CHARLES UNIVERSITY PRAGUE

faculty of mathematics and physics

Jan Butora

Department of algebra

Combinatorics on words and automated proving II

Theory behind the prover

Fall school of algebra 2015

Content

- Automatic sequences
- Logic
- Automated proving

Automatic sequences

Definition (k-automatic sequence)

An infinite sequence $\mathbf{a} = (a_n)_{n \geq 0}$ over a finite alphabet is said to be *k-automatic* if there exists a deterministic finite automaton (with output associated with the states) such that after completely processing the input n expressed in base k, the automaton reaches some state q with output a_n.

- Thue-Morse sequence: $t = t(0)t(1)t(2)\ldots = 011010011001\ldots$

Automatic sequences

Definition (k-automatic sequence)

An infinite sequence $\mathbf{a} = (a_n)_{n \geq 0}$ over a finite alphabet is said to be *k-automatic* if there exists a deterministic finite automaton (with output associated with the states) such that after completely processing the input n expressed in base k, the automaton reaches some state q with output a_n.

- Thue-Morse sequence: $t = t(0)t(1)t(2)\ldots = 011010011001\ldots$
- Fibonacci-automatic sequences
  - Infinite Fibonacci word: $f = f(0)f(1)f(2)\ldots = 01001010\ldots$

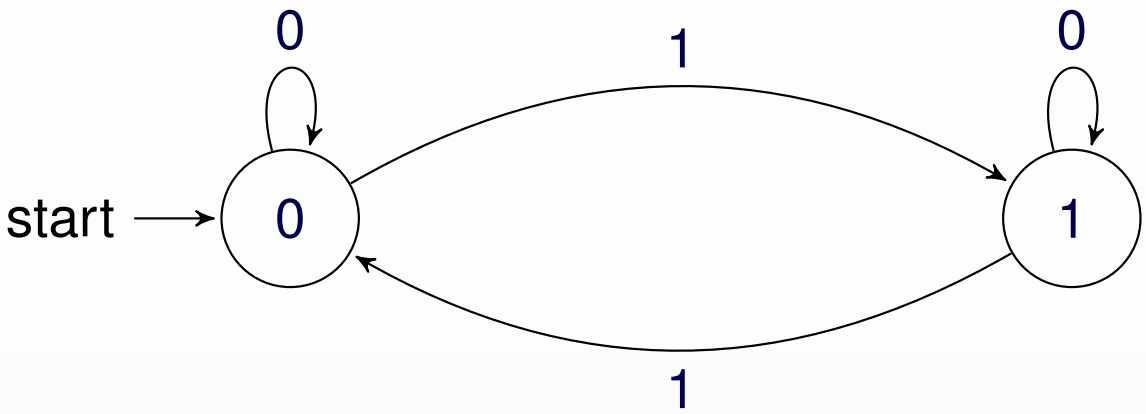
Automatic sequences

Definition (k-automatic sequence)

An infinite sequence $\mathbf{a} = (a_n)_{n \geq 0}$ over a finite alphabet is said to be *k-automatic* if there exists a deterministic finite automaton (with output associated with the states) such that after completely processing the input n expressed in base k, the automaton reaches some state q with output a_n.

- Thue-Morse sequence: $t = t(0)t(1)t(2)\ldots = 011010011001\ldots$
- Fibonacci-automatic sequences
  - Infinite Fibonacci word: $f = f(0)f(1)f(2)\ldots = 01001010\ldots$
- Tribonacci sequences
- Tetranacci sequences
- ...

