VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

AUTOMATOVÉ SYSTÉMY

AUTOMATA SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE PATRIK PETŘÍK
AUTHOR

VEDOUCÍ PRÁCE prof. RNDr. ALEXANDER MEDUNA, CSc.
SUPERVISOR

BRNO 2007

Zadání bakalářské práce/5025/2006/xpetri11

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Petřík Patrik**
Obor: Informační technologie
Téma: **Automatové systémy**
Kategorie: Teorie informatiky
Pokyny:

1. Seznamte se detailně s gramatickými systémy.
2. Analogicky s gramatickými systémy zaveďte automatové systémy na bázi automatů. Studujte vlastnosti výsledných systémů. Porovnejte je s vlastnostmi gramatických systémů.
3. Studujte užití navržených systémů (např. kompilátory, lingvistika, mikrobiologie). Navrhněte vhodnou aplikaci založenou na automatových systémech. Testujte ji.
4. Zhodnoťte dosažené výsledky a diskutujte další možný vývoj projektu.

Literatura:
- Meduna, A.: Automata and Languages, Springer, London, 2000

Při obhajobě semestrální části projektu je požadováno:
- Body 1-2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Meduna Alexander, prof. RNDr., CSc.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2
L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení:	**Patrik Petřík**
Id studenta:	84201
Bytem:	Hutník 1446, 698 01 Veselí nad Moravou
Narozen:	23. 11. 1983, Kyjov

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

....

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP:	Automatové systémy
Vedoucí/školitel VŠKP:	Meduna Alexander, prof. RNDr., CSc.
Ústav:	Ústav informačních systémů
Datum obhajoby VŠKP:	....

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: ....

....	....
Nabyvatel	Autor

Abstrakt

V této práci jsou definovány paralelní automatové systémy a paralelně komunikující automatové systémy komunikující přechody, které obsahují jako komponenty konečné automaty. Jsou zkoumány jejich vlastnosti vzhledem k jiným gramatickým systémům či gramatikám.

Klíčová slova

Paralelní gramatika, maticová gramatika, gramatický systém, automatový systém, paralelně komunikující automatový systém.

Abstract

This work deals with automata systems. We define parallel automata systems and parallel communicating automata systems communicating by transitions, whose components are finite state automata. We investigate their properties compared with grammar systems or grammars.

Keywords

Parallel grammar, matrix grammar, grammar system, automata system, parallel communicating automata system.

Citace

Patrik Petřík: Automatové systémy, bakalářská práce, Brno, FIT VUT v Brně, 2007

Automatové systémy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením prof. RNDr. Alexandra Meduny, CSc.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……………………
Patrik Petřík
15. dubna 2007

Poděkování

Na tomto místě bych rád poděkoval prof. RNDr. Alexandru Medunovi, CSc., vedoucímu bakalářské práce, za odbornou pomoc, ochotu a čas, který mi při tvorbě práce věnoval.

Obsah

Obsah ... 1
Úvod ... 2
1 Terminologie ... 3
1.1 Základní terminologie ... 3
1.2 Gramatické systémy ... 4
1.3 m-paralelní n-pravé lineární maticové gramatiky ... 5
1.4 Automatové systémy ... 5
2 Definice ... 8
2.1 Paralelní automatové systémy ... 8
2.2 Paralelně komunikující automatové systémy ... 9
2.3 Vlastnosti automatových systémů ... 14
3 Budoucí výzkum ... 27
Závěr ... 28
Literatura ... 29
Seznam příloh ... 30

Úvod

V této práci definujeme automatové systémy, které jsou založeny na paralelizmu, tedy na principu, kterému je poslední dobou věnováno hodně pozornosti a to především z toho důvodu, neb umožňuje pomocí více jednoduchých struktur docílit stejného efektu jako při použití jedné mnohem složitější struktury.

V práci je definováno více druhů automatových systémů a jsou studovány jejich vlastnosti a to jak vzhledem k jiným formám automatů, tak vzhledem ke gramatikám či gramatickým systémům, kterým byla doposud věnována větší pozornost nežli systémům složených z automatů.

V první kapitole jsou připomenuty důležité pojmy nutné pro další kapitoly. Jsou zde zmíněny především gramatické systémy a paralelní gramatiky, jež tvoří jakýsi vzor vůdči kterému porovnáváme vlastnosti námi navržených systémů. Dále jsou připomenuty jiné formy automatů, které jsou v práci taktéž využity pro možnost srovnání či jsou použity při důkazech. V neposlední řadě jsou zmíněny i jiné výsledky, kterých bylo již na poli automatových systémů dosaženo a to především distribuované automatové systémy či jiné paralelně komunikující automatové systémy.

V druhé kapitole jsou nadefinovány nejprve paralelní automatové systémy, tedy systémy, které neobsahují komunikaci jako takovou, tedy žádné komunikační přechody či stavy. Dále jsou nadefinovány paralelně komunikující automatové systémy, které komunikují pomocí komunikačních přechodů. Poté jsou studovány vlastnosti těchto systémů vzhledem jiným strukturám.

V třetí kapitole je diskutován další možný vývoj v této oblasti založen na námi definovaných automatových systémech.

1 Terminologie

V této kapitole připomeneme mnohé důležité pojmy z oblasti formálních jazyků. Pro podrobnější informace o jednotlivých pojmech odkážeme na vhodnou literaturu.

1.1 Základní terminologie

Abeceda je konečná, neprázdná množina elementů, které nazýváme symboly. Množinu všech řetězců nad abecedou Σ značíme Σ^*. ε značí prázdný řetězec. Potom Σ^+ značí $\Sigma^* - \{\varepsilon\}$. Označme REG, LIN, CF, CS a RE třídy regulárních, lineárních, bezkotextových, kontextových a rekurzivně vyčíslitelných jazyků.

Definice 1.1. Konečný automat je pětice

$$M = (Q, \Sigma, R, s, F),$$

kde Q je konečná množina stavů, Σ je abeceda vstupních symbolů, R je konečná množina pravidel tvaru $pa \to q$, kde $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $s \in Q$ je počáteční stav, $F \subseteq Q$ je množina koncových stavů.

Definice 1.2. Pravá lineární gramatika je čtveřice

$$G = (N, T, P, S),$$

kde N je konečná množina nonterminálních symbolů, T je konečná množina terminálních symbolů, $N \cap T = \phi$, P je konečná podmnožina kartézského součinu $N \times T^* \cup T^* N$, $S \in N$ je startovací symbol gramatiky.

Algoritmus 1.1. Konstrukce nedeterministického konečného automatu k pravé lineární gramatice, která obsahuje pouze pravidla typu $A \to aB$ nebo $A \to \varepsilon$.

K pravé lineární gramatice $G = (N, T, P, S)$ sestrojíme nedeterministický konečný automat $M = (Q, \Sigma, R, s, F)$ takto:

1. $Q = N$
2. $\Sigma = T$
3. $R : R(A, a)$ obsahuje B, jestliže $A \to aB$ je v P
4. $s = S$
5. $F = \{A \mid (A \to \varepsilon) \in P\}$

Algoritmus 1.2. Konstrukce pravé lineární gramatiky k nedeterministickému konečnému automatu.

K nedeterministickému konečnému automatu $M = (Q, \Sigma, R, s, F)$ sestrojíme pravou lineární gramatiku $G = (N, T, P, S)$ následujícím postupem:

1. $N = Q$

2. $T = \Sigma$

3. je-li $R(q,a) = r$, pak P obsahuje pravidlo $q \to ar$

4. je-li $p \in F$, pak P obsahuje pravidlo $p \to \varepsilon$

5. $S = s$

Podrobnější informace lze nalézt v následující literatuře: [1], [7].

1.2 Gramatické systémy

Gramatickým systémem rozumíme množinu gramatik, které spolupracují podle jistého protokolu a společně tak generují daný jazyk. Gramatické systémy dělíme na dvě základní třídy: sekvenční tzv. „Cooperation Distributed“ a paralelní tzv. „Parallel Communicating“.

Definice 1.3. CD gramatický systém stupně *n*, $n \geq 1$ je $(n+3)$-tice

$$\Gamma = (N, T, S, P_1, ..., P_n),$$

kde *N*, *T* jsou disjunktní abecedy, $S \in N$, a $P_1, ..., P_n$ jsou konečné množiny přepisovacích pravidel nad množinou $N \cup T$. Prvky množiny *N* jsou nonterminály, prvky *T* jsou terminály, $P_1, ..., P_n$ nazýváme komponenty systému.

Tyto systémy pracují sekvenčně, tedy v jednom okamžiku je aktivní pouze jediná komponenta. U těchto systémů jsou definovány podmínky, kdy je daná komponenta aktivní a mód derivace, tedy jak dlouho smí provádět derivace než předá řízení jiné komponentě.

Označíme $CD_n(f)$ třídu jazyků *CD* gramatických systémů stupně *n*, $n \geq 1$, obsahující bezkotextová pravidla a pracující v derivačním módu *f*, kde $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$. Potom z [2] plyne zajímavá vlastnost těchto systémů a to konkrétně: $CD_\infty(t) \subseteq CD_3(t)$ tedy, počet komponent takových systémů pracujících v módu derivace *t* lze zminimalizovat až na tři. Jak uvidíme dále mnohé jiné systémy podobnou vlastnost nemají.

Definice 1.4. PC gramatický systém stupně *n*, $n \geq 1$ je $(n+3)$-tice

$$\Gamma = (N, K, T, (S_1, P_1), ..., (S_n, P_n)),$$

kde *N* je abeceda nonterminálů, *T* je abeceda terminálů, $K = \{Q_1, Q_2, ..., Q_n\}$, množiny *N, T, K* jsou vzájemně disjuktní, P_i je množina přepisovacích pravidel nad $N \cup K \cup T$ a $S_i \in N$, pro všechna $1 \leq i \leq n$. Tyto systémy pracují paralelně, každá komponenta zpracovává svou větnou formu a řízení komponent je realizováno komunikačními symboly. Jazyk, který generuje první komponenta je i výsledným jazykem celého systému. Existuje několik typů těchto systémů například centrální, necentrální, vracející se apod. Více o těchto systémech v [2].

1.3 m-paralelní n-pravé lineární maticové gramatiky

Nyní připomeneme další paralelní strukturu a tou je m-paralelní n-pravá linearní maticová gramatika. I když se nejedná o gramatický systém ale pouze o gramatiku, přesto díky formě přepisovacích pravidel, má tato gramatika mnohé podobné rysy jako gramatický systém. Více o těchto gramatikách v díle [8], [9]. Zde připomeňme alespoň definici těchto systémů.

Definice 1.5. *m*-paralelní *n*-pravá linearní maticová gramatika (*m-Pn-G*) je (*mn*+3)-tice

$$G = (N_{11},...,N_{1n},...,N_{m1},...,N_{mn},T,S,P),$$

kde N_{ij}, $1 \le i \le m$, $1 \le j \le n$ jsou vzájemně disjunktní abecedy nonterminálů, T je abeceda terminálů, S je startovací symbol, $S \notin (N_{11} \cup ... \cup N_{mn} \cup T)$ a P je konečná množina maticových pravidel. Maticové pravidla mohou být následující formy:

1. $[S \to X_{11}...X_{mn}]$, $X_{ij} \in N_{ij}$, $1 \le i \le m$, $1 \le j \le n$,
2. $[X_{i1} \to a_{i1},...,X_{in} \to a_{in}]$, $X_{ij} \in N_{ij}$, $a_{ij} \in T^*$, $1 \le j \le n$, pro nějaké *i*, $1 \le i \le m$,
3. $[X_{i1} \to a_{i1}Y_{i1},...,X_{in} \to a_{in}Y_{in}]$, $X_{ij}, Y_{ij} \in N_{ij}$, $a_{ij} \in T^*$, $1 \le j \le n$, pro nějaké *i*, $1 \le i \le m$.

