L. Clemente **S. Lasota** **R. Piórkowski**
University of Warsaw

TIMED CHURCH's SYNTHESIS PROBLEM

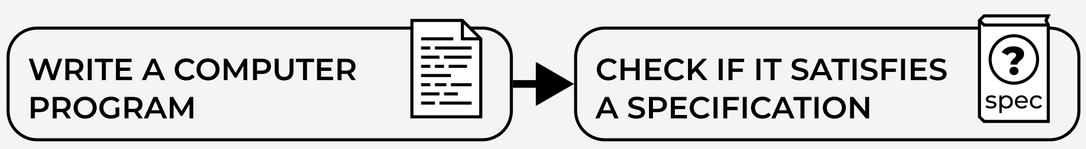
a popular scientific poster

MOTIVATION

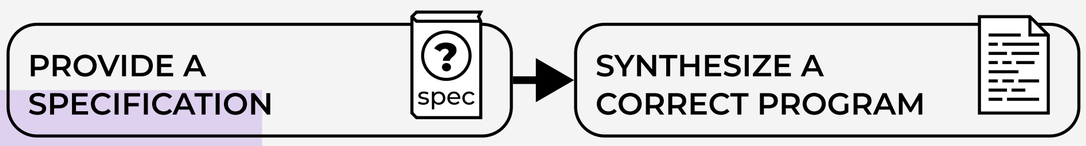
Need for provably correct programs

■ **WHY PROGRAMS NEED TO BE VERIFIED?** Computers have become an integral part of our lives and helped us to solve many previously unattainable problems. Some task we entrusted them with are very responsible. When human life is at stake – e.g. when a computer is steering an autonomous car or conducts a complicated surgery – there is no place for programming errors. A need arises for dependable software systems which are provably error-free.

■ **HOW CAN IT BE DONE?** There are many approaches to creating dependable programs. The most natural one is called **program verification**. Here, the program is written as usual, and the task is to check whether it satisfies some provided specification, which specifies desired or undesired behaviours of the program.



But there are also some more ambitious approaches. One of them is known as **program synthesis**. In this case, only a specification needs to be provided, and the goal is to automatically construct a program that correctly implements it.



PRELIMINARIES

Timed automata

■ **WHAT IS A MODEL OF COMPUTATION?** A typical computer is an incredibly complicated machine, and it is very hard to fully understand how it works inside out. To deal with this issue, scientists use models of computation. Those are simplified mathematical devices able to simulate the execution of some of the algorithms a real computer runs.

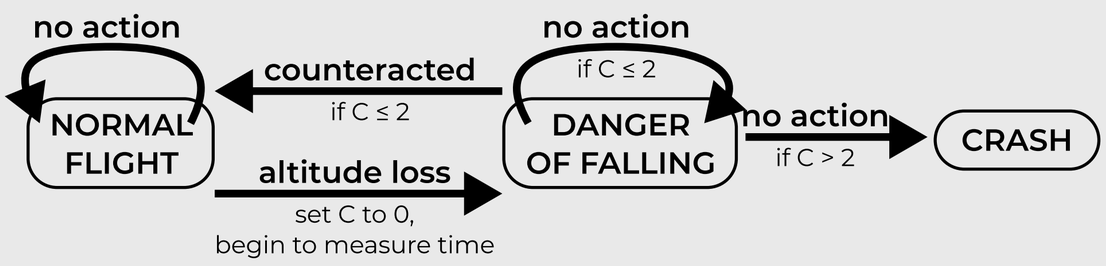
■ **WHAT IS A TIMED AUTOMATON?** It is one of the great number of models of computation. It is particularly well suited for the analysis of systems, where timing is important, e.g. self-driving cars, autopiloted drones and planes, control systems in power plants, etc.

■ **HOW THEY REACT TO TIME** A timed automaton's input is a series of events that happen at some given point in time. They use "clocks" to measure the time between different events.

