameters.
  - What's inside: a combination of machine learning and optimisation techniques.
- Similarly, in many applications, there is often no single algorithm that performs best on all problem instances.
- Automated **algorithm selection**:
  - Given a problem instance, automatically select the **best algorithm** from a portfolio of available algorithms.
  - What's inside: machine learning techniques to predict the best algorithm based on *features* of instances.

Health Informatics

Theme Lead: David Harris-Birtill (dcchb@st-andrews.ac.uk) 

Health Informatics: Overview

- The School is situated next door to the School of Medicine, with which we work closely.
- Part of our collaboration is through the **Mackenzie Institute for Early Diagnosis**:
  - https://medicine.st-andrews.ac.uk/mackenzie/
- Applications of both symbolic and sub-symbolic AI
- Examples:
  - AI and ML applied to diagnosis, fertility prediction, multimorbidity.

Health Data Science in Cancer and Fertility

Tom Kelsey
twk@st-Andrews.ac.uk

- Cancer treatment and age affect fertility-related biomarkers and long-term reproductive health.
- 40-year population-based cohort studies investigating mental health, hospital admission and fertility for cancer survivors
- AI and machine learning to personalize, streamline and improve assisted reproductive technologies, using data from over 13,000 cycles.
- The use of Electronic Health Records to optimize public health initiatives for lung, breast and bladder cancer screening.

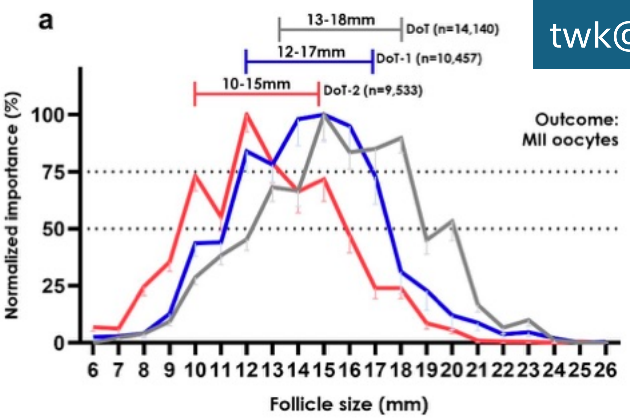
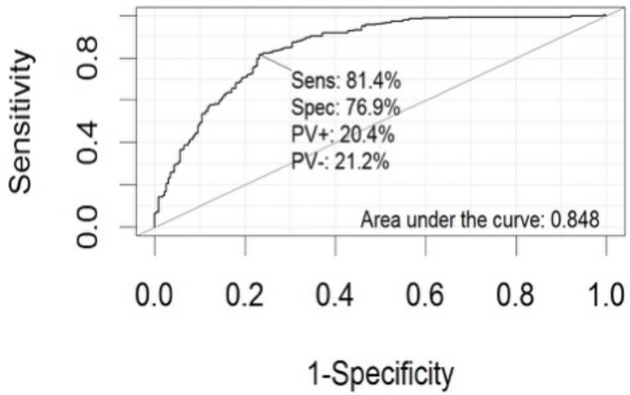


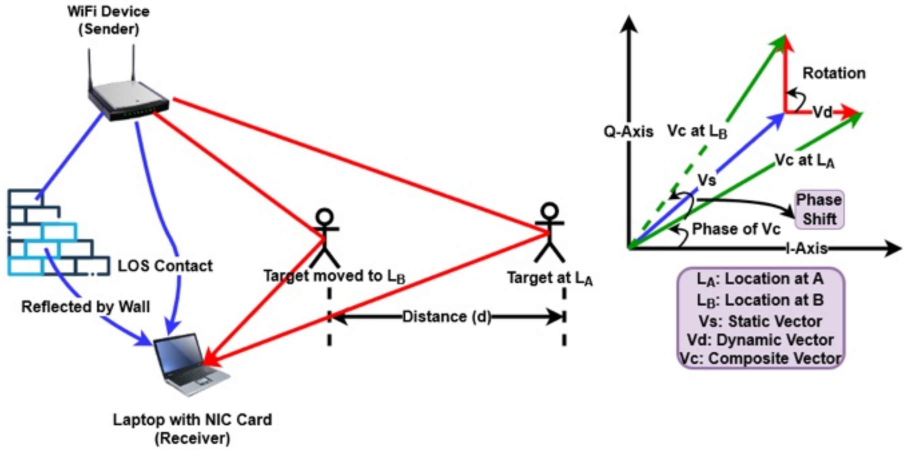
Figure 2: Area Under the Curve (AUC) for ECLS model applied to UK Biobank SMOTE data (AUC = 0.848).



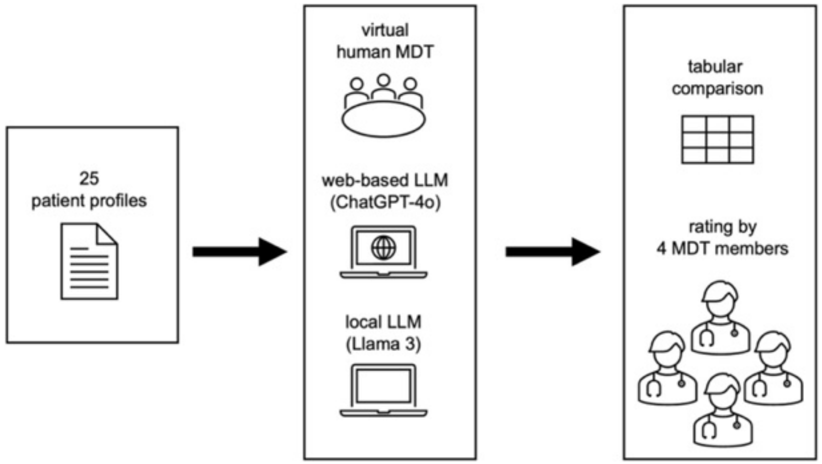
Sensors and Large Language Models

Tom Kelsey
twk@st-Andrews.ac.uk

- WiFi sensor data provides important data on the home lives of vulnerable people
- Modern deep learning and generative AI methods can be used to train and validate models that reliably detect occurrences of dangerous events



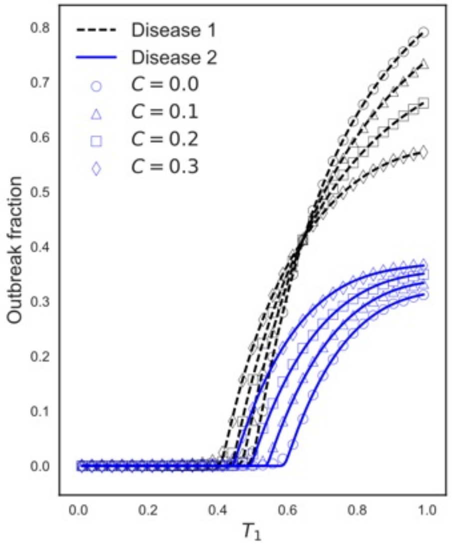
- LLMs such as ChatGTP are now pervasive in modern society
- Is their use in medical settings a danger or an opportunity?
- What is the concordance between LLM output and clinical team recommendations?



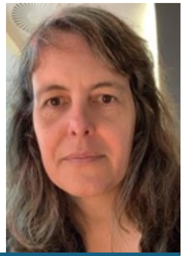
Epidemic Modelling

Simon Dobson
Simon.Dobson@st-andrews.ac.uk

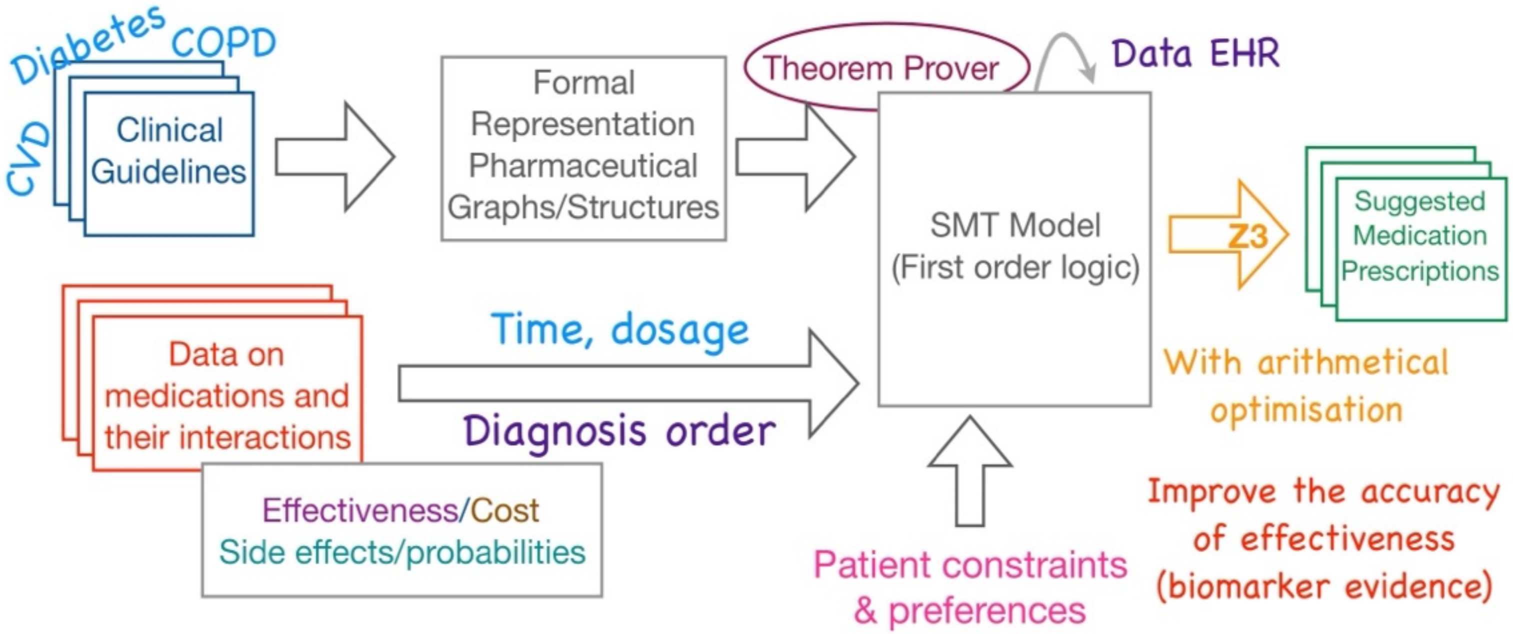
- Processes on networks and other combinatorial structures
  - SIR, SEIR, percolation, opinion dynamics, ...
  - Tooling now downloaded over 100,000 times
  - https://pyepydemic.readthedocs.io/en/latest/
- Main interests
  - New models of epidemic spread and potential (how big will it get, and how fast?)
  - Co-infection and interactions of multiple strains



Optimising Treatments for Multimorbidities



Juliana Bowles
jkfb@st-andrews.ac.uk

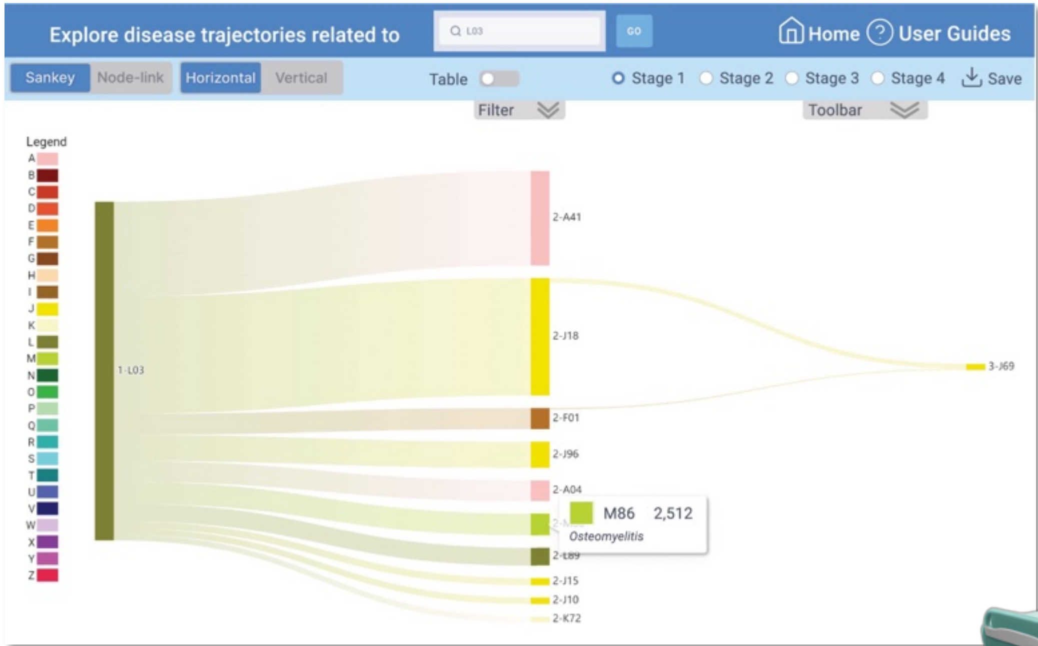


Very much an application of the constraint programming approaches we will see later.

- Search for solution(s) with maximal score such that side effects have low probability of occurrence, and minimising number of medications.

Data Science Methods to Improve the Delivery of Healthcare Services

Areti Manataki
a.manataki@st-andrews.ac.uk



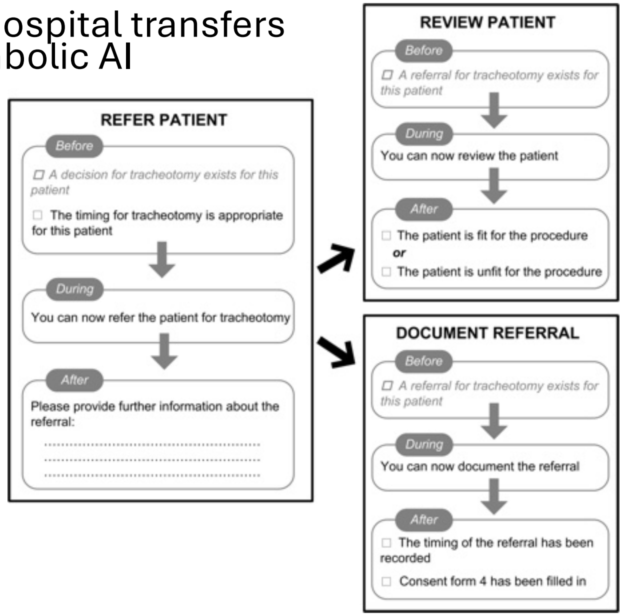
Discovering and visualising disease trajectories of multimorbidity in Scotland.

- In what order do people accumulate diseases?

Predicting length of stay in hospital

- Pediatric critical care
- Using only routinely collected data.

Safer intra-hospital transfers through symbolic AI

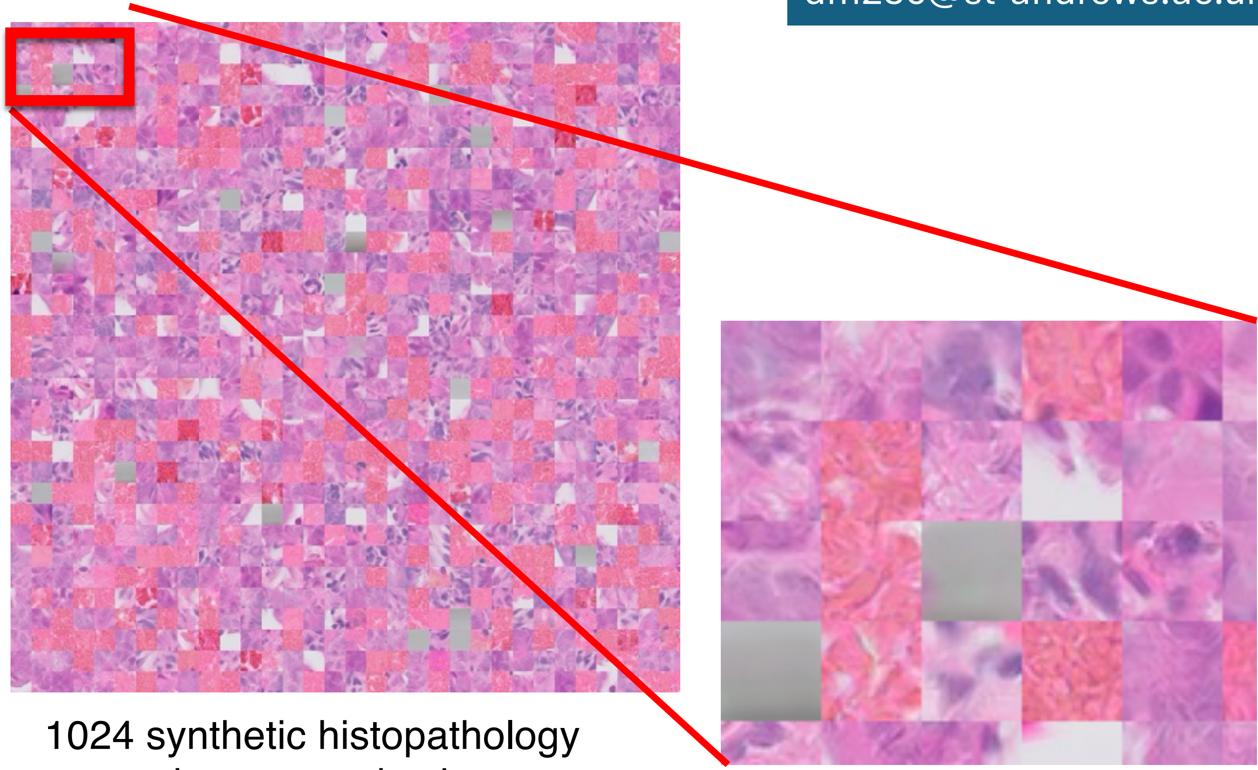


- Process modelling of the steps.
- Resulting in checklist inspired by WHO process for surgical safety.