Třída jazyků generovanou *m-Pn-G* značíme $R^m_{[n]}$. Pokud $m = 1$ dostaneme tzv. *n*-pravou lineární maticovou gramatiku (*n-RLSMG*), třídu jazyků generovanou touto gramatikou značíme $R_{[n]}$. Pokud $n = 1$ potom dostaneme *m*-paralelní pravou lineární gramatiku (*m-PRLG*), třídu jazyků pak značíme R_m.

Definice 1.6. Pro $m,n > 1$, nechť $G = (N_{11},...,N_{1n},...,N_{m1},...,N_{mn},T,S,P)$ je *m-Pn-G*, potom G_{ij}, $1 \le i \le m$, $1 \le j \le n$ tedy (i, j)-tá slot gramatika gramatiky *G*, je čtveřice:

$$G_{ij} = (N_{ij} \cup \{S\}, T, S, P_{ij}), \text{ kde}$$

$P_{ij} = \{S \to X_{ij} : [S \to X_{11}...X_{ij}...X_{mn}] \in P\} \cup \{X_{ij} \to x_{ij} : [X_{i1} \to x_{i1},...,X_{in} \to x_{in}] \in P\}$. $L(G_{ij})$ je (i, j)-tý slot jazyk gramatiky *G*.

1.4 Automatové systémy

Na poli automatových systémů již bylo také dosaženo jistých výsledků, zmíníme jak zástupce CD automatových systémů tak PC automatových systémů a to z pohledu, kdy komponentami systémů jsou konečné automaty.

Definice 1.7. CD automatový systém stupně *n*, $n \ge 1$ je pětice

$$M = (Q, V, \Delta, q_0, F),$$

kde Q je n-tice $(Q_1, Q_2, ..., Q_n)$ kde každé Q_i je konečná množina stavů i-té komponenty, V je vstupní abeceda, Δ je n-tice $(\delta_1, \delta_2, ..., \delta_n)$ přechodových funkcí, kde každá $\delta_i : Q_i \times (V \cup \{\varepsilon\}) \to 2^{\cup_i Q_i}$, $1 \le i \le n$, $q_0 \in \cup_i Q_i$ je počáteční stav, $F \subseteq \cup_i Q_i$ je množina koncových stavů.

Z [4] se dočteme, že síla těchto systémů, je totožná se sílou konečných automatů a nezáleží na módu, při němž systém předává řízení mezi komponentami.

Pozornosti hodnými články jsou [5] a [6], které také představují paralelní automatové systémy založené na bázi konečných automatů, ovšem ke komunikaci je v tomto případě přistupováno jinou metodou. V dalším textu porovnáme námi navržené automatové systémy i s těmito systémy.

Definice 1.8. PC automatový systém komunikující stavy je (n+2)-tice

$$A = (V, A_1, A_2, ..., A_n, K),$$

kde V je vstupní abeceda, $A_i = (Q_i, V, f_i, q_i, F_i), 1 \le i \le n$ jsou konečné automaty, kde Q_i je množina stavů, $q_i \in Q_i$ je počáteční stav, $F_i \subseteq Q_i$ je množina koncových stavů a f_i je přechodová funkce definovaná $f_i : Q_i \times V \cup \{\varepsilon\} \to 2^{Q_i}$, $K \subseteq \{K_1, K_2, ..., K_n\} \subseteq \bigcup_{i=1}^{n} Q_i$ je množina komunikačních stavů.

Konfigurace tohoto systému je 2n-tice $(s_1, x_1, s_2, x_2, ..., s_n x_n)$, kde pro každé $1 \le i \le n$, s_i je aktuální stav i-té komponenty a x_i je zbývající část vstupního slova, jež nebyla dosud přečtena i-tou komponentou. Více o těchto systémech viz. [5], [6], zde připomeneme pouze některé vlastnosti.

Existuje několik druhů těchto systémů, centralizované, u kterých je komunikace schopna pouze jedna komponenta tzv. master, necentralizované, u nichž mohou komunikovat všechny komponenty, dále systémy pracující v módu s návratem, což znamená, že po komunikaci se jisté komponenty vrátí do počátečního stavu a systémy v módu bez návratu. Celkem existují tedy čtyři možné varianty těchto systémů:

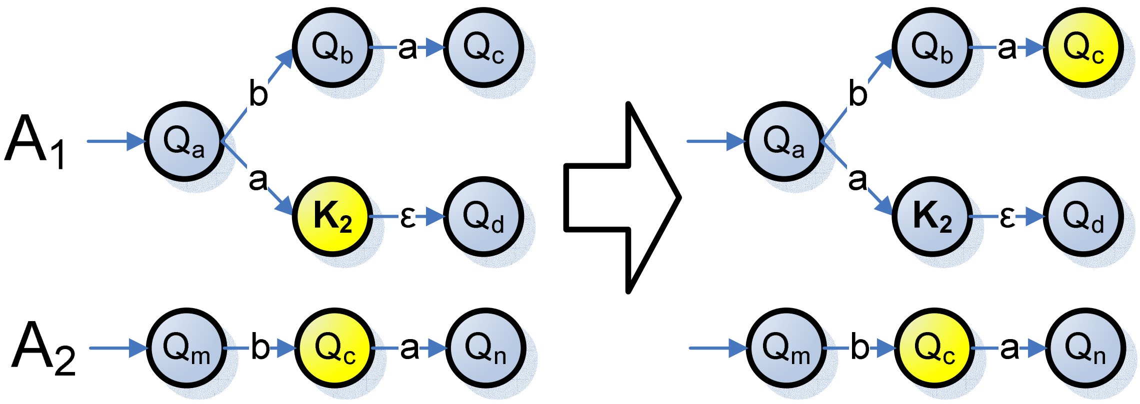
1. RCPCFA(n) – centralizované vracející se automatové systémy
2. RPCFA(n) – vracející se automatové systémy
3. CPCFA(n) – centralizované automatové systémy
4. PCFA(n) – automatové systémy.

Pro tyto systémy platí následující věta (viz. [6]):

Věta 1.1. Nechť $X(n)$ je třída jazyků akceptována automatovými systémy typu X, potom platí:

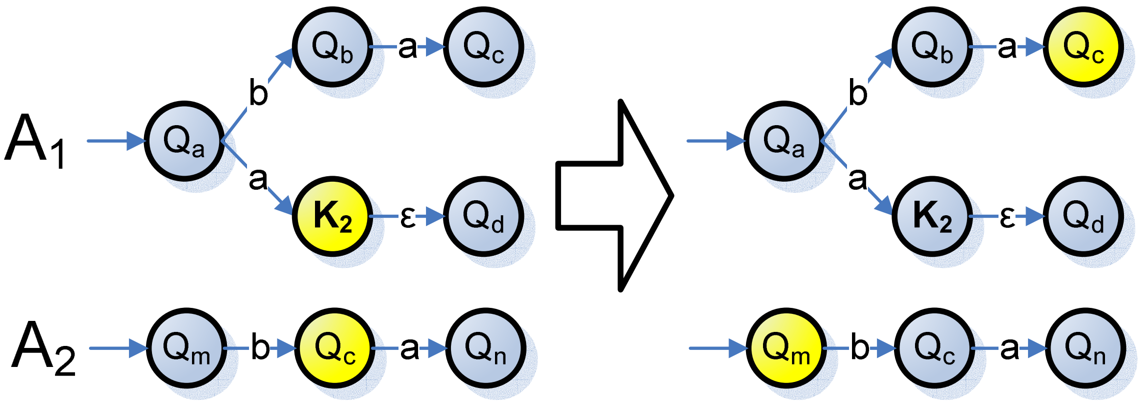
1. $X(1) = REG$, $X \in \{RCPCFA, RPCFA, CPCFA, PCFA\}$
2. $X(n) \subseteq X(n+1)$ pro všechna $X \in \{RCPCFA, RPCFA, CPCFA, PCFA\}$.

Následující dva obrázky, převzaté z [3], ukazují, jak tyto sytémy pracují při komunikaci. Na obrázku 1.1 je zobrazen nejprve systém pracující v módu bez návratu. Vidíme, jak systém provádí přechod z jedné konfigurace do jiné. Za zmínku stojí fakt, že byť se jedná o dvě komponenty, mají některé stavy pojmenovány stejně a to kvůli komunikaci.



Obrázek 1.1

Na dalším obrázku je zobrazen automatový systém pracující v módu s návratem. Tedy po provedení komunikace je dotazovaná komponenta uvedena do počátečního stavu.



Obrázek 1.2

Definice 1.9. Jazyk příjímaný automatovým systémem, který komunikuje stavy, je definován:

$$L(AS)=\{x\in V^* \mid (q_1,x,q_2,x,\ldots,q_n,x)\Rightarrow^* (s_1,\varepsilon,s_2,\varepsilon,\ldots,s_n,\varepsilon), s_i\in F_i, 1\le i\le n\}.$$

2 Definice

V této části nadefinujeme nejprve paralelní automatové systémy (*P AS*) a posléze nadefinujeme paralelně komunikující automatové systémy, které komunikují přechody (*PC AS*). Pro oba systémy uvažujeme, že komponenty jsou konečné automaty. Jak uvidíme dále, mají tyto systémy některé společné vlastnosti s výše uvedenými strukturami.

2.1 Paralelní automatové systémy

Definice 2.1. Paralelní automatový systém, jehož komponenty jsou konečné automaty (*P AS FSA*) je (n+1)-tice

$$AS = (\Sigma, A_1, ..., A_n),$$

kde Σ je vstupní abeceda, $A_i = (Q_i, \Sigma_i, R_i, s_i, F_i)$, $1 \le i \le n$ jsou konečné automaty (komponenty systému), kde Q_i je konečná množina stavů dané komponenty, $\Sigma_i = \Sigma$ je vstupní abeceda, $R_i = \{pa \to q \mid p, q \in Q_i, a \in \Sigma_i \cup \{\varepsilon\}\}$ je přechodová funkce, $s_i \in Q_i$ je počáteční stav, $F_i \subseteq Q_i$ je množina konečných (akceptujících) stavů a platí $\bigcap_{j=1}^{n} Q_j = \phi$.

Definice 2.2. Nechť $AS = (\Sigma, A_1, ..., A_n)$ je *P AS FSA*(n). Konfigurace *P AS FSA*(n) *AS* je řetězec

$$\chi = q_1 v_1 q_2 v_2 ... q_n v_n,$$

kde $q_i \in Q_i, v_i \in \Sigma^*, i \in \{1, ..., n\}$.

Definice 2.3. Nechť $q_1 v_1 q_2 v_2 ... q_n v_n$ a $q_{1'} v_{1'} q_{2'} v_{2'} ... q_{n'} v_{n'}$ jsou dvě konfigurace *P AS FSA*(n) *AS*, kde $q_i, q_{i'} \in Q_i, a_i \in \Sigma \cup \{\varepsilon\}, v_i = a_i v_{i'}, v_i, v_{i'} \in \Sigma^*$ pro všechna $i, 1 \le i \le n$. Nechť pro každé $i, i \in \{1, ..., n\}$ platí $\exists r_i = (q_i a_i \to q_{i'}) \in R_i$. Potom *AS* může provést přechod z konfigurace $q_1 v_1 q_2 v_2 ... q_n v_n$ do $q_{1'} v_{1'} q_{2'} v_{2'} ... q_{n'} v_{n'}$ za použití $r_1, r_2, ..., r_n$, zapsáno $q_1 v_1 q_2 v_2 ... q_n v_n \Rightarrow q_{1'} v_{1'} q_{2'} v_{2'} ... q_{n'} v_{n'} [r_1, r_2, ..., r_n]$ nebo zjednodušeně $q_1 v_1 q_2 v_2 ... q_n v_n \Rightarrow q_{1'} v_{1'} q_{2'} v_{2'} ... q_{n'} v_{n'}$.

