VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Formální modely ve verifikaci

2002

Luděk Haša

Abstrakt

Automaty nad nekonečnými řetězci existují v různých variantách. Büchiho automat je jeden z nejjednodušších automatů nad nekonečnými řetězci. Jako další nekonečná struktura pro automaty se používá nekonečný strom. Automaty nad nekonečnými strukturami se ve verifikaci používají, jelikož úzce souvisí s temporálními logikami, jichž se používá pro specifikaci chování reaktivních systému.

Vedle systémů s konečným stavovým prostorem existují také systémy nekonečně-stavové a parametrické. Tyto systémy lze modelovat tak, že jednotlivé stavy se reprezentují jako konečné řetězce určité délky nad konečnou abecedou a jejich chování pomocí převodníků. Hlavní problémy ve verifikaci nekonečně-stavových a parametrických systémů je výpočet množiny dosažitelných stavů a výpočet tranzitivního uzávěru přechodové relace.

Tuto práci jsem vypracoval v rámci předmětu *Moderní teoretická informatika* za použití uvedené literatury.

Obsah

1 Konečné automaty nad nekonečnými strukturami

1.1 ω-automaty

Automaty nad nekonečnými vstupními řetězci (ω-automaty) byly zavedeny v roce 1960. Automaty byly využity pro řešení některých základních problémů v matematické logice. Nyní se používají pro specifikaci a verifikaci reaktivních systémů. Převedením systému na automat převedeme verifikaci systému na verifikaci automatů.

Tak jako automaty nad konečnými vstupními řetězci mají ω-automaty konečný počet stavů a přijímají nebo odmítají vstupní řetězce. Nemají však koncový stav, přijetí řetězce je provedeno s respektováním nekonečné posloupnosti navštívených stavů.

1.2 Büchiho automat

Nejjednodušší případ ω-automatu je Büchiho automat. Nekonečný řetězec Σ^ω je nekonečná sekvence znaků $a_0, a_1, \ldots$ z abecedy Σ.

Definice 1.1: Nedeterministický Büchiho automat $\mathcal{A}$ je pětice $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, kde

- Σ je konečná abeceda.
- Q je konečná množina stavů.
- $\delta : Q \times \Sigma \rightarrow 2^Q$ je přechodová funkce.

- q_0 je počáteční stav
- F je konečná množina akceptujících stavů.

Běh τ automatu $\mathcal{A}$ nad řetězcem $w = a_0, a_1 \ldots$ z abecedy Σ je sekvence $q_0, q_1, \ldots$, kde q_0 je počáteční stav a $q_{i+1} \in \delta(q_i, a_i)$ pro všechna $i > 0$. Dále definujeme množinu $inf(\tau)$, která obsahuje všechny stavy, které se vyskytují v běhu τ v nekonečném počtu. Automat $\mathcal{A}$ přijímá řetězec $w \in \Sigma^\omega$ právě, když $inf(\tau) \cap F \neq \emptyset$. To znamená, že některý z akceptujících stavů se objevuje v τ v nekonečném počtu. Jazyk $L(\mathcal{A})$ přijímaný automatem $\mathcal{A}$ je množina řetězců přijímaných automatem $\mathcal{A}$.

Věta 1.1: Nedeterministický Büchiho automat je uzavřen vzhledem ke komplementu.
První konstrukci komplementárního automatu provedl J. R. Büchi. Tato konstrukce byla netriviální a nárůst velikosti automatu byl značný. První optimální konstrukcí provedl S. Safra. Počet stavů komplementárního automatu je $O(2^{n.\log n})$, kde n je počet stavů původního automatu. Algoritmus je však velmi těžko implementovatelný.

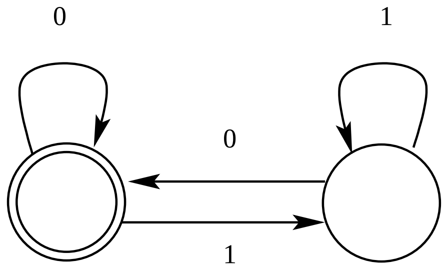
Věta 1.2: Nedeterministický Büchiho automat je uzavřen vzhledem k průniku.