■ **EXAMPLE** Consider a plane flying with an autopilot program. To guarantee safety, it has to respond quickly to the varying atmospheric conditions. To model its desired properties, we may use a timed automaton. It becomes a specification for our real-time system. Let us assume that we want to express the following property:

THE AUTOPILOT COUNTERACTS A POSSIBLE ALTITUDE LOSS AT MOST 2 SECONDS AFTER IT IS DETECTED

The corresponding automaton using one clock C looks as follows:



PRELIMINARIES

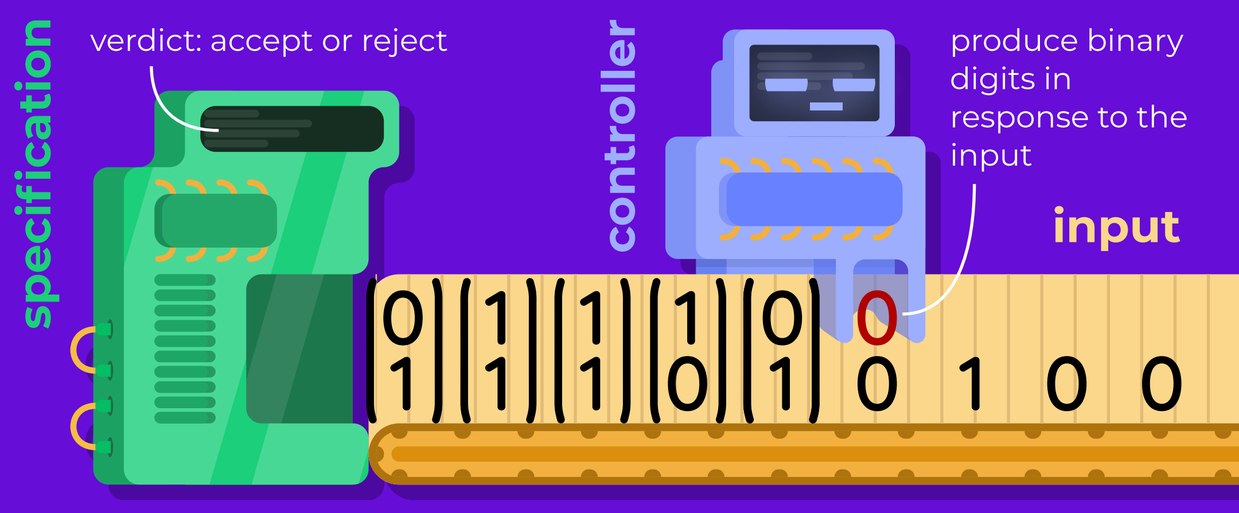
The Church's synthesis problem

■ **HISTORY** The Church's problem was first stated by Alonzo Church in 1957. It is now seen by many as a cornerstone of program synthesis.

■ **STATEMENT** Given a **specification S**, construct a **controller C** such that for every possible **input** data (say, an infinite sequence of zeros and ones) **C** will satisfy the specification.

■ **IMPORTANT THEOREM** (Büchi-Landweber) If the specification is given by a **ω-regular language**, the synthesis problem can be solved.
In other words, in that case, there is a way to construct a correct controller if it exists.

■ **GRAPHIC DEPICTION OF SYNTHESIS PROBLEM**



Above, the controller (to be synthesised) needs to respond to the binary digits it observes at the input. The effects of its work (together with the input) are verified by the specification.

OUR CONTRIBUTION

Timed synthesis problem

■ **THE MODELS WE WORK WITH** In our paper, we have defined a generalised version of the Church's synthesis problem – adapted to the setting of timed automata. Now, the controller and specification both become timed (more precisely: **a timed automaton with an output** and **a timed ω-automaton**).

■ **OUR THEOREM**
Given a **specification S (a timed ω-automaton)**
that can be fulfilled,
it is possible to automatically construct
a **controller C (a timed automaton with an output)**
such that for every possible **timed input**
C will produce results matching the **specification**.

This is a natural generalisation of the Büchi-Landweber theorem to the world of timed automata.

■ **OTHER RESULTS**
Additionally, using the above results, we also solve an interesting problem of "separability" for timed automata.

https://arxiv.org/abs/2004.12868