Definice 2.4. Nechť $AS = (\Sigma, A_1, ..., A_n)$ je *P AS FSA*(n). Jazyk přijímaný paralelním automatovým systémem *AS*, *L*(*AS*), je definován:

$$L(AS) = \{w \mid s_1 v_1 s_2 v_2 ... s_n v_n \Rightarrow^* f_1 f_2 ... f_n, w \in \Sigma^*, v_1 v_2 ... v_n = w, f_i \in F_i, i \in \{1, ..., n\}\}.$$

Označme A_n třídu jazyků generovanou *P AS FSA*(n).

Příklad 2.1. Nechť $AS = (\{a, b, c\}, A_1, A_2, A_3)$ je *P AS FSA*, kde

$$A_1 = (\{Q_1, Q_{f1}\}, \{a, b, c\}, R_1, Q_1, \{Q_{f1}\}), \; R_1 = \{Q_1 a \to Q_1, Q_1 \varepsilon \to Q_{f1}\},$$

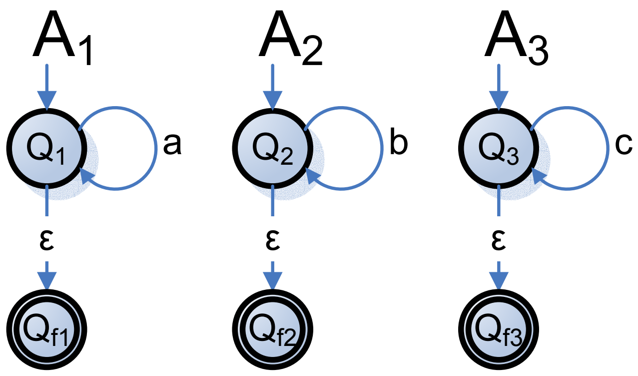
$$A_2 = (\{Q_2, Q_{f2}\}, \{a,b,c\}, R_2, Q_2, \{Q_{f2}\}),\ R_2 = \{Q_2 b \to Q_2, Q_2\varepsilon \to Q_{f2}\},$$

$$A_3 = (\{Q_3, Q_{f3}\}, \{a,b,c\}, R_3, Q_3, \{Q_{f3}\}),\ R_3 = \{Q_3 c \to Q_3, Q_3\varepsilon \to Q_{f3}\}.$$

AS má tedy tři komponenty a jazyk generovaný tímto systémem je $L(AS) = \{a^n b^n c^n \mid n \geq 0\}$. Tento systém příjme slovo *aaabbbccc* následovně:

$$Q_1 aaa Q_2 bbb Q_3 ccc \Rightarrow Q_1 aa Q_2 bb Q_3 cc \Rightarrow Q_1 a Q_2 b Q_3 c \Rightarrow Q_1 Q_2 Q_3 \Rightarrow Q_{f1} Q_{f2} Q_{f3}.$$

Následuje grafické znázornění systému:



Obrázek 2.1

2.2 Paralelně komunikující automatové systémy

Definice 2.5. Paralelně komunikující automatový systém komunikující přechody, jehož komponenty jsou konečné automaty (*PC AS FSA*) je (*n*+1)-tice:

$$AS = (\Sigma, A_1, \ldots, A_n),$$

kde Σ je vstupní abeceda, $A_i = (Q_i, \Sigma_i, R_i, s_i, F_i)$, $1 \leq i \leq n$ jsou konečné automaty (komponenty systému), kde Q_i je konečná množina stavů dané komponenty, $\Sigma_i = \Sigma$ je vstupní abeceda, $R_i = \{pa \to q \mid p, q \in Q_i, a \in \Sigma_i \cup \{\varepsilon\}\} \cup \left\{ p \xrightarrow{c} q \mid p, q \in Q_i, c \in \bigcup_{j=1}^{n} Q_j \right\}$ je přechodová funkce obsahující „normální“ pravidla typu $pa \to q$ a „komunikační“ pravidla typu $p \xrightarrow{c} q$, $s_i \in Q_i$ je počáteční stav, $F_i \subseteq Q_i$ je množina konečných (akceptujících) stavů a platí $\bigcap_{j=1}^{n} Q_j = \phi$.

Definice 2.6. Nechť $AS = (\Sigma, A_1, \ldots, A_n)$ je *PC AS FSA*(*n*). Konfigurace *PC AS FSA*(*n*) *AS* je řetězec

$$\chi = q_1 v_1 q_2 v_2 \ldots q_n v_n,$$

kde $q_i \in Q_i, v_i \in \Sigma^*, i \in \{1, \ldots, n\}$.

Definice 2.7. Nechť $AS = (\Sigma, A_1, \dots, A_n)$ je *PC AS FSA*(n). Množina komunikačních stavů i-té komponenty, tedy stavů, ze kterých je možno provést komunikační přechod, je množina

$$C_i = \left\{ q \;\middle|\; q \in Q_i \wedge \exists (q \xrightarrow{c} r) \in R_i, c \in \bigcup_{j=1}^{n} Q_j, r \in Q_i \right\}.$$

Definice 2.8. Nechť $AS = (\Sigma, A_1, \dots, A_n)$ je *PC AS FSA*(n). Množina dotazovaných stavů i-té komponenty je množina

$$D_i = \left\{ c \mid c \in Q_i \wedge \exists j \mid (q \xrightarrow{c} r) \in R_j, q, r \in Q_j, j \in \{1, \dots, n\} \right\}.$$

Jsou to takové stavy, na které se může některá komponenta teoreticky dotázat, tj. existuje u jiné komponenty komunikační přechod, při němž je dotazováno na tento stav.

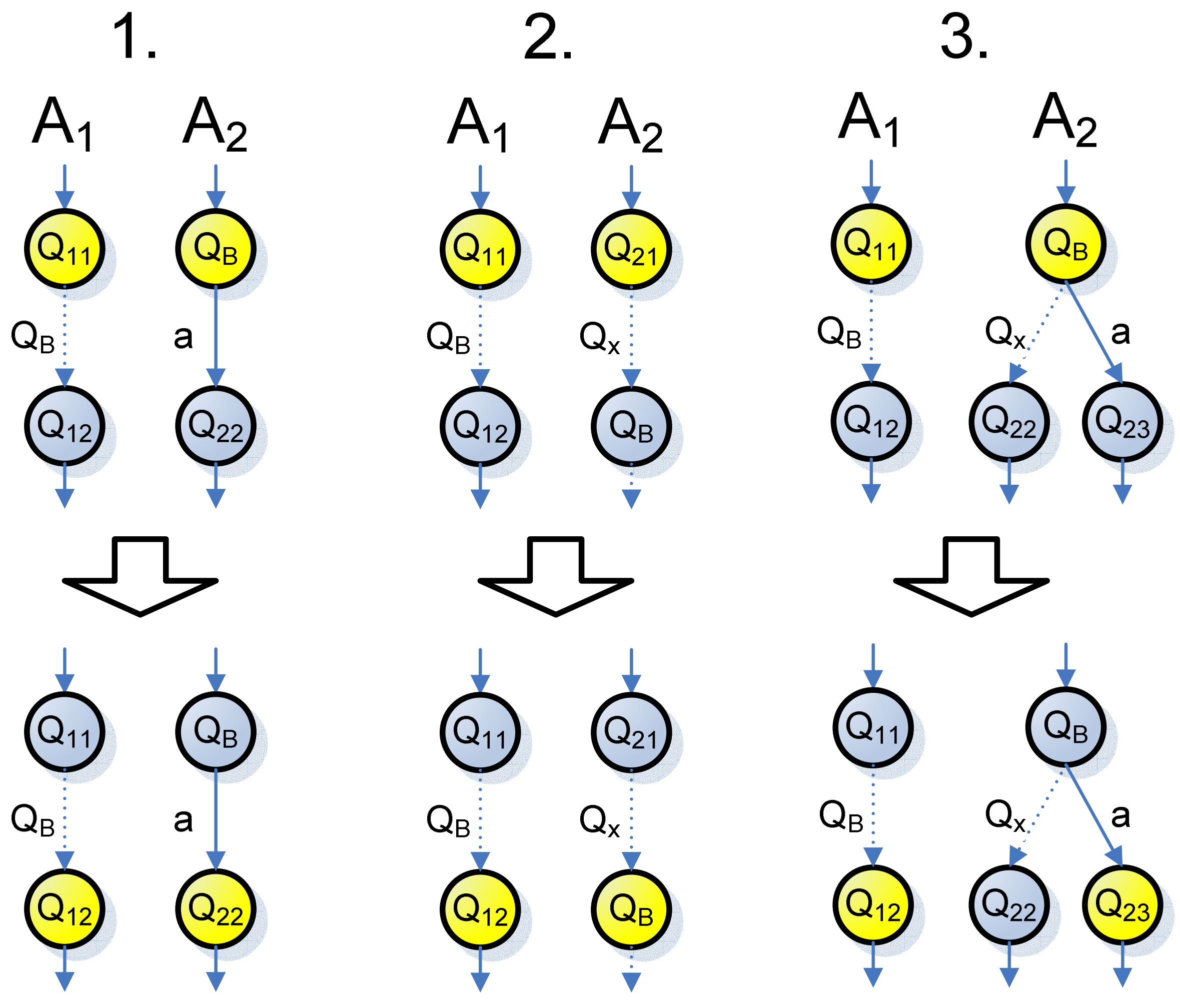
Definice 2.9. Nechť $q_1 v_1 q_2 v_2 \dots q_n v_n$ a $q_{1'} v_{1'} q_{2'} v_{2'} \dots q_{n'} v_{n'}$ jsou dvě konfigurace *PC AS FSA*(n) *AS*, kde $q_i, q_{i'} \in Q_i, a_i \in \Sigma \cup \{\varepsilon\}, v_i = a_i v_{i'}, v_i, v_{i'} \in \Sigma^*$ pro všechna $i, 1 \le i \le n$. Potom *AS* může provést přechod z konfigurace $q_1 v_1 q_2 v_2 \dots q_n v_n$ do $q_{1'} v_{1'} q_{2'} v_{2'} \dots q_{n'} v_{n'}$ za použití přechodů $r_1, r_2, \dots, r_n$, zapsáno $q_1 v_1 q_2 v_2 \dots q_n v_n \Rightarrow q_{1'} v_{1'} q_{2'} v_{2'} \dots q_{n'} v_{n'} [r_1, r_2, \dots, r_n]$ nebo zjednodušeně $q_1 v_1 q_2 v_2 \dots q_n v_n \Rightarrow q_{1'} v_{1'} q_{2'} v_{2'} \dots q_{n'} v_{n'}$, pokud platí jeden z následujích případů:

1. Pro každé i, $1 \le i \le n$ platí, že $q_i \notin C_i$ a $\exists r_i = (q_i a_i \to q_{i'}) \in R_i$.
2. Pro všechna j, $1 \le j \le n$, pro které $q_j \notin C_j$ existuje $r_j = (q_j a_j \to q_{j'}) \in R_j$ a pro každé i, $1 \le i \le n$,pro které platí $q_i \in C_i$ platí buď:

   1. $r_i = \left(q_i \xrightarrow{c} q_{i'} \right) \in R_i$ a existuje k, $1 \le k \le n$, pro které platí $c = q_k$ a $q_k \notin C_k$ a $r_k = (q_k a_k \to q_{k'}) \in R_k$ potom $a_i = \varepsilon$ nebo
   2. $r_i = \left(q_i \xrightarrow{c} q_{i'} \right) \in R_i$ a existuje k, $1 \le k \le n$, pro které platí $c = q_{k'}$ a $r_k = \left(q_k \xrightarrow{q_m} q_{k'} \right) \in R_k$, pro nějaké $m \in \{1, \dots, n\}$ potom $a_i = \varepsilon$ nebo
   3. $r_i = \left(q_i \xrightarrow{c} q_{i'} \right) \in R_i$ a existuje k, $1 \le k \le n$, pro které platí $c = q_k$ a $r_k = (q_k a_k \to q_{k'}) \in R_k$ potom $a_i = \varepsilon$.
   4. Pokud není splněná žádná z uvedených možností, musí platit, že $r_i = (q_i a_i \to q_{i'}) \in R_i$.