Algoritmus 1.1: Nechť existují dva automaty $\mathcal{A}_1 = (\Sigma, Q_1, \delta_1, q_{01}, F_1)$ a $\mathcal{A}_2 = (\Sigma, Q_2, \delta_2, q_{02}, F_2)$. Můžeme sestrojit automat, který přijímá $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$, takto: $\mathcal{A}_1 \cap \mathcal{A}_2 = (\Sigma_1, Q_1 \times Q_2 \times \{0, 1, 2\}, \delta, q_{01} \times q_{02} \times \{0\}, F_1 \times F_2 \times \{2\})$. $\langle r_m, q_n, y\rangle \in \delta(\langle r_i, q_j, x\rangle, a)$ právě když jsou splněny tyto podmínky:

- $r_m \in \delta(r_i, a)$ a $q_n \in \delta(q_j, a)$.
- Třetí komponenta je určena podle:
  - jestliže $x = 0$ a $r_m \in F_1$, potom $y = 1$.
  - jestliže $x = 0$ a $q_n \in F_2$, potom $y = 2$.
  - jestliže $x = 2$, potom $y = 0$

Definice 1.2: Büchiho automat je deterministický, jestliže pro všechny $a \in \Sigma, q \in Q$ platí $|\delta(q, a)| = 1$. Tedy platí, že pro každý řetězec existuje právě jeden běh v deterministickém automatu.

Věta 1.3: Deterministický Büchiho automat není uzavřen vzhledem ke komplementu



Obrázek 1: Nedeterministický Büchiho automat pro $L = (01^*)^\omega$

Důkaz : Mějme jazyk $L = (01^*)^\omega$, jazyk všech řetězců, kde se 0 vyskytuje v nekonečném počtu. Tento jazyk přijímá deterministický Büchiho automat. Komplementární jazyk k jazyku L je jazyk všech řetězců s konečným počtem 0 — $\overline{L} = (0+1)^*1^\omega$. Tento jazyk nelze přijmout žádným deterministickým Büchiho automatem. Předpokládejme, že existuje deterministický automat $\overline{\mathcal{A}}$, který přijímá jazyk $\overline{L}$, a předpokládejme řetězec 1^ω. Jestliže je přijat automatem $\overline{\mathcal{A}}$, existuje $i_1 > 0$ takové, že existuje běh přes 1^{i_1}, který vede k akceptujícímu stavu $f_1 \in \overline{\mathcal{A}}$. Nyní uvažujme řetězec $1^{i_1}01^\omega$. Jestliže je tento řetězec přijat, existuje $i_2 > 0$ takové, že existuje běh přes $1^{i_1}01^{i_2}$, který vede k akceptujícímu stavu $f_2 \in \overline{\mathcal{A}}$. Stav f_2 musí být různý od stavu f_1, jinak by byl přijat i řetězec $1^{i_1}(01^{i_2})^\omega$. Takto můžeme pokračovat a sestrojit nekonečnou sekvenci stavů $f_1, f_2 \ldots$. Tento počet však nemůže být konečný, což neodpovídá tomu, že $\overline{\mathcal{A}}$ je konečný automat.

Věta 1.4: Problém prázdnosti jazyka Büchiho automat je rozhodnutelný v lineárním čase.

Důkaz : Daný Büchiho automat $\mathcal{A}$ předpokládejme jako graf. Jelikož množina akceptujících stavů je konečná, musí každý akceptující běh navštívit akceptující stav nekonečně krát. Automat musí obsahovat cyklus, ve kterém bude obsažen některý akceptující stav. Tento cyklus musí byt dosažitelný z počátečního stavu.

Neprázdnost jazyka $L(\mathcal{A})$ je tedy ekvivalentní nalezení silně souvislé komponenty, která je dosažitelná z počátečního stavu a obsahuje akceptující stav. Pro dekompozici grafu na silně souvislé komponenty existuje DFS algoritmus pracující v lineárním čase (viz. [1]).

1.3 Další ω-automaty

Vedle Büchiho automatu existují další modifikace ω-automatů. Největší modifikace se provádí v podmínkách pro akceptování stavů.

Definice 1.3: Mullerův automat $\mathcal{A}$ je pětice $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, kde

- $F \subseteq 2^Q$ je konečná množina akceptujících stavů.

