FORMALIZED MATHEMATICS
Vol. 17, No. 2, Pages 193–199, 2009
DOI: 10.2478/v10037-009-0023-9

Equivalence of Deterministic and Nondeterministic Epsilon Automata

Michał Trybulec
YAC Software
Warsaw, Poland

Summary. Based on concepts introduced in [14], semiautomata and left-languages, automata and right-languages, and langauges accepted by automata are defined. The powerset construction is defined for transition systems, semiautomata and automata. Finally, the equivalence of deterministic and nondeterministic epsilon automata is shown.

MML identifier: `FSM_3`, version: `7.11.02 4.125.1059`

The terminology and notation used in this paper have been introduced in the following articles: [1], [8], [2], [11], [6], [18], [7], [9], [17], [16], [15], [4], [10], [13], [3], [12], [5], and [14].

1. Preliminaries

For simplicity, we adopt the following convention: x, y, X denote sets, E denotes a non empty set, e denotes an element of E, u, u_1, v, v_1, v_2, w denote elements of E^ω, F denotes a subset of E^ω, i, k, l denote natural numbers, $\mathfrak{T}$ denotes a non empty transition-system over F, and S, T denote subsets of $\mathfrak{T}$.

One can prove the following propositions:

(1) If $i \geq k + l$, then $i \geq k$.

(2) For all finite sequences a, b such that $a \frown b = a$ or $b \frown a = a$ holds $b = \emptyset$.

(3) For all finite sequences p, q such that $k \in \operatorname{dom} p$ and $\operatorname{len} p + 1 = \operatorname{len} q$ holds $k + 1 \in \operatorname{dom} q$.

(4) If $\operatorname{len} u = 1$, then there exists e such that $\langle e \rangle = u$ and $e = u(0)$.

ISSN 1426–2630(p), 1898-9934(e)

(5) If $k \neq 0$ and $\operatorname{len} u \leq k+1$, then there exist v_1, v_2 such that $\operatorname{len} v_1 \leq k$ and $\operatorname{len} v_2 \leq k$ and $u = v_1 \frown v_2$.

(6) For all finite 0-sequences p, q such that $\langle x \rangle \frown p = \langle y \rangle \frown q$ holds $x = y$ and $p = q$.

(7) If $\operatorname{len} u > 0$, then there exist e, u_1 such that $u = \langle e \rangle \frown u_1$.

Let us consider E. One can verify that $\operatorname{Lex} E$ is non empty.

Next we state three propositions:

(8) $\langle\rangle_E \notin \operatorname{Lex} E$.

(9) $u \in \operatorname{Lex} E$ iff $\operatorname{len} u = 1$.

(10) If $u \neq v$ and u, $v \in \operatorname{Lex} E$, then it is not true that there exists w such that $u \frown w = v$ or $w \frown u = v$.

2. Transition Systems over Lex E

The following propositions are true:

(11) For every transition-system $\mathfrak{T}$ over $\operatorname{Lex} E$ holds $\langle\rangle_E \notin \operatorname{rng} \operatorname{dom}$ (the transition of $\mathfrak{T}$).

(12) For every transition-system $\mathfrak{T}$ over $\operatorname{Lex} E$ such that the transition of $\mathfrak{T}$ is a function holds $\mathfrak{T}$ is deterministic.

3. Powerset Construction for Transition Systems

Let us consider E, F, $\mathfrak{T}$. The functor $\operatorname{bool} \mathfrak{T}$ yielding a strict transition-system over $\operatorname{Lex} E$ is defined by the conditions (Def. 1).

(Def. 1)(i) The carrier of $\operatorname{bool} \mathfrak{T} = 2^{\text{the carrier of } \mathfrak{T}}$, and

(ii) for all S, w, T holds $\langle\langle S, w \rangle, T \rangle \in$ the transition of $\operatorname{bool} \mathfrak{T}$ iff $\operatorname{len} w = 1$ and $T = w\text{-succ}_{\mathfrak{T}}(S)$.

Let us consider E, F, $\mathfrak{T}$. Note that $\operatorname{bool} \mathfrak{T}$ is non empty and deterministic.

Let us consider E, F and let $\mathfrak{T}$ be a finite non empty transition-system over F. One can check that $\operatorname{bool} \mathfrak{T}$ is finite.

The following two propositions are true:

(13) If $x, \langle e \rangle \Rightarrow^*_{\operatorname{bool} \mathfrak{T}} y, \langle\rangle_E$, then $x, \langle e \rangle \Rightarrow_{\operatorname{bool} \mathfrak{T}} y, \langle\rangle_E$.