Generative Machine Learning for Synthetic Histopathology Slides

David Morrison
dm236@st-andrews.ac.uk

- Anonymising medical data for use in machine learning is important to preserve patient privacy and, in many circumstances, is a requirement before data can be made available.
- One approach to anonymising image data is to train a generative model to produce data that is statistically similar to the input data and use the synthetic data in place of the real.
- A study of the effects of such a process on an exemplar downstream task, histology image classification:

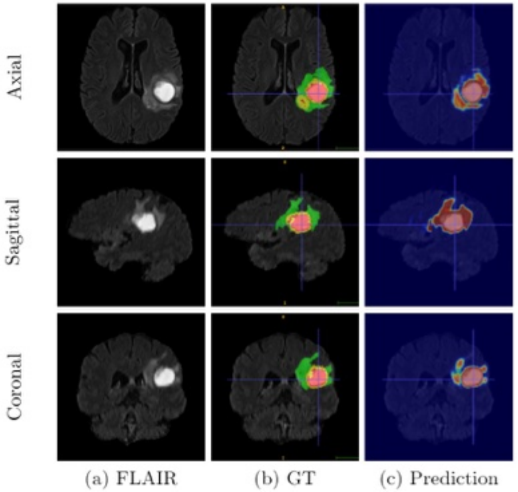


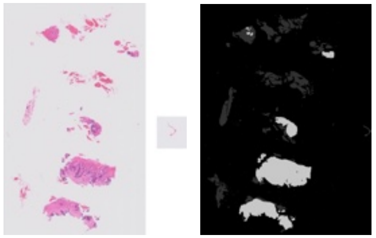
1024 synthetic histopathology patches created using a Generative Adversarial Network

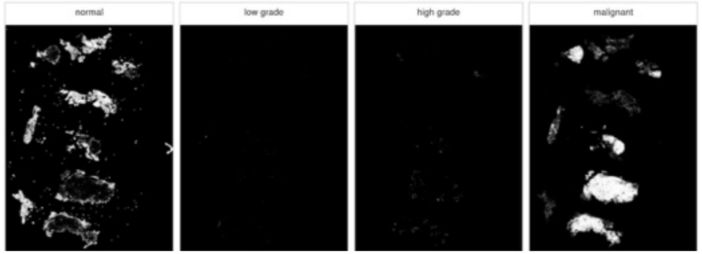
Medical Imaging and Sensing

David Harris-Birtill
dcchb@st-andrews.ac.uk

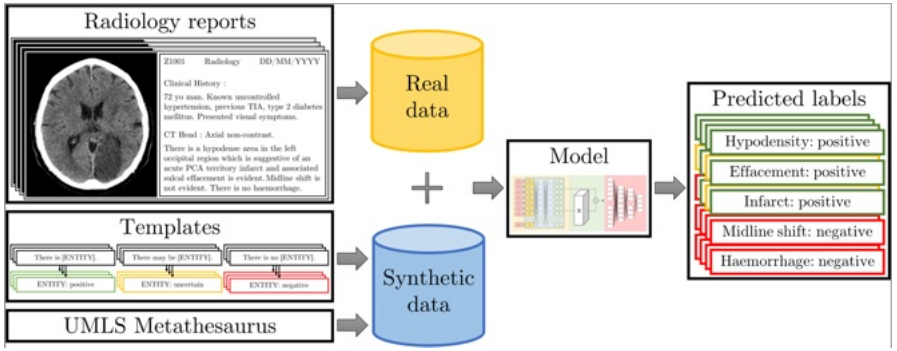
- Cancer detection and segmentation:







- Automated labelling of radiology reports:



NHS Fife

beyond medics

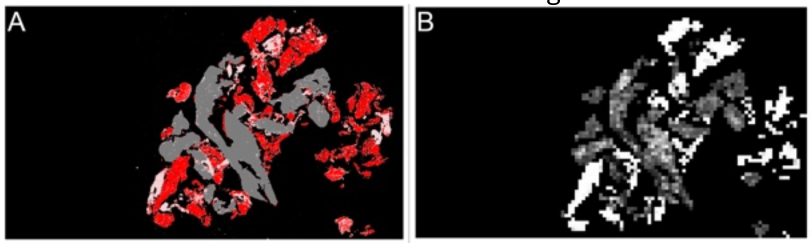
Cancer Detection & Risk Estimation

Chrissy Fell
cmf21@st-andrews.ac.uk

Deep learning for cancer detection

Annotation

Probability Malignant

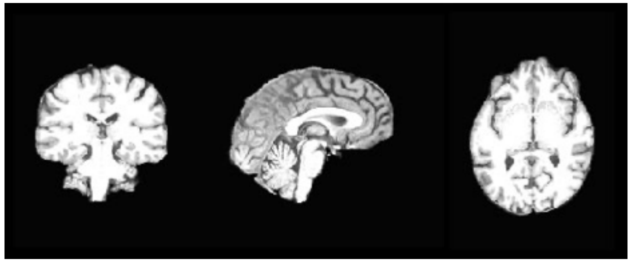


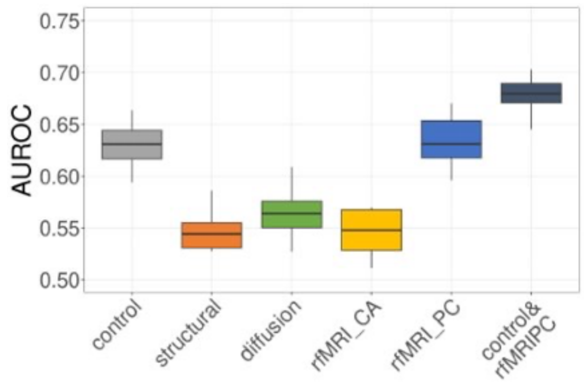
= Benign
= Malignant
= Blood or mucus
= Background

Probability Malignant = 1

Probability Malignant = 0

Machine learning to discriminate high and low genetic risk





Medical Imaging and Sensing

David Harris-Birtill
dcchb@st-andrews.ac.uk

- Measure vital signs at a distance.
- Automated Remote Pulse Oximetry – measuring peoples vital signs at a distance using cameras:

Human-Computer Interaction

Theme Lead: Loraine Clarke (lec24@st-andrews.ac.uk)

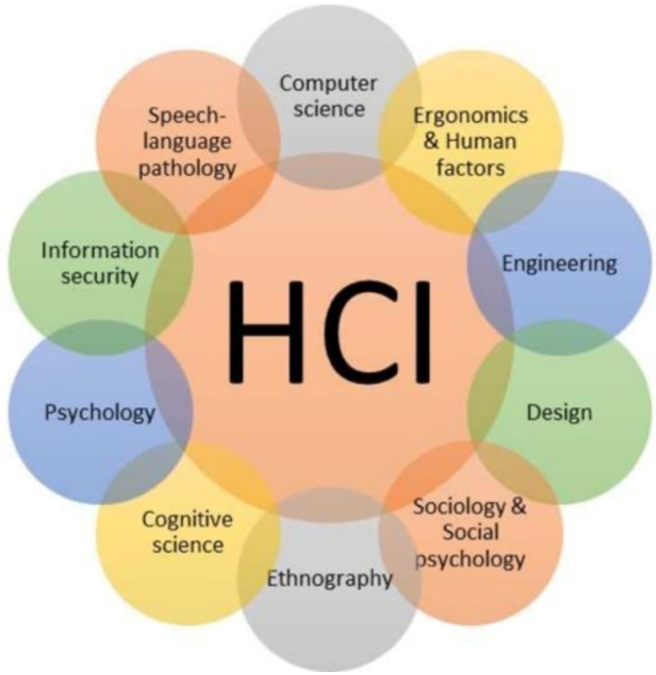
Human-Computer Interaction: Overview

- Exploring what interactions and technologies should be made to support a positive impact upon society and a quality user experience.
- Different User Needs

- Quality User Experience

Human-Computer Interaction: Overview

- The Interdisciplinary Design Science of Human-Computer Interaction (HCI) combines knowledge and methods associated with professionals including:
- **Psychologists** (incl. Experimental, Educational, Social and Industrial Psychologists)
- Computer Scientists
- Instructional and Graphic Designers
- Technical Writers.
- **Human** Factors and **Ergonomics** Experts
- **User** experience designers
- Anthropologists and **Sociologists**

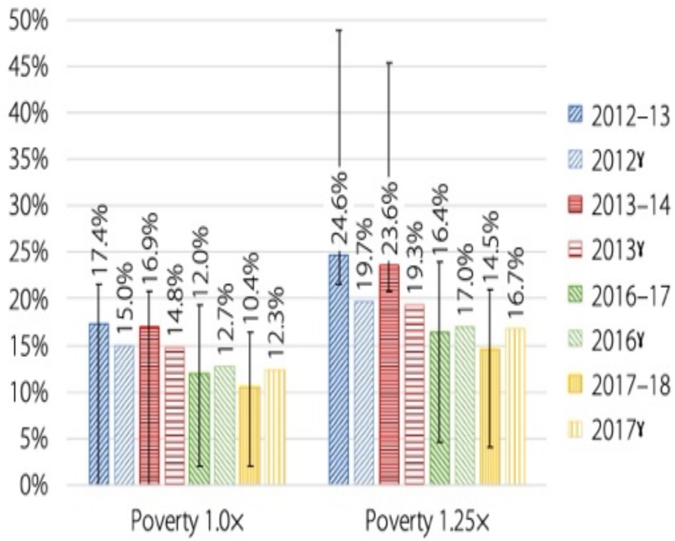


Research Strengths within the HCI Theme

- Primary Research Areas:
  - Privacy, Ethics and Equality
  - Sustainability & Biodiversity
  - Digital Inclusion & Education
- Research Approaches:
  - Tangible Interaction & Physical Computing
  - Storytelling & Data Visualization
  - User-centred & Participatory Design.

Understanding the Impact of ”Crowdwork” and the Expanding Gig Economy

Jason Jacques
jtj2@st-andrews.ac.uk



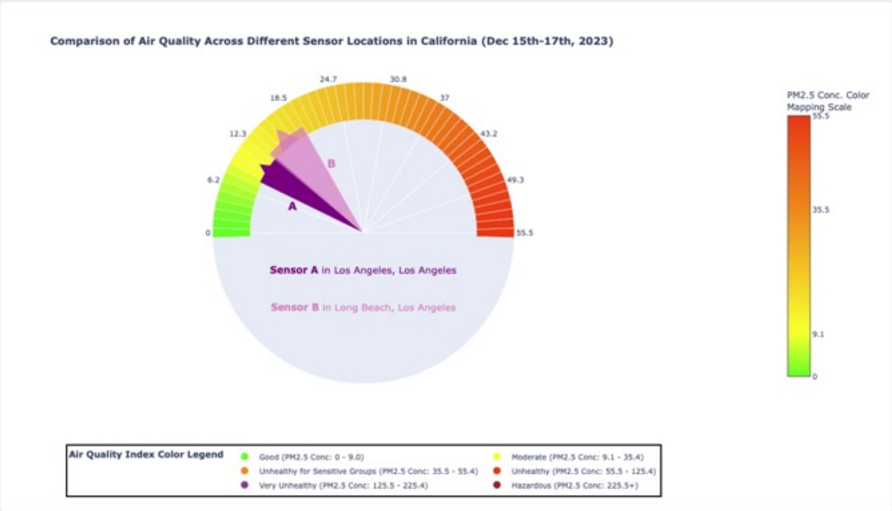
Estimated poverty rates for respondents at 1.0x and 1.25x the official threshold, compared to national figures (γ).

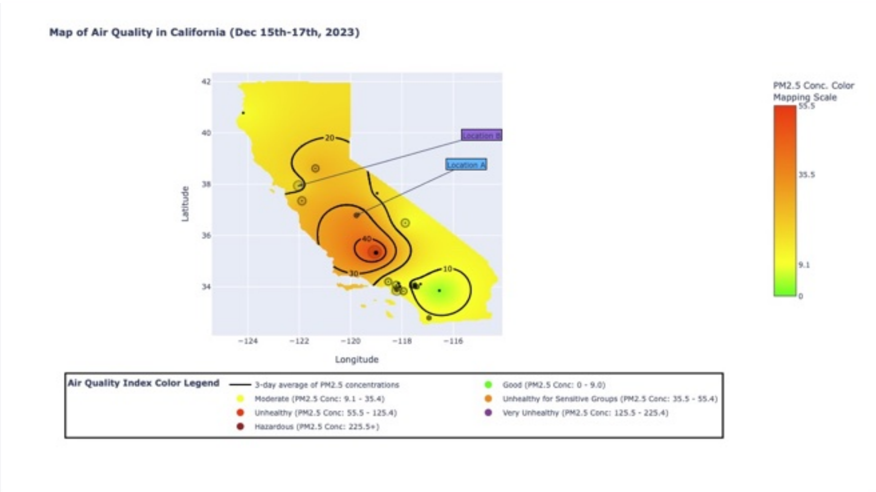
Longitudinal analysis of the **impact of platforms** (apps) on work and the gig economy:

- Graph shows long term analysis on **unemployment rates**
- Previously, digital knowledge workers (crowdworkers) were far more likely to be at risk of unemployment
- More recently, the expansion of higher-skill knowledge work and the physical gig economy have offered more viable/rewarding work
- However, challenges remain for gig-economy work (benefits: pensions, healthcare; job security; disability/equity; taxation and revenue)

Exploring Interactive and Passive Data Visualisations for Understanding Environmental Data

Jason Jacques
jtj2@st-andrews.ac.uk





- Accurate communication of environmental data is important:
  - Public health, climate change, trust in science.
- Communication of uncertainty in data (sensor noise, model error, etc.) can influence decision making and trust
- We (Jason, STARIS intern, Emory University) have explored the trade-offs in how visualisations can accurately communicate data
- Our experiments show that increased interactivity results in lower user comprehension/accuracy
- However, interactivity increases user insight into variability in the data which can improve trust

Exploring experiences of AI facial recognition technologies

Loraine Clarke
lec24@st-andrews.ac.uk

- Work undertaken in India with the public to discuss the concerns they have about large tech giants and governments collecting **biometric data on a mass scale**.
- The device in the centre image is a raspberry pi that takes photos of you and sends this Photo to an **AI** to determine what age it thinks you are, what gender, your facial expression….. all from one image.
- This is an example of how we are exploring public opinion, knowledge and perceptions of what data is being collecting about them daily by tech giants.

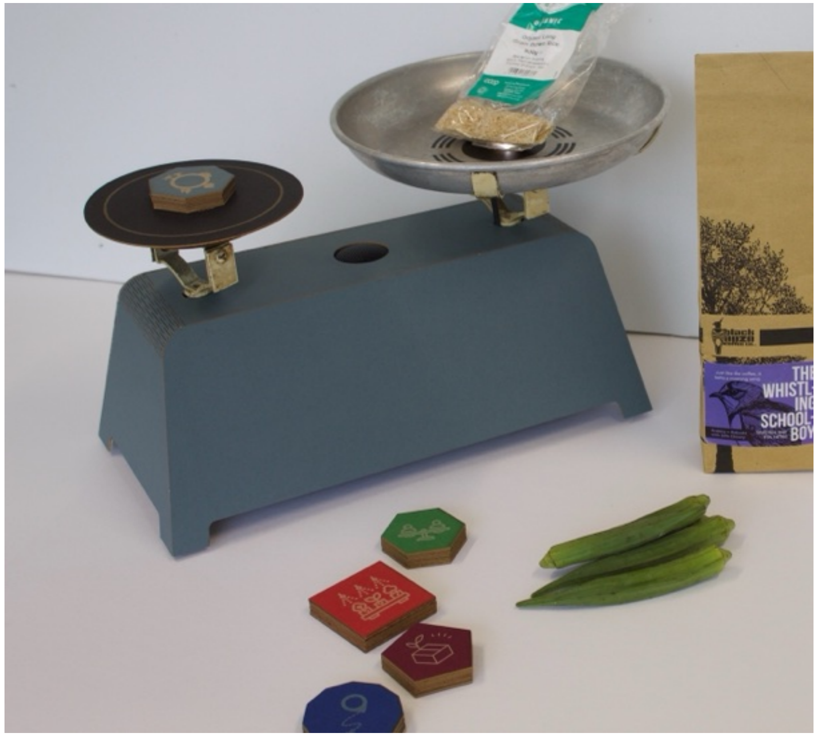
Sensing Biodiversity at the Botanic Gardens

Loraine Clarke
lec24@st-andrews.ac.uk

- One of our collaborators is the botanic gardens in St Andrews.
- To explore ways in which we can **sense changes in biodiversity** and then use this information to **engage the public** with biodiversity changes that often go unnoticed.
- Left, we have a raspberry pi set up with a powerbank and camera taking pictures throughout the day to detect visual changes in the woodpile.
- We've seen slugs, bugs, a frog and a wasp burrowing into the wood using this system.
- The next phrase we're interested in is creating playful interactive technologies for the public to engage with these changes.

Tangible Interactive Experiences Encouraging Reflection on the Sustainable Values of our Food Choices

Loraine Clarke
lec24@st-andrews.ac.uk



- Part of a project working with Indian farmers investigating the different perspectives **farmers** and **consumers** have of food.
- Where a traditional weighting scales only allows people to find out the weight of food, this scales allows people to investigate other sustainable attributes of the food such as:
  - Distance travelled from the farm,
  - Amount of pesticide used,
  - How much money the grower received in comparison to the price a consumer is paying.

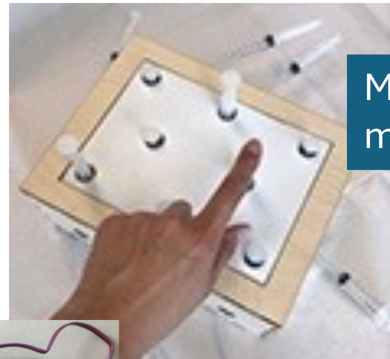
Tangible Interactions & Physical Computing

Loraine Clarke
lec24@st-andrews.ac.uk

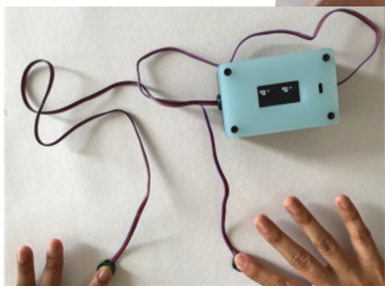
- Using physical computing, digital fabrication and tangible interfaces to address our research.
- For example, exploring ways to engage the public with the impact of human activity on biodiversity.
- A tangible interface to enable people to scroll through the increase in flights and decrease of lake size over the years.
- Rather than having screen based content we're interested in alternative technologies such as tangible interfaces.

Miriam Sturdee
ms535@st-andrews.ac.uk

Miriam Sturdee



Loraine Clarke



- More examples of tangible interfaces and shape changing interfaces.

