KYBERNETIKA ČÍSLO 2, ROČNÍK 2/1966

O jednom typu konečného automatu

VÁCLAV PINKAVA

V článku se popisuje zařízení o poměrně jednoduché vnitřní struktuře schopné realizovat velmi širokou třídu fixních konečných automatů. Je-li vybaveno dalším pomocným zařízením, je popisované zařízení schopno realizovat rostoucí logické sítě s omezením.

Definice 1. *Konečným automatem, univerzálním vzhledem k n nazveme zařízení, schopné simulovat chování libovolného konečného automatu, pokud tento automat splňuje podmínku*: $\tilde{\varrho}, \tilde{\varkappa}, \tilde{\lambda} \leq 2^n$ (*kde* $\tilde{\varrho}, \tilde{\varkappa}, \tilde{\lambda}$ *značí respektive počet vstupních, vnitřních a výstupních stavů simulovaného automatu*).

Označíme toto zařízení symbolem $\mathcal{U}_n$.

Principiální realizaci $\mathcal{U}_n$ si lze představit takto:

Zařízení má tyto části:

Vstupní a výstupní kanály, které zde budeme chápat jako binární. V tom případě je n vstupních a n výstupních kanálů. Stavy těchto kanálů nazveme vlastními vstupy a vlastními výstupy, kanály vlastními vstupními a vlastními výstupními kanály.

Dále má $\mathcal{U}_n$ čtecí hlavici s čidly, která zde budeme chápat rovněž jako binární. Kanály, předávající informaci přečtenou čtecí hlavicí, nazveme čtecími kanály a jejich stavy čtecími vstupy. Binárních čtecích kanálů je $2^{n+1}n$.

Dále má $\mathcal{U}_n$ zařízení posunující čtecí hlavici.

Kanály předávající informaci, na níž závisí posun čtecí hlavice, nazveme posunujícími výstupními kanály a jejich stavy posunujícími výstupy. Posun hlavice předpokládá zařízení, které jej provádí v závislosti na posunujících výstupech. Posunujících výstupních kanálů jest n.

Čtecí hlavice čte údaje, týkající se zadání simulovaného konečného automatu, které zde budeme chápat jako reprezentované sjednocenou kanonickou tabulkou binárně zakódovanou. Příkladem tabulky je tab. 1 pro automat $\mathcal{U}_2$. V horizontální hlavičce tabulky leží v každém poli jedno vstupní písmeno simulovaného automatu (zakódované binárním náborem), ve vertikální hlavičce leží v každém poli (buňce, políčku) tabulky jedno vnitřní písmeno, zakódované rovněž binárním náborem. V každém vnitřním poli tabulky leží dvě písmena, zakódovaná binárními nábory, a to údaj pro výstup a údaj pro vnitřní stav v dalším taktu simulovaného automatu. V tabulkách 1 a 2 níže uvedených konkrétních příkladů jsou tyto dvojice písmen (zakódovaných binárními nábory neboli současnými slovy) reprezentovány zlomky, v jejichž čitatelích leží vždy údaj pro další vnitřní stav a ve jmenovatelích údaj pro výstup v daném taktu (vzhledem k simulovanému automatu).

Bylo by si ovšem možno představit $\mathscr{U}_n$ pracující pomocí jiného způsobu zadání simulovaného automatu a to jak ve smyslu jiné úpravy čtené tabulky, tak ve smyslu zadání simulovaného automatu v jiném jazyce. Zde budeme vycházet ze základní představy, dané tabulkou typu příkladů tab. 1 a tab. 2.

Za předpokladu binárního zakódování tabulek simulovaných automatů lze vnitřní zařízení automatu $\mathscr{U}_n$ zadat binární logickou sítí $L_{\mathscr{U}_n}$ o rovnicích typu (1).

