We model stochastic systems through Markov chains and specify and evaluate their properties using probabilistic temporal logics and automata. In general, logics offer a clearer syntax and automata provide better performance in terms of computability. Therefore, it is important to define classes of logics and automata that have the same expressive power and can be used interchangeably. Here, we focus on two such formalisms known as $\mu^p$-calculus and $p$-automata.

### Background

**$\mu^p$-Calculus**

The $\mu^p$-calculus [1] is a probabilistic temporal logic.

- Atomic propositions $p, \neg p$
- Boolean connectives $\vee, \wedge$
- Next operator $\nu p$
- Probabilistic quantification $[\mu]p$

Using the fixpoint operators this logic can express finite and infinite iterations of properties:

- Least fixpoint $\mu$ finitely many iterations
- Greatest fixpoint $\nu$ infinitely many iterations

A formula $\nu p$ contained inside a probabilistic quantification are associated with a probability value in $[0,1]$. The operator $\mu$, checks whether the value of the formula meets the bound $J$ (of the form $\geq \alpha$), and gets the value $1$ or $0$ accordingly. Therefore, top-level formulas are qualitative: either true or false.

When a $\mu^p$-calculus formula is true on a Markov chain, we say that the Markov chain satisfies the formula.

**$p$-Automata**

A $p$-automaton [2] is an automaton that reads a Markov chain as input and decides whether to accept it or not. It is characterised by five components:

1. **States** are the elementary blocks and, to handle probabilities, may be enclosed in a probabilistic quantification $[\mu]$.
2. **Alphabet** contains symbols that are read by the automaton, triggering a specific transition.
3. **Transitions** allow the automaton to move from one state to a Boolean (and/or) combination of them, depending on the symbol read.
4. **Initial condition** is a state, or a combination thereof, from which the automaton begins its computation.
5. **Acceptance** assigns a number to each state. Only states that are marked by an even number can be visited infinitely often.

### Equivalence

**$\mu^p$-Calculus $\rightarrow$ $p$-Automata**

For every $\mu^p$-calculus formula we can construct a $p$-automaton that accepts exactly those Markov chains that satisfy the formula [3].

- **States** originate from sub-formulas of the form: propositions, negated propositions, next, and quantified next; plus accepting and rejecting states.
- **Alphabet** is the power set of propositions appearing in the formula.
- **Transitions** preserve the Boolean connectives $\vee, \wedge$ and unfold the next operators into their nested sub-formulas.
- **Initial condition** derives from the main formula without fixpoints.
- **Acceptance** reflects the type of fixpoints that enclose the sub-formula/state ($\mu \rightarrow$ odd, $\nu \rightarrow$ even) and their potential nesting. Accepting and rejecting states are assigned numbers 0 and 1, respectively.

**$p$-Automata $\rightarrow$ $\mu^p$-Calculus**

For every $p$-automaton we can construct a $\mu^p$-calculus formula satisfied in exactly those Markov chains accepted by the automaton [3].

We have summarised the analogies that allow the translation from $\mu^p$-calculus to $p$-automata and backwards. The mutual correspondence of the two languages implies their equivalence in expressive power; thus, lifting the well-known connection between logics and automata theory to a probabilistic scenario.