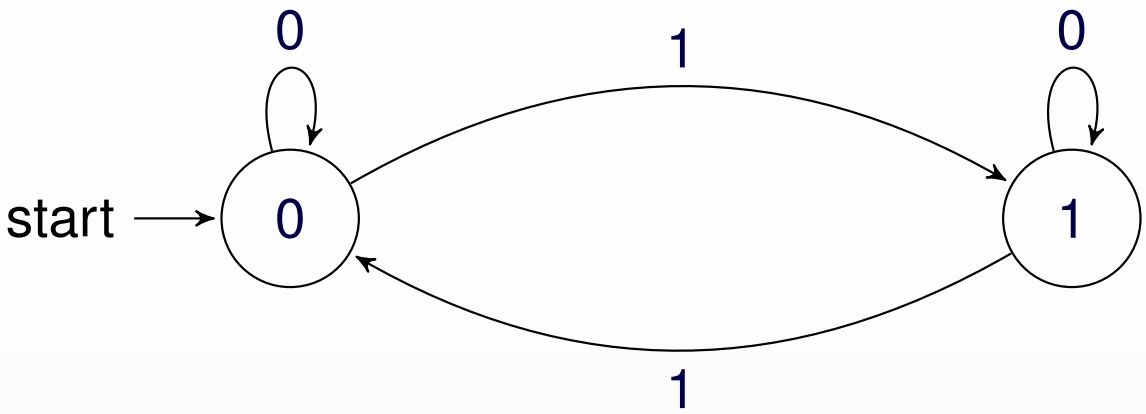
Generating automatic sequence



Finite automaton generating the Thue-Morse sequence **t**

- Thue-Morse sequence **t** = $t(0)t(1)t(2)\ldots = 011010011001\ldots$

Generating automatic sequence



Finite automaton generating the Thue-Morse sequence **t**

- Thue-Morse sequence **t** = $t(0)t(1)t(2)\ldots = 011010011001\ldots$
- $t(n)$ is the sum, modulo 2, of the binary digits of n

Definition (Structure)

We call following S a *structure*

$$S = \langle D, (R_i)_{i \in I}, (f_j)_{j \in J}, (c_k)_{k \in K} \rangle$$

$D \ldots$ *a domain* (some set)
$(R_i)_{i \in I} \ldots$ a family of relations on D
$(f_j)_{j \in J} \ldots$ a family of functions from D^{n_j} to D
$(c_k)_{k \in K} \ldots$ constants of D
The set $\{(R_i)_{i \in I}, (f_j)_{j \in J}, (c_k)_{k \in K}\}$ is called the *language* of the structure S. In addition, symbols $x, y, z, \ldots, \vee, \wedge, \neg, \forall, \exists, \rightarrow, \leftrightarrow, =$ are in S.

Structures and terms

Definition (Structure)

We call following S a *structure*

$$S = \langle D, (R_i)_{i\in I}, (f_j)_{j\in J}, (c_k)_{k\in K} \rangle$$

D . . . *a domain* (some set)
$(R_i)_{i\in I}$ . . . a family of relations on D
$(f_j)_{j\in J}$ . . . a family of functions from D^{n_j} to D
$(c_k)_{k\in K}$ . . . constants of D
The set $\{(R_i)_{i\in I}, (f_j)_{j\in J}, (c_k)_{k\in K}\}$ is called the *language* of the structure S. In addition, symbols $x, y, z, \ldots, \vee, \wedge, \neg, \forall, \exists, \rightarrow, \leftrightarrow, =$ are in S.

Definition (Term)

The *terms* are defined by induction following two rules:

1. any variable and constant is a term,
2. if f_j is a n-ary function and $t_1, \ldots, t_n$ are terms, then $f_j(t_1, \ldots, t_n)$ is a term.

Formulae

Definition (Formula)

The *formulae* are generated by four rules:

1. if t_1, t_2 are terms, then $t_1 = t_2$ is a formula,
2. if R_i is a n-ary relation and $t_1, \ldots, t_n$ are terms, then $R_i(t_1, \ldots, t_n)$ is a formula,
3. if φ, Φ are formulae, then $\varphi \vee \Phi, \varphi \wedge \Phi, \neg\varphi, \varphi \to \Phi, \varphi \leftrightarrow \Phi$ are formulae,
4. if φ is a formula and x is a variable, then $\forall x\varphi, \exists x\varphi$ are formulae.

Formulae

Definition (Formula)

The *formulae* are generated by four rules:

1. if t_1, t_2 are terms, then $t_1 = t_2$ is a formula,
2. if R_i is a n-ary relation and $t_1, \ldots, t_n$ are terms, then $R_i(t_1, \ldots, t_n)$ is a formula,
3. if φ, Φ are formulae, then $\varphi \vee \Phi, \varphi \wedge \Phi, \neg\varphi, \varphi \to \Phi, \varphi \leftrightarrow \Phi$ are formulae,
4. if φ is a formula and x is a variable, then $\forall x\varphi, \exists x\varphi$ are formulae.