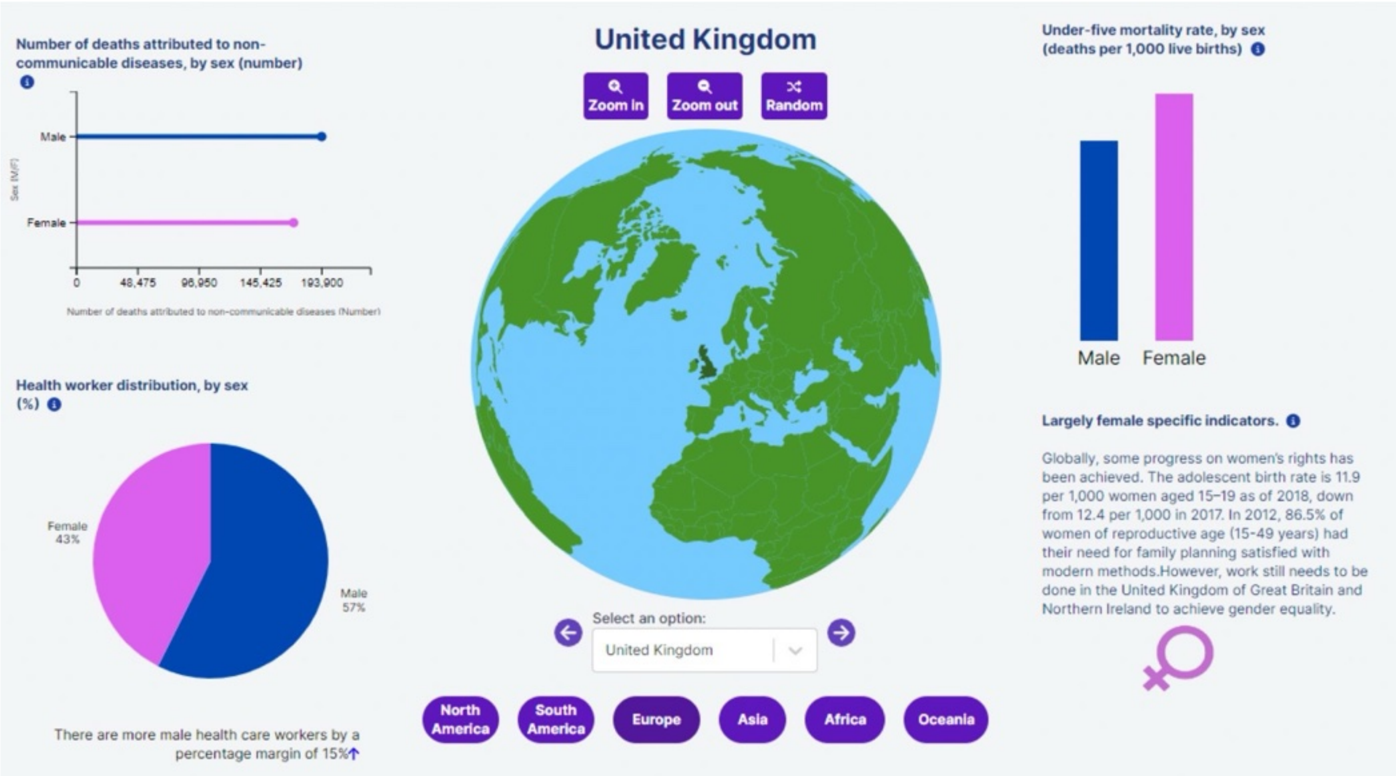
Storytelling

Miriam Sturdee
ms535@st-andrews.ac.uk

- Sketching as a tool for designing new technologies.
- Also the creation process when using newer novel technologies, such as:
    - Augmented reality,
    - VR,
    - haptics.

Data Visualization

- A dashboard to raise awareness of **“Gender bias in healthcare”**.



https://www.ghcdp.co.uk/

Areti Manataki
a.manataki@st-andrews.ac.uk

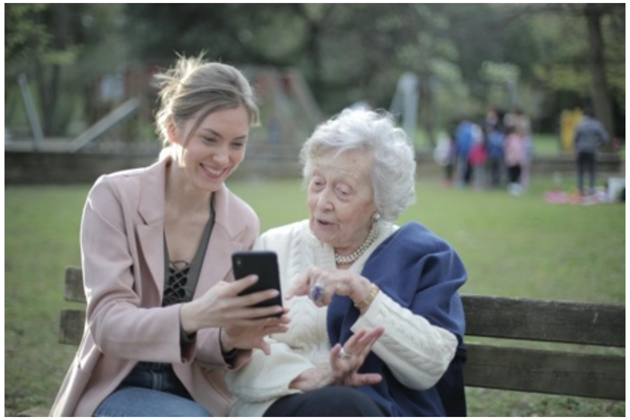
- Data visualisation in the context of health care settings.
- 78% of participants who evaluated the dashboard who said they knew little about gender bias in healthcare felt their knowledge had increased.
- All participants maintained the same level of importance or their belief in the importance became stronger of addressing gender bias in healthcare.

Digital inclusion

Dharini Balasubramaniam
dharini@st-andrews.ac.uk

Digital Inclusion in Later Life
- Understanding the age-based digital divide
- Improving the digital literacy of older adults
- Developing accessible digital services

- DILL is a global challenge, recognised by the UN, governments and NGOs.
- In the context of population ageing, continuing digitalisation, and fast-changing technologies, many older adults are **digitally excluded** from services and resources that are needed to maintain their independence, wellbeing and social connections.
- This research aims to explore the barriers and enablers to digital technology adoption in later life and create effective and scalable solutions to improving the **digital literacy** of older adults and developing and deploying accessible digital services.



User studies

Participatory design

Community engagement

Digital inclusion

Abd Ardati
Abd.ardati@st-andrews.ac.uk

Community-Led Digital Inclusion

- Ethical digital transformation and digital inclusion, particularly for marginalised communities.
- His work explores ways to build digital skills, reduce exclusion, and address AI bias.
- He collaborates with policymakers, industry, and civil society to support responsible digital innovation.
- Through initiatives like the Scottish Collective Intelligence Community (SCIC), he examines how **participatory approaches**, and AI can enhance decision-making and governance.
- Contributes to shaping ethical, inclusive digital policies and practices that ensure technology benefits all communities.

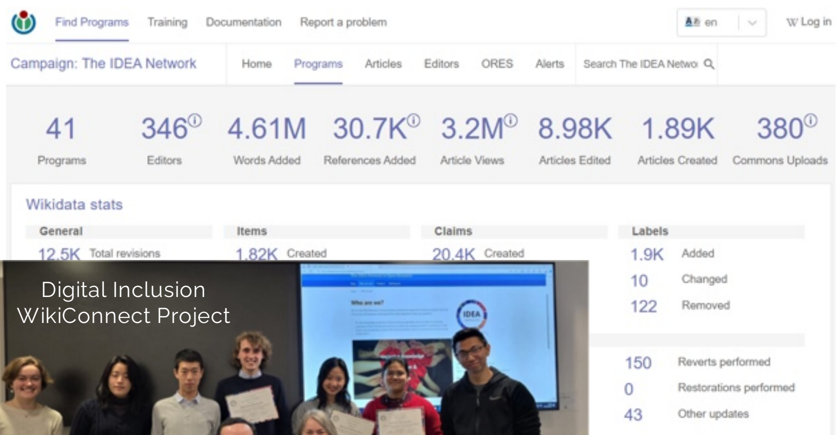
User studies

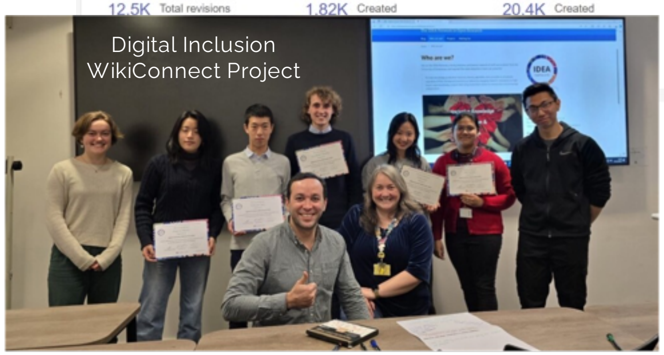
Participatory design

Community engagement

Kirsty Ross
ksrh1@st-andrews.ac.uk

Abd Ardati
Abd.ardati@st-andrews.ac.uk





The **Inclusion, Diversity, Equity, and Accessibility in Open Research Network** (IDEA Network for short):

- **To make knowledge production inclusive, diverse, equitable, and accessible** to all people, regardless of their background, location, or culture
- **by engaging research institutions** in high-impact open knowledge projects that **bring researchers closer to communities** and encourage collaboration.
- IDEA Network has led several impactful projects, including The Role of Universities in an Ethical Digital Nation, which explores how universities can align with the government's vision for ethical digital transformation.

Website

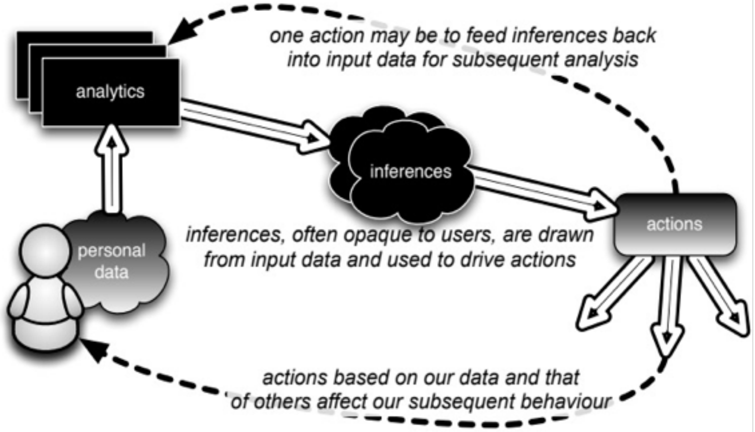
Human-data Interaction

- HCI evolved from interactions between humans and computers as artefacts
- but we increasingly interact with data rather than computers
- see e.g. Encyclopedia of HCI

Participatory Design for Data-centred Projects

Tristan Henderson
tnhh@st-andrews.ac.uk

- How do stakeholders (e.g. data subjects, data protection officers) engage with technology law and related technologies?
- See e.g. GoodIT 2024 doi:10.1145/3677525.3678675



Systems

Theme Lead: Stephen McQuistin (sjm55@st-andrews.ac.uk)

Systems: Overview

- Research Challenges:
  - How do we design computer systems that scale to millions or billions of users?
  - How do we balance the trade-offs between performance, energy consumption, and security in computer systems?
  - How do we ensure that the increasingly complex and interconnected systems that we build are sustainable and provide social benefit?
- The Systems Research Theme:
  - Systems is one of the largest research areas in the School, covering the broad areas of distributed systems, networked systems, sensor systems and data-intensive systems
  - The group takes a very practical approach to research, by building and evaluating real systems, whilst publishing in many of the top-tiered systems research conferences and journals

Computer Systems

Tom Spink
tcs6@st-andrews.ac.uk

Broad interests:

- Dynamic Binary Translation
- Virtualisation
- Operating Systems
- Compilers
- Hardware Acceleration

Simulation and Virtualisation

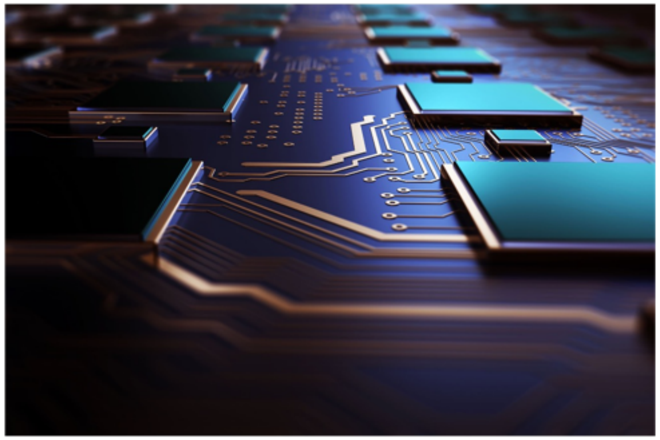
- Generating Fast and Efficient Instruction Set Simulators from Formal Semantics
- Virtualising the Internet-of-Things
- Hybrid Static/Dynamic Binary Translation

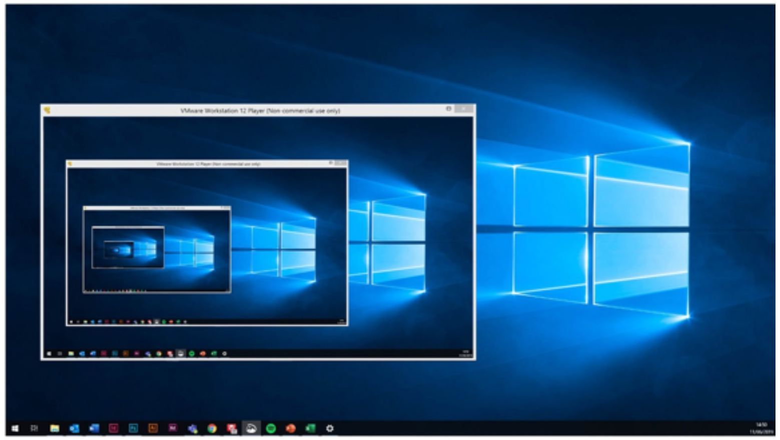
Hardware Acceleration

- Hardware Accelerated JIT Compilation
- Sparse Matrix Acceleration

Operating Systems

- **Make Unikernels Great:** Making unikernels a viable alternative to containers





Generating Fast and Efficient Instruction Set Simulators from Formal Semantics

Tom Spink
tcs6@st-andrews.ac.uk

Ferdia McKeogh
tcs6@st-andrews.ac.uk

- Formal semantics can accurately describe the Instruction Set Architecture of a processor, but they are often verbose, containing a significant amount of detail.
- Such detail is not required for functional simulation of the ISA, and instead gets in the way of doing things fast.
- This project aims to identify parts of formal semantics that do not directly contribute to ISS, and generate a simulator from these descriptions that is significantly faster than existing.

Virtualising the Internet-of-Things

Tom Spink
tcs6@st-andrews.ac.uk

The Internet-of-Things is growing! There are literally billions of connected devices, from light bulbs to toasters.

But, companies are not testing their products at scale – or if they are, it's using a "patchwork" of simulator tools that don't give a holistic overview of what's going on.

Combining my expertise in fast simulator generation, I'm looking to change that by developing high-performance, configurable simulation infrastructure to model and simulate large IoT device deployments, from the hardware to the software.

Hybrid Static/Dynamic Binary Translation

Tom Spink
tcs6@st-andrews.ac.uk

Work in collaboration with: TU Munich

- Binary Translation is an important tool in modern systems, facilitating the transition from one processor architecture to another.
- Dynamic Binary Translation introduces a lot of runtime overhead, and Static Binary Translation is often impossible in a lot of cases.
- Combining these strategies, we have developed techniques that provide a "hybrid" translation environment.
  - Translate as much statically as possible, and fall back to dynamic translation when required.

Hardware Accelerated JIT Compilation

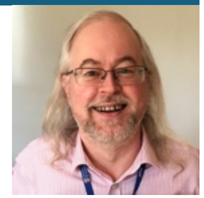
Tom Spink
tcs6@st-andrews.ac.uk

Work with PhD student: Kim Stonehouse (University of Edinburgh)

- JIT compilers are prevalent in dynamic language runtimes, such as Java, C#, WebAssembly, Python, PHP, etc – and the popularity of these language is growing.
- These languages are found on devices ranging from mobile phones, through to scale-out applications running across data centres.
- Recognising this, we aim to build **faster** and more **energy efficient** JIT compilers in hardware (i.e. specialised hardware for JIT compilation), so that the performance of applications written in these languages improves significantly.

Sparse Matrix Acceleration

Tom Spink
tcs6@st-andrews.ac.uk



Simon Dobson
Simon.Dobson@st-andrews.ac.uk

- Sparse Matrices are an important **underlying data structure** in many scientific and ML-based application.
- Recognising that with sparse matrix computations comes a number of **potential optimisations**, and that scientific workloads are demanding more and more performance, we aim to build an accelerator specifically designed for performing fundamental operations on sparse matrices.

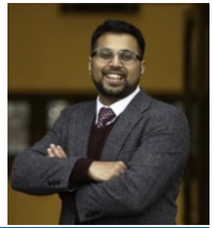
Make Unikernels Great: Making unikernels a viable alternative to containers

Tom Spink
tcs6@st-andrews.ac.uk

The container ecosystem is highly prevalent, and used significantly in both development and end-user workflows.

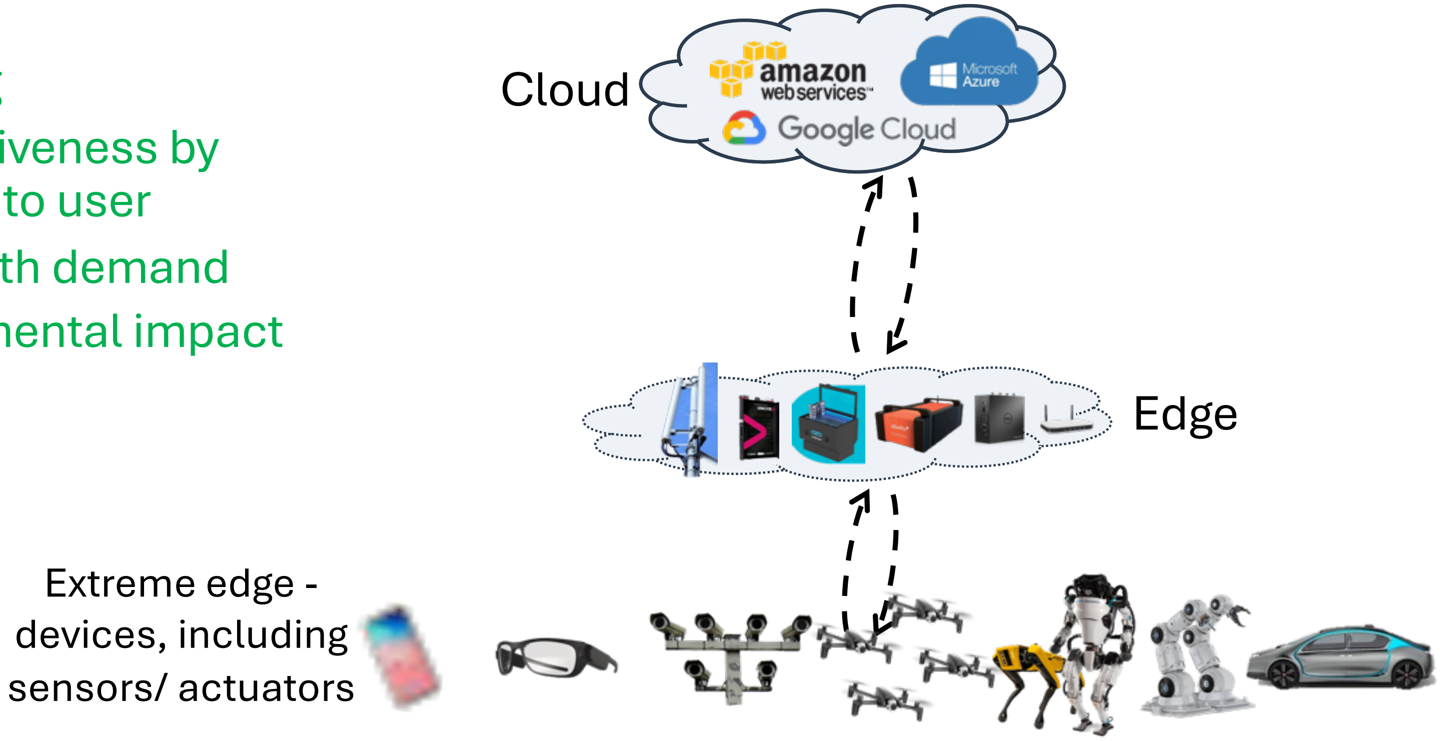
- They allow applications to be bundled together with their dependencies, support files, and assets in a convenient package that can be distributed and orchestrated locally, or across a computing cluster.
- However, they incur significant overheads in terms of storage utilisation, and can also have certain performance implications too.
- This project proposes Unikernels as a viable alternative to the container ecosystem, playing off their strengths as bare-metal applications with full access to virtualised hardware.

