Machine Learning for Computer Algebra

Rashid Barket, Tereso del Río and Matthew England

Centre for Computational Science and Mathematical Modelling

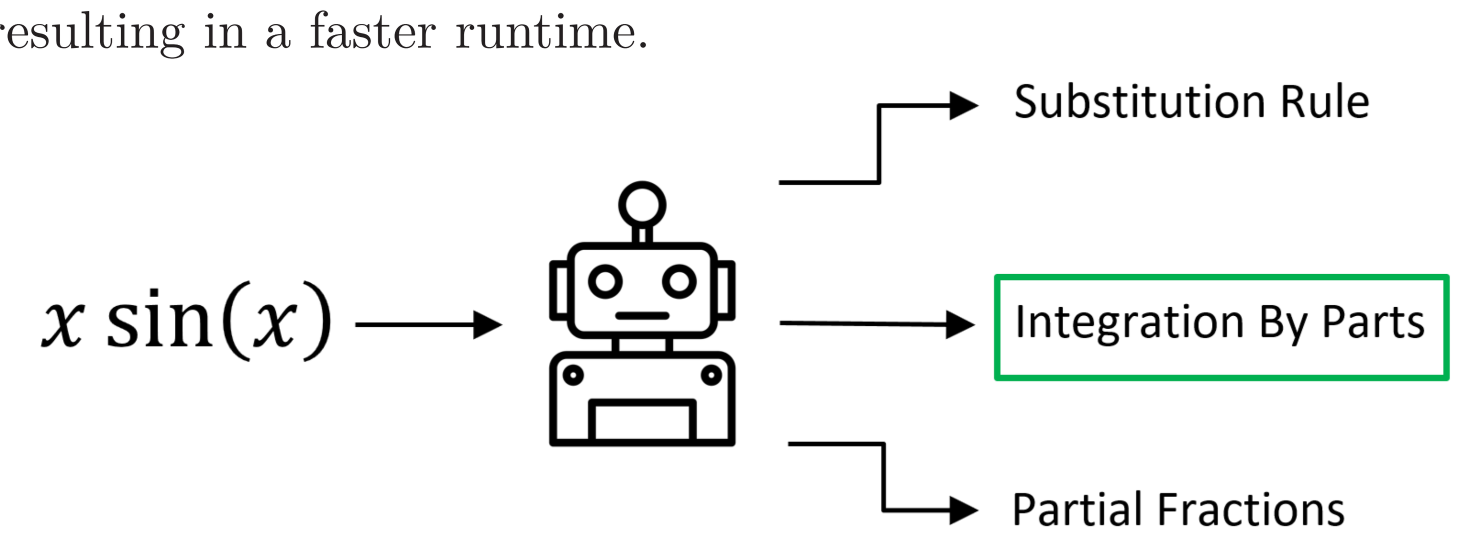
barketr@coventry.ac.uk, delriot@coventry.ac.uk, Matthew.England@coventry.ac.uk

