1. Arcadian Automata

A is defined as a tuple \( \langle A, Q, q^0, ϕ^0, I, i, fv \rangle \), where

- \( A = \langle A, \leq \rangle \) is a finite tree;
- \( Q \) is a set of states;
- \( q^0 \in Q \) is an initial state of the automaton;
- \( ϕ^0 \in A \) is an initial tree node of the automaton;
- \( I \) is a set of all instructions;
- \( i: Q \to P(I) \) is a function which gives the set of instructions available in a given state;
- \( fv: A \to P(A) \) is a binding function.

Each state may be either existential or universal and belongs to an element \( a \in A \), so \( Q = Q^1 \cup Q^2 \), and \( Q^1 = \bigcup_{a \in A} Q^1_a \) and \( Q^2 = \bigcup_{a \in A} Q^2_a \).

2. Instantaneous description

An ID of A is a tuple \( \langle q, κ, w, w', S, V \rangle \) where

- \( q \in Q \) is the current state,
- \( κ \) is the current node in \( A \),
- \( w: A \to V \) is the interpretation of bindings associated with \( κ \) by \( fv(κ) \), in particular we require here that \( fv(κ) \subseteq \text{dom}(w) \),
- \( w': A \to V \) is the auxiliary interpretation of bindings
- \( S \) is the store of the automaton, which is a set of pairs \( \langle ρ, v \rangle \) where \( ρ \in A \) and \( v: A \to V \) and we require that \( fv(ρ) \subseteq \text{dom}(v) \),
- \( V \) is the working domain of the automaton, i.e. a set of eigenvariables,

3. Instructions

1. \( q: \text{store} \rho, ρ', q' \) adds a a new fact \( \langle \rho, (w \ll w')|_{v(ρ)} \rangle \) to the store;
2. \( q: \text{jmp} \rho, q' \)
3. \( q: \text{new} \rho, q' \) extends the discourse domain \( V \) with a new element \( X \);
4. \( q: \text{check} \rho, ρ', q' \) checks if appropriate fact is stored in \( S \);
5. \( q: \text{instL} \rho, ρ', q' \) checks if the current node is an appropriate successor of \( ρ \);
6. \( q: \text{load} \rho, q' \) loads an appropriately constructed interpretation to the additional register.

4. Structural decomposition instructions

(1) \( ϕ_1 \to ϕ_2 \)
(2) \( ϕ_1 \land ϕ_2 \)
(3) \( ϕ_1 \lor ϕ_2 \)
(4) \( ∀X.ϕ \)
(5) \( ∃X.ϕ \)

5. Example

We built the Arcadian automaton for \( ϕ_{pos} \)
\[ ∀x(P(x)) \to ∀y∃zP(x) \]
(by tree shown in Fig. 1).

The set \( I \) of available instructions consists of elements:

- \( q_1: \text{jmp} 2, q^3_1 \)
- \( q_2: \text{new} 5, q^3_2 \)
- \( q_3: \text{jmp} 5, q^3_3 \)
- \( q_4: \text{instR} 6, q^3_4 \)
- \( q_5: \text{store} 2, 4, q^3_5 \)

We use some instructions available for any state. Here we spell out only the ones used in our successful run:

- \( q_1: \text{jmp} 1, q^1_1 \)
- \( q_2: \text{jmp} 1, q^1_2 \)
- \( q_3: \text{jmp} 5, q^3_3 \)
- \( q_4: \text{check} 2, q^4_4 \)

The red arrows show a successful run of the automaton.