Logická síť $L_{\mathscr{U}_n}$ se netýká zařízení posunujícího čtecí hlavici, které by při technické realizaci $\mathscr{U}_n$ bylo nejspíše mechanické.

$$
(1) \qquad \left\{
\begin{aligned}
q_j^{t(+1)} &= \bigvee_{i=1}^{2^n} \Big[\bigwedge_{j=1}^{n} (x_j^t \equiv \xi_{ij}^t) \Big] \wedge \eta_{ij}^t \,, \\
z_j^t &= \bigvee_{i=1}^{2^n} \Big[\bigwedge_{j=1}^{n} (x_j^t \equiv \xi_{ij}^t) \Big] \wedge \zeta_{ij}^t \,.
\end{aligned}
\right.
$$

$\mathscr{U}_n$ má tedy logickou síť $L_{\mathscr{U}_n}$ o $2n$ kanonických rovnicích. V systému (1) značí $\bigvee$ – booleovskou (logickou) sumu, $\bigwedge$ – booleovský (logický) součin. Ostatní symboly mají běžný význam. Časové indexy jsou psány jako horní. Smysl závorky u $q_j^{t(+1)}$ bude vysvětlen dále. Proměnné x_j^t, z_j^t ($j = 1, 2, \ldots, n$) reprezentují jednotlivé binární kanály vlastních vstupů (x_j^t) a vlastních výstupů (z_j^t). Proměnné η_{ij}^t a ζ_{ij}^t ($i = 1, 2, \ldots, 2^n$) značí kanály čtecích vstupů resp. jednotlivá binární čidla. Proměnné $q_j^{t(+1)}$ značí kanály posunovacích výstupů. Symboly ξ_{ij}^t značí konstanty 0 nebo 1, reprezentující elementy jednotlivých binárních náborů, jimiž jsou zakódována vstupní písmena tabulky simulovaného automatu (v horizontální hlavičce tabulky) tak, že každé ξ_{ij}^t značí hodnotu, ležící v i-tém náboru (v i-tém poli neboli buňce hlavičky) na j-tém místě. Jestliže chápeme věc tak, že pořadí vstupních písmen v tabulce každého simulovaného automatu je stanoveno jednou provždy, je zápis pomocí konstant ξ_{ij}^t v systému (1) ryze formální. Při konstrukci logické sítě pro daný konkrétní $\mathscr{U}_n$ by se v tom případě postupovalo tak, že větvení kanálů vlastních vstupů, vstupující do určitého konjunktu by byla vybavena negačními elementy tam, kde příslušné $\xi_{ij}^t = 0$. V případě však, že by pořadí písmen (binárních náborů) v hlavičce tabulky každého simulovaného automatu nebylo jednou provždy určeno, reprezentovala by jednotlivá ξ_{ij}^t další čtecí vstupní kanály resp. binární čidla čtoucí vstupní údaje simulovaného automatu v hlavičce jeho tabulky. Hodnoty proměnných reprezentujících tato další čidla by byly konstantní po dobu, po kterou by $\mathscr{U}_n$ pracoval na jedné určité tabulce.

V konkrétním příkladě automatu $\mathscr{U}_2$, uvedeném níže, předpokládáme první eventualitu.

Automat $\mathscr{U}_n$ pracuje takto: V každém taktu pokrývá čtecí hlavice jeden řádek tabulky. Při tom čidla η^t_{ij} čtou údaje, týkající se vnitřního stavu simulovaného automatu pro další takt, a čidla ζ^t_{ij} údaje, týkající se výstupních stavů simulovaného automatu pro daný takt. (Každé η^t_{ij} resp. ζ^t_{ij} nabývá stavu, odpovídajícího hodnotě, která leží v i-tém poli (buňce) daného řádku na j-tém místě. Při tom η^t_{ij} čtou údaje v čitateli, ζ^t_{ij} údaje ve jmenovateli zlomku.) Nábor hodnot, reprezentující vlastní vstup podaný v taktu t, způsobí, že v témž taktu t se na vlastním výstupu objeví nábor hodnot, který odpovídá výstupu simulovaného automatu v tom poli daného řádku tabulky, v jejíž (horizontální) hlavičce leží nábor hodnot, který se objevil na vlastním vstupu. Zároveň se na posunovacím výstupu automatu $\mathscr{U}_n$ objeví nábor hodnot, odpovídající vnitřnímu stavu pro simulovaný automat v taktu $t + 1$. Tímto náborem je zakódován rozkaz pro posun čtecí hlavice na další řádek tabulky simulovaného automatu. (Vyhledání dalšího pole vertikální hlavičky tabulky.)