Introduction

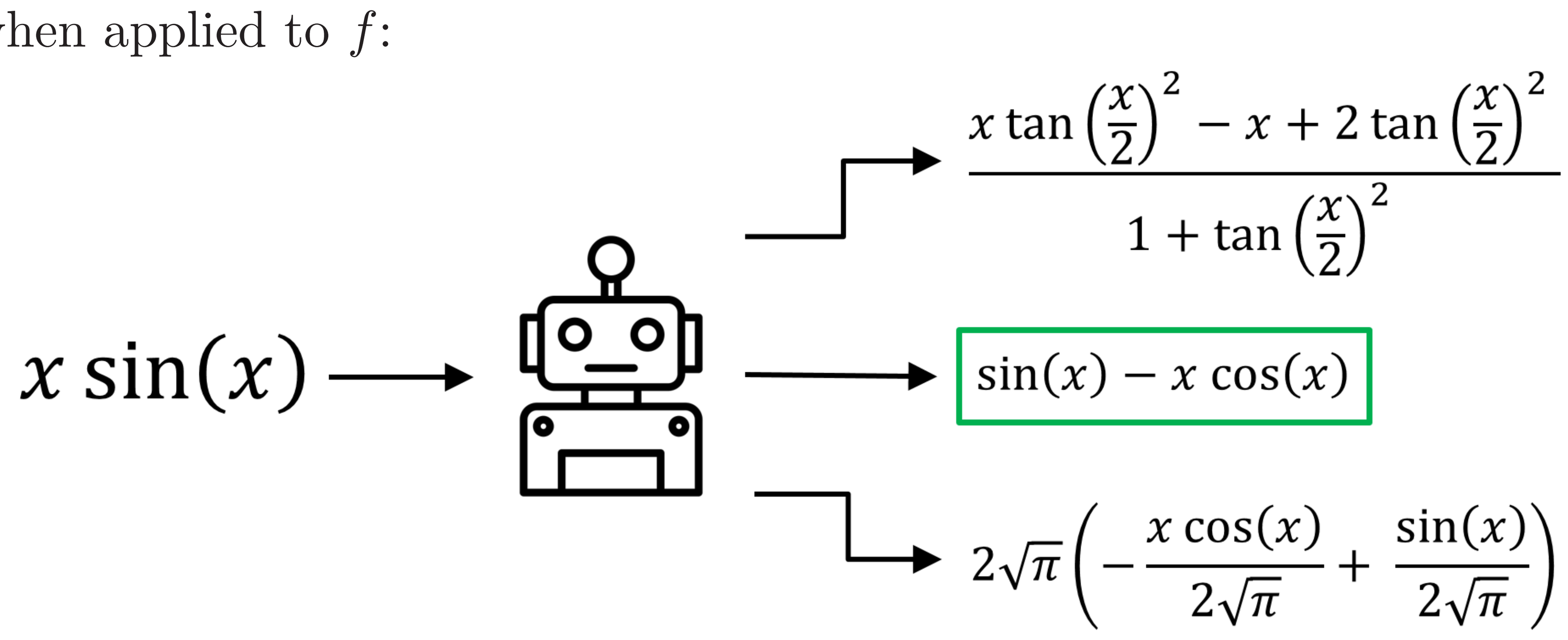
A Computer Algebra System (CAS) such as Maple implements algorithms and data structures for computing exact mathematics in a computer. A CAS prioritises mathematical correctness, allowing for confidence in results, with symbolic results often giving more insight into the problem at hand.
In a CAS, there are usually many possible paths that could all solve a problem correctly. These choices can have a significant impact on resources, in some cases affecting the tractability of the problem.It can also affect how the answer is represented to the user.
Historically, these choices have been made using heuristics designed by experts in the field. Machine Learning (ML) is a separate field of Computer Science, where models are given data and used to make predictions.
Unlike Computer Algebra, ML can produce incorrect answers. It seems counter-intuitive to use ML in a CAS where producing an incorrect answer is not acceptable. However, we can instead use ML to guide algorithms to make optimal choices within these algorithms. This allows us to improve the algorithm efficiency without risking the answer correctness.

Rashid's Project: ML in Integration

Maple has a list of algorithms to use when trying to integrate an expression. Currently, it tries all of these algorithms in a sequence until one of them works. One of the goals of the project is to use ML to have Maple make smart choices about which algorithm to try first depending on the input, resulting in a faster runtime.



Another issue is the form of the output from integration. Consider $f = x\sin(x)$. Maple has three integration algorithms that give different results when applied to f:



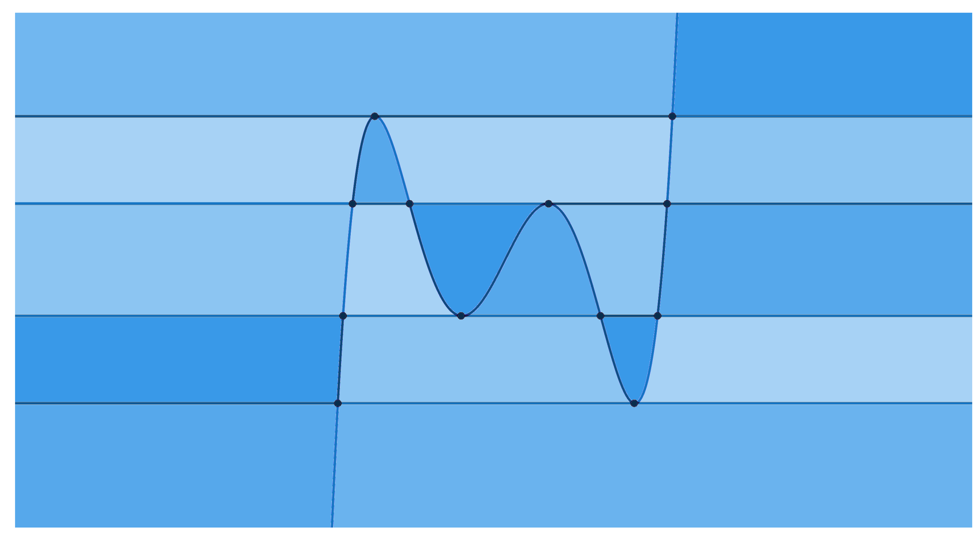
These outputs all look different to each other despite them being the same mathematically. The output is important to the user; the one selected above is easiest for a user to understand. Thus, another use of ML is to teach Maple that there is a preference for how the output looks when displaying the answer to the user.

Tereso's Project: ML in CAD

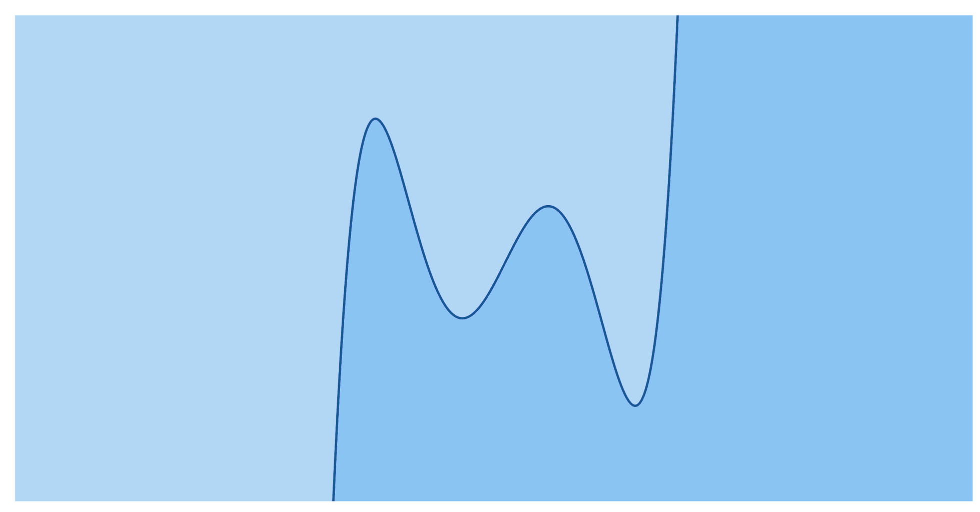
Cylindrical Algebraic Decomposition (CAD) is an important algorithm that divides space into regions where the input polynomials don't change their sign. Solving problems throughout science and engineering.
When computing a CAD, the variable ordering has a great impact on the number of cells in which the space is divided. To illustrate this, we consider a CAD for the polynomial:

$$x^5 + 5x^4 + 5x^3 - 5x^2 - 6x - 2y.$$

Using ordering $x \succ y$, we can obtain a CAD with 57 cells (the 18 shaded areas, 27 lines and 12 points).