Řetězec w je přijat Mullerovým automatem, jestliže množina stavů $inf(\tau)$ běhu τ nad řetězcem w je obsažena v F.

Věta 1.5: Každý jazyk přijímaný deterministickým Büchiho automatem je přijímán Mullerovým automatem.

Důkaz : Nechť $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ je Büchiho automat. Nechť existuje množina $F_M = \{S \subseteq Q | S \cap F \neq \emptyset\}$. Potom Mullerův automat je $\mathcal{A}_M = (\Sigma, Q, \delta, q_0, F_M)$.

Věta 1.6: Třída jazyků přijímaná nedeterministickým Büchiho automatem a třída jazyků přijímaná Mullerovým automatem jsou stejné.

Toto bylo dokázáno v roce 1966 a je to jeden z fundamentálních výsledků teorie ω-automatů. Nejsložitější část důkazu bylo ukázání, že každý nedeterministický automat má ekvivalentní deterministickou verzi. Toto bylo dokázáno přímou konstrukcí.

Definice 1.4: Rabinův automat $\mathcal{A}$ je pětice $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, kde

- F je konečná množina $\{(L_1, U_1), \ldots (L_n, U_n)\}$, taková, že platí pro každé $i : 1 \leq i \leq n, L_i, U_i \subseteq Q$.

Běh τ je akceptující, jestliže pro některé $i : 1 \leq i \leq n$ existuje nějaký stav v L_i, který se vyskytuje v $inf(\tau)$, a zároveň všechny stavy z U_i se vyskytují v konečném počtu $(\exists i : inf(\tau) \cap L_i \neq \emptyset \wedge inf(\tau) \cap U_i = \emptyset)$.

Definice 1.5: Streetův automat $\mathcal{A}$ je pětice $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, kde

- F je konečná množina $\{(L_1, U_1), \ldots (L_n, U_n)\}$, taková, že platí pro každé $i : 1 \leq i \leq n, L_i, U_i \subseteq Q$.

Běh τ je akceptující, jestliže pro všechny $i : 1 \leq i \leq n$ platí, jestliže běh navštíví některý stav z L_i v nekonečném počtu, potom také U_i je navštíveno v nekonečném počtu $(\forall i : inf(\tau) \cap L_i \neq \emptyset \rightarrow inf(\tau) \cap U_i \neq \emptyset)$. Vlastností Streetova automatu je, že akceptující podmínky zajišťují, že automat splňuje podmínky silné spravedlnosti (silná spravedlivost: výpočet je nespravedlivý pokud existuje událost, která je proveditelná nekonečně krát, je provedena jen v konečném počtu). Pro každou událost e akceptující podmínky by obsahovali dvojici (L_i, U_i), kde L_i je množina stavů, ve kterých je e proveditelná, a U_i je množina stavů, kde je e provedena. Podmínky slabé spravedlnosti (slabá spravedlnost: výpočet je nespravedlivý pokud existuje událost, která je spojitě proveditelná, a je provedena pouze v konečném počtu) splňuje například Büchiho automat.

Věta 1.7: Nedeterministický Büchiho, Rabinův, Mullerův a Streetův automat přijímají stejnou třídu ω-jazyků Tuto věta lze dokázat tak, že jednotlivé automaty lze simulovat pomocí Mullerova automatu. Shodnost jazyků Mullerova a Büchiho byla uvedena ve větě 1.6. Automaty lze mezi sebou převádět a využít tak specifických vlastností jednotlivých automatů, které jsou dány různými podmínkami pro přijetí řetězce. Algoritmy pro převody lze najít např. v [3]. Při převodu musíme však počítat s růstem složitosti automatu.

1.4 Automaty nad nekonečnými stromy

Vstupem takového automatu je nekonečný strom (předpokládáme binární strom, existují však automaty i pro stromy jiných stupňů). Každý uzel je označen symbolem ze vstupní abecedy Σ a lze ho identifikovat řetězci z $0, 1^*$, strom T:$0, 1^* \to \Sigma$.