Definition (Free variable)

We call *free variable* a variable in formula without quantifier.

Formulae

Definition (Formula)

The *formulae* are generated by four rules:

1. if t_1, t_2 are terms, then $t_1 = t_2$ is a formula,
2. if R_i is a n-ary relation and $t_1, \ldots, t_n$ are terms, then $R_i(t_1, \ldots, t_n)$ is a formula,
3. if φ, Φ are formulae, then $\varphi \vee \Phi, \varphi \wedge \Phi, \neg\varphi, \varphi \to \Phi, \varphi \leftrightarrow \Phi$ are formulae,
4. if φ is a formula and x is a variable, then $\forall x\varphi, \exists x\varphi$ are formulae.

Definition (Free variable)

We call *free variable* a variable in formula without quantifier.

Definition (Sentence)

We call *sentence* a formula without free variable.

Theories and equivalent structures

Definition (Decidable theory)

Given a structure S, the set of the sentences true for S is the *theory* of S, denoted by $Th(S)$.
The theory $Th(S)$ is called *decidable* if there exists an algorithm which decides if any sentence of S is true or false for S.

Theories and equivalent structures

Definition (Decidable theory)

Given a structure S, the set of the sentences true for S is the *theory* of S, denoted by $Th(S)$.
The theory $Th(S)$ is called *decidable* if there exists an algorithm which decides if any sentence of S is true or false for S.

Definition (Equivalent structures)

We say that the structures S and S' with the same domain D are *equivalent* if the sets definable in S are the same as in S'.

Theories and equivalent structures

Definition (Decidable theory)

Given a structure S, the set of the sentences true for S is the *theory* of S, denoted by $Th(S)$.
The theory $Th(S)$ is called *decidable* if there exists an algorithm which decides if any sentence of S is true or false for S.

Definition (Equivalent structures)

We say that the structures S and S' with the same domain D are *equivalent* if the sets definable in S are the same as in S'.

Lemma

Structures $\langle \omega, + \rangle$ *and* $\langle \omega, +, \leq \rangle$ *are equivalent.*

Theories and equivalent structures

Definition (Decidable theory)

Given a structure S, the set of the sentences true for S is the *theory* of S, denoted by $Th(S)$.
The theory $Th(S)$ is called *decidable* if there exists an algorithm which decides if any sentence of S is true or false for S.

Definition (Equivalent structures)

We say that the structures S and S' with the same domain D are *equivalent* if the sets definable in S are the same as in S'.

Lemma

Structures $\langle \omega, + \rangle$ *and* $\langle \omega, +, \leq \rangle$ *are equivalent.*

- Just consider formula $(\exists z)(x + z = y)$ in $\langle \omega, + \rangle$.

Presburger artihmetic

Definition (Presburger arithmetic)

We call the theory $Th(\langle \omega, + \rangle)$ *Presburger arithmetic*.

Presburger artihmetic

Definition (Presburger arithmetic)

We call the theory $Th(\langle \omega, + \rangle)$ *Presburger arithmetic*.

"Chicken McNuggets" theorem can be described in Presburger arithmetic:

$$(\forall n > 43 \, \exists x, y, z \geq 0 \text{ such that } n = 6x + 9y + 20z) \wedge$$

$$\neg(\exists x, y, z \geq 0 \text{ such that } 43 = 6x + 9y + 20z)$$

Number representation

$\Sigma = \{0, 1\}$ . . . binary alphabet

$$\Sigma_n = \left\{ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \right\} \ldots n\text{-tuples of integers}$$