Identifikujeme-li slovo, dané náborem hodnot $q_j^{t(+1)}$ s *faktem* posunu hlavice, můžeme v systému (1) chápat jednotlivá q_j jako zapsaná s indexem $t + 1$. Z tohoto hlediska představuje $L_{\mathscr{U}_n}$ spolu se čtecí hlavicí konečný automat s pamětí, ale bez cyklu. Pokládáme-li nábor hodnot $q_j^{t(+1)}$ v daném tlaku za *rozkaz* pro posun hlavice, vydaný již v taktu t (ačkoliv jeho "efekt" trvá do dalšího taktu), můžeme q_j^t chápat jako výstupy a $L_{\mathscr{U}_n}$ o sobě se jeví jako konečný automat bez paměti.

$\mathscr{U}_n$ může být vybaven zařízením, umožňujícím nastavení čtecí hlavice na libovolný řádek tabulky simulovaného automatu pro takt $t = 1$. Tím lze zajistit určení počátečního vnitřního stavu simulovaného automatu pro libovolné chování.

Svrchu popsaný proces čtení údajů v řádku tabulky, na který je hlavice v daném taktu nastavena, výběr údajů na základě vlastního vstupu, objevení se přečteného výstupu na vlastním výstupním kanálu a nastavení hlavice na další řádek tabulky podle přečteného údaje pro další vnitřní stav, se opakuje v každém taktu, pokud $\mathscr{U}_n$ pracuje na nějaké tabulce.

Tak $\mathscr{U}_n$ simuluje vlastně chování osoby, instruované jak číst sjednocenou kanonickou tabulku konečného automatu, které byla dána nějaká taková tabulka s další instrukcí, zahrnující určení počátečního vnitřního stavu (řádku tabulky) a povel, aby tato osoba hlásila výstupní stavy automatu, jehož tabulku čte, podle toho, jaké vstupy jí budou hlášeny. Osoba si v každém taktu vyhledá uvnitř tabulky vnitřní stav pro další takt, který si zapamatuje. (Posune prst na ten řádek, v jehož hlavičce je symbol, který nalezla uvnitř tabulky.) Taková osoba je zřejmě schopna simulovat libovolné chování libovolného konečného automatu, pokud počet symbolů pro jejich stavy nepřevýší její diskriminační schopnost. Meze diskriminační schopnosti automatu $\mathscr{U}_n$ jsou určeny právě číslem n. (Z ryze psychologického hlediska má automat $\mathscr{U}_n$, jak je zde popsán, proti lidskému čtenáři tabulky tu výhodu, že „přehlédne jedním pohledem" celý řádek čtené tabulky, což lidský čtenář nedokáže během jednoho „taktu", jakmile počet symbolů převýší cca 5.)

Princip $\mathscr{U}_n$ a jeho činnosti demonstrujeme nyní na elementárním příkladě automatu $\mathscr{U}_2$.

Automat $\mathscr{U}_2$ má logickou síť $L_{\mathscr{U}_2}$ zadatelnou rovnicemi:

$$
(2) \qquad \left\{ \begin{aligned}
q_2^{t(+1)} &= \bar{x}_1^t \bar{x}_2^t \eta_{11}^t \vee \bar{x}_1^t x_2^t \eta_{21}^t \vee x_1^t \bar{x}_2^t \eta_{31}^t \vee x_1^t x_2^t \eta_{41}^t \,, \\
q_2^{t(t+1)} &= \bar{x}_1^t \bar{x}_2^t \eta_{12}^t \vee \bar{x}_1^t x_2^t \eta_{22}^t \vee x_1^t \bar{x}_2^t \eta_{32}^t \vee x_1^t x_2^t \eta_{42}^t \,, \\
z_1^t &= \bar{x}_1^t \bar{x}^t \zeta_{11}^t \vee \bar{x}_1^t x_2^t \zeta_{21}^t \vee x_1^t \bar{x}_2^t \zeta_{31}^t \vee x_1^t x_2^t \zeta_{41}^t \,, \\
z_2^t &= \bar{x}_1^t \bar{x}_1^t \zeta_{12}^t \vee \bar{x}_1^t x_2^t \zeta_{22}^t \vee x_1^t \bar{x}_2^t \zeta_{32}^t \vee x_1^t x_2^t \zeta_{42}^t \,.
\end{aligned} \right.
$$