Edge Computing



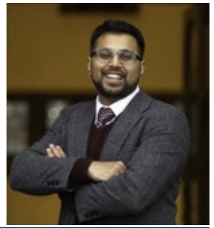
Blesson Varghese
blesson@st-andrews.ac.uk

- Key benefits
  - Privacy preserving
  - Improves responsiveness by processing closer to user
  - Reduces bandwidth demand
  - Reduces environmental impact



Edge AI research

- Edge AI - bridging machine learning (ML) and edge computing
- How to run ML on small devices with limited compute and memory?
- Host and direct the Rakuten-funded Edge Computing Hub
- Partners in the recently awarded UKRI National Edge AI Hub (~£10M)
    - 12 UK universities and 55+ industry partners
    - We lead a research theme and the Industry Engagement Directorate in Scotland.



Blesson Varghese
blesson@st-andrews.ac.uk

Developed a range of new techniques to train and run ML models:

- Offloading
- Compression
- Pipeline parallelism
- Local learning

Neural Networks

Transformers

MBs

...

a few GBs

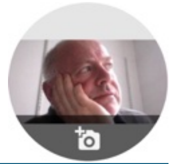
...

a few more GBs

Similarity Search

- Find, from within a very large collection of objects, those few that are most similar to another object presented as a query.

Al Dearle
Alan.dearle@st-andrews.ac.uk

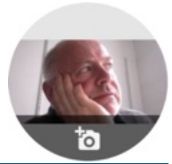


Richard Connor
rchc@st-andrews.ac.uk

- Modern neural networks very good at producing **embeddings**:
  - High-dimensional vectors that contain dense information about the network input.
- Typically used to perform classification tasks,
  - but usually distance metrics over the embeddings give a good proxy to similarity within the original domain.
- In the example image, we are querying a large set of random photographs with queries representing a border collie dog, and the results are mostly also border collies, a tiny minority of all the photos present in the set.
- We have pioneered the use of **polyadic queries**, where more than a single object is presented as a single, composite query.
- The results shown here are better than those returned by any individual element of the query set.
- Most current techniques for search use a notion of graph navigation to achieve fast query time, but such techniques are inherently unscalable to, for example, billion-scale data.
- We are investigating various locality sensitivity techniques, which typically require much less pre-processing time, to address this issue.

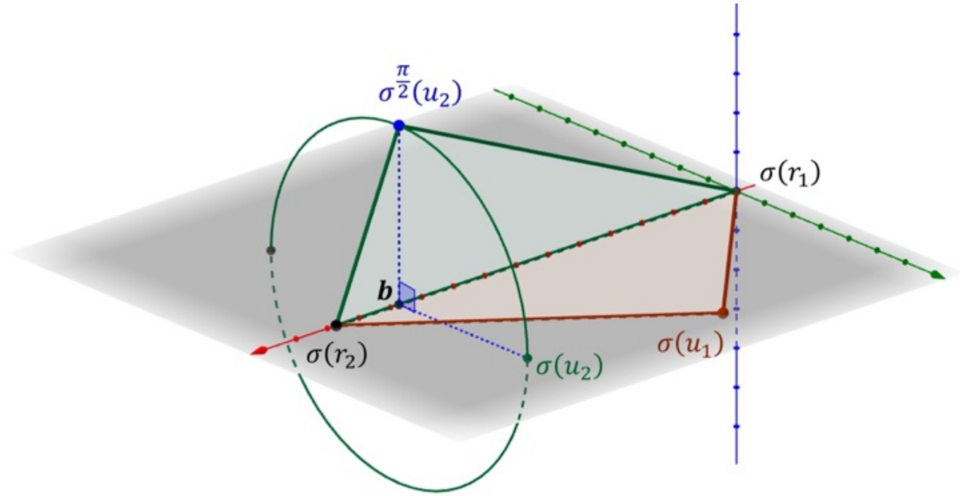
Similarity Search

Al Dearle
Alan.dearle@st-andrews.ac.uk



Richard Connor
rchc@st-andrews.ac.uk

- Also interested in more theoretical aspects of high-dimensional vector spaces.
- We have shown some startling correlations among various commonly used **loss functions** in neural networks, and have some early results suggesting that more principled information-theoretic metrics may perform better than those commonly used.
- The **nSimplex-Zen** transform has been developed over Hilbert spaces in general, which include most commonly-used metric spaces.
- Its performance, in terms of accuracy, is markedly better than any of principle component analysis, multidimensional scaling, and random projects for almost all spaces.
- The picture is an abstraction of a high-dimensional polytope being rotated in a one-smaller dimension to demonstrate an upper-bound property and a powerful estimator function.



(b) Adding **b**, the centre of the locus of $\sigma^{\theta}(u_2)$, and setting the angle $\theta = \angle\sigma(u_2)\mathbf{b}\sigma^{\theta}(u_2)$ to $\pi/2$.

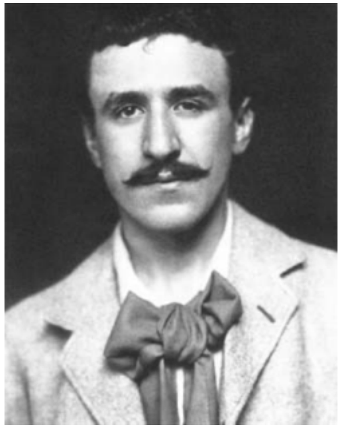
Linkage

- Scottish Historic Population Platform **SHiPP** project (with Edinburgh)
- The aim is to link records of Scotland 1855-1973 to reconstitute the genealogical population structure for use in other research (medical, historical, sociological etc.)
  - 14 million births (1.9Gb)
  - 11 million deaths (2.3Gb)
  - 4.2 million marriages (0.8Gb)
  - 18 million individuals

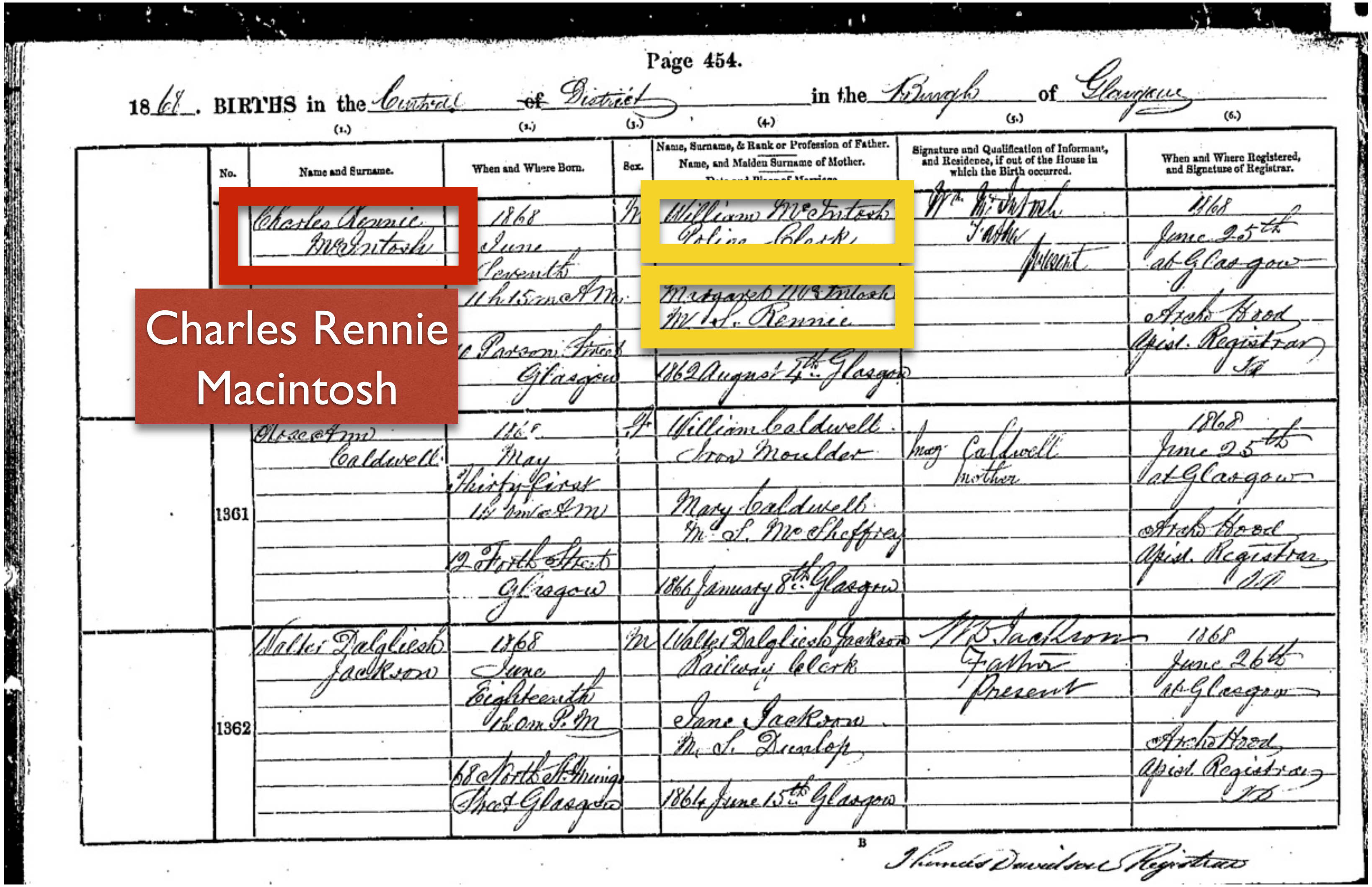
Al Dearle
Alan.dearle@st-andrews.ac.uk

Graham Kirby
Graham.kirby@st-andrews.ac.uk

Ozgur Akgun
Ozgur.akgun@st-andrews.ac.uk



A 19th Century Scottish architect, designer, water colourist and artist.



Page 454.

1868. BIRTHS in the Central of District in the Burgh of Glasgow

No.	(1.) Name and Surname.	(2.) When and Where Born.	(3.) Sex.	(4.) Name, Surname, & Rank or Profession of Father. Name, and Maiden Surname of Mother. Date and Place of Marriage.	(5.) Signature and Qualification of Informant, and Residence, if out of the House in which the Birth occurred.	(6.) When and Where Registered, and Signature of Registrar.
	Charles Rennie McIntosh	1868 June Seventh 11h 15m A.M. 70 Parson Street Glasgow	M	William McIntosh Police Clerk Margaret McIntosh M.S. Rennie 1862 August 4th Glasgow	Wm McIntosh Father Present	1868 June 25th at Glasgow Arch Hood Asst. Registrar
1361	Rose Ann Caldwell	1868 May Thirty-first 1h 0m A.M. 12 Forth Street Glasgow	F	William Caldwell Iron Moulder Mary Caldwell M.S. McSheffrey 1866 January 8th Glasgow	Mary Caldwell mother	1868 June 25th at Glasgow Arch Hood Asst. Registrar
1362	Walter Dalgliesh Jackson	1868 June Eighteenth 1h 0m P.M. 68 North St Mungo Street Glasgow	M	Walter Dalgliesh Jackson Railway Clerk Jane Jackson M.S. Dunlop 1864 June 15th Glasgow	W D Jackson Father Present	1868 June 26th at Glasgow Arch Hood Asst. Registrar

Thomas Davidson Registrar

www.scotlandspeople.gov.uk/content/images/famousscots/birthCharlesRMcIntosh1868.tif

Linkage: Metric Indexing

- Comparing each record with every other record would be prohibitively expensive many use blocking but blocking potentially misses potential matches.
- Metric indexing does not suffer from that problem
- We therefore use a metric indexing system called BitPart to index the data (from our similarity search work)
- The BitPart index structure creates a set of inclusion zones encoding the inclusion of data points in (or out of) a set of database partitions in a binary fashion with respect to a set of reference points.
- A relatively small number of reference points (in the order to 20-40) is enough to characterise the search space and each metric query only requires distances to be calculated to the reference points.
- For large datasets such as the Scottish vital event records this represents a large performance increase.
- Furthermore, the index is highly compressed (as a set of bits) and obviates the need to directly interact with the stored database records when making queries.

Al Dearle
Alan.dearle@st-andrews.ac.uk

Graham Kirby
Graham.kirby@st-andrews.ac.uk

Ozgur Akgun
Ozgur.akgun@st-andrews.ac.uk

Identifier Locator Network Protocol (ILNP)

https://ilnp.cs.st-andrews.ac.uk

Saleem Bhatti
saleem@st-andrews.ac.uk

Scalability

Internet addressing architecture, routing state, multiple connectivity, ubiquitous connectivity.
https://ilnp.cs.st-andrews.ac.uk

Performance Optimisation

Protocol performance, mobile / wireless, quality of service for end-systems.
https://sites.cs.st-andrews.ac.uk/people/snb6//tcp.html
https://sites.cs.st-andrews.ac.uk/people/snb6//ieee_802_11-wifi.html
https://sites.cs.st-andrews.ac.uk/people/snb6//qos.html

Security & Privacy

Architectural mechanisms for privacy and security, without new cryptographic techniques.
https://sites.cs.st-andrews.ac.uk/people/snb6//privacy.html
https://sites.cs.st-andrews.ac.uk/people/snb6//security.html

Energy Efficiency

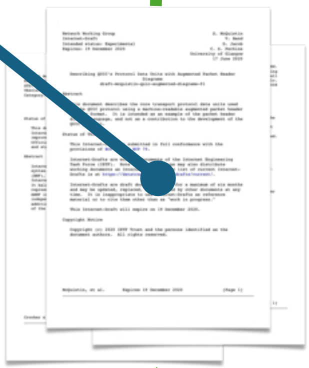
Client-side energy efficiency, mobile / wireless Internet connectivity, energy usage in video.
https://sites.cs.st-andrews.ac.uk/people/snb6//energy.html

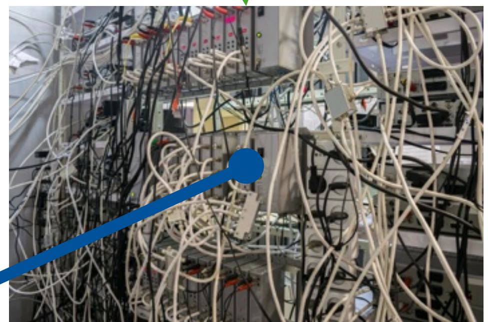
Building a Trustworthy Internet

Stephen McQuistin
sjm55@st-andrews.ac.uk

Developing protocol description languages for Internet standards documents

Analysing the social and political process of standards development





Measuring the network to understand the impact of standardisation

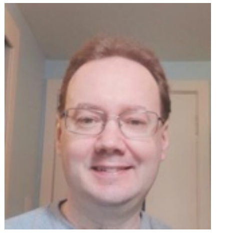
How can we use technology law as a measurement tool?

Tristan Henderson
tnhh@st-andrews.ac.uk

- We can use for instance data portability rights requests to obtain data from data controllers in a machine-readable format.
  - Can be used to understand how systems work, and also to audit them to make sure that they do what they are supposed to be doing.
- Data rights as a tool to measure how the law works in practice
  - e.g. using data rights to measure compliance with GDPR Art 20 (data portability)
  - see IDPL 2019 doi:10.1093/idpl/ipz008
- Data rights as a tool to audit large-scale systems:
  - e.g. using data rights to gather ground truth data on cloud-based health systems
  - See WRAPS 2021 doi:10.1145/3460418.3479343

Programming Languages

Theme Lead: Edwin Brady (ecb10@st-andrews.ac.uk)

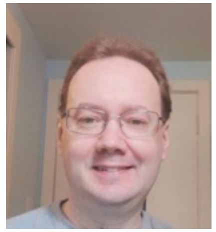


Programming Languages: Overview

- Type Systems (type-driven development), refactoring.
- Parallel and Concurrent programming.
- Energy Efficiency
- Runtime systems, GPUs, hardware architecture.

Type Systems

- Types are a key concept in language design.
- A way of classifying values
  - e.g. numbers, text, functions, . . .
- A well-understood formal method
- We can think of them as lightweight specifications
  - Safe type-directed editing and refactoring
  - Expressing concepts such as: what a program can do, when it is allowed to do it (protocol verification)
- Research problems: expressivity, ergonomics, applications
- Idris (http://idris-lang.org/) is a functional programming language with first class types, supporting type-driven development.
- Development led in St Andrews.
  - Hundreds of contributors (academic and industry)
- Programming as a conversation, led by types as a lightweight specification.



Edwin Brady
ecb10@st-Andrews.ac.uk

Chris Brown
cmb21@st-Andrews.ac.uk

Adam Barwell
adb23@st-Andrews.ac.uk

Parallel and Concurrent Programming

Chris Brown
cmb21@st-Andrews.ac.uk

Adam Barwell
adb23@st-Andrews.ac.uk

- Multi-core CPUs are everywhere, but much easier to program sequentially How to take advantage of parallel architectures
- Language design choices Refactoring, preserving semantics
- Reliability of concurrent systems
- Energy efficiency: can we predict and reduce energy usage of a program, by static analysis?

Hardware Architecture and Runtime

Tom Spink
ecb10@st-Andrews.ac.uk

Susmit Sarkar
ecb10@st-Andrews.ac.uk

- Programs are translated to a form which runs on hardware Semantics must be preserved.
- Challenging on modern, heterogeneous systems!
- Memory consistency models for ARM, RISC-V, IBM POWER machines
- Cache protocol verification Assembly code semantics Efficiency of binary translation

Simon Dobson
Simon.Dobson@st-andrews.ac.uk

Can we Afford AI (and the Rest of Computing)?

Computing is amazing!