Number representation

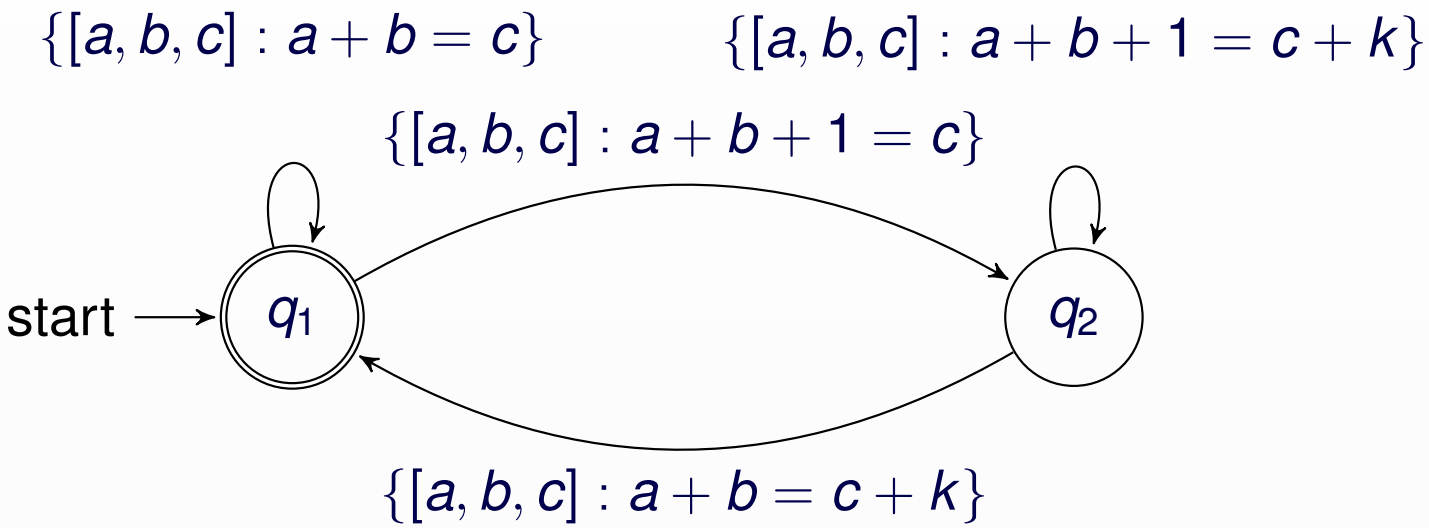
$\Sigma = \{0, 1\}$ . . . binary alphabet

$$\Sigma_n = \left\{ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \right\} \ldots n\text{-tuples of integers}$$

$B = \{w \in \Sigma_3^* | \text{the bottom row of } w \text{ is the sum of the top two rows}\}$

For example $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in B$

Checking addition in Presburger arithmetic



Checking addition with carry in base k

Definability of sequences

Let $p \geq 2$ and $s : \omega \to A$ be a *sequence* with values in a finite alphabet $A \subset \omega$.

Definition (p-definability)

Consider the structure $\langle \omega, +, V_p \rangle$, where the function V_p is defined as $V_p(x) = p^n$, where p^n is the greatest power of p dividing x ($x \neq 0$) $V_p(0) = 1$.
A sequence s is *p-definable* if for each letter $a \in A$, there exists a first-order formula ϕ_a of $\langle \omega, +, V_p \rangle$ such that

$$s^{-1}(a) = \{n \in \omega \mid \phi_a(n) \text{ is true in } \langle \omega, +, V_p \rangle\}$$

Example: 2-automatic sequence

Let $p : \omega \to \{0, 1\}$ be the characteristic sequence of the powers of 2:

$$01101000100000010\ldots$$

$p_n = 1$ if n is power of 2, $p_n = 0$ otherwise

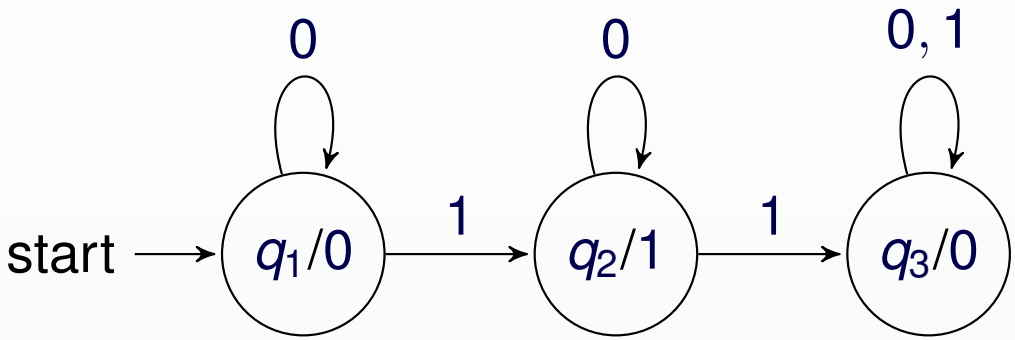
Example: 2-automatic sequence

Let $p : \omega \to \{0, 1\}$ be the characteristic sequence of the powers of 2:

$$01101000100000010\ldots$$

$p_n = 1$ if n is power of 2, $p_n = 0$ otherwise

- p is 2-automatic:
  The sequence p is computed by the following finite automaton with output.



An automaton computing p in base 2

Example: 2-definable sequence

Let $p : \omega \to \{0, 1\}$ be the characteristic sequence of the powers of 2:

$$011010001000000010\ldots$$

$p_n = 1$ if n is power of 2, $p_n = 0$ otherwise

Example: 2-definable sequence

Let $p : \omega \to \{0, 1\}$ be the characteristic sequence of the powers of 2:

$$011010001000000010\ldots$$

$p_n = 1$ if n is power of 2, $p_n = 0$ otherwise

- p is 2-definable:
  Let $P_2(x)$ be the formula $V_2(x) = x$; then

$$p^{-1}(1) = \{n \in \omega \mid P_2(n) \text{ is true in } \langle \omega, +, V_2 \rangle\}$$

$$p^{-1}(0) = \{n \in \omega \mid P_2(n) \text{ is false in } \langle \omega, +, V_2 \rangle\}$$

p-automatic vs. p-definable

Theorem

Let $p \geq 2$ and $s : \omega \to A$ be a sequence with values in a finite alphabet $A \subset \omega$. Then s is p-automatic if and only if s is p-definable.

p-automatic vs. p-definable

Theorem

Let $p \geq 2$ and $s : \omega \to A$ be a sequence with values in a finite alphabet $A \subset \omega$. Then s is p-automatic if and only if s is p-definable.

Theorem

Let $p \geq 2, m \geq 1$ and $s : \omega^m \to A$ be a sequence. Then s is p-automatic if and only if s is p-definable.

p-automatic sets

Definition

Let $p \geq 2, m \geq 1$ and $M \subseteq \omega^m$. We say that M is *p-automatic* if its characteristic sequence $s_M : \omega^m \to \{0, 1\}$ defined by

$$s_n = 1 \leftrightarrow n = (n_1, \ldots, n_m) \in M$$

is p-automatic.

p-automatic sets

Definition

Let $p \geq 2, m \geq 1$ and $M \subseteq \omega^m$. We say that M is *p-automatic* if its characteristic sequence $s_M : \omega^m \to \{0, 1\}$ defined by

$$s_n = 1 \leftrightarrow n = (n_1, \ldots, n_m) \in M$$

is p-automatic.

Equivalently M is p-automatic if and only if there is a finite automaton accepting set $\{w \in \{0, \ldots, p-1\}^* \mid [w]_p \in M\}$
where $[w]_p = w_0 p^k + w_1 p^{k-1} + \ldots + w_k p^0$

p-definable sets

Definition

Let $m \geq 1$. We say that set $M \subseteq \omega^m$ is *p-definable* if there exists a formula ϕ such that

$$M = \{(n_1, \ldots, n_m) \in \omega^m \mid \phi(n_1, \ldots, n_m) \text{ is true in } \langle \omega, +, V_p \rangle\}$$

The set of powers of 2 is 2-definable.

Characterisation of p-automatic sets

Theorem

Let $m \geq 1$ and $M \subseteq \omega^m$. Let $p \geq 2$. Then M is p-automatic if and only if M is p-definable.

Characterisation of p-automatic sets

Theorem

Let $m \geq 1$ and $M \subseteq \omega^m$. Let $p \geq 2$. Then M is p-automatic if and only if M is p-definable.

$\leftarrow$: Take formula ϕ of $\langle \omega, +, V_p \rangle$ defining

$$M_\phi = \{(n_1, \ldots, n_m) \in \omega^m \mid \phi(n_1, \ldots, n_m) \text{ is true in } \langle \omega, +, V_p \rangle\}$$

Characterisation of p-automatic sets

Theorem

Let $m \geq 1$ and $M \subseteq \omega^m$. Let $p \geq 2$. Then M is p-automatic if and only if M is p-definable.