(Jelikož předpokládáme zmíněnou první eventualitu, totiž že pořadí vstupních písmen v hlavičce tabulky každého automatu, na níž $\mathscr{U}_2$ pracuje, jest určeno jednou pro vždy, jsou místo výrazů $(x_j^t \equiv \xi_{ij}^t)$ psány jen vstupní proměnné v asertivním nebo v negačním tvaru, neboť platí: $(x \equiv 0) = \bar{x}$, $(x \equiv 1) = x$. Při obecném zápisu v (1) tohoto zkrácení nelze použít, neboť n není specifikováno.)

Bude-li $\mathscr{U}_2$ pracovat na tabulce 1, bude simulovat konečný automat, zadatelný kanonickým systémem (3)

$$
(3) \qquad \left\{ \begin{aligned}
q_1^{t+1} &= \bar{x}_1^t x_2^t \vee [x_1^t \bar{x}_2^t \,.\, (\bar{q}_1^t \vee q_2^t)] \,, \\
q_2^{t+1} &= q_1^t \,.\, (x_2^t \equiv q_2^t) \,, \\
z_1^t &= (x_1^t \vee x_2^t) \,.\, q_1^t q_2^t \,, \\
z_2^t &= \bar{q}_1^t \,.
\end{aligned} \right.
$$

Dejme tomu, že v taktu 1 bude čtecí hlavice automatu $\mathscr{U}_2$ nastavena na první řádek tabulky 1, tj. simulovaný automat začne pracovat z vnitřního stavu 00. Nechť se v témž taktu objeví na vlastním vstupu binární nábor 10.

Tab. 1.

x_1, x_2 \\ q_1, q_2	00	01	10	11
00	00/01	10/01	10/01	00/01
01	00/01	10/01	10/01	00/01
10	01/00	10/00	01/00	00/00
11	00/00	11/10	10/00	01/10

Dosadíme-li do systému (2) udané hodnoty vlastních binárních vstupů a zároveň za η_{ij}^t a ζ_{ij}^t hodnoty, které čtecí hlavice čte v prvním řádku tabulky 1, počítáno shora, dostáváme:

$$\begin{aligned}
q_1^{1(+1)} &= 0.1.0 \vee 0.0.1 \vee \underbracket{1.1.1} \vee 1.0.0 = 1\,,\\
q_2^{1(+1)} &= 0.1.0 \vee 0.0.0 \vee \underbracket{1.1.0} \vee 1.0.0 = 0\,,\\
z_1^1 &= 0.1.0 \vee 0.0.0 \vee \underbracket{1.1.0} \vee 1.0.0 = 0\,,\\
z_2^1 &= 0.1.1 \vee 0.0.1 \vee \underbracket{1.1.1} \vee 1.0.1 = 1\,.
\end{aligned}$$

Všechny konjunkce vstupních proměnných, kromě té, které odpovídá hodnota vlastního vstupu v taktu 1 budou anulovány. Tím je zajištěno, že vlastní výstupy z_1^1, z_2^1 a posunovací výstupy $q_1^{1(+1)}, q_2^{1(+1)}$ nabudou hodnot, které hlavice přečtla v tab. 1 v tom poli (buňce) řádku o hlavičce 00, v jehož hlavičce pro vstup je hodnota vstupu 10.

Hodnoty 01 se objeví na vlastním výstupu v taktu 1 (jsou vydány ven) a hodnoty posunovací výstupů v taktu 1 způsobí posunutí hlavice na třetí řádek shora tabulky tab. 1.

Je-li v taktu 2 podán na vlastní vstup $\mathscr{U}_2$ nábor hodnot 01, vydá automat $\mathscr{U}_2$ vlastní výstup 00 a hlavice zůstane na témž (třetím) řádku (provede identický posun), jak se přesvědčíme novým dosazením do systému (2), analogickým tomu, jaké jsme provedli pro takt 1.

Bude-li v taktu 3 podán vlastní vstup 11, vydá automat $\mathscr{U}_2$ vlastní výstup 00 a zároveň se hlavice posune na prvý řádek shora tabulky tab. 1, jak zjistíme dalším dosazením.