V prvním případě žádná z komponent není schopna provést komunikační přechod. Proto aby systém mohl provést přechod z jedné konfigurace do jiné, musí existovat příslušná pravidla v R_i u jednotlivých komponent.

Druhý případ je složitější neb existují komponenty, které jsou ve stavech, v nichž jsou schopné komunikovat, tj. dotazovat se na stavy jiných komponent. Z definice plyne, že nejprve se upřednostní komunikace a teprve po komunikaci provedou přechody i zbylé komponenty. Přitom je rozlišeno několik případů, které mohou při komunikaci nastat. Pokud se komponenta dotazuje na stav komponenty, který nepatří do množiny komunikačních stavů dané komponenty, provede se komunikační přechod v tom případě, byla-li dotazovaná komponenta v dotazovaném stavu před přechodem do nového stavu viz. první případ v následujícím obrázku. Další možností je, že se komponenta dotazuje na stav komponenty, která ovšem taky provádí komunikační přechod, v tom případě musí nejprve počkat, nežli dotazovaná komponenta komunikační přechod provede viz. druhý případ v následujícím obrázku. Pokud se komponenta dotazuje na stav, který patří do komunikačních stavů dotazované komponenty, která ovšem komunikaci provést nemohla, provede se komunikační přechod, pokud dotazovaná komponenta byla v dotazovaném stavu, který seč byl komunikační, přesto dotazovaná komponenta provedla přechod normální viz. třetí případ v obrázku. Z tohoto faktu plyne další možnost, jak se může celý systém zaseknout, tedy případ, kdy komponenty v kruhu čekají než následující komponenta provede komunikaci.



Obrázek 2.2

Komunikační přechody jsou v grafické podobě znázorněny přerušovanou čarou.

Definice 2.10. Nechť $AS = (\Sigma, A_1, ..., A_n)$ je *PC AS FSA*(n). Jazyk přijímaný paralelně komunikujícím automatovým systémem *AS*, *L*(*AS*), je definován:

$$L(AS) = \{w | s_1v_1s_2v_2...s_nv_n \Rightarrow^* f_1q_2...q_n, w \in \Sigma^*, v_1v_2...v_n = w, f_1 \in F_1, q_i \in Q_i, i \in \{2,...,n\}\}.$$

Označme $A_{[n]}$ třídu jazyků generovanou *PC AS FSA*(n).

Systém začíná v počáteční konfiguraci, postupně čte slovo a komunikuje. Dané slovo přijme, pokud první komponenta tzv. „master“ přejde do koncového stavu a celé slovo je přečteno. Na rozdíl od paralelních systémů komunikující stavy (viz. [5], [6]), kde každá komponenta operuje nad celým vstupním slovem systému, v těchto námi definovaných systémech, má každá komponenta k dispozici pouze jistou část vstupního slova. Za povšimnutí stojí také fakt, že nezáleží v jakém stavu skončí ostatní komponenty, zdali také v koncovém stavu či nikoliv.

Doposud jsme uvažovali paralelně komunikující systémy, které neměli žádná omezení týkající se komunikace. Přesněji řečeno, každá komponenta systému mohla obsahovat komunikační přechody kterými se dotazovala, zdali některá komponenta je v jistém stavu. V souvislosti s tím zaveďme tyto druhy automatových systémů.

Definice 2.11. Nechť $AS = (\Sigma, A_1, ..., A_n)$ je *PC AS FSA*(n). Pokud pouze jedna komponenta, tedy komponenta A_1, obsahuje komunikačni pravidla čili pravidla typu $p \xrightarrow{c} q$, potom říkáme, že *AS* je centralizovaný *PC AS FSA*(n), což značíme *CPC AS FSA*(n), jinak *AS* je necentralizovaný *PC AS FSA*(n).

Příklad 2.2. Nechť $AS = (\{a, b, c\}, A_1, A_2, A_3)$ je *PC AS FSA*, kde

$$A_1 = (\{Q_{11}, Q_{12}, Q_{13}, Q_{f1}\}, \{a, b, c\}, R_1, Q_{11}, \{Q_{f1}\}),\ R_1 = \Big\{Q_{11}a \rightarrow Q_{11}, Q_{11}\varepsilon \rightarrow Q_{12},$$

$$Q_{12} \xrightarrow{Q_B} Q_{13}, Q_{13} \xrightarrow{Q_C} Q_{f1}\Big\},$$

$$A_2 = (\{Q_{21}, Q_B, Q_{22}, Q_{23}\}, \{a, b, c\}, R_2, Q_{21}, \phi),\ R_2 = \{Q_{21}b \rightarrow Q_{21}, Q_{21}\varepsilon \rightarrow Q_B,$$

$$Q_B\varepsilon \rightarrow Q_{22}, Q_{22}\varepsilon \rightarrow Q_{23}\},$$

$$A_3 = (\{Q_{31}, Q_{32}, Q_C, Q_{33}\}, \{a, b, c\}, R_3, Q_{31}, \phi),\ R_3 = \{Q_{31}c \rightarrow Q_{31}, Q_{31}\varepsilon \rightarrow Q_{32},$$

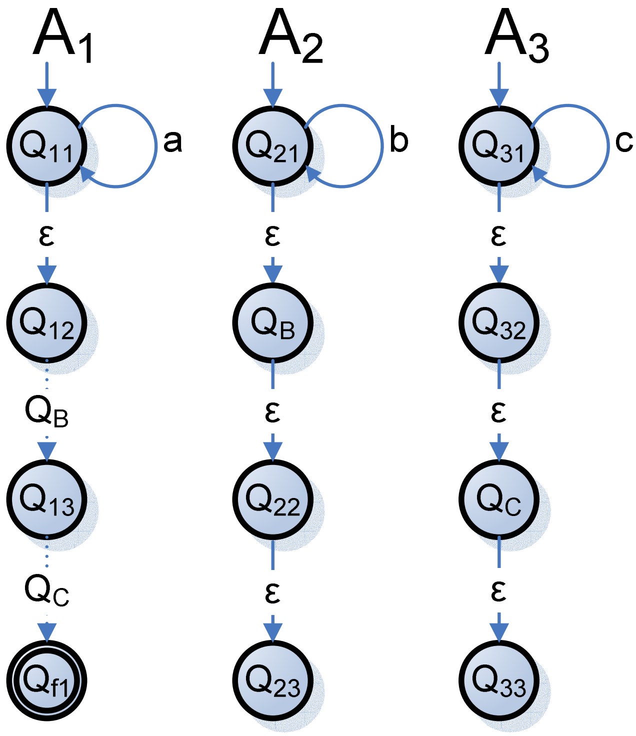
$$Q_{32}\varepsilon \rightarrow Q_C, Q_C\varepsilon \rightarrow Q_{33}\}.$$

AS má tedy tři komponenty, je centralizovaný, první komponenta je tzv. master a jazyk generovaný tímto systémem je $L(AS) = \{a^n b^n c^n \mid n \geq 0\}$.Tento systém příjme slovo *aaabbbccc* následovně:

$$Q_{11}aaaQ_{21}bbbQ_{31}ccc \Rightarrow Q_{11}aaQ_{21}bbQ_{31}cc \Rightarrow Q_{11}aQ_{21}bQ_{31}c \Rightarrow .$$

$$\Rightarrow Q_{11}Q_{21}Q_{31} \Rightarrow Q_{12}Q_BQ_{32} \Rightarrow Q_{13}Q_{22}Q_C \Rightarrow Q_{f1}Q_{23}Q_{33}$$

Následuje grafické znázornění systému:



Obrázek 2.3

Příklad 2.3. Nechť $AS = (\{a,b,c\}, A_1, A_2)$ je *PC AS FSA*, kde

$$A_1 = (\{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7\}, \{a,b,c\}, R_1, Q_1, \{Q_5\}),$$

$$R_1 = \{Q_1\varepsilon \to Q_2, Q_2\varepsilon \to Q_3, Q_3\varepsilon \to Q_5, Q_5c \to Q_6, Q_6c \to Q_7, Q_4b \to Q_1,$$

$$Q_3 \xrightarrow{A} Q_4, Q_7 \xrightarrow{B} Q_5\},$$

$$A_2 = (\{P_1, P_2, P_3, P_4, P_5, P_6, A, B\}, \{a,b,c\}, R_2, P_1, \phi),$$

$$R_2 = \{P_1a \to P_2, P_2a \to A, A\varepsilon \to P_3, P_3\varepsilon \to P_1, P_1b \to P_4, P_4\varepsilon \to B, B\varepsilon \to P_5,$$

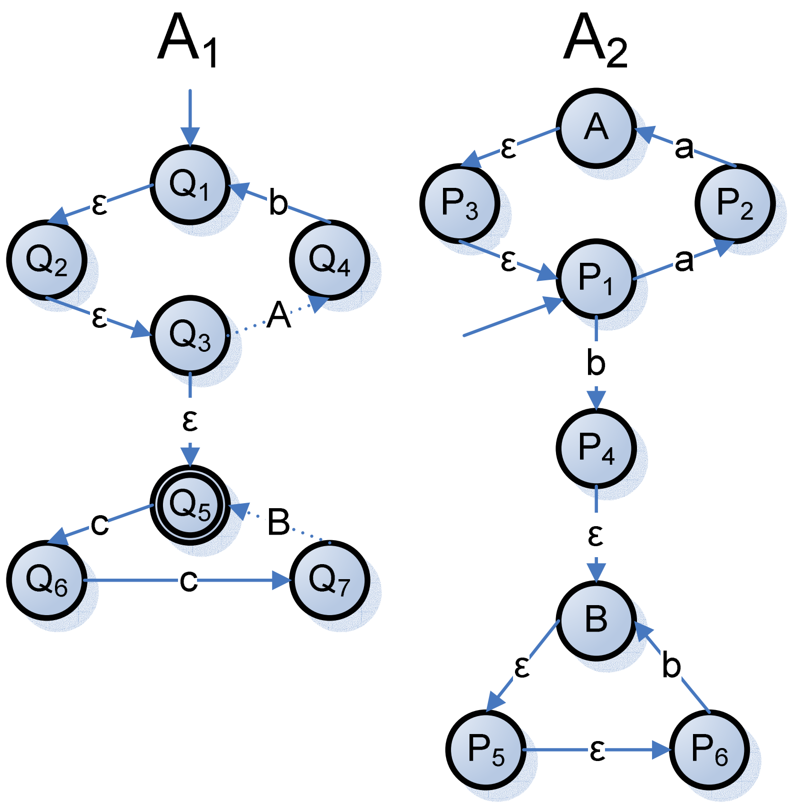
$$P_5\varepsilon \to P_6, P_6b \to B\}.$$

AS má dvě komponenty, opět se jedná o centralizovaný systém, první komponenta je tzv. master a jazyk generovaný tímto systémem je $L(AS) = \{b^n c^{2m} a^{2n} b^{m+1} \mid m, n \geq 0\}$. Tento systém příjme slovo *bccaabb* následovně:

$$Q_1bccP_1aabb \Rightarrow Q_2bccP_2abb \Rightarrow Q_3bccAbb \Rightarrow Q_4bccP_3bb \Rightarrow Q_1ccP_1bb\,.$$

$$\Rightarrow Q_2ccP_4b \Rightarrow Q_3ccBb \Rightarrow Q_5ccP_5b \Rightarrow Q_6cP_6b \Rightarrow Q_7B \Rightarrow Q_5P_5$$

Následuje grafické znázornění systému:



Obrázek 2.4

2.3 Vlastnosti automatových systémů

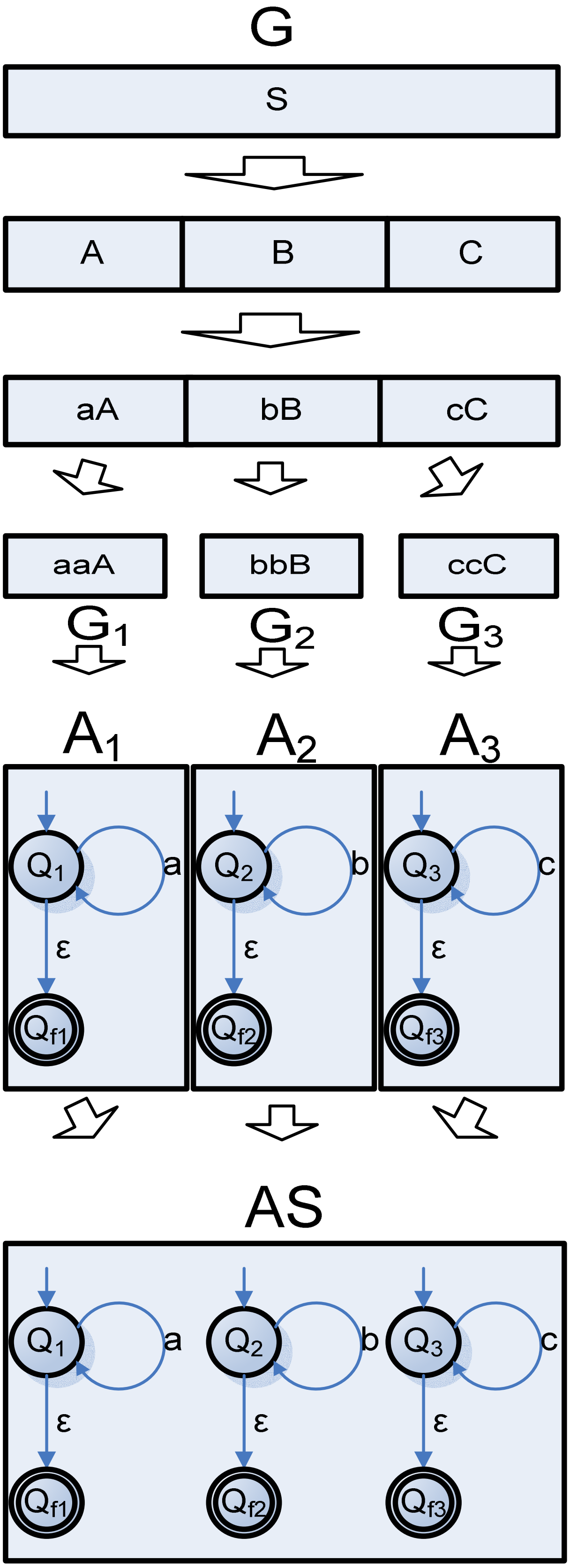
Věta 2.1. Nechť $X(n)$ je třída jazyků akceptována automatovými systémy typu X mající n komponent, potom platí: $X(1) = REG$, $X \in \{P\ AS\ FSA, PC\ AS\ FSA, CPC\ AS\ FSA\}$.

Důkaz. Je zřejmé, že jakýkoliv takový automatový systém mající pouze jednu komponentu, která je rovna konečnému automatu má minimálně stejnou sílu jako konečný automat a větší sílu ani mít nemůže, protože systém neobsahuje nic navíc, než to co obsahuje konečný automat. Komunikační přechody nemají smysl, jedná-li se pouze o jednu komponentu.

Věta 2.2. Pro každou *n-PRLG* existuje ekvivalentní *P AS FSA*(*n*).

Důkaz. Nechť *G* je *n-PRLG*, tj. $G = (N_1,...,N_n,T,S,P)$ a nechť pojem „slot“ gramatika má obvyklý význam (více viz.kapitola 1.3 nebo [8]). Potom je zřejmé, že *G* je „složena“ z *n* slot gramatik. Pokud každou tuto slot gramatiku převedeme pomocí známého algoritmu na převod pravé lineární gramatiky na konečný automat, dostáváme *n* konečných automatů, který každý zvlášť přijímá stejný jazyk, jako generuje jemu příslušná slot gramatika. Další důležitou skutečností je i to, že zůstává zachována jakási „synchronizace“ všech automatů a to na základě toho, jak je algoritmus převodu vystavěn, přesněji řečeno, každé přepisovací pravidlo, je v automatu reprezentováno hranou či koncovým stavem. Proto, pokud vytvoříme následující *P AS FSA*(*n*), označme jej *AS*, tedy

$AS = (\Sigma, A_1, ..., A_n)$, kde $\Sigma = T$ a A_i, $1 \leq i \leq n$ jsou konečné automaty převedené ze slot gramatik, dostáváme ekvivalentní *P AS FSA*(*n*).



Obrázek 2.5

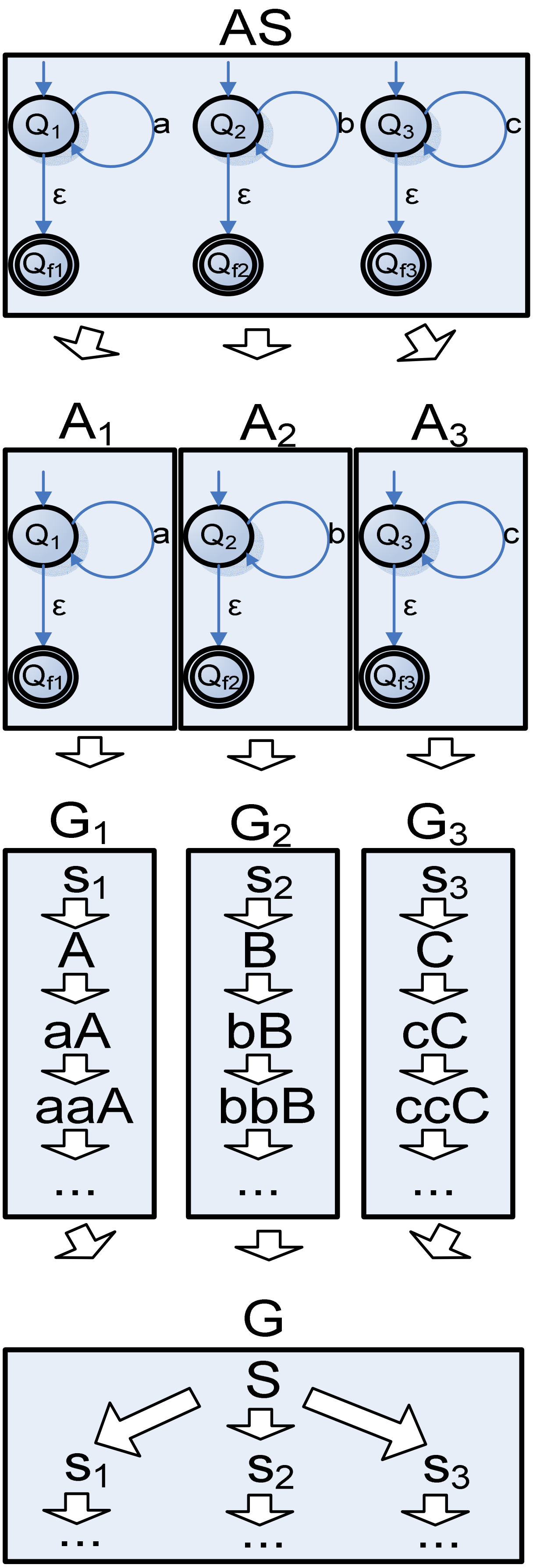
Na obrázku 2.5 vidíme, jak je G složena z jednotlivých slot gramatik G_1, G_2, G_3, které pomocí algoritmu na převedení pravé linearní gramatiky převedeme na konečné automaty A_1, A_2, A_3, tvořící dohromady paralelní automatový systém *AS*.

Věta 2.3. Pro každý *P AS FSA*(*n*) existuje ekvivalentní *n-PRLG*.

Důkaz: Nechť $AS = (\Sigma, A_1, ..., A_n)$ je *P AS FSA*(*n*), kde $A_i = (Q_i, \Sigma_i, R_i, s_i, F_i)$, $1 \leq i \leq n$ jsou konečné automaty. Převedeme-li jednotlivé komponenty *AS* pomocí algoritmu na převod konečného automatu na pravou lineární gramatiku dostáváme gramatiky ve tvaru $G_i = (N_i, T, s_i, P_i)$. Nyní nechť G je *n-PRLG*, tj. $G = (N_1, ..., N_n, T, S, P)$, kde N_i je abeceda nonterminálů dané gramatiky, $T = \Sigma$, S je počáteční symbol, $S \notin (N_1 \cup N_2 \cup ... \cup N_n)$, P je množina pravidel obsahující následující pravidla:

1. $S \rightarrow s_1 s_2 ... s_n$,
2. pro každé pravidlo z $\bigcup_{i=1}^{n} P_i$ ve tvaru $X_i \rightarrow \varepsilon$ nebo ve tvaru $X_i \rightarrow aY_i$, kde $X_i, Y_i \in N_i$, $a \in T \cup \{\varepsilon\}$, $i \in \{1, ..., n\}$ existuje v P stejné pravidlo,

potom G je ekvivalentní *AS*.



Obrázek 2.6

Věta 2.4. Pro každý *P AS FSA*(n) existuje ekvivalentní *CPC AS FSA*(n).

Důkaz: Mějme tedy *P AS FSA*(n), označme jej *AS*, potom $AS = (\Sigma, A_1, ..., A_n)$, kde $A_i = (Q_i, \Sigma_i, R_i, s_i, F_i)$, $1 \le i \le n$ jsou konečné automaty. K tomuto automatovému systému existuje ekvivalentní *CPC AS FSA*(n), označme jej *AS‘*, tedy $AS' = (\Sigma, A'_1, ..., A'_n)$, kde $A'_1 = (Q'_1, \Sigma'_1, R'_1, s'_1, F'_1)$ je tzv. „master“ tedy jediná komponenta systému, která je schopna komunikovat a pro kterou platí:

$$Q'_1 = Q_1 \cup \{q'_{1_1}, q'_{1_2}, ..., q'_{1_n}\}, \text{ kde } q'_{1_i} \notin \bigcup_{j=1}^{n} Q_j \text{ pro } \forall i, i \in \{1, ..., n\},$$

$$\Sigma'_1 = \Sigma_1,$$

$$R'_1 = R_1 \cup \{fa \to q \mid f \in F_1, a = \varepsilon, q = q'_{1_1}\} \cup \left\{ q'_{1_1} \xrightarrow{x_2} q'_{1_2}, ..., q'_{1_{n-1}} \xrightarrow{x_n} q'_{1_n} \right\},$$

kde $x_i \in F'_i$ pro $2 \le i \le n$,

$$s'_1 = s_1,$$

$$F'_1 = \{q'_{1n}\},$$

$A'_j = (Q'_j, \Sigma'_j, R'_j, s'_j, F'_j)$ pro $j \in \{2, ..., n\}$ jsou ostatní komponenty, pro které platí:

$$Q'_j = Q_j \cup \{q'_{j_1}, ..., q'_{j_j}\}, \text{ kde } q'_{j_i} \notin \bigcup_{k=1}^{n} Q_k \text{ pro } \forall i, i \in \{1, ..., j\},$$