- The only technology that can save the planet
  - Simulate and analyse our societal challenges

NASA

- Cloud, machine learning, and artificial intelligence
  - Extend the power of available computing, and the range of potential algorithms we can apply

Wikimedia Commons

- Data collection through sensing
  - Cheap, in the field, embedded, …

The effects

- Changes the way we do science
  - Collect *lots* of data
  - ...and be confident we can actually process it afterwards
  - Machine learning as well as more traditional curve-fitting
- Change the science we do
  - Simulation as a "third pillar" alongside theory and experiment
  - The computer as the *new microscope*

- ...and surveillance capitalism :-(

sciencemuseum.org.uk

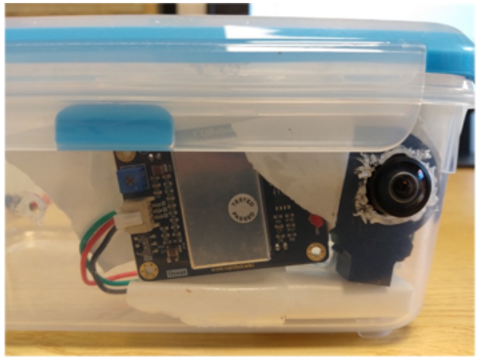
Ubiquitous computing

- General-purpose computing
  - Fast, convenient, elastic
  - Don’t have to commit ahead of time

Chilton computing

Marco Herrera *via* Wikipedia

- Some applications can “pre-commit”
  - Sensors in the field, wearables, embedded, …
  - The environment constrains the power envelope
  - Cheap and flexible for field use



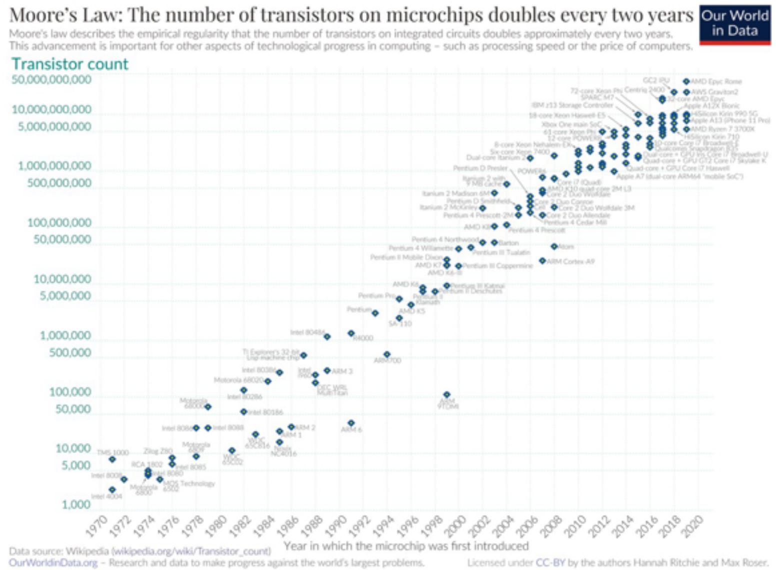
But There’s a Problem

- Power consumption is a massive constraint on computing futures
  - Unsustainable energy, water, and resource footprints
- At scale
  - “In the cloud” simply means “out of sight, out of mind”
  - ...but AI compute demands double every 100 days
  - ...and currently accounts for 2.1–3.9% of all greenhouse gas emissions
- Pervasively, in people and the environment
  - Can’t collect, store, or process the data as we want to, sustainably

According to the World Economic Forum

Where Does It Come From?

- (Computing) power costs (electrical) power
  - Moore's law has a power analogue, *via* the Second Law of Thermodynamics
  - Modern chips are massively more efficient – but still draw massively more power and need to shed massively more heat



Wikipedia

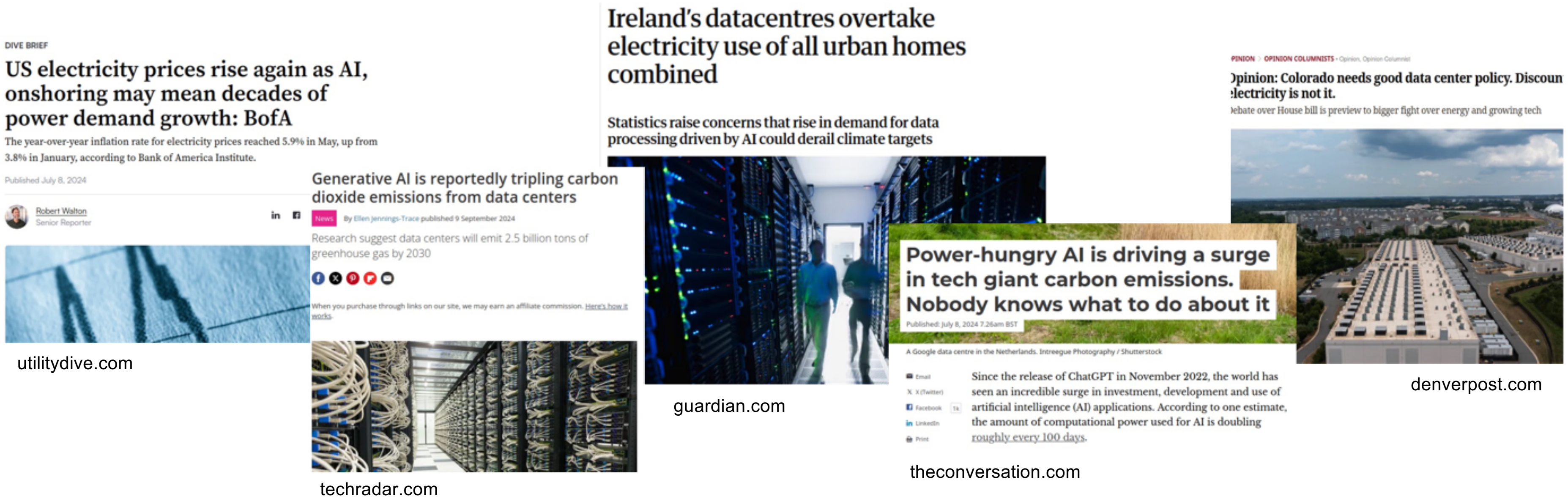
Wikipedia

quietpc.co.uk

techpowerup.com

...and it *needs* fixing urgently

- We're starting to see pushback
  - Economics favours consolidation into "hyperscalers"
  - And the "digital exhaust" remains too valuable to lose



...and is often amazingly mis-diagnosed

Extreme Weather Is Taxing Utilities More Often. Can A.I. Help?

nytimes.com

...and there are pressures the other way

- Commercial concerns mitigate against even *trying* to address the issue
  - Infinite demand for AI implies an infinite power demand
  - Grows faster than we can reduce power consumption in (all) other areas

- Should we *really* give up before we've properly tried?

https://x.com/tsarnick/status/1842401670225125539

Where is the power consumed?

- Computing in data centres can be very efficient and secure
  - ...but also concentrates the load on the grid
  - ...and makes a valuable target

- Distributed computing reduces (literal) hot-spots
  - ...at a cost of requiring more power *everywhere*
  - ...and requiring more user effort to keep secure

Distributing AI

- “Older” models like ChatGPT require enormous computing and storage
  - Especially during training, ut increasingly during operation
  - Recent advances like “test-time training” increase power load

- “New” models like DeepSeek-v3 appear to be far cheaper
  - Less training time, less storage, …
  - Some very novel architectural decisions, like using 8-bit floating-point numbers for weights

Our interest

- A topic we're becoming increasingly interested in
  - Keep the benefits of computing...
  - ...but stop exacerbating the problems we're trying to solve

What do ***advanced computing futures*** with ***radically reduced environmental footprints*** look like?

- Across all our research
  - Computer systems
  - Programming languages
  - AI
  - HCI
  - Digital health

Vision: A holistic view of computing futures

- How should we program?
- Do we need new platforms?
- Where do AI and machine learning fit?
- How do I simulate complex systems?
- How do analyse lots of data?

- Impart sustainability at the centre of our world-leading student experience

St Andrews / CS

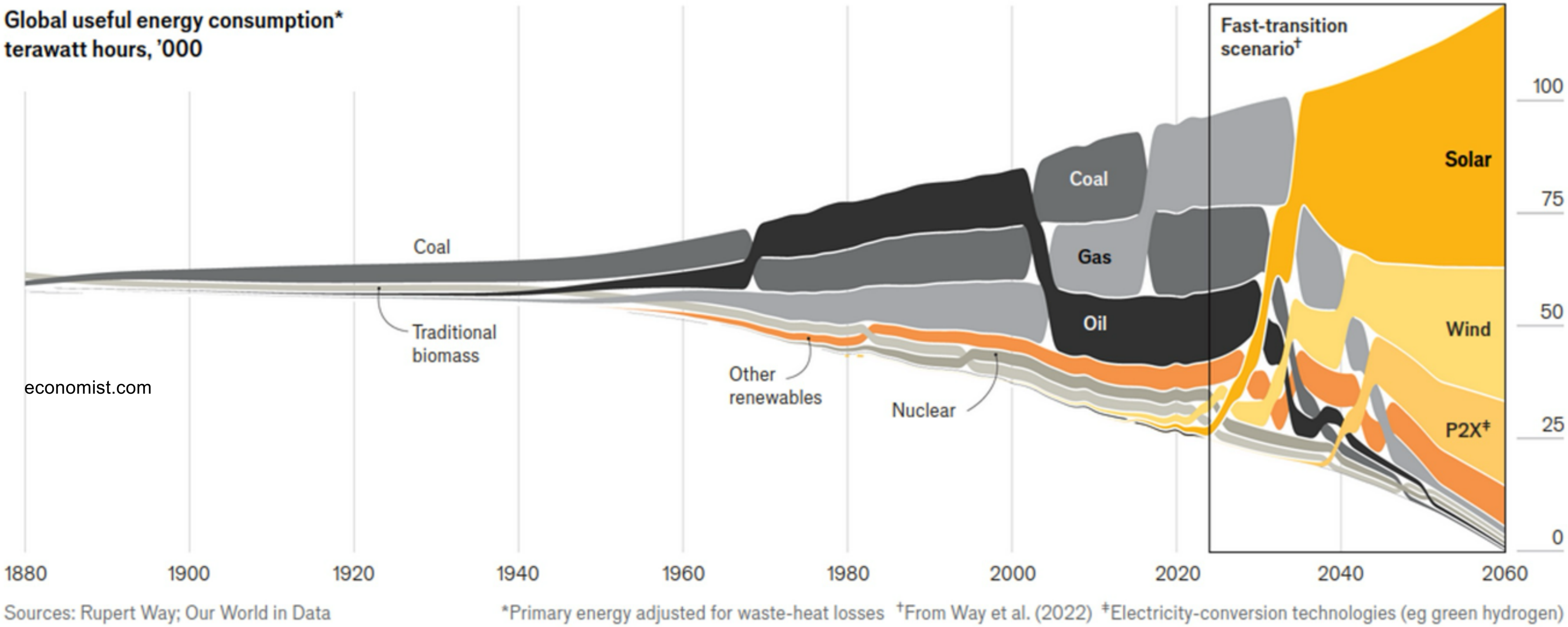
W I C S

WOMEN IN COMPUTER

StACCS
St Andrews Centre for Critical Sustainabilities

The entwinedness of low power

- Solutions come from interactions across our main research themes:
- A lot of computing in the field can be powered from local low-voltage solar – even in Scotland. If:
  - We can design algorithms to work in a compatible way.
  - We can design algorithms to work in a compatible way.
  - We keep machine learning for what it's really good at.



Low power @ St Andrews

- There’s a global challenge to be addressed
  - Arguably the *only* significant computing challenge is power consumption
- A holistic take
  - Leverage our small size to work together
  - Positioning to train the next generation of young minds
  - Re-centre the computing curriculum while
    driving the technology research

Part I: Summary

Summary

- We are **eager to collaborate**!
- If you have seen anything that might interest you, please do get in touch and I will connect you to the right person/group.

Part II: Automated Constraint Modelling & Solving

Constraint Programming in a Nutshell

Constraint Programming

- An active field of **Artificial Intelligence** in which we study how to model and solve constraint satisfaction problems.
- Subject of major investment from industry.
- E.g. Google OR-tools, IBM CP-optimizer

Constraints: A Natural Means of Knowledge Representation

- $x + y = 30$
- Adjacent countries on map cannot be coloured same.
- The telescope must be observing a particular star at a particular time.
- The deployed application:
  - Requires at least 2GB of memory.
- This set of applications:
  - Must be deployed in the same region.

St Andrews University Observatory

The Constraint Satisfaction Problem

- A general way in which we can represent and solve decision-making problems:
- **Given:**
  1. A set of **decision variables**.
  2. For each decision variable, a **domain** of potential values.
  3. A set of **constraints** on the decision variables.
- **Find:**
  - An **assignment** of values to variables such that all constraints are satisfied.

1. Decision Variables

- A decision variable corresponds to a **choice** that must be made in solving a problem.
- In university timetabling we must decide, for example:
  - The time for each lecture.
  - The venue for each lecture.
  - The lecturer for each lecture.
  - …

2. Domains

- **Values** in the domain of a decision variable correspond to the **options** for a particular choice.
- E.g. Decide lecture time.
  - Values in this domain:
    9am, 10am, ..., 5pm
- E.g. lecture venue.
  - Values in this domain:
    theatre A, theatre B, ...
- A decision variable is **assigned** a **single** value from its domain.
  - Equivalently: the choice associated with that variable is made.

3. Constraints

- **scope**: subset of the decision variables a constraint involves.
- Of the possible combinations of assignments to the variables in its scope, a constraint specifies:
  - Which are allowed.
    Assignments that **satisfy** the constraint.
  - Which are disallowed.
    Assignments that **violate** the constraint
  - I.e. can think of a constraint as a relation.
- E.g. if variables t_A, t_B, represent time for lectures A, B, both taken by student S:
  - $\mathbf{t_A \neq t_B}$ (student S can't be in two places at once!)

The Constrained Optimisation Problem (COP)

Given:

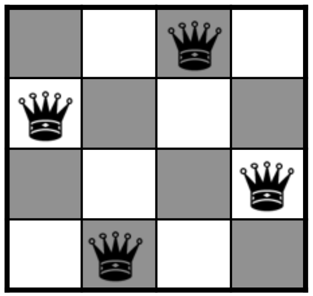
- A CSP + an **objective function**.
  - E.g. maximise/minimise value of some variable/expression.
  - In our example: maximise preferences of lecturers.

Find:

- An assignment of values to variables such that:
  - All constraints are satisfied.
  - The objective is optimised.

Problem Classes

- A problem class describes a family of problems, related by a common set of **parameters**.
- Obtain an instance: give values for the parameters.
- Example: *n*-queens problem **class**.
  Place *n* queens on an *n* x *n* chess board such that no pair of queens attack each other.
- Here is a solution to the 4-queens **instance**.



The Sudoku Problem Class

- Sudoku is parameterised by the set of filled-in cells in the grid:

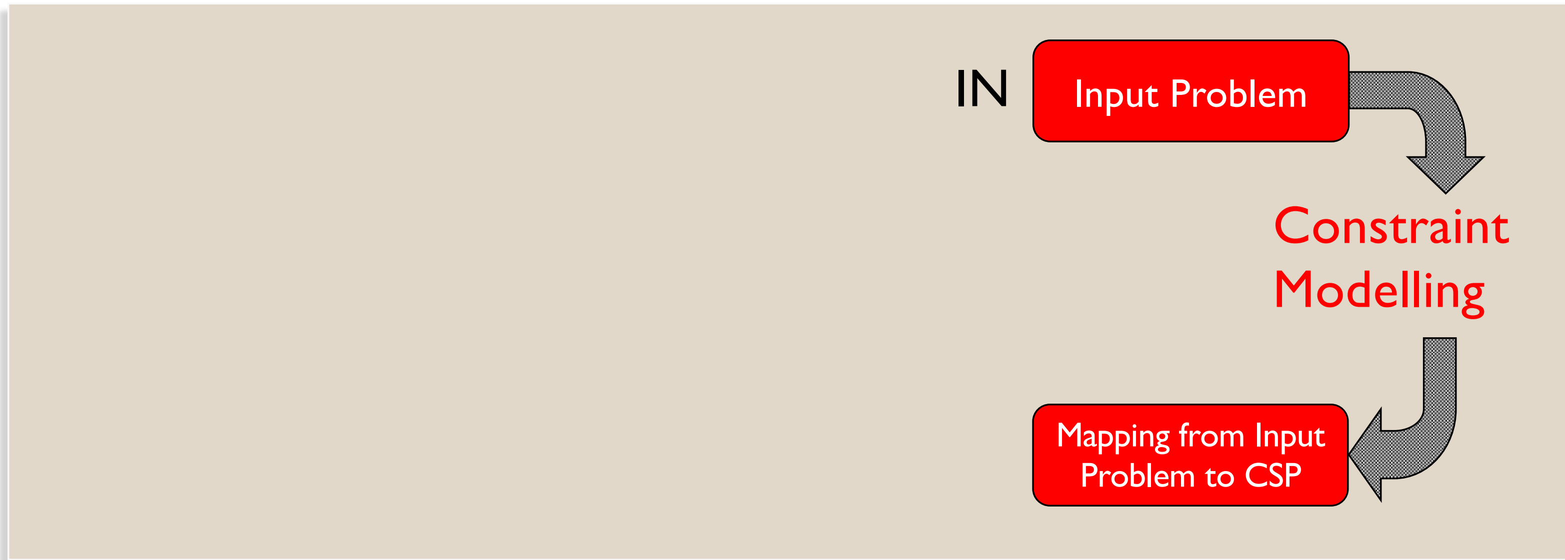
	2	6				8	1	
3			7		8			6
4				5				7
	5		1		7		9	
		3	9		5	1		
	4		3		2		5	
1				3				2
5			2		4			9
	3	8				4	6	

Solving Problems with Constraints

- An efficient means of finding solutions to combinatorial problems.
  - Planning, Scheduling, Design, Configuration, …
- Two phases:

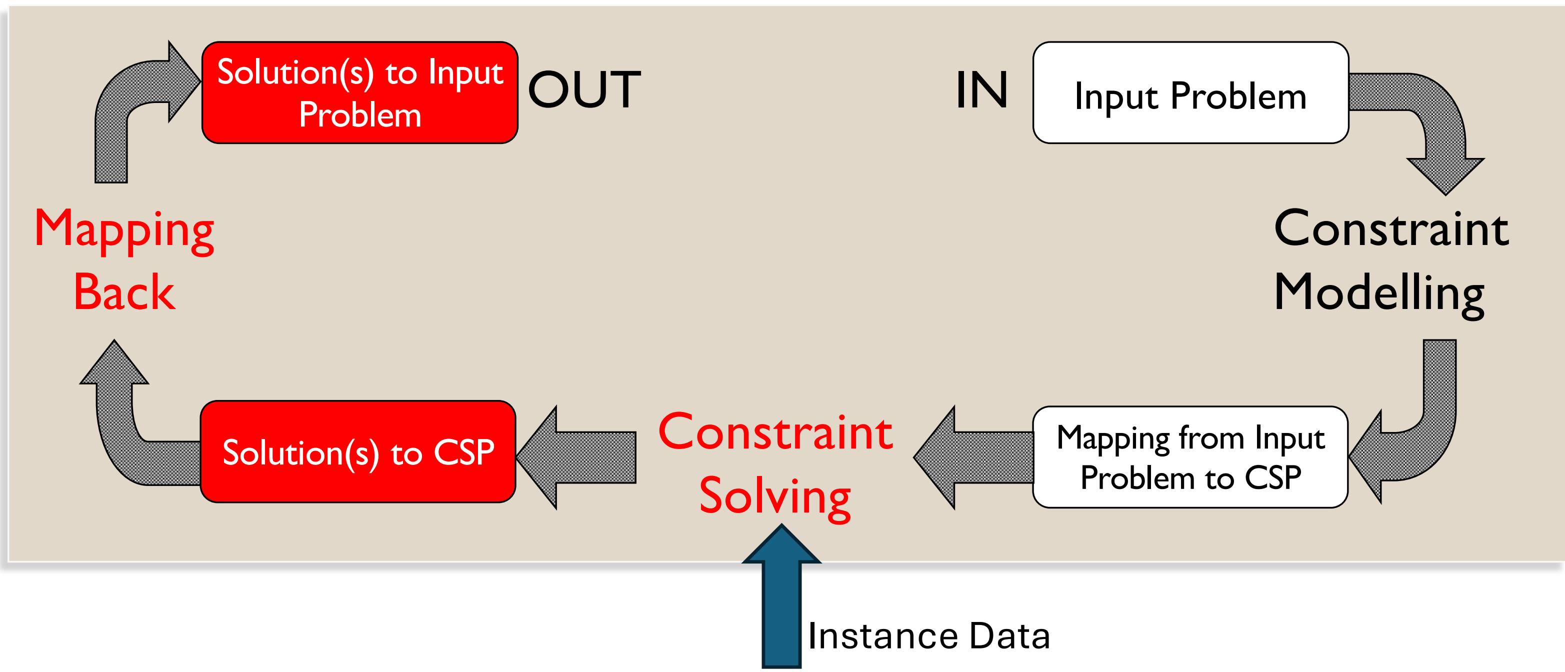
1. **Describe** the problem to be solved as a **constraint model**, a format suitable for input to a **constraint solver**.
2. **Search** (automatically) for solutions to the model with a constraint solver.