Dosadíme-li do systému (3) pro takt 1 hodnoty: $x_1^1 = 1, x_2^1 = 0, q_1^1 = 0, q_2^1 = 0$, pro takt 2 vstupní hodnoty 01 a hodnoty vnitřních proměnných vypočtené v taktu 1, pro takt 3 vstupní hodnoty 11 a hodnoty vnitřních proměnných, vypočtené v taktu 2, dostaneme stejnou výstupní posloupnost, totiž 01, 00, 00, jako při popsaném chování automatu $\mathscr{U}_2$, pracujícího na tabulce 1.

Tak by bylo možno pokračovat dále.

Jiný příklad:

Automat $\mathscr{U}_2$ pracující pomocí tabulky 2 bude realizovat (simulovat) chování konečného automatu o kanonických rovnicích:

$$(4) \qquad \begin{cases} q^{t+1} = (x^t \not\equiv q^t)\,, \\ z^t \phantom{{}^{+1}} = x^t q^t\,. \end{cases}$$

Jde o dvojkový sčítač následných vstupů.

Vzhledem k vlastním vstupům, vlastním výstupům a posunovacím výstupům (tj. „vlastním" vnitřním stavům simulovaného automatu) platí zde pro vztah proměnných automatu $\mathscr{U}_2$ a automatu o rovnicích (4): $x^t = x_1^t$, $q^\tau = q_1^\tau$, $z^t = z_1^t$ ($\tau = t, t + 1$). Stavy proměnných x_2^t, q_2^τ, z_2^t automatu $\mathscr{U}_2$ jsou vzhledem k simulování a utomatu (4) irrelevantní.

Tab. 2.

q_1, q_2 \ x_1, x_2	00	01	10	11
00	00/00	00/00	10/00	10/00
01	00/00	00/00	10/00	10/00
10	10/00	10/00	00/10	00/10
11	10/00	10/00	00/10	00/10

Tabulka 2 je zkonstruována tak, že v každém taktu $t \geqq 1$ může být na vstupu x_2^t libovolná hodnota a v taktu $t = 1$ může být hodnota q_2^t rovněž libovolná. V každém taktu $t > 1$ bude $q_2^t = 0$ a v každém taktu $t \geqq 1$ bude $z_2^t = 0$.

Učiníme-li konvenci, že stavy vlastního vstupu automatu $\mathscr{U}_2$ zakódované nábory 01, 11 jsou nepřípustné pro $t \geqq 1$ (nemohou se vyskytnout, nepodáváme je) a nastavení hlavice na řádky 2 a 4 tabulky 2 je rovněž nepřípustné pro takt $t = 1$, může být tab. 2 zadána také tak, že všechna pole (buňky) kromě těch, která jsou zde silně orámována, se ponechají prázdná. O tom, že automat $\mathscr{U}_2$ pracující na tab. 2 simuluje automat (4), se můžeme přesvědčit dosazováním, analogickým tomu, jaké jsme prováděli v prvním uvedeném příkladě.

Právě uvedený příklad demonstruje případ, kdy $\mathscr{U}_n$ realizuje automat o počtu stavů $\tilde{\varrho}, \tilde{\varkappa}, \tilde{\lambda} < 2^n$ (viz definici 1).

Jak je vidno, je automat $\mathscr{U}_2$ schopen simulovat chování všech konečných automatů zadatelných logickými sítěmi o rovnicích typu:

$$\begin{aligned} q_1^{t+1} &= \varphi_1(x_1^t, x_2^t, q_1^t, q_2^t)\,, \\ q_1^{t+1} &= \varphi_2(x_1^t, x_2^t, q_1^t, q_2^t)\,, \\ z_1^t &= \psi_1(x_1^t, x_2^t, q_1^t, q_2^t)\,, \\ z_2^t &= \psi_2(x_1^t, x_2^t, q_1^t, q_2^t)\,, \end{aligned}$$

kde symboly x_1^t, x_2^t, q_1^t, q_2^t jsou buď binární proměnné nebo konstanty 0 nebo 1.

Dále se zmíníme o některých vlastnostech $\mathscr{U}_n$, které vyplývají z jeho povahy.