Definice 1.6: Automat nad nekonečnými stromy $\mathcal{T}$ je pětice $\mathcal{T} = (\Sigma, Q, \delta, q_0, F)$, kde

- Σ je konečná abeceda.
- Q je konečná množina stavů.
- δ je přechodová funkce, která pro daný stav $q \in Q$ a vstupní symbol $a \in \Sigma$ určuje množinu dvojic obsahujících levého následníka q_l a pravého následníka q_r. $\delta : Q \times \Sigma \to 2^{Q \times Q}$. Pro deterministický automat platí $\delta : Q \times \Sigma \to Q \times Q$.
- q_0 je počáteční stav.
- F je konečná množina akceptujících stavů.

Běh pro automatu $\mathcal{T}$ je strom $\Gamma : 0, 1^* \to Q$, kde kořen je stav q_0 a všechny další uzly jsou určeny podle δ, pro každý uzel $p \in 0, 1^*$

platí $(\Gamma(p0), \Gamma(p1)) \in \delta(\Gamma, T(p))$. Běh je akceptující, jestliže každá nekonečná cesta splňuje akceptační podmínky. Akceptační podmínky jednotlivých automatů odpovídají akceptačním podmínkám ω-automatů. Např. Rabinův automat: F je konečná množina $\{(L_1, U_1), \ldots (L_n, U_n)\}$, taková, že platí pro každé $i : 1 \leq i \leq n, L_i, U_i \subseteq Q$. Běh Γ je akceptující, jestliže pro všechny cesty v Γ existuje nějaké $i : 1 \leq i \leq n$ takové, že existuje nějaký stav v L_i, který se vyskytuje v nekonečném počtu, a zároveň všechny stavy z U_i se vyskytují v konečném počtu.

Rabinův, Mullerův a Streetův automat nad nekonečnými stromy má stejnou výpočetní sílu a jsou uzavřeny vzhledem k průniku a problém prázdnosti jazyka automatů je rozhodnutelný.

1.5 Využití automatů ve verifikaci

Temporální logika je silný nástroj pro specifikaci a verifikaci paralelních programů. Velmi důležité bylo, že pro verifikaci temporálních vlastností systémů s konečným počtem stavů byly vyvinuty patřičné verifikační algoritmy. Verifikované systémy lze modelovat jako přechodový systém, třeba jako *Kripkeho strukturu*, jehož vlastnosti jsou specifikovány temporální logikou. Verifikace se pak provádí na tomto modelu a ověřuje zda temporální formule splňuje požadované vlastnosti.

Temporální logiku lze rozdělit na dva druhy: lineárním a větvená. U lineární logiky má každý okamžik v čase jednu možnou budoucnost, kdežto u větvené každý okamžik v čase se může rozdělit do několik možných budoucností. Pro obě logiky existuje spojitost s teorií automatů nad nekonečnými objekty. Základní myšlenka je asociovat temporální formuli s automatem, který přijímá všechny výpočetní posloupnosti, které splňují formuli. Lineární logice odpovídají ω-automaty a větvené automaty nad nekonečnými stromy.

Ověření požadovaných vlastností modelu lze potom provést tak, že převedeme model na konečný automat $\mathcal{A}$. Toto lze jednoduše z Kripkeho struktury a postup lze naleznout v [4]. Kripkeho struktura odpovídá ω-automatu. Pro specifikaci modelu (temporální formuli) S vytvoříme také automat s jazykem $L(S)$. Jazyk $L(S)$ je množina požadovaného chování. Systém $\mathcal{A}$ splňuje S jestliže platí:

$$L(A) \subseteq L(S)$$

Z toho plyne, že $\mathcal{A}$ obsahuje jen povolené chování specifikací. Nechť $\overline{L(S)}$ je komplement $L(S)$, pak můžeme předcházející vlastnost přepsat:

$$L(A) \cap \overline{L(S)} = \emptyset$$

To znamená, že $\mathcal{A}$ neobsahuje žádné chování, které není povolené specifikací.

Vzhledem k uzávěrovým vlastnostem automatů (průnik, komplement), lze navrhnout tento postup ověření modelu:

1. Vytvoříme automat $\overline{S}$ s jazykem $\overline{L(S)}$.
2. Vytvoříme automat, který přijímá jazyk $L(A) \cap \overline{L(S)}$

Jestliže je jazyk výsledného automatu prázdný, model $\mathcal{A}$ splňuje specifikaci S.