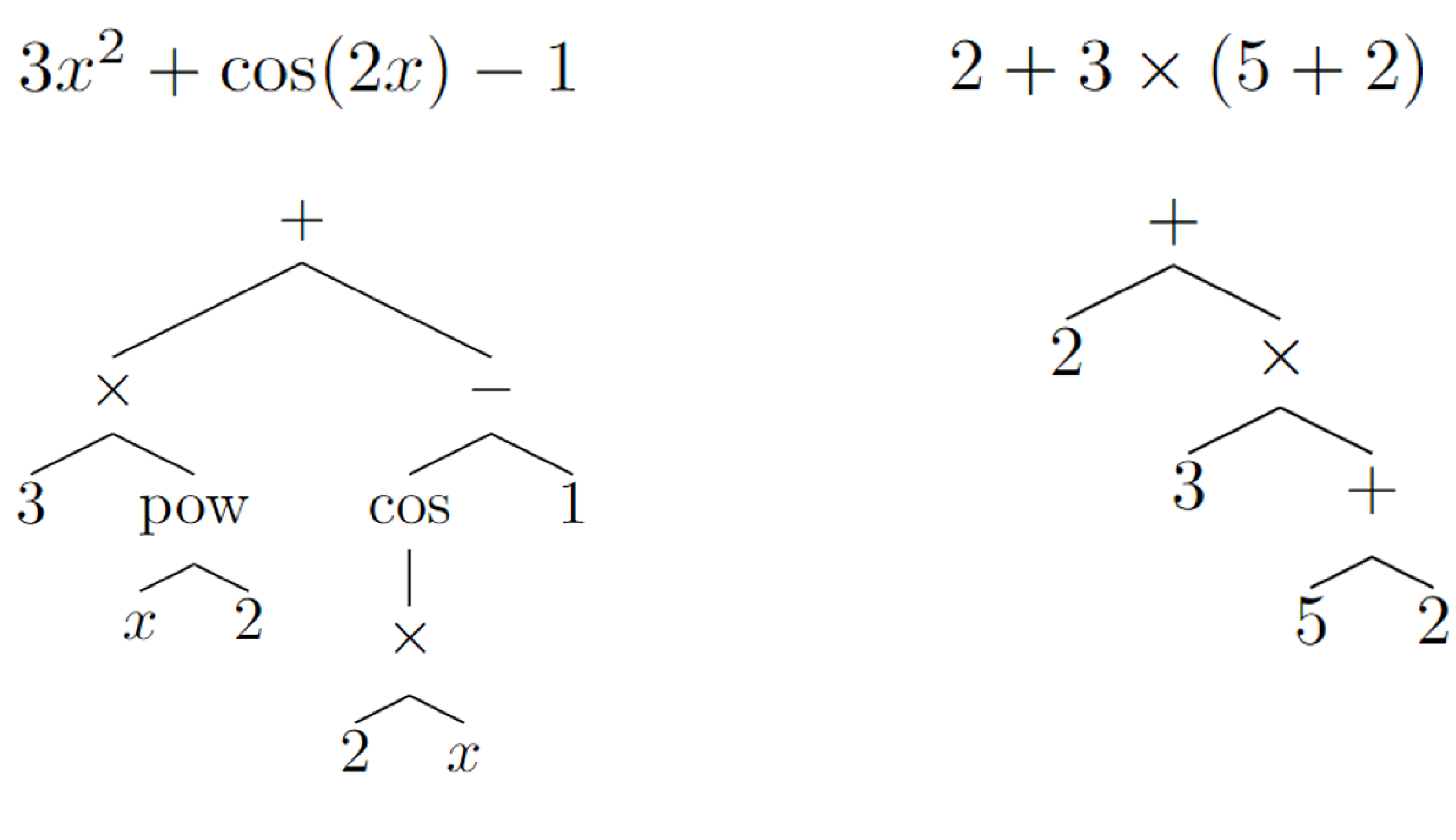
Using the ordering $y \succ x$ only generates 3 cells (2 areas and one line).



Having fewer cells means less time is needed for the algorithm to finish. The goal is to use ML to help in picking a good variable ordering.