1. *Počet konečných automatů, realizovatelných automatem $\mathscr{U}_n$ je 2^N, kde $N = 2^{2n+1}n$.*

D ů k a z: Automat $\mathscr{U}_n$ pracuje pomocí tabulky o nejvíce 2^n vstupních a 2^n vnitřních symbolech. Tabulka takového automatu má zřejmě 2^{2n} vnitřních polí (buněk), kde v každém poli leží $2n$ binárních symbolů (vždy n pro další vnitřní stav a n pro výstup – srovnej tab. 1 a tab. 2 pro $\mathcal{U}_2$).

Každou tabulku lze tedy reprezentovat binárním náborem (posloupností) o délce: $2^{2n+1}n$. Všech tabulek konečných automatů, na kterých může daný $\mathcal{U}_n$ pracovat, je tedy tolik, kolik je různých takových posloupností. Je jich zřejmě 2^N, kde $N = 2^{2n+1}n$. Tak automat $\mathcal{U}_2$ může simulovat již 2^{64} konečných automatů. (Některé automaty jsou ovšem izomorfní navzájem a liší se pouze způsobem reprezentace stavů a některé jsou triviální.)

2. *Automat $\mathcal{U}_n$ je schopen v rámci dané třídy simulovat automat libovolného typu, tj. Mealyho stroj, Moorův automat a dva zbývající typy.*

Důkaz vyplývá z interpretovatelnosti každého konečného automatu v termínech Mealyho stroje. Důkaz toho je v [1] str. 181–183 a nebudeme ho zde opakovat. Tabulka, na které $\mathcal{U}_n$ pracuje, je tabulkou Mealyho stroje, resp. $\mathcal{U}_n$ ji tímto způsobem čte (interpretuje). Z těchto dvou faktů vyplývá správnost uvedeného tvrzení.

3. *Automat $\mathcal{U}_n$ je schopen simulovat v rámci dané třídy chování automatu bez vstupu, bez (samostatného) výstupu a bez paměti.*

Konečný automat bez vstupu lze chápat jako konečný automat s jediným vstupem. Bude-li $\mathcal{U}_n$ pracovat na tabulce, ve které budou údaje ve všech sloupcích stejné, nebude jeho chování záviset na druhu vstupu, nýbrž libovolný vstup se bude uplatňovat jen jako signál pro nástup taktu. Takto se bude $\mathcal{U}_n$ chovat jako automat bez vstupu.

Analogicky: Bude-li $\mathcal{U}_n$ pracovat na tabulce, ve které budou údaje ve všech řádcích stejné, bude se chovat jako automat bez paměti. (V těchto případech nemusí být tabulka vyplněna ve všech polích.)

Konečně, bude-li $\mathcal{U}_n$ pracovat na takové tabulce, ve které se v každém políčku (buňce) budou údaje pro vnitřní stav a výstup navzájem shodovat, bude se $\mathcal{U}_n$ chovat jako automat bez samostatného výstupu.

4. *Automat $\mathcal{U}_n$ není schopen simulovat sám sebe.*

Důkaz: Počet všech (binárních) vstupů automatu $\mathcal{U}_n$ jest $n + 2^{n+1}n$. Vlastních vstupů je n. Vstupů čtecí hlavice je tolik, kolik je údajů (binárních míst) v jednom řádku tabulky. Řádek tabulky pro $\mathcal{U}_n$ má 2^n polí (buněk) a v každém leží $2n$ binárních míst. Je tedy všech vstupů automatu (binárně zakódovaných), tj. vlastních i čtecích, dohromady $n + 2^{n+1}n$. Automat $\mathcal{U}_n$ jest však schopen simulovat jen automaty, jejichž počet vstupů nepřesahuje 2^n.

Není tedy $\mathcal{U}_n$ schopen simulovat sám sebe.

Definice 2.* *Fixní (konečný) automat je každé zařízení, reprezentovatelné (řádně vytvořenou) logickou sítí.*

* Viz [2].