Constraint Modelling



- A constraint model maps the features of a given problem onto the features of a constraint satisfaction problem.

Constraint Solving

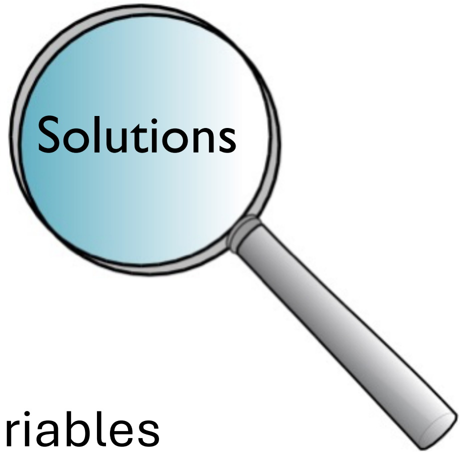


- The CSP is input to a **constraint solver**, which produces a solution (or solutions).

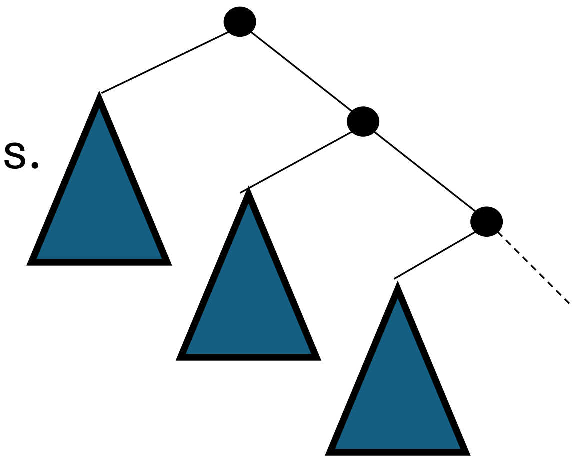
Finding Solutions

- How does a constraint solver go about finding solutions?
- It combines:
  - **Search** (guesses ☺), with
  - **Deduction** (ruling out values it can prove cannot be part of a solution, based on the decisions made so far).

In More Detail



1. Systematic **Search** through a space of partial assignments.
   - Extend an assignment to a subset of the variables incrementally.
   - Backtrack if establish that current partial assignment **cannot** be extended to a solution.
2. Constraint **Propagation**.
   - Deduction based on constraints, current domains.
   - Usually recorded as reductions in domains.



Representing Constraints

- Constraints are relations and can be represented by listing allowed/disallowed tuples of values (called “Table” constraints).
  - Cumbersome, e.g. Sudoku.
- Instead, constraint solvers provide a **library** of commonly-occurring constraints that can be specified much more concisely.
  - E.g. AllDifferent.
- Internally, the solver usually represents these constraints **Intensionally**:
  - An expression that can be evaluated:
    - E.g. **=, <, ≤, ≠.**
  - An algorithm that can be executed:
    - AllDifferent, various kinds of counting constraints.

	2	6				8	1	
3			7		8			6
4				5				7
	5		1		7		9	
		3	9		5	1		
	4		3		2		5	
1				3				2
5			2		4			9
	3	8				4	6	

Constraint Languages and Constraint Programs

- We do not usually work directly with CSP/COPs, which can be large and cumbersome.
- Instead we write **constraint programs** (also known as **constraint models**) in **constraint languages**.
- A constraint program/model is a **recipe**.
  - When followed, produces a CSP/COP.

Constraint Languages: Common Features

- Allow us to declare decision variables, and their domains.
- Often support arrays of variables.
  - And **iteration** over these arrays for concision.
- Allow us to model problem **classes**.
  - I.e. allow us to specify parameters.

Constraint Languages: Common Features

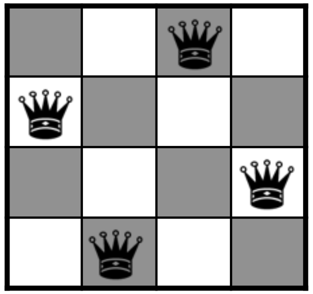
- All support **table** constraints.
  - These are our basic building blocks.
- Equality, disequality, inequality.
- Operators allow us to build constraint expressions:
  - Arithmetic: $+$, $-$, $\times$, absolute value.
  - Logical: AND, OR, NOT
- These constraints are represented **intensionally**.

Automated Constraint Modelling

The Importance of Modelling

- There are typically many ways to formulate a constraint model of a problem.
  - What should the **variables** be? Their **domains**?
  - How should we express the **constraints**?
- These choices have a substantial effect on the efficiency of the solving process.

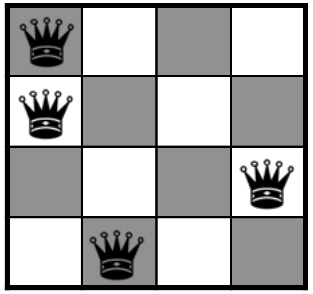
Importance of Modelling: Example



- How should we model this problem?
- We should always begin by asking ourselves what decisions we need to make, and so what the **variables** are:
  - A variable per square, domain is on/off for queen or not.
  - A variable per queen, domain is square.
  - A variable per row, domain is position of queen in that row.
    - Since we know that each row must contain a queen.

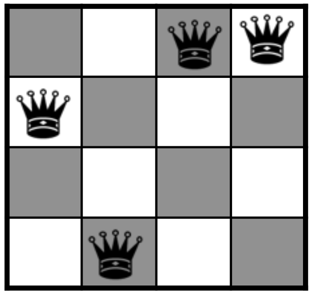
Importance of Modelling: Example

- A variable per row, domain is position of queen in that row.
  - Since we know that each row must contain a queen.
- The reason this is the best of the three is symmetry:
  - A symmetry is a solution-preserving transformation.



Not a solution

Rotate 90 degrees clockwise:



Also not a solution
BUT **cannot** be represented in our row-based model

Importance of Modelling: Example II

- Let’s say our problem requires us to model a set with four elements.
  - Could be the contents of a bin, packages assigned to a van, items held by a robot, …
- And our constraint modelling language doesn’t support sets directly.
- So it is natural to use a 1-dimensional array of four elements:

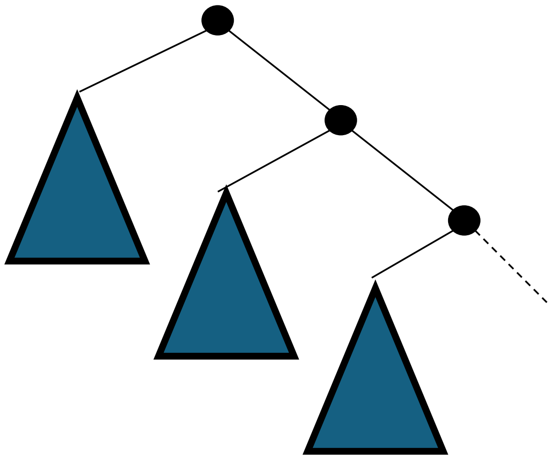
1	2	3	4
a	b	c	d

And add an all-different constraint

Importance of Modelling: Example II

- But there’s a problem. The original set has no indices but the array does.
- This is the same set:

1	2	3	4
a	b	c	d



- as:

1	2	3	4
b	c	a	d

- Why a problem? Because if the set {a,b,c,d} is not part of a solution the solver may have to search through all of its symmetrical equivalents to establish that.

Importance of Modelling: Example II

- So what can we do?
- One option is to insist the set is presented in ascending order:

1	2	3	4
a	< b	< c	< d

Notice hear that we can dispense with the all-different constraint

- Or switch to a characteristic function-like representation:

a	b	c	d	e	f
1	1	1	1	0	0

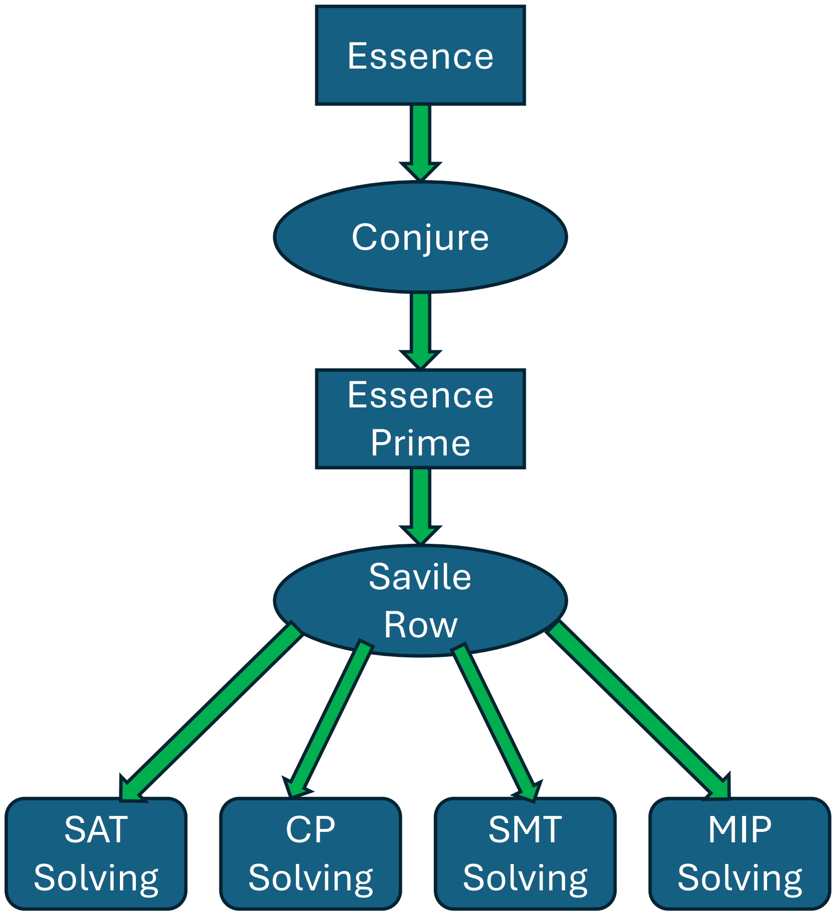
- Which is best depends on the set we want to represent, the constraints on the set…

Automating Modelling.

- The preceding examples illustrate that some expertise is needed in formulating a model.
- Can we automate any of this process?
- Several ideas:
  - Automate the improvement of a given model.
  - ”Learn” the model by querying the user with example assignments.
    - ConAcq and its descendants.
  - Get LLMs to model the problem.
- We have a different approach…

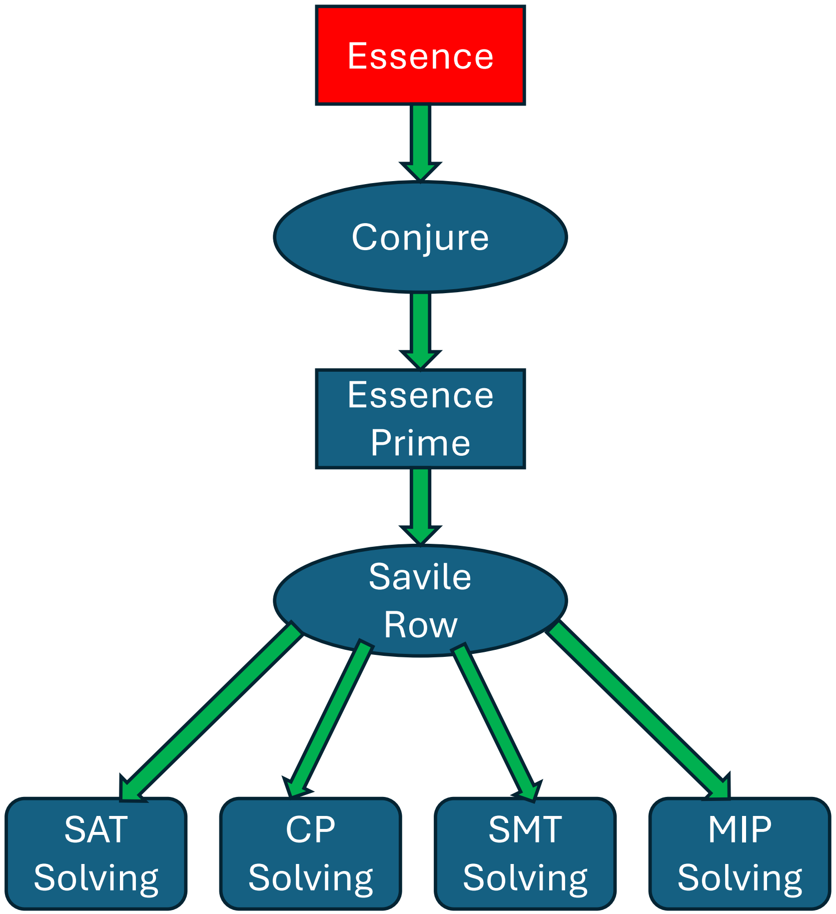
A Constraint Modelling Pipeline

Our Constraint Modelling Pipeline



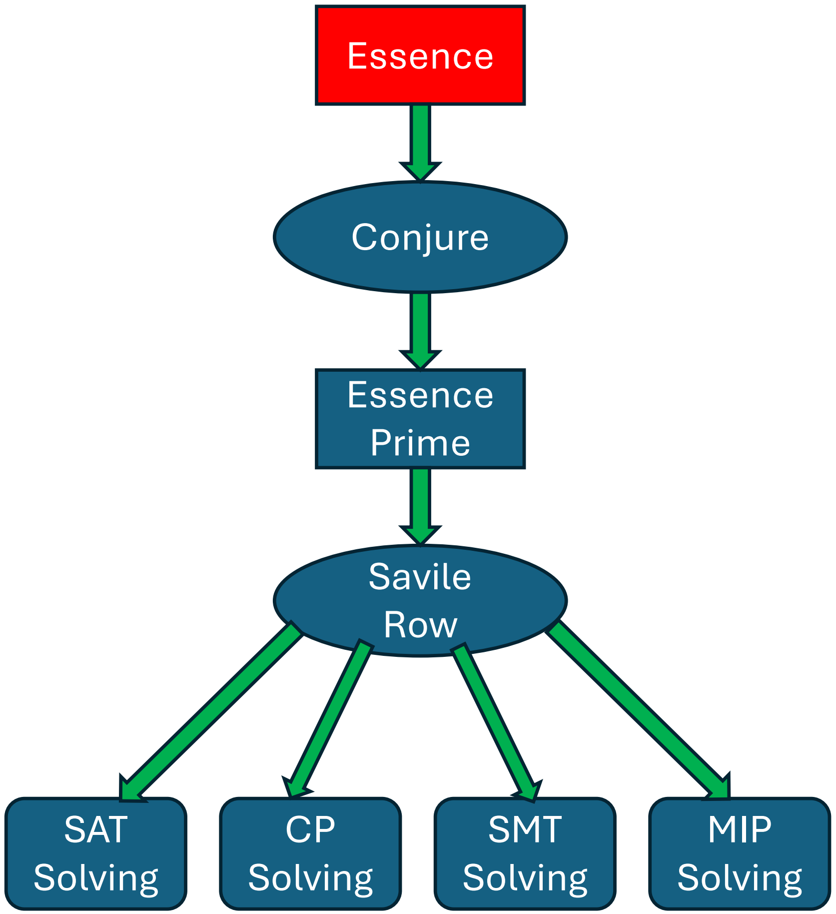
- Overview:
  - We capture an abstract specification of a constraint problem in the **Essence** language.
    - Abstract: without committing to detailed modelling decisions.
  - This is automatically refined into a solver-independent constraint model in **Essence Prime**.
  - And then tailored for a particular solver type.

Constraint Modelling Pipeline: Essence



- An abstract constraint specification language.
- Domain constructors, such as set, function, sequence, partition, relation, …
  - Arbitrary nesting of these: set of sets, sequence of functions, …
- Attributes of these domains:
  - Injective function, symmetric relation.
- Constraints/Operators on these domains:
  - Projection on relations.
  - Range of function.