$\leftarrow$: Take formula ϕ of $\langle \omega, +, V_p \rangle$ defining

$$M_\phi = \{(n_1, \ldots, n_m) \in \omega^m \mid \phi(n_1, \ldots, n_m) \text{ is true in } \langle \omega, +, V_p \rangle\}$$

Construct a finite automaton A_ϕ computing M_ϕ

Characterisation of p-automatic sets

Theorem

Let $m \geq 1$ and $M \subseteq \omega^m$. Let $p \geq 2$. Then M is p-automatic if and only if M is p-definable.

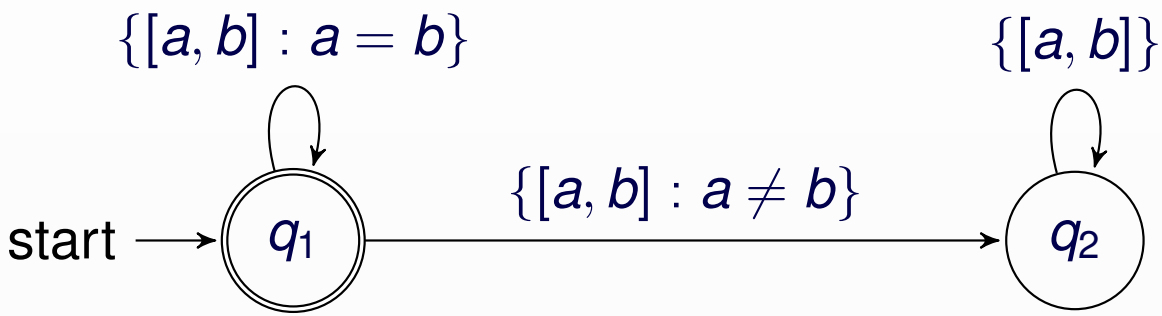
$\leftarrow$: Take formula ϕ of $\langle \omega, +, V_p \rangle$ defining

$$M_\phi = \{(n_1, \ldots, n_m) \in \omega^m \mid \phi(n_1, \ldots, n_m) \text{ is true in } \langle \omega, +, V_p \rangle\}$$

Construct a finite automaton A_ϕ computing M_ϕ
For simplicity, we'll work with structure $\langle \omega, R_+, R_{V_p} \rangle$
$R_+(x, y, z)$ is relation $x + y = z$
$R_{V_p}(x, y)$ is relation $V_p(x) = y$
$\langle \omega, R_+, R_{V_p} \rangle$ is equivalent to $\langle \omega, +, V_p \rangle$