Definice 3.* *Rostoucí automat je posloupnost fixních automatů, z nichž každý je určen počátečním konečným automatem a pravidlem, podle něhož fixní automat v čase t určuje fixní automat v čase $t + 1$.*

Definice 4. *Rostoucím automatem s omezením nazveme takový rostoucí automat (posloupnost fixních automatů), není-li žádný z fixních automatů, který se může v posloupnosti vyskytnout, schopen většího počtu stavů, než jaký je dán určitým konečným (přirozeným) číslem.*

Z definic 1 a 4 plyne bezprostředně:

5. *Automat $\mathscr{U}_n$ vybavený zařízením na střídání tabulek v závislosti na vlastním chování simuluje rostoucí automat s omezením.*

Specifikací způsobů, jak lze formálně zadat rostoucí automat (realizovaný pomocí $\mathscr{U}_n$ nebo jinak) se zde zabývat nebudeme. Základní myšlenku obsahuje citovaná práce [2].

Jestliže by takové zadání rostoucího automatu s omezením mělo formu tabulky, na které by pracoval nějaký k tomu vhodný automat $\mathscr{U}'_n$ spojený s automatem $\mathscr{U}_n$ pomocí regulárních cyklů, přičemž vlastní výstup automatu $\mathscr{U}'_n$ by řídil střídání tabulek pro automat $\mathscr{U}_n$, simulovala by tato dvojice automatů rostoucí automat s omezením.

Otázkami s tím spojenými se hodláme zabývat jindy.

Studium sítí, jejichž elementy jsou automaty typu $\mathscr{U}_n$ (případně s malými n) by bylo možno pokládat za určitý přístup k modelování adaptivních nervových sítí.

Automat $\mathscr{U}_n$ lze dále pokládat za abstraktní model počítače s programovým řízením a vzniká otázka, zda by tohoto pojmu nebylo možno využít k určitému přístupu k teorii programových automatů.

Jak dalece by bylo možno pojetí $\mathscr{U}_n$ využít prakticky při konstrukci technických cifrových zařízení, neodvažujeme se rozhodnout. Mnoho by zde pravděpodobně záleželo na technickém vyřešení zadání simulovaných automatů, které jsme zde chápali jako provedené pomocí kánonických sjednocených tabulek.

Vzhledem k tomu, že počet automatů, jež může daný $\mathscr{U}_n$ simulovat, je v pravdě astronomický již při malých n, přičemž vnitřní logika je poměrně jednoduchá, stála by snad otázka praktického využití automatu $\mathscr{U}_n$ za úvahu.

Poznamenáme ještě, že hypotéza fyziologa Galambose, týkající se vztahu mezi funkcí neuronových sítí a funkcí glie, má nebo by mohla mít zřejmou souvislost s pojetím automatu $\mathscr{U}_n$, jak hodláme ukázat v jiné práci.

Závěrem autor děkuje panu prof. dr. O. Zichovi, DrSc. za cenné připomínky týkající se formulací a úpravy textu.

(Došlo dne 8. června 1965.)

* Viz [2].

[1] М. А. Айзерман, Л. А. Гусев, Л. И. Розоноэр, И. М. Смирнова, А. А. Таль: Логика, автоматы, алгорифмы. Москва 1963.
[2] A. W. Burks: The logic of fixed and growing automata. Internat. Symp. on the Theory of Switching, vol. *29* (1959).

SUMMARY

On a Certain Type of Finite Automata

VÁCLAV PINKAVA

A device is described simulating the behaviour of a person knowing how to read canonic (united) tables of finite automata and instructed to enounce output values found in a given table according to the initial inner state indicated and according to the input values told to him successively. Obviously, such a person is able to simulate any behaviour of any finite automaton the table of which it is still able to read, i.e. to discriminate the symbols of this table.

It consists of input and output channels, called proper input and outputs channels, these communicating the information corresponding to the information told to and told by the table-reading person. The automaton has further a reading head moving across the table and a device, realizing the movements of the head according to the information read by and "told" to the automaton.

The logic of the automaton is given by the canonic system (1). The system (2) represents the logical net of a small device of the type described, capable of simulating finite automata the nomber of states of which does not exceed 2^2. It is able to simulate 2^{64} finite automata.

Under certain conditions involving cheafly an auxiliary device (which may be realized by another automaton of the type described) the machine is able to simulate growing automata.

Dr. Václav Pinkava, Psychiatrická klinika KU, Praha 2, Ke Karlovu 11