(14) If $\operatorname{len} w = 1$, then $X = w\text{-succ}_{\mathfrak{T}}(S)$ iff $S, w \Rightarrow^*_{\operatorname{bool} \mathfrak{T}} X$.

4. Semiautomata

Let us consider E, F. We consider semiautomata over F as extensions of transition-system over F as systems

$\langle$ a carrier, a transition, an initial state $\rangle$,

where the carrier is a set, the transition is a relation between the carrier$\times F$ and the carrier, and the initial state is a subset of the carrier.

Let us consider E, F and let $\mathfrak{S}$ be a semiautomaton over F. We say that $\mathfrak{S}$ is deterministic if and only if:

(Def. 2) The transition-system of $\mathfrak{S}$ is deterministic and Card (the initial state of $\mathfrak{S}$) $= 1$.

Let us consider E, F. One can check that there exists a semiautomaton over F which is strict, non empty, finite, and deterministic.

In the sequel $\mathfrak{S}$ is a non empty semiautomaton over F.

Let us consider E, F, $\mathfrak{S}$. Observe that the transition-system of $\mathfrak{S}$ is non empty.

Let us consider E, F, $\mathfrak{S}$. The functor bool $\mathfrak{S}$ yields a strict semiautomaton over Lex E and is defined by the conditions (Def. 3).

(Def. 3)(i) The transition-system of bool $\mathfrak{S}$ = bool (the transition-system of $\mathfrak{S}$), and

(ii) the initial state of bool $\mathfrak{S} = \{\langle\rangle_E$-succ$_{\mathfrak{S}}$(the initial state of $\mathfrak{S})\}$.

Let us consider E, F, $\mathfrak{S}$. Observe that bool $\mathfrak{S}$ is non empty and deterministic.

The following proposition is true

(15) The carrier of bool $\mathfrak{S} = 2^{\text{the carrier of } \mathfrak{S}}$.

Let us consider E, F and let $\mathfrak{S}$ be a finite non empty semiautomaton over F. Observe that bool $\mathfrak{S}$ is finite.

5. Left-languages

Let us consider E, F, $\mathfrak{S}$ and let Q be a subset of $\mathfrak{S}$. The functor left-Lang Q yields a subset of E^ω and is defined as follows:

(Def. 4) left-Lang $Q = \{w : Q$ meets w-succ$_{\mathfrak{S}}$(the initial state of $\mathfrak{S})\}$.

Next we state the proposition

(16) For every subset Q of $\mathfrak{S}$ holds $w \in$ left-Lang Q iff Q meets w-succ$_{\mathfrak{S}}$(the initial state of $\mathfrak{S}$).

6. Automata

Let us consider E, F. We consider automata over F as extensions of semiautomaton over F as systems

$\langle$ a carrier, a transition, an initial state, final states $\rangle$,

where the carrier is a set, the transition is a relation between the carrier$\times F$ and the carrier, the initial state is a subset of the carrier, and the final states constitute a subset of the carrier.

Let us consider E, F and let $\mathfrak{A}$ be a automaton over F. We say that $\mathfrak{A}$ is deterministic if and only if:

(Def. 5) The semiautomaton of $\mathfrak{A}$ is deterministic.

Let us consider E, F. Observe that there exists a automaton over F which is strict, non empty, finite, and deterministic.

In the sequel $\mathfrak{A}$ denotes a non empty automaton over F and p, q denote elements of $\mathfrak{A}$.

Let us consider E, F, $\mathfrak{A}$. One can check that the transition-system of $\mathfrak{A}$ is non empty and the semiautomaton of $\mathfrak{A}$ is non empty.

Let us consider E, F, $\mathfrak{A}$. The functor $\operatorname{bool}\mathfrak{A}$ yields a strict automaton over $\operatorname{Lex} E$ and is defined by the conditions (Def. 6).

(Def. 6)(i) The semiautomaton of $\operatorname{bool}\mathfrak{A} = \operatorname{bool}$ (the semiautomaton of $\mathfrak{A}$), and

(ii) the final states of $\operatorname{bool}\mathfrak{A} = \{Q; Q$ ranges over elements of $\operatorname{bool}\mathfrak{A} : Q$ meets the final states of $\mathfrak{A}\}$.

Let us consider E, F, $\mathfrak{A}$. One can check that $\operatorname{bool}\mathfrak{A}$ is non empty and deterministic.

The following proposition is true

(17) The carrier of $\operatorname{bool}\mathfrak{A} = 2^{\text{the carrier of } \mathfrak{A}}$.