2 Modelování parametrických a nekonečně-stavových systémů

Vedle systémů s konečným stavovým prostorem existují i systémy s nekonečným počtem stavů, např. systémy pracující s frontou. I pro ně existují formalismy, které umožňují verifikaci. Tyto formalismy lze také použít pro parametrické systémy, které se skládají ze systémů s konečným počtem stavů a jsou v lineární nebo kruhové topologii. Stavy systému jsou kódovány pomocí regulární množiny řetězců nad konečnou abecedou. Přechody mezi těmito stavy jsou reprezentovány převodníky, které zachovávají délku řetězce.

2.1 Model

Jako model se používá program. Nechť Σ je konečná abeceda. Relace R nad Σ^* je *regulární* a *zachovávající délku*, jestliže w a w' jsou stejné délky, kdykoliv $(w, w') \in R$ a množina $\{(a_1, a'_1) \ldots (a_n, a'_n) : (a_1 \ldots a_n, a'_1 \ldots a'_n) \in R\}$ je regulární podmnožina $(\Sigma \times \Sigma)^*$.

Definice 2.1: Program je trojice $P = (\Sigma, \phi_I, R)$, kde

- Σ je konečná abeceda.
- ϕ_I je regulární množina nad Σ, značící počáteční konfiguraci.
- R je regulární relace nad Σ^*, někdy je nazývaná přechodová funkce.

Konfigurace w programu P je řetězec $a_1 a_2 \ldots a_n$ nad Σ.

Parametrický systém

Předpokládejme parametrický systém, který se skládá z libovolného počtu procesů, které jsou spojeny v lineární topologii. Konfiguraci takového systému lze reprezentovat řetězcem nad abecedou, která se skládá z množiny stavů každého procesu a délka řetězce udává počet procesů.

Systémy komunikující neomezeným FIFO kanálem

Uvedeme příklad pro definici systémů, které mezi sebou komunikují neomezeným FIFO kanálem. Předpokládejme, že se používá pro zjednodušení jen jeden kanál. Kanál obsahuje sekvenci zpráv z množiny M. Nechť Q je konečná množina řídících stavů. Konfigurace systému lze vyjádřit jako řetězec této formy:

$$q \perp\perp \ldots \perp m_1 m_2 \ldots m_n \perp\perp \ldots \perp$$

kde $q \in Q$ je aktuální stav a $m_1 m_2 \ldots m_n$ je sekvence zpráv v kanále. Aby bylo možné přidávat a odebírat zprávy, jsou přidány symboly $\perp$. Potom systém lze namodelovat jako program $P = (\Sigma, \phi_I, R)$, kde:

- $\Sigma = Q \cup M \cup \{\perp\}$.
- ϕ_I je regulární množina $q_i \perp^*$.
- R je sjednocení několika relací: jedna pro každou operaci v systému. Operace, která zasílá zprávu $m \in M$ a přechází ze stavu q do stavu q' je modelována relací:

  $$(q, q').(\perp, \perp)^*.(R_{id} \cap M^2).(\perp, m).(\perp, \perp)^*$$

  R_{id} je relace identity.

2.2 Verifikace

Nechť R je regulární relace, potom R^+ značí tranzitivní uzávěr R, R^* značí reflexivní a tranzitivní uzávěr, R^{-1} značí inversní relaci k R. Pro regulární množinu ϕ, $R(\phi)$ značí obraz $w' : \exists w \in \phi, (w, w') \in R$ množiny ϕ v relaci R.

Mezi hlavní verifikační problémy patří:

- *výpočet množiny dosažitelnosti* $R^*(\phi)$ — tuto množinu lze použít pro řešení dosažitelnosti určité konfigurace z množiny ϕ_F z počáteční konfigurace ϕ_I.
- *výpočet tranzitivního uzávěru* R^+ — uzávěr se používá k zjištění smyček parametrizovaných systémů.