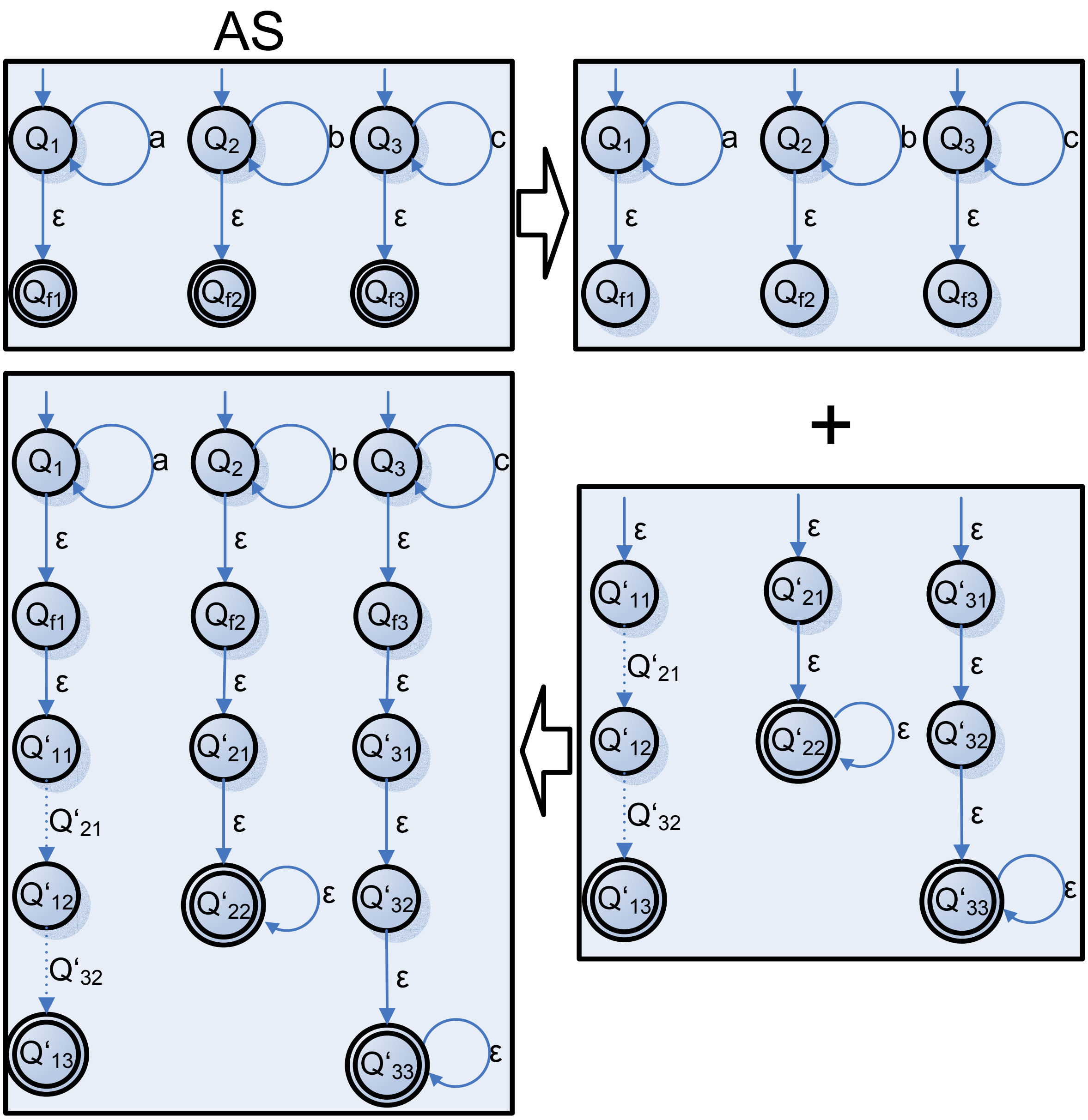
$$\Sigma'_j = \Sigma_j,$$

$$R'_j = R_j \cup \{fa \to q \mid f \in F_j, a = \varepsilon, q = q'_{j_1}\} \cup \left\{ q'_{j_1} \xrightarrow{\varepsilon} q'_{j_2}, ..., q'_{j_{j-1}} \xrightarrow{\varepsilon} q'_{j_j} \right\} \cup \left\{ q'_{j_j} \xrightarrow{\varepsilon} q'_{j_j} \right\},$$

$$s'_j = s_j,$$

$$F'_j = \{q'_{j_j}\}.$$

Jak vidíme z následujícího obrázku, pouze jsme přidali komunikaci na konec jednotlivých komponent, čímž jsme vytvořili ekvivalentní *CPC AS FSA.*



Obrázek 2.7

Věta 2.5. Pro každou *n-RLSMG* existuje ekvivalentní *PC AS FSA*(*n*).

Důkaz: Mějme *n-RLSMG*, označme ji *G*, potom $G = (N_1, ..., N_n, T, S, P)$, kde N_i, $1 \le i \le n$ jsou vzájemně disjunktní abecedy nonterminálů, *T* je abeceda terminálů, *S* je startovací symbol, $S \notin (N_1 \cup ... \cup N_n \cup T)$ a *P* je konečná množina maticových pravidel. Víme, že *P* obsahuje pravidla tvaru

1. $[S \to X_1 ... X_n]$, $X_i \in N_i$, $1 \le i \le n$,
2. $[X_1 \to a_1, ..., X_n \to a_n]$, $X_i \in N_i$, $a_i \in T^*$, $1 \le i \le n$,
3. $[X_1 \to a_1 Y_1, ..., X_n \to a_n Y_n]$, $X_i, Y_i \in N_i$, $a_i \in T^*$, $1 \le i \le n$.

Nyní, nechť P obsahuje k maticových pravidel, potom pro každé i-té pravidlo tvaru $[X_1 \to a_1Y_1, ..., X_n \to a_nY_n]$, pro $1 \le i \le k$, provedeme převod pravidla na následujících 2n-1 pravidel:

$$[X_1 \to a_1 X^i_{1_1}, X_2 \to \varepsilon X^i_{2_1}, X_3 \to \varepsilon X^i_{3_1}, ..., X_n \to \varepsilon X^i_{n_1}],$$

$$[X^i_{1_1} \to \varepsilon X^i_{1_2}, X^i_{2_1} \to X^i_{1_1} X^i_{2_2}, X^i_{3_1} \to X^i_{1_1} X^i_{3_2}, ..., X^i_{n_1} \to X^i_{1_1} X^i_{n_2}],$$

$$[X^i_{1_2} \to \varepsilon X^i_{1_3}, X^i_{2_2} \to a_2 X^i_{2_3}, X^i_{3_2} \to \varepsilon X^i_{3_3}, ..., X^i_{n_2} \to \varepsilon X^i_{n_3}],$$

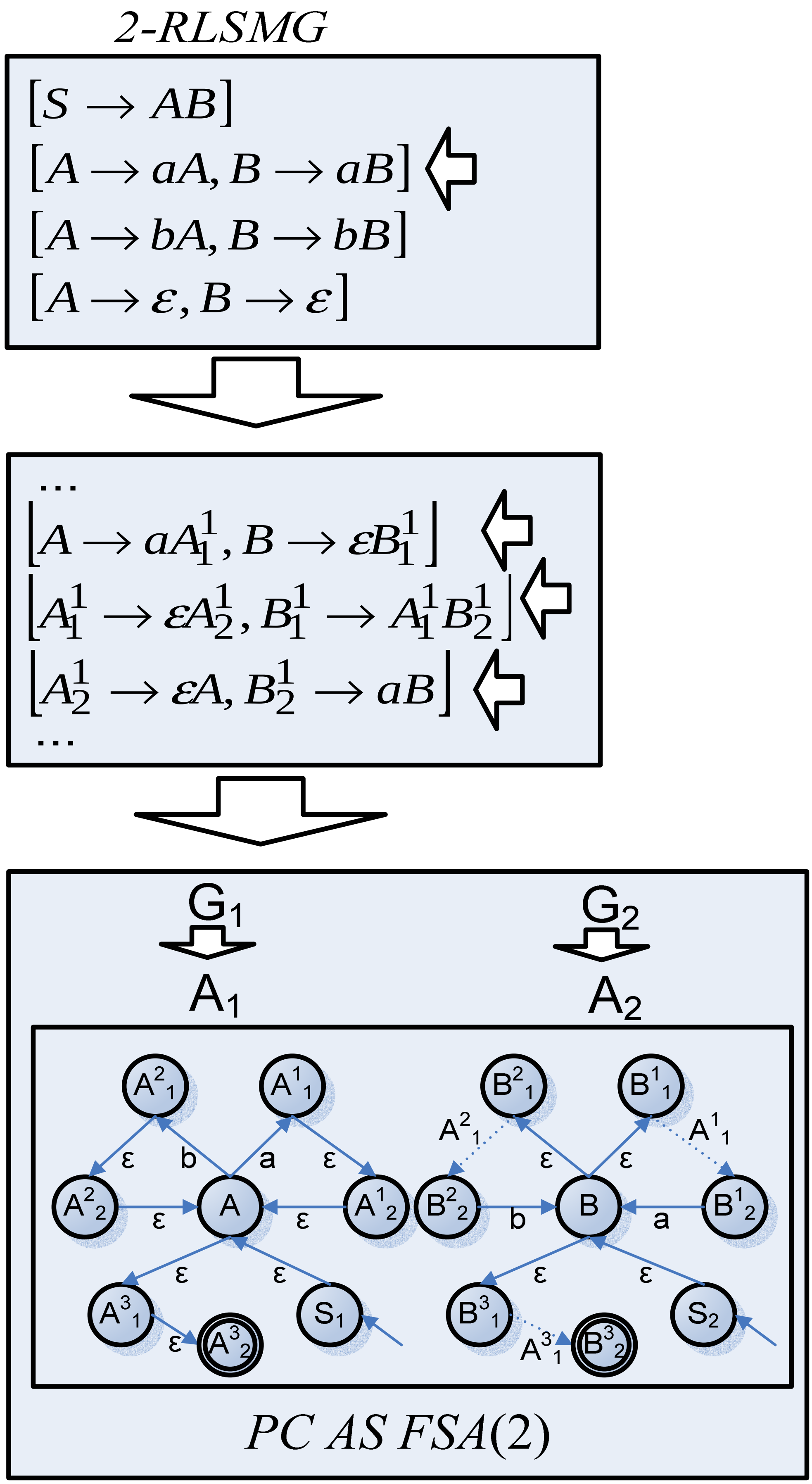
$$[X^i_{1_3} \to \varepsilon X^i_{1_4}, X^i_{2_3} \to \varepsilon X^i_{2_4}, X^i_{3_3} \to X^i_{2_3} X^i_{3_4}, ..., X^i_{n_3} \to X^i_{2_3} X^i_{n_4}],$$

$$[X^i_{1_4} \to \varepsilon X^i_{1_5}, X^i_{2_4} \to \varepsilon X^i_{2_5}, X^i_{3_4} \to a_5 X^i_{3_5}, ..., X^i_{n_4} \to \varepsilon X^i_{n_5}],$$

...

$$[X^i_{1_{2n-2}} \to \varepsilon Y_1, X^i_{2_{2n-2}} \to \varepsilon Y_2, X^i_{3_{2n-2}} \to \varepsilon Y_3, ..., X^i_{n_{2n-2}} \to a_n Y_n].$$

Pro pravidla typu $[X_1 \to a_1, ..., X_n \to a_n]$ provedeme převod obdobně. Takto jsme maticové pravidla upravili do tvaru, kdy „skrytě“ obsahují komunikační přechody. Jsou to ty přepisovací pravidla tvaru $X^i_{2_1} \to X^i_{1_1} X^i_{2_2}$. Tyto pravidla potřebujeme proto, abychom zachovali skutečnost, že jsou vždy provedena pouze pravidla, která jsou v daném maticovém pravidle. Převedeme-li nyní jednotlivé slot gramatiky na automaty, vzniknou nám pak z pravidel tvaru $X^i_{2_1} \to X^i_{1_1} X^i_{2_2}$ komunikační hrany. Tedy dostaneme *PC AS FSA*(n). Následující obrázek osvětlí princip konstrukce.


2-RLSMG
[S → AB]
[A → aA, B → aB]
[A → bA, B → bB]
[A → ε, B → ε]
...
[A → aA¹₁, B → εB¹₁]
[A¹₁ → εA¹₂, B¹₁ → A¹₁B¹₂]
[A¹₂ → εA, B¹₂ → aB]
...
G1
A1
G2
A2
PC AS FSA(2)

Obrázek 2.8

Věta 2.6. Pro každý *PC AS FSA*(n) existuje ekvivalentní *n-RLSMG*.