Let us consider E, F and let $\mathfrak{A}$ be a finite non empty automaton over F. Note that $\operatorname{bool}\mathfrak{A}$ is finite.

7. Right-languages

Let us consider E, F, $\mathfrak{A}$ and let Q be a subset of $\mathfrak{A}$. The functor right-Lang Q yields a subset of E^{ω} and is defined as follows:

(Def. 7) right-Lang $Q = \{w : w\text{-succ}_{\mathfrak{A}}(Q)$ meets the final states of $\mathfrak{A}\}$.

The following proposition is true

(18) For every subset Q of $\mathfrak{A}$ holds $w \in$ right-Lang Q iff $w\text{-succ}_{\mathfrak{A}}(Q)$ meets the final states of $\mathfrak{A}$.

8. Languages Accepted by Automata

Let us consider E, F, $\mathfrak{A}$. The language generated by $\mathfrak{A}$ yielding a subset of E^ω is defined by the condition (Def. 8).

(Def. 8) The language generated by $\mathfrak{A} = \{u : \bigvee_{p,q} (p \in$ the initial state of $\mathfrak{A} \wedge q \in$ the final states of $\mathfrak{A} \ \wedge \ p, u \Rightarrow^*_{\mathfrak{A}} q)\}$.

The following propositions are true:

(19) $w \in$ the language generated by $\mathfrak{A}$ if and only if there exist p, q such that $p \in$ the initial state of $\mathfrak{A}$ and $q \in$ the final states of $\mathfrak{A}$ and $p, w \Rightarrow^*_{\mathfrak{A}} q$.

(20) $w \in$ the language generated by $\mathfrak{A}$ if and only if w-succ$_{\mathfrak{A}}$(the initial state of $\mathfrak{A}$) meets the final states of $\mathfrak{A}$.

(21) The language generated by $\mathfrak{A} =$ left-Lang (the final states of $\mathfrak{A}$).

(22) The language generated by $\mathfrak{A} =$ right-Lang (the initial state of $\mathfrak{A}$).

9. Equivalence of Deterministic and Nondeterministic Epsilon Automata

In the sequel $\mathfrak{T}$ denotes a non empty transition-system over $\mathrm{Lex}\, E \cup \{\langle\rangle_E\}$.

One can prove the following three propositions:

(23) For every reduction sequence R w.r.t. $\Rightarrow_{\mathfrak{T}}$ such that $R(1)_{\mathbf{2}} = \langle e \rangle \frown u$ and $R(\mathrm{len}\, R)_{\mathbf{2}} = \langle\rangle_E$ holds $R(2)_{\mathbf{2}} = \langle e \rangle \frown u$ or $R(2)_{\mathbf{2}} = u$.

(24) For every reduction sequence R w.r.t. $\Rightarrow_{\mathfrak{T}}$ such that $R(1)_{\mathbf{2}} = u$ and $R(\mathrm{len}\, R)_{\mathbf{2}} = \langle\rangle_E$ holds $\mathrm{len}\, R > \mathrm{len}\, u$.

(25) For every reduction sequence R w.r.t. $\Rightarrow_{\mathfrak{T}}$ such that $R(1)_{\mathbf{2}} = u \frown v$ and $R(\mathrm{len}\, R)_{\mathbf{2}} = \langle\rangle_E$ there exists l such that $l \in \mathrm{dom}\, R$ and $R(l)_{\mathbf{2}} = v$.

Let us consider E, u, v. The functor $\mathrm{chop}(u, v)$ yielding an element of E^ω is defined by:

(Def. 9)(i) For every w such that $w \frown v = u$ holds $\mathrm{chop}(u, v) = w$ if there exists w such that $w \frown v = u$,

(ii) $\mathrm{chop}(u, v) = u$, otherwise.

The following propositions are true:

(26) Let p be a reduction sequence w.r.t. $\Rightarrow_{\mathfrak{T}}$. Suppose $p(1) = \langle x,\, u \frown w \rangle$ and $p(\mathrm{len}\, p) = \langle y,\, v \frown w \rangle$. Then there exists a reduction sequence q w.r.t. $\Rightarrow_{\mathfrak{T}}$ such that $q(1) = \langle x,\, u \rangle$ and $q(\mathrm{len}\, q) = \langle y,\, v \rangle$.

(27) If $\Rightarrow_{\mathfrak{T}}$ reduces $\langle x,\, u \frown w \rangle$ to $\langle y,\, v \frown w \rangle$, then $\Rightarrow_{\mathfrak{T}}$ reduces $\langle x,\, u \rangle$ to $\langle y,\, v \rangle$.