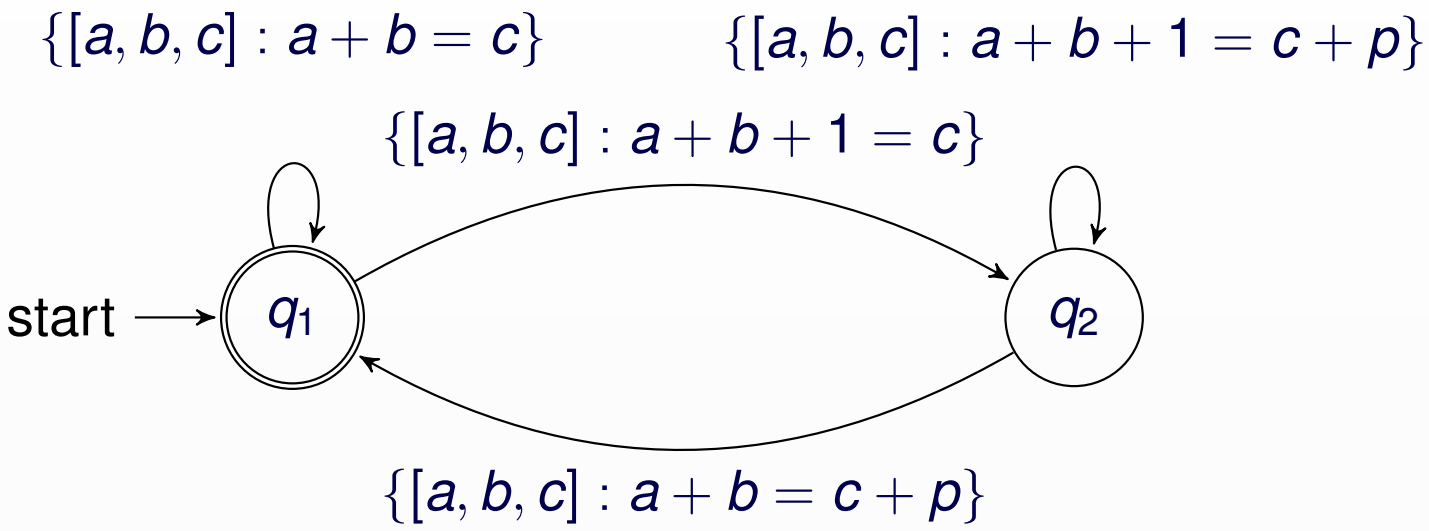
Proof - equality

Sets $M_=, M_+, M_{V_p}$ are p-automatic. They are accepted by automata $A_=, A_+, A_{V_p}$



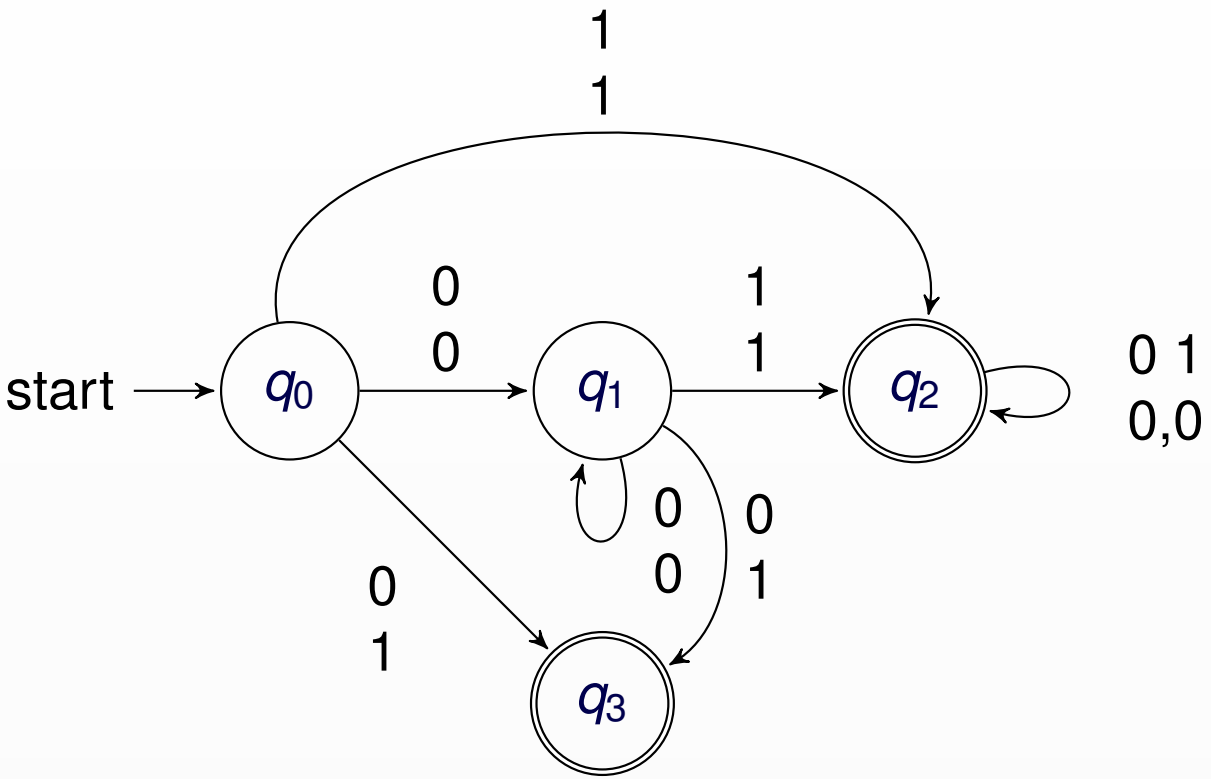
Automaton $A_=$ checking equality in base p