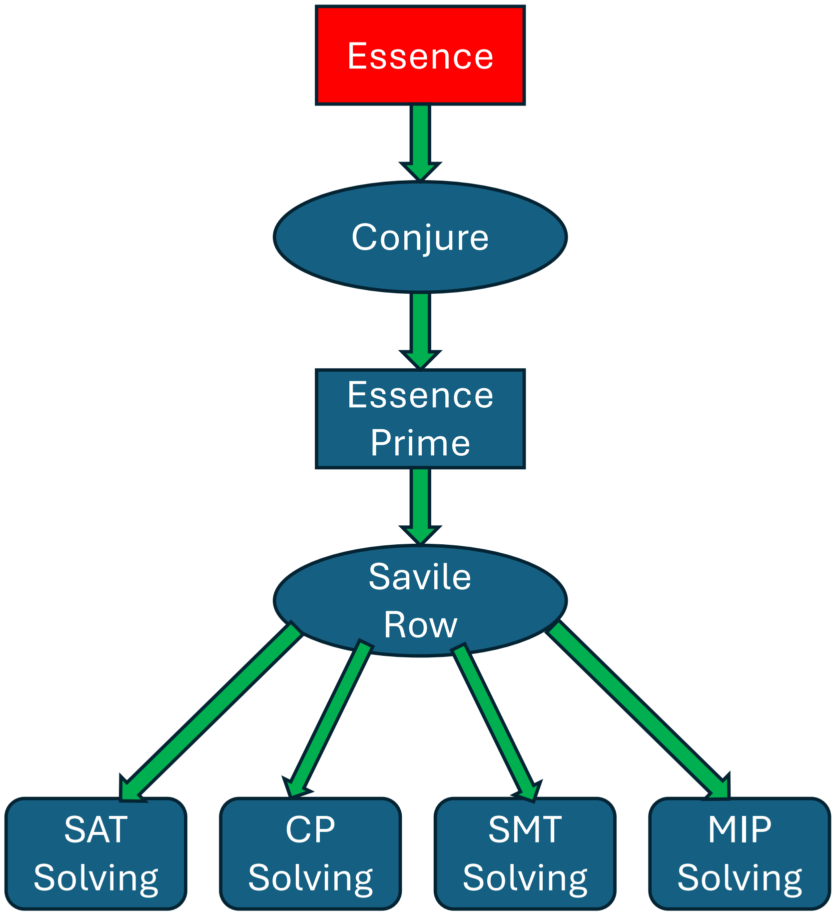
Constraint Modelling Pipeline: Essence



- Example: Social Golfers Problem.
- In a golf club there are a number of golfers who wish to play together in **g** groups of size **s**. Find a schedule of play for **w** weeks such that no pair of golfers play together more than once

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
                 partition (regular, numParts g, partSize s)
                 from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Essence

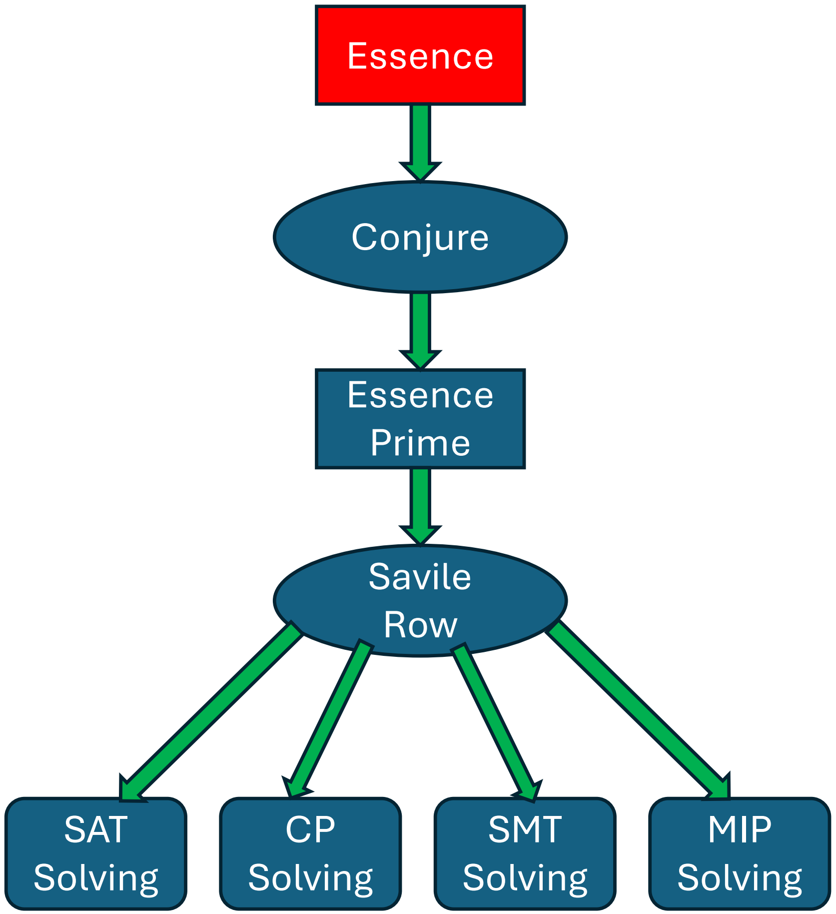


- Example: Social Golfers Problem.
- In a golf club there are a number of golfers who wish to play together in **g** groups of size **s**. Find a schedule of play for **w** weeks such that no pair of golfers play together more than once

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
             partition (regular, numParts g, partSize s)
             from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Integer parameters

Constraint Modelling Pipeline: Essence

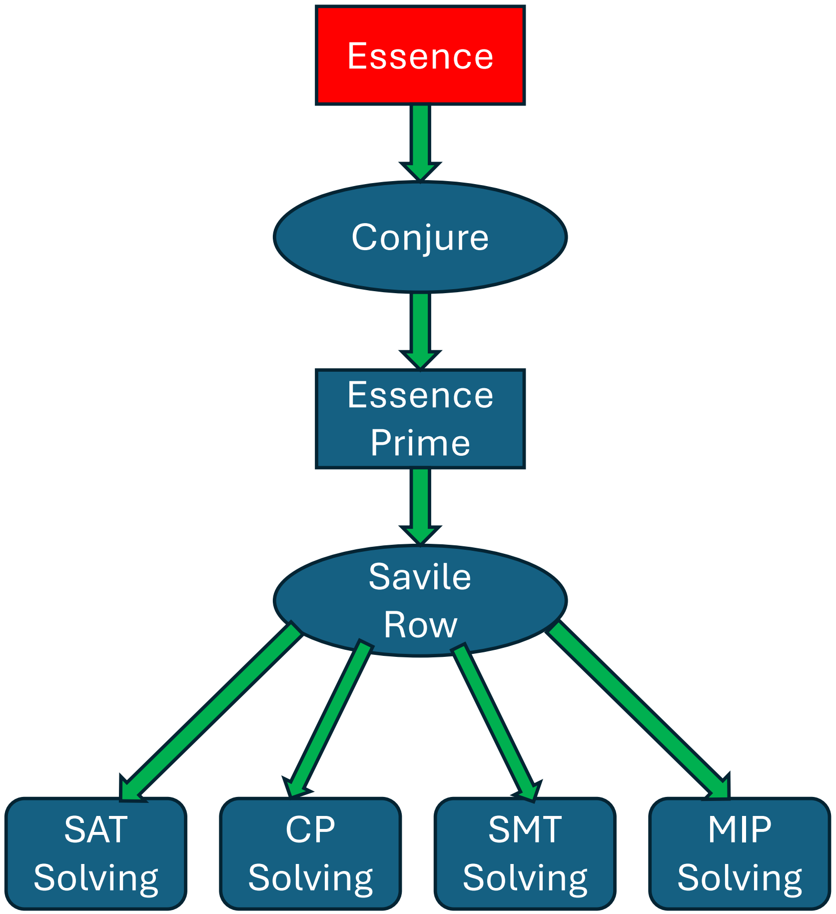


- Example: Social Golfers Problem.
- In a golf club there are a number of golfers who wish to play together in **g** groups of size **s**. Find a schedule of play for **w** weeks such that no pair of golfers play together more than once

Individual golfers don't need to be identified
Symmetry avoided.

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
             partition (regular, numParts g, partSize s)
             from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Essence

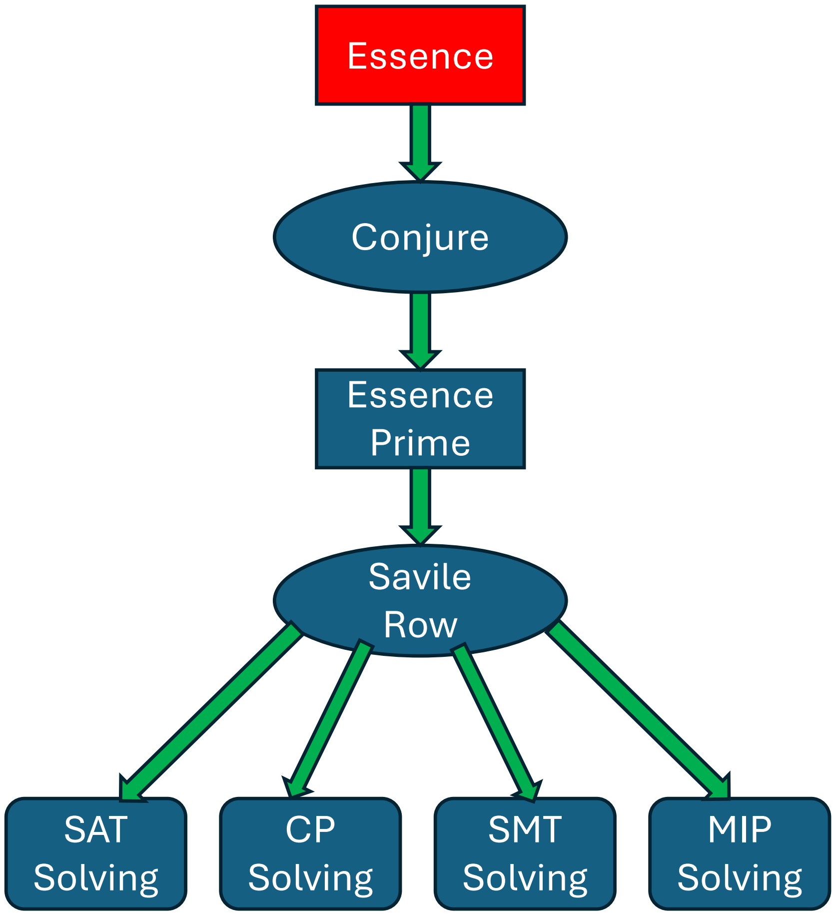


- Example: Social Golfers Problem.
- In a golf club there are a number of golfers who wish to play together in **g** groups of size **s**. Find a schedule of play for **w** weeks such that no pair of golfers play together more than once

One highly-structured decision variable.

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
             partition (regular, numParts g, partSize s)
             from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Essence



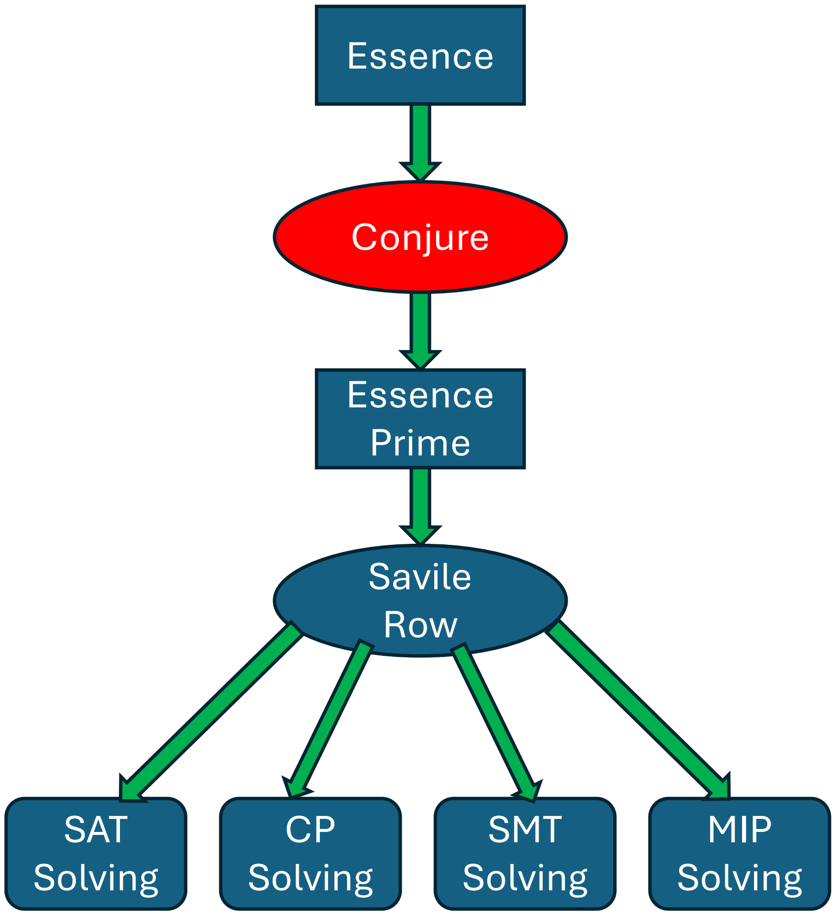
- Example: Social Golfers Problem.
- In a golf club there are a number of golfers who wish to play together in **g** groups of size **s**. Find a schedule of play for **w** weeks such that no pair of golfers play together more than once

The **socialisation** constraint

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
                partition (regular, numParts g, partSize s)
                from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

NB Having described the combinatorial structure to be found using Essence's types this is the only constraint left to be stated.

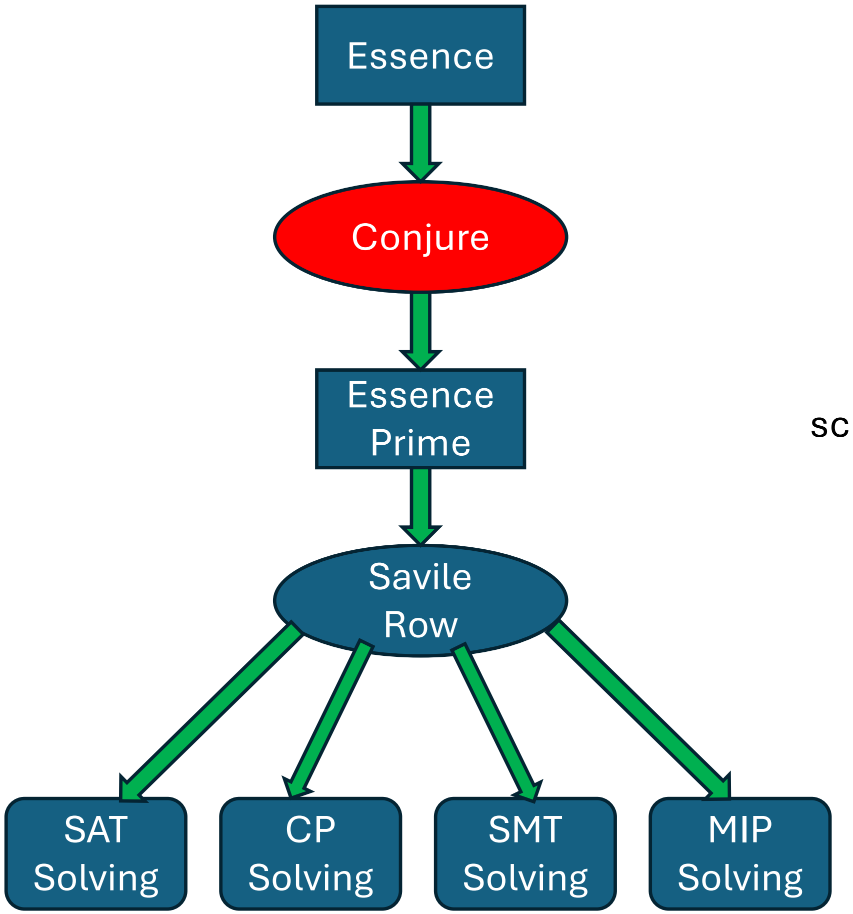
Constraint Modelling Pipeline: Conjure



- We can’t typically solve an Essence specification directly.
- We use the **Conjure** system to **refine** an Essence specification into **Essence Prime**.
  - A subset of Essence with facilities common in constraint modelling languages.
  - (Matrices of) Integer, Boolean variables.
  - Logical, Arithmetic, Global Constraints.

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
               partition (regular, numParts g, partSize s)
               from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure



- Refinement proceeds from the choice of representation of the decision variables.
- The outer structure of sched here is a **fixed-cardinality set**.
- A natural model is via a matrix:

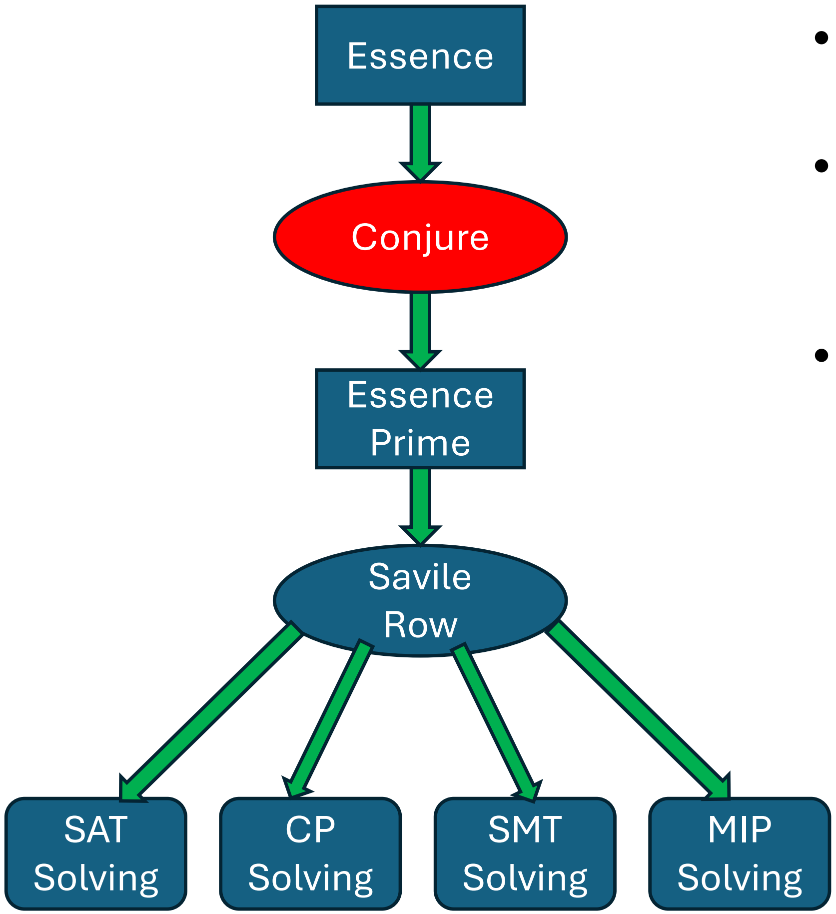
sched

1	2	…	w-1	w
<partition>	<partition>	…	<partition>	<partition>

Structural constraint: AllDifferent(sched)

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
             partition (regular, numParts g, partSize s)
             from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure



- Key **advantage** of refinement-based approach:
  - Recognise and break **symmetry** as it enters the model.
  - By refining a set to an indexed matrix we introduce symmetry: permuting the weeks is solution-preserving.
  - Conjure knows this and adds constraints to break this symmetry:

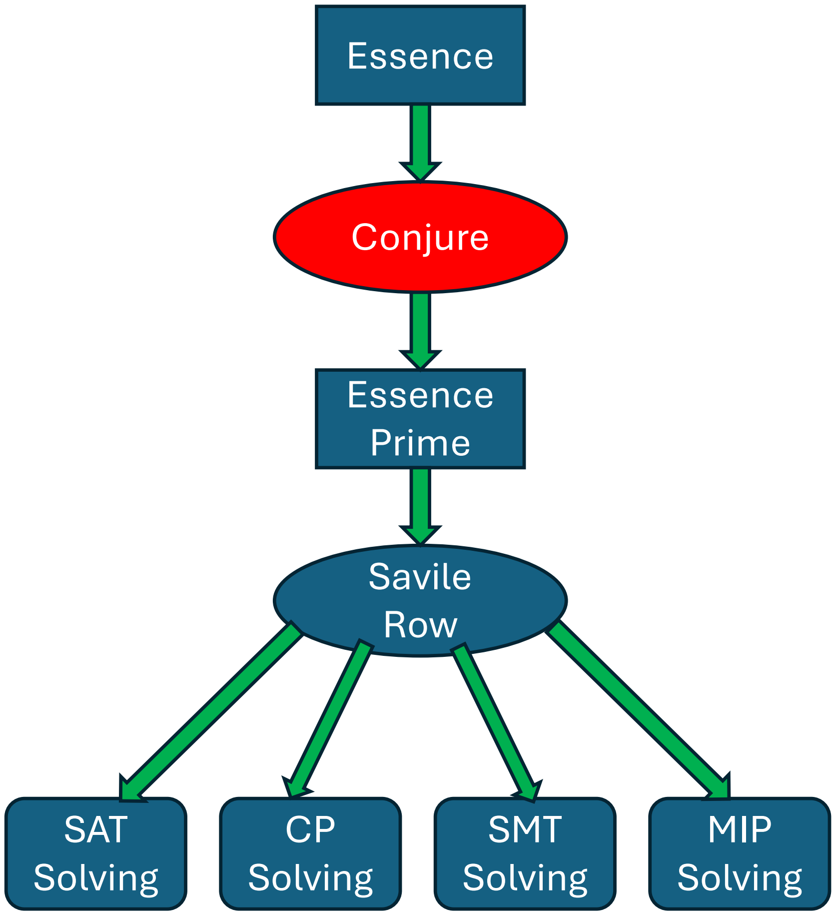
< < < <

1	2	...	w-1	w
<partition>	<partition>	...	<partition>	<partition>

(and the AllDifferent is automatically removed)