(28) If $x, u \frown w \Rightarrow^*_{\mathfrak{T}} y, v \frown w$, then $x, u \Rightarrow^*_{\mathfrak{T}} y, v$.

(29) For all elements p, q of $\mathfrak{T}$ such that $p, u \frown v \Rightarrow^*_{\mathfrak{T}} q$ there exists an element r of $\mathfrak{T}$ such that $p, u \Rightarrow^*_{\mathfrak{T}} r$ and $r, v \Rightarrow^*_{\mathfrak{T}} q$.

(30) $w \frown v\text{-succ}_{\mathfrak{T}}(X) = v\text{-succ}_{\mathfrak{T}}(w\text{-succ}_{\mathfrak{T}}(X))$.

(31) bool $\mathfrak{T}$ is a non empty transition-system over Lex $E \cup \{\langle\rangle_E\}$.

(32) $w\text{-succ}_{\text{bool}\,\mathfrak{T}}(\{v\text{-succ}_{\mathfrak{T}}(X)\}) = \{v \frown w\text{-succ}_{\mathfrak{T}}(X)\}$.

In the sequel $\mathfrak{S}$ denotes a non empty semiautomaton over Lex $E \cup \{\langle\rangle_E\}$.

One can prove the following proposition

(33) $w\text{-succ}_{\text{bool}\,\mathfrak{S}}(\{\langle\rangle_E\text{-succ}_{\mathfrak{S}}(X)\}) = \{w\text{-succ}_{\mathfrak{S}}(X)\}$.

In the sequel $\mathfrak{A}$ denotes a non empty automaton over Lex $E \cup \{\langle\rangle_E\}$ and P denotes a subset of $\mathfrak{A}$.

Next we state several propositions:

(34) If $x \in$ the final states of $\mathfrak{A}$ and $x \in P$, then $P \in$ the final states of bool $\mathfrak{A}$.

(35) If $X \in$ the final states of bool $\mathfrak{A}$, then X meets the final states of $\mathfrak{A}$.

(36) The initial state of bool $\mathfrak{A} = \{\langle\rangle_E\text{-succ}_{\mathfrak{A}}(\text{the initial state of } \mathfrak{A})\}$.

(37) $w\text{-succ}_{\text{bool}\,\mathfrak{A}}(\{\langle\rangle_E\text{-succ}_{\mathfrak{A}}(X)\}) = \{w\text{-succ}_{\mathfrak{A}}(X)\}$.

(38) $w\text{-succ}_{\text{bool}\,\mathfrak{A}}(\text{the initial state of bool}\,\mathfrak{A}) = \{w\text{-succ}_{\mathfrak{A}}(\text{the initial state of } \mathfrak{A})\}$.

(39) The language generated by $\mathfrak{A} =$ the language generated by bool $\mathfrak{A}$.

(40) Let $\mathfrak{A}$ be a non empty automaton over Lex $E \cup \{\langle\rangle_E\}$. Then there exists a non empty deterministic automaton $\mathfrak{A}_1$ over Lex E such that the language generated by $\mathfrak{A} =$ the language generated by $\mathfrak{A}_1$.

(41) Let $\mathfrak{F}$ be a non empty finite automaton over Lex $E \cup \{\langle\rangle_E\}$. Then there exists a non empty deterministic finite automaton $\mathfrak{A}_2$ over Lex E such that the language generated by $\mathfrak{F} =$ the language generated by $\mathfrak{A}_2$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[4] Grzegorz Bancerek. Reduction relations. *Formalized Mathematics*, 5(**4**):469–478, 1996.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[8] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[9] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[10] Karol Pąk. The Catalan numbers. Part II. *Formalized Mathematics*, 14(**4**):153–159, 2006, doi:10.2478/v10037-006-0019-7.
[11] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[12] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[13] Michał Trybulec. Formal languages – concatenation and closure. *Formalized Mathematics*, 15(**1**):11–15, 2007, doi:10.2478/v10037-007-0002-y.

[14] Michał Trybulec. Labelled state transition systems. *Formalized Mathematics*, 17(**2**):163–171, 2009, doi: 10.2478/v10037-009-0019-5.
[15] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[16] Tetsuya Tsunetou, Grzegorz Bancerek, and Yatsuka Nakamura. Zero-based finite sequences. *Formalized Mathematics*, 9(**4**):825–829, 2001.
[17] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[18] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received May 25, 2009