Proof - addition



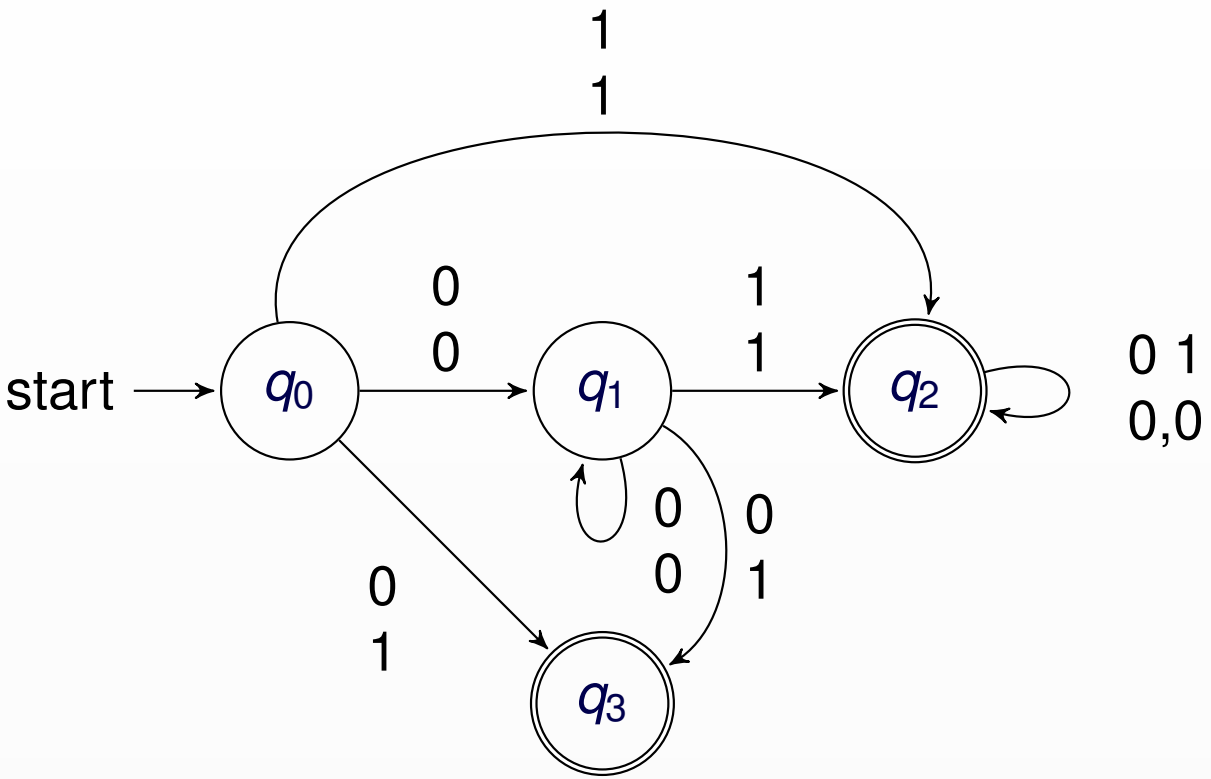
Automaton A_+ checking addition with carry in base p

Proof - the greatest power of p



Automaton A_{V_2} in base 2

Proof - the greatest power of p



Automaton A_{V_2} in base 2

Now by induction assume that automata A_φ and A_ψ are constructed. Obtain automata $A_{\varphi\vee\psi}, A_{\exists x\varphi}, A_{\neg\varphi}$.

The finish

Corollary

$Th(\langle \omega, + \rangle)$ and $Th(\langle \omega, +, V_p \rangle)$ are decidable theories.

The finish

Corollary

$Th(\langle \omega, + \rangle)$ and $Th(\langle \omega, +, V_p \rangle)$ are decidable theories.

Corollary

*There is an algorithm that, given a predicate phrased using only the universal and existential quantifiers, indexing into a given automatic sequence **a**, addition, subtraction, logical operations, and comparisons, will decide the truth of that proposition.*