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
             partition (regular, numParts g, partSize s)
             from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure



- We can think of a partition as a constrained set of cardinality **g** of sets of cardinality **s**:

Parts (i.e. groups)

Domain: {1, …, g x s}
Where g x s is the number of golfers

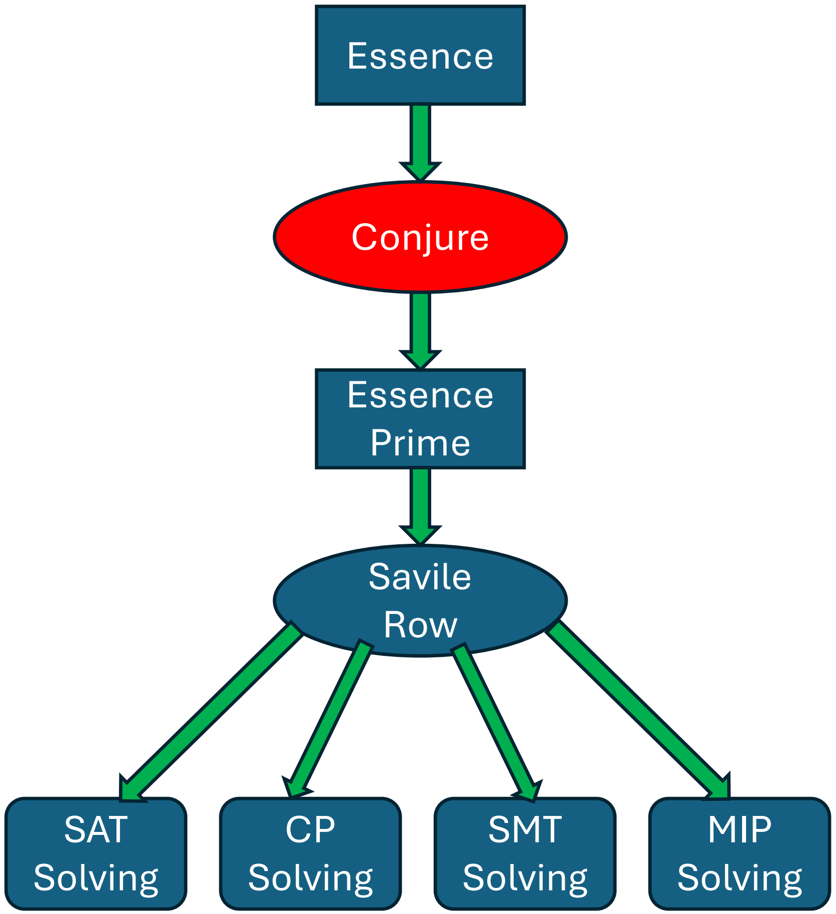
Elements of each part (i.e. golfers)

Again, breaking **symmetry**. Among columns using **lexicographic ordering.**

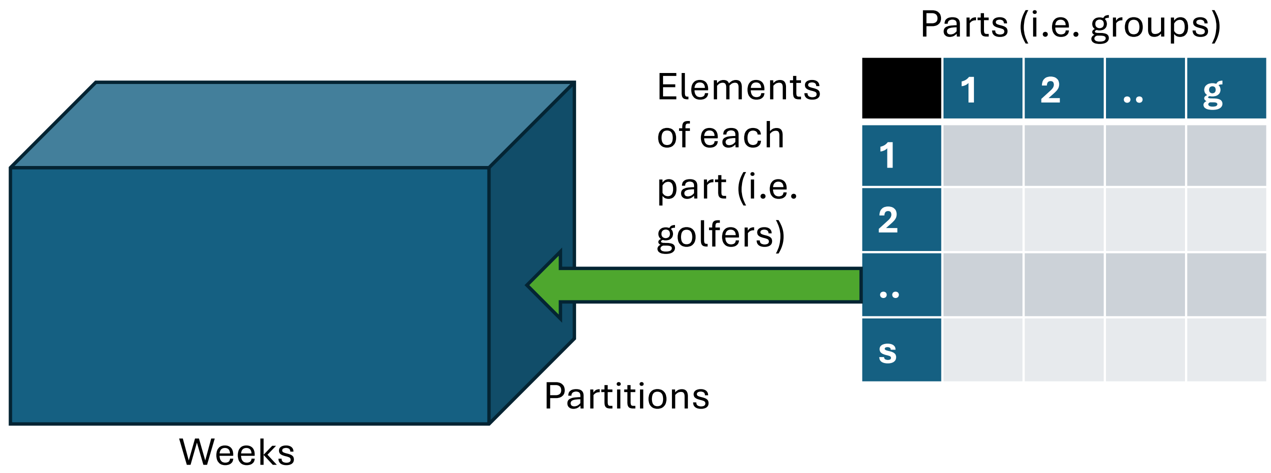
Structural constraint: AllDifferent

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
               partition (regular, numParts g, partSize s)
               from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure

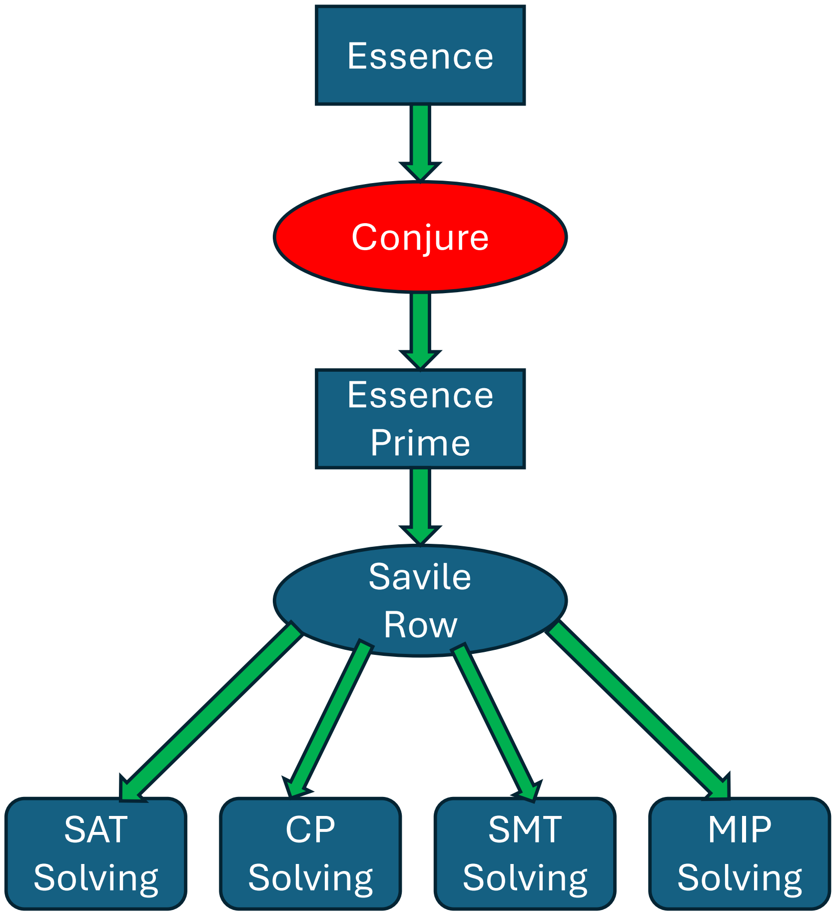


- Giving (a) representation of sched:




```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
               partition (regular, numParts g, partSize s)
               from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure



- Conjure then refines the constraints to suit the representation chosen:

Parts (i.e. groups)

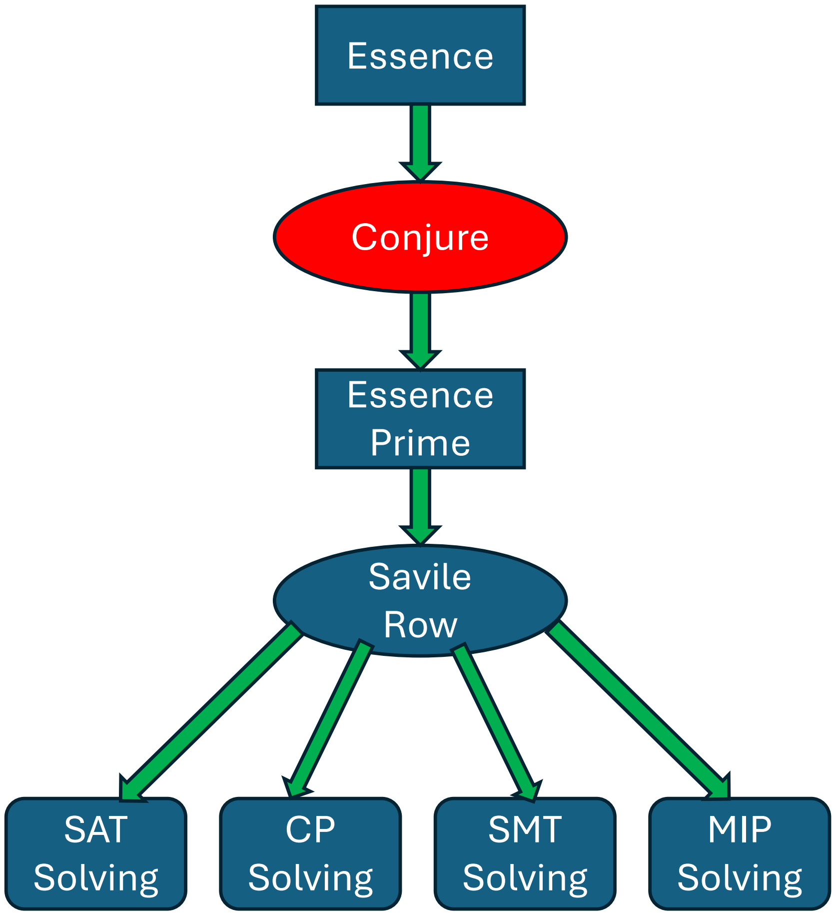
Elements of each part (i.e. golfers)

	1	2	..	g
1	4			
2	5			
..				
s				

- **Disallow 4, 5 in the same group in any other week**
- **How:**
    - **Represent the intersection between parts in different weeks.**
    - **Ensure size at most 1.**

```
language Essence 1.3
given w, g, s : int(1..)
letting Golfers be new type of size g * s
find sched : set (size w) of
                 partition (regular, numParts g, partSize s)
                 from Golfers
such that
  forAll g1, g2 : Golfers, g1 < g2 .
    (sum week in sched . toInt(together({g1, g2}, week))) <= 1
```

Constraint Modelling Pipeline: Conjure



- Conjure has alternative refinement rules for both decision variable and constraint representation.
- Allows us to explore the **space of models**.
- Heuristics to select models likely to be effective.

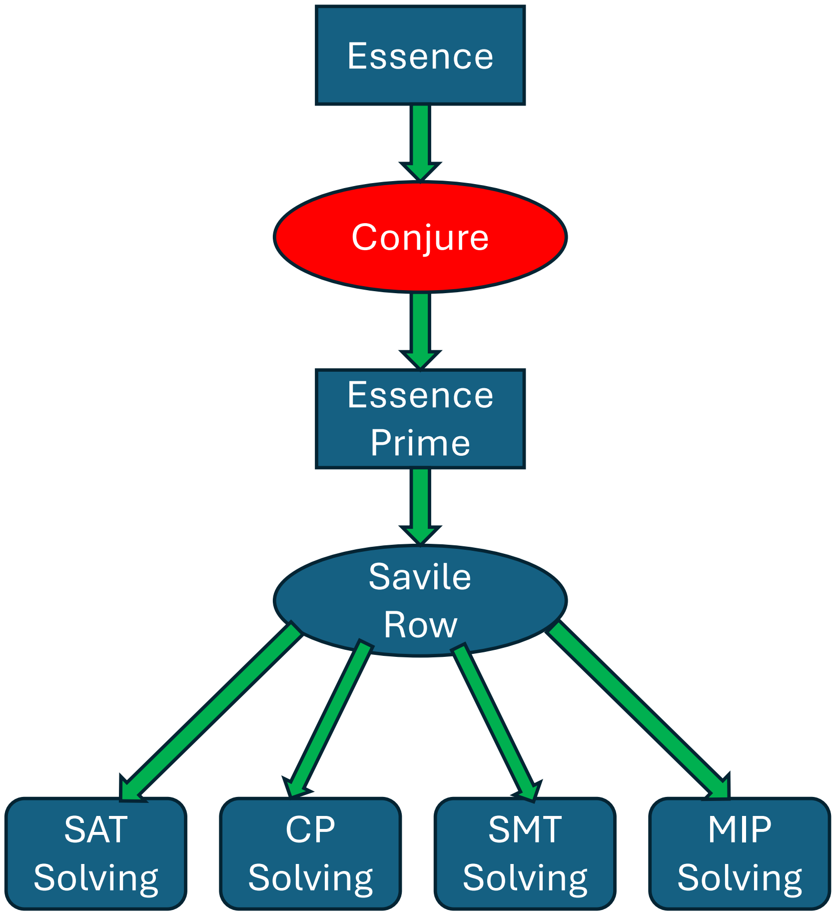
Parts (i.e. groups)

Elements of each part (i.e. golfers)

	1	2	..	g
1	0/1			
2	0/1			
..				
gxs	0/1			

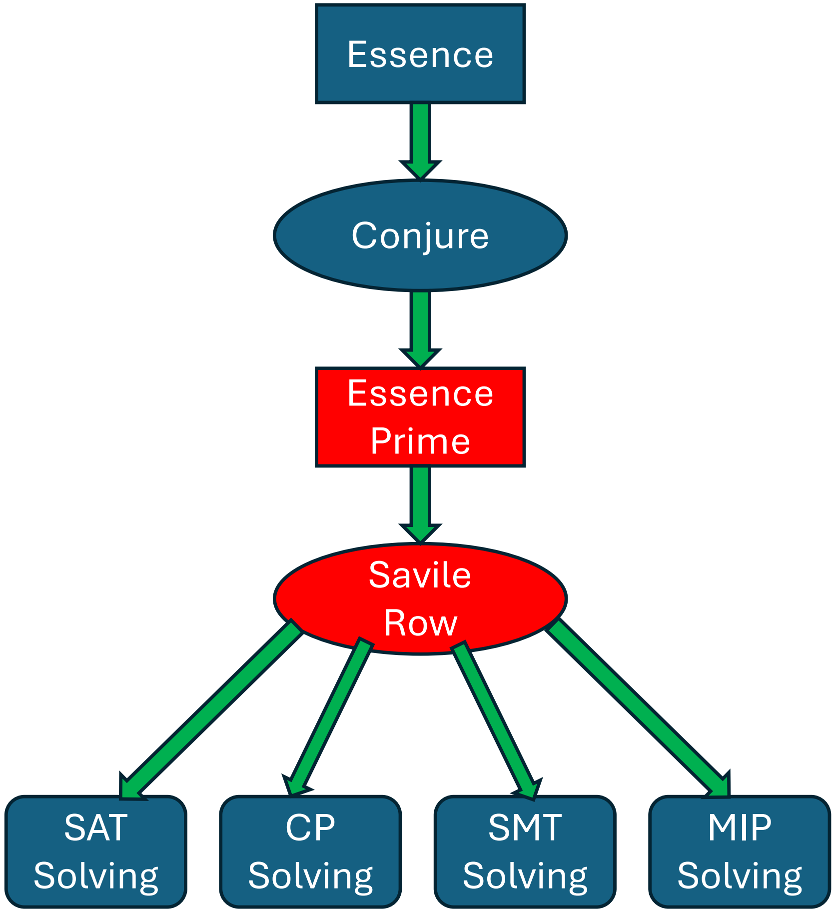
Structural: each column sums to s.

Constraint Modelling Pipeline: Conjure



- We refined Essence specifications of 42 benchmark problems from CSPLib.
- In doing so, naturally employed all six of the abstract types in Essence:
  - Set, multiset, sequence, function, relation, and partition.
- Confirmed that Conjure can generate the kernels of models written by human experts.
  - I.e. not including implied constraints that might require arbitrarily complex chains of reasoning.

Constraint Modelling Pipeline: Savile Row



Social Golfers	3 weeks		
3 groups, size 3	[1, 2, 3]	[1, 4, 7]	[1, 5, 9]
	[4,5,6]	[2,5,8]	[2,6,7]
	[7,8,9]	[3,6,9]	[3,4,8]

Solution to an instance of Social Golfers

- The Essence Prime model is close to the input of a constraint solver.
- **Savile Row** is responsible for:
  - Tailoring this model to a particular solver
  - Or encoding to a different formalism.
- While further enhancing the model.
  - E.g. **Common subexpression elimination**, **tabulation**.

Key References

https://github.com/conjure-cp

- **Essence: A Constraint Language for Specifying Combinatorial Problems.** AM Frisch, W Harvey, C Jefferson, B Martinez-Hernandez, I Miguel. Constraints 13, 268-306, 2008.
- **Conjure: Automatic generation of constraint models from problem specifications**. Ö Akgün, AM Frisch, IP Gent, C Jefferson, I Miguel, P Nightingale. Artificial Intelligence 310, 103751, 2022.
- **Automatically improving constraint models in Savile Row**. P Nightingale, Ö Akgün, IP Gent, C Jefferson, I Miguel, P Spracklen. Artificial Intelligence 251, 35-61, 2017.

Directions

- Model selection:
  - Conjure has heuristics, we are exploring using ML.
  - This in turn requires training data:
    - **A framework for generating informative benchmark instances**. N Dang, O Akgun, J Espasa, I Miguel and P Nightingale. CP 2022.
- Solving Essence directly:
  - A new solver called Athanor:
    - **Athanor: Local Search over Abstract Constraint Specifications**. S Attieh, N Dang, C Jefferson, I Miguel, P Nightingale. Artificial Intelligence 340, 104277, 2025.
- Streamlining:
  - **Automated streamliner portfolios for constraint satisfaction problems**. P Spracklen, N Dang, Ö Akgün, I Miguel. Artificial Intelligence 319, 103915, 2023.


University of
St Andrews
www.st-andrews.ac.uk