Důkaz: Nechť $AS = (\Sigma, A_1, \dots, A_n)$ je *PC AS FSA*(n), kde Σ je vstupní abeceda a $A_i = (Q_i, \Sigma_i, R_i, s_i, F_i)$ jsou komponenty systému. Zaveďme si následující pětici $T = (\Sigma', Q', R', s', F')$, kde $\Sigma' = \Sigma$ je vstupní abeceda. Zaveďme si nejprve pomocné množiny $\overline{Q}$, $\overline{R}$ takové, že $\overline{Q} = \{(q_1, \dots, q_n) \mid q_i \in Q_i, i \in \{1, \dots, n\}\}$ tj. Q' je $Q_1 \times Q_2 \times \dots \times Q_n$. Jedná se tedy o množinu všech možných n-tic stavů. $\overline{R} = \{(qa \to r) \mid q \in \overline{Q},\ \ q = (q_1, \dots, q_n),\ \ r \in \overline{Q},\ \ r = (r_1, \dots, r_n),$ $a = (a_1, \dots, a_n),\ \ a_i \in \left\{\Sigma' \cup \left\{\bigcup_{j=1}^{n} Q_j\right\}\right\},\ \ i \in \{1, \dots, n\} \wedge pro \forall i \in \{1, \dots, n\} \exists (q_i a_i \to r_i) \in R_i\}$. $\overline{R}$ je tudíž přechodová funkce $\overline{Q} \times \left(\underbrace{\Sigma \cup \left\{\bigcup_{j=1}^{n} Q_j\right\} \times \Sigma \cup \left\{\bigcup_{j=1}^{n} Q_j\right\} \times \dots \times \Sigma \cup \left\{\bigcup_{j=1}^{n} Q_j\right\}}_{n}\right) \to \overline{Q}$.

$R' = \{(qa \to r) \mid (qa \to r) \in \overline{R}, q = (q_1, q_2, \dots, q_n), q_i \in Q_i, r = (r_1, r_2, \dots, r_n),\ \ r_i \in Q_i,\ \ a = (a_1, a_2, \dots, a_n),$

$a_i \in \left\{\Sigma' \cup \left\{\bigcup_{j=1}^{n} Q_j\right\}\right\}, i \in \{1, \dots, n\} \wedge (\neg \exists j, k \mid j, k \in \{1, \dots, n\},\ j \neq k \wedge \neg \exists (qa' \to r') \in R' \mid$

$a' = (a_1, \dots, a_{k-1}, a'_k, a_{k+1}, \dots, a_n) \wedge a'_k \in \bigcup_{i=1}^{n} Q_i \wedge q_j = a'_k \wedge a_k \in \Sigma' \Big) \wedge$

$\left(\left(\exists k \mid k \in \{1, \dots, n\}, a_k \in \bigcup_{i=1}^{n} Q_i\right) \Rightarrow \left(\exists j \mid j \in \{1, \dots, n\}, j \neq k, q_j \in \bigcup_{i=1}^{n} Q_i\right)\right)$

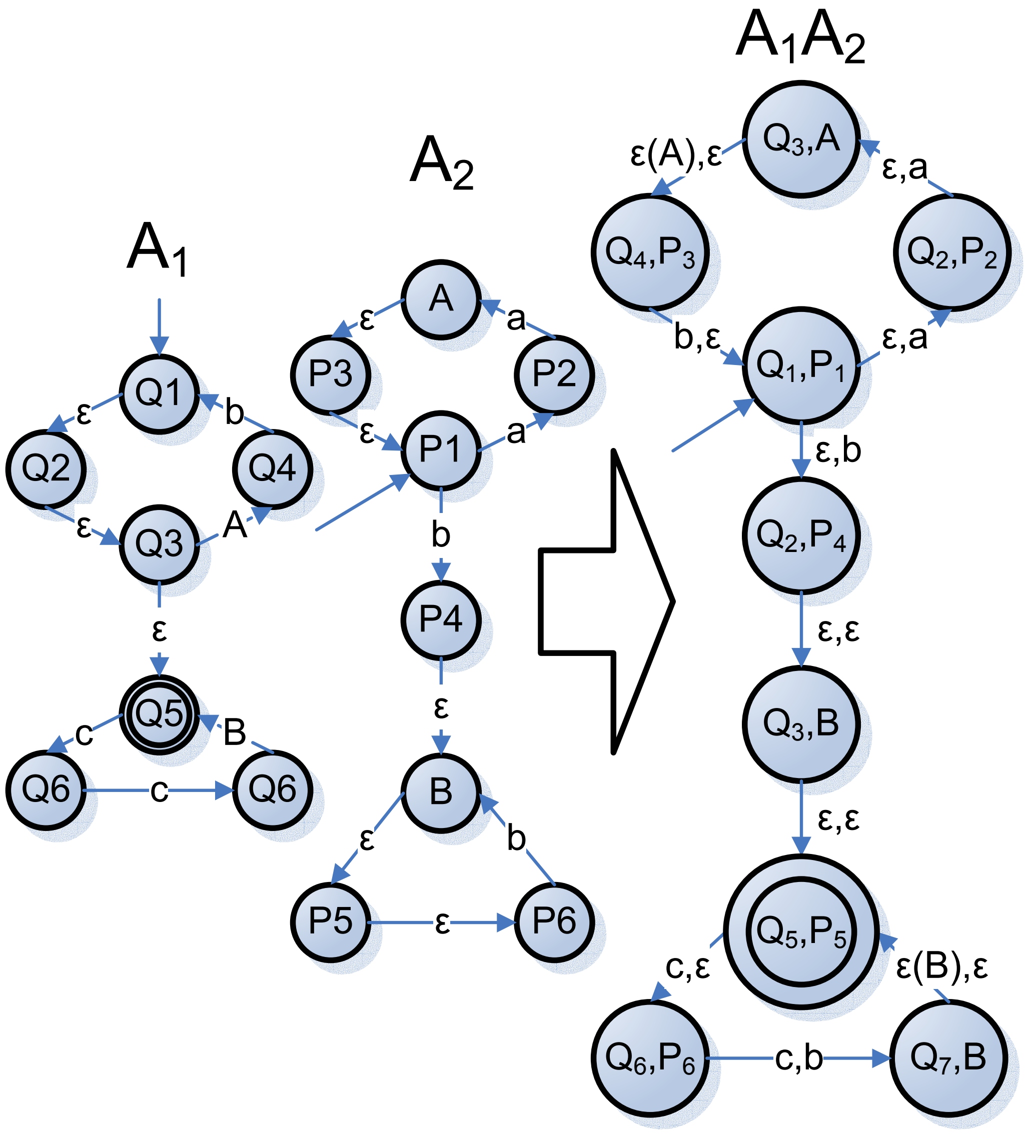
$\wedge ((\exists ((q_0 a_0 \to q_1) \in R' \wedge (q_1 a_1 \to q_2) \in R' \wedge \dots \wedge (q_m a_m \to q) \in R') \mid q_0 = (s_1, s_2, \dots, s_n),\ m \geq 0) \vee$

$(q = (s_1, s_2, \dots, s_n))) \wedge ((\exists ((rb_0 \to r_0) \in R' \wedge (r_0 b_1 \to r_1) \in R' \wedge \dots \wedge (r_{k-1} b_k \to r_k) \in R') \mid$

$r_k = (p_1, p_2, \dots, p_n), p_1 \in F_1, k \geq 0) \vee (r = (q_1, q_2, \dots, q_n), q_1 \in F_1)))\ \}$, R' je přechodová funkce obsahující pouze proveditelná pravidla a stavy, které jsou dostupné a ukončující. $Q' = \{q \mid q \in \overline{Q} \wedge (\exists (pa \to r) \in R' \mid (p = q) \vee (r = q))\}$ je množina dostupných a ukončujích stavů. $s' = (s_1, s_2, \dots, s_n)$ je počáteční stav. $F' = \{q \mid q \in Q', q = (q_1, q_2, \dots, q_n) \wedge q_1 \in F_1\}$ je množina obsahující prvky tvořeny *n*-ticemi stavů, kdy každá *n*-tice obsahuje koncový stav.

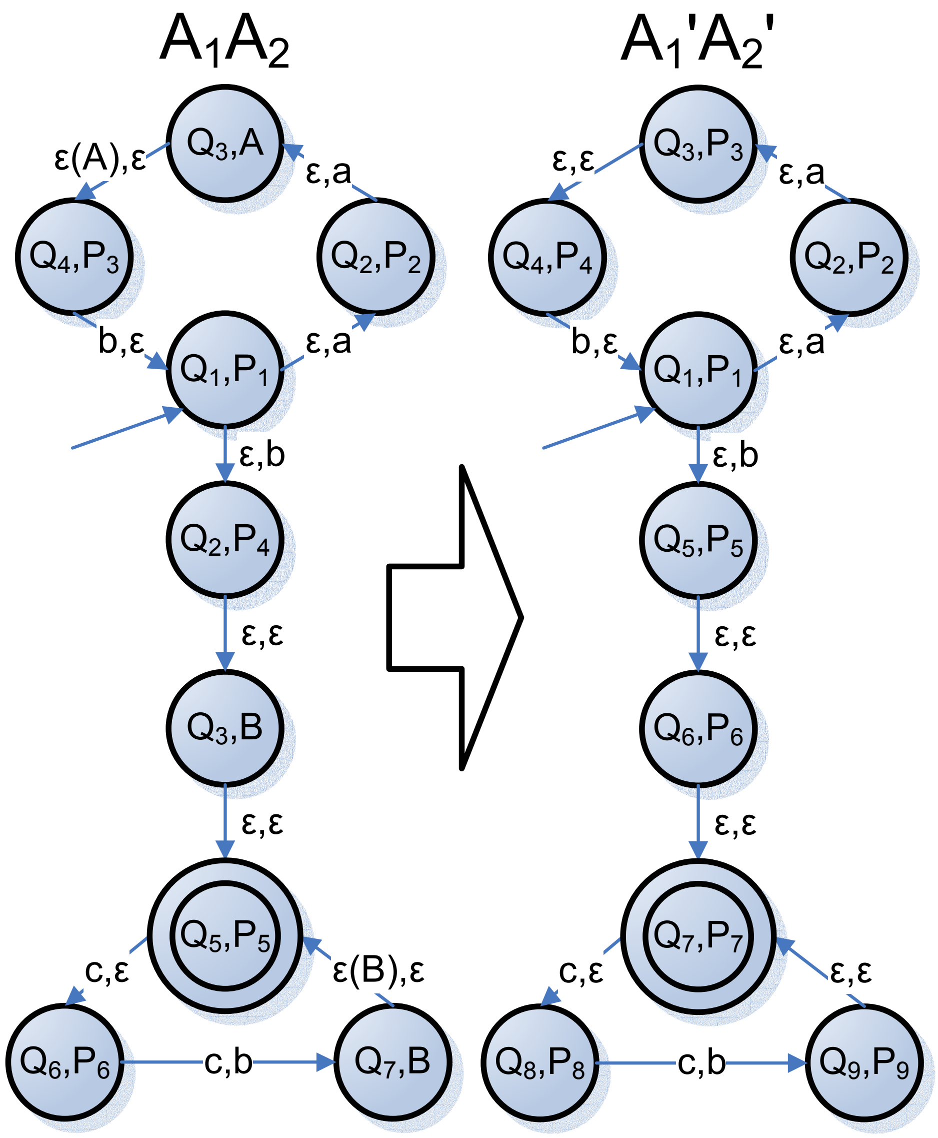
V důkazu je využita myšlenka, že původní automatový systém *PC AS FSA*(n) je složen z komponent, kdy každá má konečný počet stavů, z toho plyne, že pokud bychom celý systém považovali za jednu entitu, která je v určitém okamžiku v jednom stavu je tento jeden stav reprezentovám právě *n*-ticí stavů komponent automatového systému, a protože ty mají konečný počet stavů, má i celkový systém jako jeden celek konečný počet stavů. A mezi těmito stavy jsou vždy provedeny přechody jako v původním automatovém systému.