2.3 Konstrukce tranzitivního uzávěru R^+

Máme relaci R, kterou lze reprezentovat konečně-stavovým převodníkem $R = (Q, q_0, \delta, F)$, kde Q je množin stavů,q_0 počáteční stav, $\delta : (Q \times ((\Sigma \times \Sigma)) \to 2^Q$ je přechodová funkce a $F \subseteq Q$ jsou koncové stavy. Pro každý pár (w, w') v R^+ existuje sekvence konfigurací $w^0, w^1, \ldots, w^m$ takových, že $w = w^0, w' = w^m$ a $(w^{i-1}, w^i) \in R$ pro $0 < i \leq m$. Nechť w^i je řetězec $a_1^i a_2^i \ldots a_n^i$ délky n, potom $(w^{i-1}, w^i) \in R$ znamená, že existuje běh $q_0^i q_1^i \ldots q_n^i$ převodníku R, který přijímá řetězec $(a_1^{i-1}, a_1^i), (a_2^{i-1}, a_2^i), \ldots (a_n^{i-1}, a_n^i)$.Uspořádáme tyto běhy do matice:

$$
\begin{array}{ccccccc}
q_0^1 & \xrightarrow{(a_1^0,a_1^1)} & q_1^1 & \xrightarrow{(a_2^0,a_2^1)} & q_2^1 \ldots q_{n-1}^1 & \xrightarrow{(a_n^0,a_n^1)} & q_n^1 \\
q_0^2 & \xrightarrow{(a_1^1,a_1^2)} & q_1^2 & \xrightarrow{(a_2^1,a_2^2)} & q_2^2 \ldots q_{n-1}^2 & \xrightarrow{(a_n^1,a_n^2)} & q_n^2 \\
 & \vdots & & \vdots & & \vdots & \\
q_0^m & \xrightarrow{(a_1^{m-1},a_1^m)} & q_1^m & \xrightarrow{(a_2^{m-1},a_2^m)} & q_2^m \ldots q_{n-1}^m & \xrightarrow{(a_n^{m-1},a_n^m)} & q_n^m
\end{array}
$$

Každý řádek ukazuje běh převodníku R s n přechody.

Pro relaci R^+ sestrojíme automat, jehož stavy jsou sloupce $q^1 q^2 \ldots q^m$ pro $m \geq 1$. Jeho přechody reprezentují vztahy mezi sousedními sloupci v matici. Definujeme převodník pro R^+ jako čtveřici $(Q^+, q_0^+, \Delta, F^+)$, kde

- Q^+ je množina neprázdných sekvencí stavů R.
- q^+ je množina neprázdných sekvencí počátečních stavů R.
- $\Delta : (Q^+ \times (\Sigma \times \Sigma)) \to 2^{Q^+}$ je definována: pro sloupce $t^0, t^1, \ldots, t^m$, $q^0, q^1, \ldots, q^m$ a dvojici (a, a') platí $t^0, t^1, \ldots, t^m \in \Delta(q^0, q^1, \ldots, q^m, (a, a'))$, jestliže existují $a^0, a^1, \ldots, a^m$, kde $a = a^0, a' = a^m$ tak, že $t^i \in \delta(q^i, (a^i - 1, a^i))$ pro $1 \leq i \leq m$.

Lze ukázat, že tento převodník přijímá právě R^+. Obsahuje však nekonečný počet stavů. Snížení stavů lze provést determinizací. Postup lze nalézt v [5].

2.4 Konstrukce dosažitelných konfigurací

Pro výpočet $R^*(\phi)$, kde ϕ je regulární množina konfigurací, se provede modifikace matice, která byla ukázaná v předchozím odstavci. Předpokládejme, že regulární množina konfigurací ϕ je přijímaná konečným automatem. Nahrazení se provede pro běh v prvním řádku matice během pro ϕ (viz. [5]).

Reference

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press and McCraw-IIill Book Company, 6th edition, 1990.

[2] M. Dam. *Temporal logic, automata and classical theories.* Notes for the Sixth European Summer School in Logic, Language and Information, Copenhagen 1994.

[3] S. Safra. *Complexity of Automata on Infinite Objects.* Ph.D. thesis, The Weizman Institute of Science, Rehovot, Israel, March 1989..

[4] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, London, England, 2000.

[5] A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili *Regular Model Checking.* In Proc. 12th Int. Conf. on Computer Aided Verification, Lecture Notes in Computer Science, Springer-Verlag, 2000.