Undecidability in resource theory: Can you tell theories apart?

Matteo Scandi and Jacopo Surace

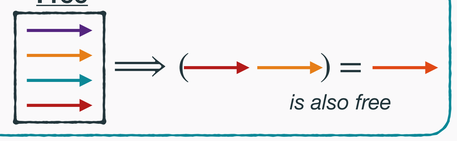
RESOURCE THEORY

A quantum resource theory describes the possibility of action of an agent whose capabilities are constrained to a set of quantum channels.

Example: Two agents (Alice and Bob) work in far-away labs. They share an entangled state, and they can communicate classically over the phone. What transformations can they achieve?
This question is at the center of the resource theory of LOCC (Local Operations and Classical Communication).

The allowed channels are called free operations. This set is closed under composition (performing two allowed channels sequentially is allowed) and contains the identity channel (doing nothing is allowed). Hence, mathematically the **free operations are a semigroup (with identity) of CPTP maps** (Completely Positive and Trace Preserving).

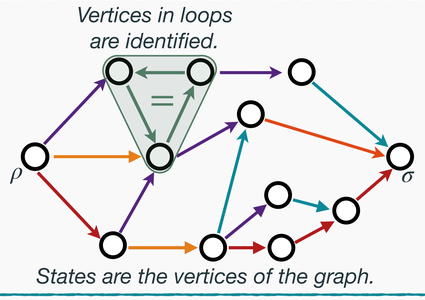
We depict free operations as arrows. The box around them indicates the corresponding semigroup.



Free operations are not the only part of the theory. One is mostly interested in what transformations are possible given an initial resourceful state. One says that the **transition $\rho \to \sigma$ is allowed** if there exists a free operation between the two.

Example: In the theory of LOCC it is possible to teleport a quantum state from Alice to Bob. Anyways, this is only possible if at the beginning of the protocol they also share an entangled state.

One can associate to each state ρ a directed graph, showing all the possible transitions. This structure is the central object studied by resource theory.



In order to quantify how resourceful a state is, one introduces the concept of monotone functions:

$$f : \text{states} \to \mathbb{R} \,|\quad \rho \to \sigma \implies f(\rho) \geq f(\sigma)$$

These are non-increasing under the free operations. A set of monotones is called **complete** if it entirely characterizes the possible transitions.

Example: In the theory of LOCC the resource that one wants to quantify is entanglement. All kind of measure have been introduced (concurrence, robustness, etc.) but no complete set has been found.

MAIN THEOREM

Given a set of free operations and a given channel is it possible to decide whether the latter can be generated just by composing elements from the first? This is called the **membership problem.** It is particularly relevant for applications: once this issue is settled, one can start constructing an explicit realization, or give up completely on the task. Unfortunately, this is not possible:

The membership problem for CPTP semigroups is undecidable

This theorem is proven by reduction of the Post correspondence problem and subsumes all the other results. Whereas the main theorem is relevant by itself, it is not strictly referring to resource theories. In fact, arguably the most important information about a resource theory is contained in the allowed transitions.

COROLLARIES

Consider the problem of deciding whether the transition $\rho \to \sigma$ can be constructed out of free operations. This goes under the name of **reachability problem**. Even in this case, there is no algorithm that can solve this problem in general:

The reachability problem for CPTP semigroups is undecidable

Another relevant issue is the one of comparing the capabilities of two different resource theories. In particular, a subset of this question is whether two resource theories are the same or not. This is also not possible to decide:

Saying whether two CPTP semigroups induce the same transitions (or contain the same maps) is undecidable

Finally, can one generically construct a complete set of monotones? This is relevant when one wants to assess the value of a state, but, again, it is shown that:

There is no general algorithm to associate to a CPTP semigroup the corresponding complete set of monotones

REFERENCES

arXiv:2105.09341

UNDECIDABILITY

In 1937 Alan Turing publishes the paper "On Computable Numbers, with an Application to the Entscheidungsproblem". In there the idea of **computation is formalised**, and it is shown that there exist an algorithm able to simulate any other algorithm (the **Universal Turing Machine**). Finally, he discovered that there are some decision problems that cannot be answered by any algorithm in finite time. The most famous example for this is the **halting problem**.

The halting problem

In order to solve the halting problem one has to construct an algorithm that takes as input the description of any other algorithm, and as output it tells whether the computation will terminate or not.

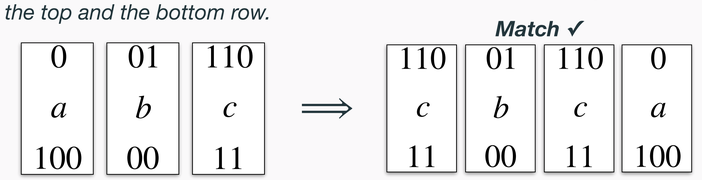
Surprisingly, ***this algorithm cannot exist.***

This result shows that there are some problems for which a general answer cannot be constructed. It is important to point out that undecidability is not just a feature of computer programs: there are a wide range of problems showing this feature, ranging from classical dynamical systems to matrix multiplication, from polynomials to cellular automata.

Undecidability is usually proven by reduction of the halting problem. This consists in two steps: first, one shows that the system in consideration can simulate any computation; second, it is shown that certifying some feature of the system is equivalent to decide the halting of the corresponding algorithm.

The Post correspondence problem

Given an arbitrary set of dominoes, one says that there is a matching if they can be rearranged with repetition so that the same string appears on the top and the bottom row.



Post showed that any computation can be simulated through matching of dominoes. For this reason, deciding whether there is a matching is equivalent to deciding whether an algorithm halts. Hence the matching of dominoes is undecidable in general.