Výše uvedená konstrukce obsahuje pouze dostupné a ukončující stavy, z toho vyplývá, že pokud obsahuje pouze dostupné stavy, tak buď bylo provedeno, co se týče původního automatového systému, *n* přechodů v nichž se nevyskytla komunikace, nebo mohlo být provedeno několik přechodů komunikačních. Z těchto dvou faktů (stavy jsou dostupné a mohla být provedena komunikace) plyne, že podmínka, kvůli niž byla komunikace zavedena, je splněna vždy. Tudíž komunikace, o níž je předem známo jak skončí, je zbytečná. A tak můžeme komunikační přechody nahradit „normálními“ přechody s epsilon. Což je přesně to, oč nám jde, tedy odstranit komunikační pravidla, abychom mohli použít algoritmus na převod konečného automatu na pravou lineární gramatiku.



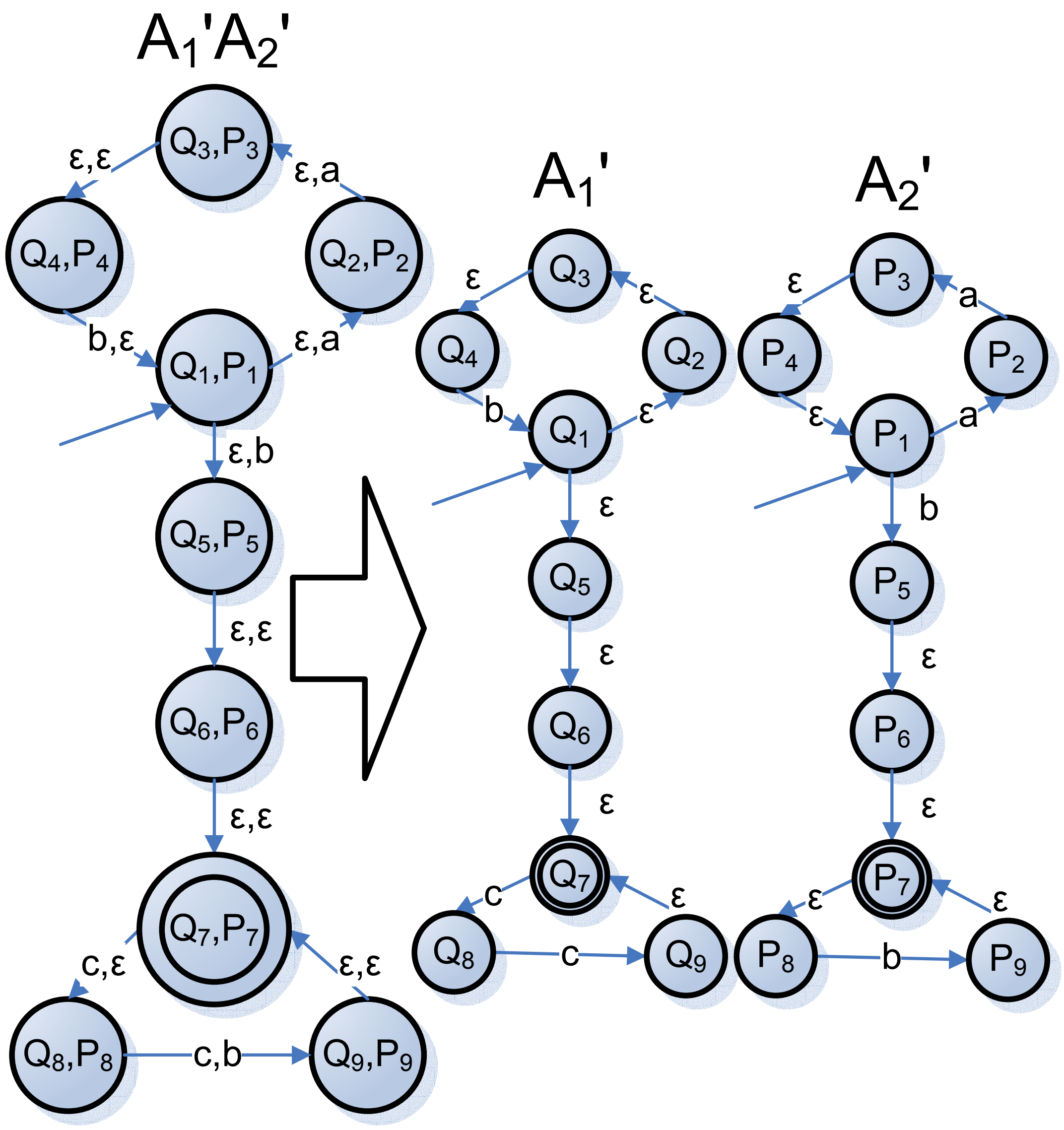
Obrázek 2.9

Nyní proveďme přejmenování stavů podle následujícího obrázku. Přejmenování provadíme především kvůli názornosti, aby byl důkaz jasný.



Obrázek 2.10

Na obrázku 2.10 vidíme, že přejmenované stavy jsou složeny ze stavů, které mají stejné dolní indexy, tohoto faktu využijeme později při sestavování maticových pravidel. Nyní převedeme námi vytvořenou strukturu zpět na jednotlivé konečné automaty, které nyní již neobsahují žádná komunikační pravidla. Viz. následující obrázek.



Obrázek 2.11

Převedeme jednotlivé konečné automaty pomocí algoritmu na převod konečného automatu na pravou lineární gramatiku a dostáváme gramatiky ve tvaru $G_i = (N_i, T, s_i, P_i)$.

Nyní nechť G je *n-RLSMG*, tj. $G = (N_1, \dots, N_n, T, S, P)$, kde N_i je abeceda nonterminálů dané gramatiky a platí, že jednotlivé abecedy nonterminálů jsou vzájemně disjunktní, $T = \Sigma$, S je počáteční symbol, $S \notin (N_1 \cup N_2 \cup \dots \cup N_n)$, P je množina pravidel obsahující následující pravidla:

1. $[S \to s_1 s_2 \dots s_n]$,
2. $[X_1 \to a_1 Y_1, X_2 \to a_2 Y_2, \dots, X_n \to a_n Y_n]$, kde $X_i \to a_i Y_i \in P_i$, $X_i, Y_i \in N_i$, $a_i \in T \cup \{\varepsilon\}$, $i \in \{1, \dots, n\}$ a platí, že $X_1, X_2, \dots, X_n$ mají v pojmenování stavů stejný dolní index a $Y_1, Y_2, \dots, Y_n$ mají v pojmenování stavů stejný dolní index,

3. $[X_1 \to \varepsilon, X_2 \to \varepsilon, ..., X_n \to \varepsilon]$, kde $X_i \to \varepsilon \in P_i$, $X_i \in N_i$, $i \in \{1,...,n\}$ a platí, že $X_1, X_2, ..., X_n$ mají v pojmenování stavů stejný dolní index,

potom *G* je ekvivalentní *AS*.

Věta 2.7. Pro $n \geq 1$ platí

1. $A_n \subset A_{n+1}$
2. $A_{[n]} \subset A_{[n+1]}$

a pro $n \geq 2$ platí

3. $A_n \subset A_{[n]}$.

Důkaz: Tvrzení plyne z předchozích vět, tedy, že pro každou *n-PRLG* existuje ekvivalentní *P AS FSA*(*n*), pro každý *P AS FSA*(*n*) existuje ekvivalentní *n-PRLG*, pro každou *n-RLSMG* existuje ekvivalentní *PC AS FSA*(*n*), pro každý *PC AS FSA*(*n*) existuje ekvivalentní *n-RLSMG* a z faktů uvedených v [8], tedy $R_n \subset R_{n+1}$, $R_{[n]} \subset R_{[n+1]}$, $R_n \subset R_{[n]}$.

Poslední větou jsme ukázali přímý vztah mezi gramatikami studovanými v díle [8], automatovými systémy z [5] a námi navrženými systémy. Stejně jako tyto gramatiky tvoří hierarchickou strukturu, tak i námi navržené systémy tvoří hierarchickou strukturu. Tedy přidáním další komponenty zvýšíme sílu systému, z čehož také plyne, že tak jako u některých gramatických systému lze minimalizovat počet komponent se zachováním generovaného jazyka, tak u těchto automatových systému toto neplatí. Dalším přínosem je nalezení dalších fundamentálních modelů k těmto gramatikám.

3 Budoucí výzkum

V práci byly definovány automatové systémy a jejich různé verze. Přesto po vzoru gramatických systémů zbývají oblasti, které studovány nebyly.

Proto další budoucí vývoj v oblasti automatových systémů může kromě definovaných automatových systémů studovat i další formy těchto systémů tzn. vracející se automatové systémy, hybridní automatové systémy, automatové systémy založené na jiném způsobu komunikace a podobně. Dále je možné uvažovat systémy obsahující i jiné komponenty než jen konečné automaty, jako například různé druhy zásobníkových automatů, Turingových strojů či systémy obsahující různé typy komponent, tedy hybridní automatové systémy.

Tak jako byla práce zaměřená spíše teoreticky je možné v budoucnu studovat vlastnosti těchto systémů z pohledu praktického. Tj. především využití těchto systémů například v oblasti překladačů a v oblastech, kde jsou využívány gramatické systémy.

Závěr

Gramatické systémy, gramatiky přepisující v rámci jednoho derivačního kroku více nonterminálů a další struktury pracující paralelně. To je moderní trend v teorii formálních jazyků. Naší prací jsme chtěli přispět k tomuto bádání také v oblasti automatů, kterým nebyla doposud věnována taková pozornost jako gramatickým systémům.

Po seznámení se s gramatickými systémy, paralelními gramatikami a dalšími objevy v oblasti ať už gramatických či automatových systémů, jsme zavedli paralelní automatové systémy a paralelně komunikující automatové systémy komunikující přechody. Zkoumali jsme vlastnosti těchto systémů a porovnávali je s vlastnostmi již známých systémů. Ukázali jsme přímý vztah mezi gramatikami studovanými v díle [8] a našimi systémy.

V této práci byly studovány základní vlastnosti automatových systémů. Budoucí výzkum může být zaměřen jak teoreticky a studovat tak další vlastnosti automatových systémů nebo může studovat využití automatových systémů po praktické stránce.

Přínosem pro autora je prohloubení znalostí z oblasti formálních jazyků, které, jak doufá, bude moci využít v budoucím profesním životě.

Literatura

[1] Češka, M.: *Teoretická informatika*, učební text FIT VUT v Brně, 2002, [online], [cit. 18-04-2007], [rev. 18-04-2007], URL < http://www.fit.vutbr.cz/study/courses/TIN/ >.

[2] Dassow, J., Paun, Gh., Rozenberg, G.: *Grammar Systems*, in: G. Rozenberg and A. Salomaa, (eds.), *Handbook of Formal Languages*, Vol. 2, Springer, Berlin 1997.

[3] Goehring, M.: *Parallel communicating automata systems*, učební text University of Potsdam, 2006, [online], [cit. 18-04-2007], [rev. 18-04-2007], URL < http://www.ift.cs.uni-potsdam.de/iuk/lehre/ss06/fmvr/ParalleleAutomaten.pdf>.

[4] Krithivasan, K., Sakthi Balan, N., Harsha, P.: *Distributed processing in automata*, in: International Journal of Foundations of Computer Science, Vol. 10, No. 4, 1999, s. 443-463.

[5] Martín-Vide, C., Mateescu, A., Mitrana, V.: *Parallel finite automata systems communicating by states*, in: International Journal of Foundations of Computer Science, Vol. 13, No. 5, 2002, s. 733-749.

[6] Martín-Vide, C., Mitrana, V.: *Some undecidable problems for parallel communicating finite automata systems*, in: Information Processing Letters, Vol. 77, 2001, s. 239-245.

[7] Meduna, A.: *Automata and Languages, Theory and Applications*, Springer, London, 2000.

[8] Wood, D.: *m-parallel n-right linear simple matrix languages*, in: Utilitas Mathematica, Vol. 8, 1975, s. 3-28.

[9] Wood, D.: *Properties of n-parallel finite state languages*, in: Utilitas Mathematica, Vol. 4, 1973, s. 103-113.

Seznam příloh

Příloha 1. CD