Challenge 1: Synthetic Data Generation

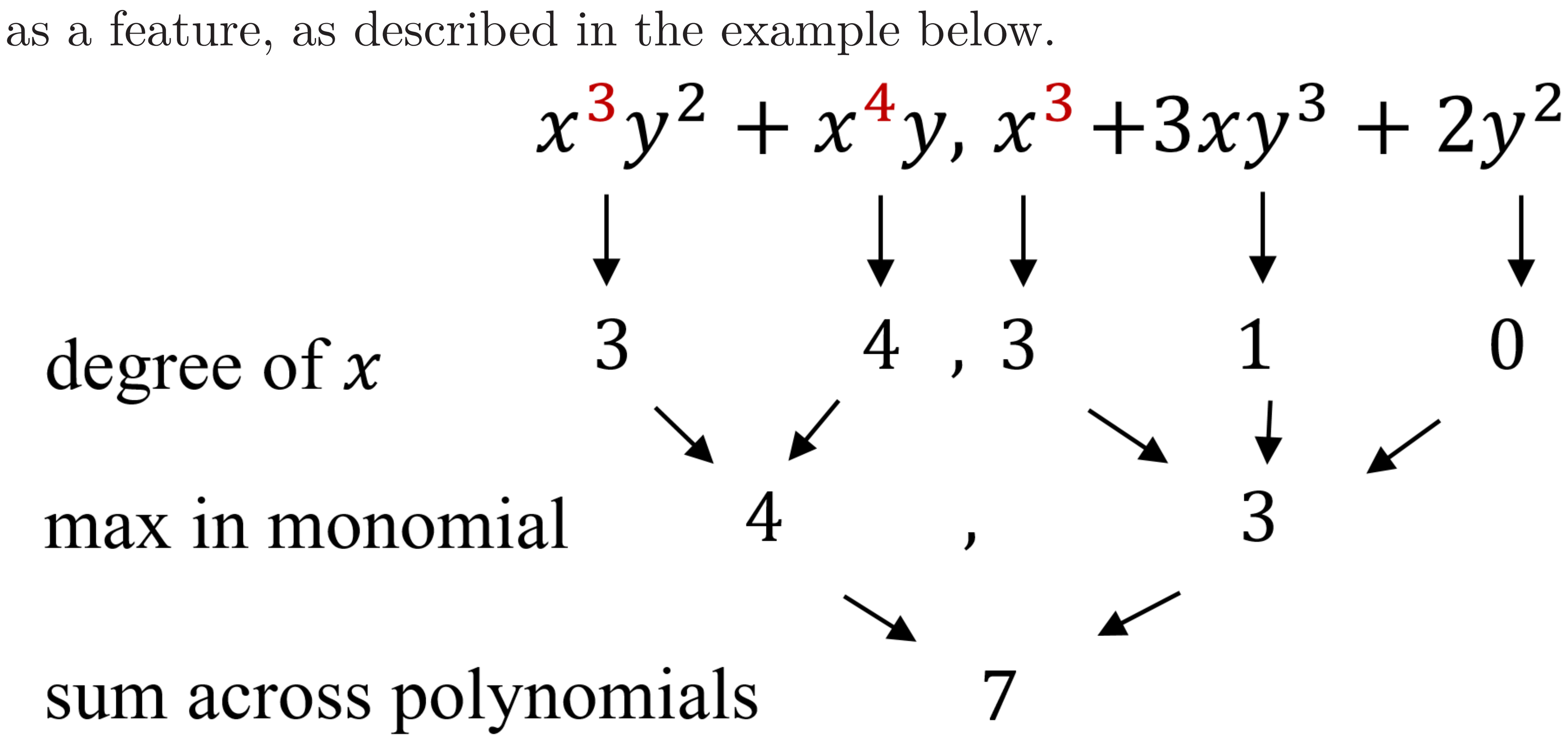
Any ML problem requires lots of data. Mathematics data appears inexpensive to produce, but care is needed to create a dataset representative of all possible use cases without overfitting.
For CAD, we create data using existing sets of example polynomials from examples to which we make random modifications of e.g. their degree, the coefficients, and the number of monomials. For integration the data can be more general, any integrable mathematical expression. We can represent such an expression as a tree, like in the following examples.



The tree nodes can be randomly picked from a set of operators and integers, but care must be taken to ensure the expression is actually integrable.

Challenge 2: Representing Mathematical Data

There are no ML models that accept a set of polynomials or a mathematical function directly as an input. One strategy is to convert the mathematics into plain text and view it as natural language.
Another available approach is to represent a mathematical object by a vector of real valued features extracted from it. For example, the sum of the maximum degree on one variable in each polynomial can be extracted as a feature, as described in the example below.



We focus on the latter approach, using features easily understood by a human. This reduces the risk of over-fitting and allows for a better understanding of how the ML is making choices. Using Explainable ML techniques we aim to gain insight into the underlying mathematics from the patterns picked up by ML.