Vybrané problémy z oblasti formální verifikace

Autor: Martin Kot

Školitel: prof. RNDr. Petr Jančar, CSc.

Autoreferát k dizertační práci
Fakulta elektrotechniky a informatiky
VŠB - Technická univerzita Ostrava
2009

Abstrakt

Tato disertační práce prezentuje výsledky dosažené autorem v oblasti verifikace systémů. Jedna část práce se zaměřuje na otázky složitosti problémů ověřování ekvivalencí (equivalence checking), tzn. rozhodování behaviorálních ekvivalencí na přechodových systémech. Druhá část práce se zabývá praktickým využitím ověřování modelů (model checking) na real-time databázových systémech. Ověřování modelů znamená rozhodování platnosti formulí v nějaké temporální logice popisujících nějakou vlastnost systému.

V první části je uvedeno několik výsledků týkajících se rozhodování bisimulační ekvivalence. Všechny tyto výsledky se týkají takzvaných základních paralelních procesů (Basic Parallel Processes – BPP). První prezentovaný algoritmus pracující v čase $O(n^4)$ rozhoduje bisimulační ekvivalenci mezi BPP a konečně stavovým systémem. Druhý algoritmus rozhoduje bisimulační ekvivalenci mezi dvěma normovanými BPP v čase $O(n^3)$. Třetí algoritmus rozhoduje v polynomiálním prostoru pro BPP, jestli existuje nějaký bisimulačně ekvivalentní konečně stavový systém. Tento problém se nazývá regularita BPP a spolu s dříve známou PSPACE-obtížností dostáváme jeho PSPACE-úplnost. Poslední prezentovaný algoritmus z oblasti ověřování ekvivalencí rozhoduje bisimulační ekvivalenci mezi normovaným BPP a normovaným BPA systémem. Tento algoritmus je polynomiální, jeho podrobnější analýza vede k odhadu $O(n^7)$.

Druhá část práce ukazuje nějaké možnosti, jak může být použit verifikační nástroj Uppaal pro modelování a verifikaci real-time databázových systémů. Prezentované modely se zaměřují na řízení souběžného přístupu k datům, které je v databázích používáno pro zabránění nekonzistence v případě paralelního zpracovávání více transakcí současně. Jsou uvedeny modely několika (známých) variant pesimistických a optimistických protokolů a následně je na těchto modelech ověřeno několik jednoduchých, po protokolech vyžadovaných, vlastností vyjádřených ve formě formulí temporální logiky.

Abstract

The thesis presents results obtained by the author in the area of verification of systems. One part of the thesis concentrates on questions of complexity of equivalence checking, , i.e., of deciding behavioral equivalences on transition systems. The other part concentrates on practical use of model checking on real time database systems. Model checking means deciding validity of temporal logic formulae which express properties of a system.

In the first part, several results on deciding bisimulation equivalence are shown. All these results concern with so called Basic Parallel Processes (BPP). The first of presented algorithms decides bisimulation equivalence between a BPP and a finite-state system. There is also presented time complexity analysis of this algorithm which shows up an upper bound $O(n^4)$. The second algorithm decides bisimulation equivalence between two normed BPPs in $O(n^3)$. The third algorithm decides for a given BPP whether there exists some equivalent finite-state system with respect to bisimulation equivalence. This problem is called Regularity of BPP. Presented algorithm works in polynomial space which, together with previously known PSPACE-hardness of regularity of BPP, gives PSPACE-completeness of this problem. The last presented equivalence checking algorithm decides bisimulation equivalence between a normed BPP and a normed BPA system. This algorithm is polynomial, its detailed analysis leads to an upper bound $O(n^7)$.

The second part of the thesis shows some possibilities how verification tool Uppaal can be used on modeling and verification of real-time database systems. Presented models are focused on concurrency control used in databases to avoid inconsistency when several transactions can be executed in parallel. There are models of several well known variants of pessimistic and optimistic protocols presented and some simple demanded properties of those protocols expressed as temporal logic formulae are checked on the models.

Shrnutí obsahu dizertační práce

V dnešní době jsou softwarové a hardwarové systémy všudypřítomné, rozsáhlé a složité a každá chyba v nich může mít vážné a drahé následky. V systémech pro řízení dopravy, řídících systémech atomových elektráren apod. může chyba způsobit ztrátu mnoha životů. Dalším případem systémů, kde bychom se rádi vyhnuli chybám, jsou operační systémy, síťové komunikační protokoly, mikroprocesory a jiné čipy, automobilové systémy a mnoho jiných. Velké úsilí se tedy věnuje zajištění bezchybnosti takových systému. Bezchybností se obvykle myslí, že *implementace* systému se chová přesně podle popisu ve *specifikace* požadovaného chování. Proces ověřování, jestli implementace odpovídá specifikaci, se nazývá *verifikace*.

Nejrozšířenější techniky pro verifikaci jsou testování a simulace. Testování znamená, že systém běží se zvolenými vstupy a kontroluje se jeho chování. Existuje mnoho možností, jak volit vstupy pro testování, např. náhodné hodnoty, všechny možné hodnoty, krajní hodnoty apod. Při simulaci je postup podobný, neprobíhá ale přímo na systému samotném, nýbrž na nějakém modelu systému. Výhodou simulace je, že zkušební běhy modelu mohou být levnější, jednodušší a často i rychlejší. Nevýhodou je, že reálný systém může obsahovat chyby, které v modelu nejsou, a tedy je simulace nemůže odhalit.

Testování i simulace se mohou používat během téměř všech fází vývoje a mohou odhalit velké množství chyb v systému. V počátečních fázích jsou velmi efektivní, protože chyb bývá velké množství. Ale jejich efektivita rychle klesá s tím, jak se množství chyb v systému zmenšuje. Potom potřebují velké množství času na odhalení každé další chyby. Jejich společnou velkou nevýhodou je to, že obvykle nemohou zaručit bezchybnost v každé možné situaci. Počet vstupů, možných interakcí s okolím apod. je obvykle tak velký (a často nekonečný), že není možné všechny vyzkoušet během testování a si-

mulace. Tento problém je ještě větší v případě systémů složených z několika paralelně pracujících komponent, které jsou v dnešní době velmi časté. Interakce mezi těmito souběžně běžícími komponentami může znamenat navenek nederministické chování celého systému jako celku. Počet možných chování roste velmi rychle s každou přidanou komponentou a může být velmi obtížné i pouze reprodukovat chyby v těchto systémech, protože ty mohou nastat jen za nějakých unikátních okolností.

Alternativou k testování a simulaci jsou *formální metody* nebo *formální verifikace*, které prochází všechna možná chování pro zajištění bezchybnosti.

Formální metody nám poskytují teoretické prostředky pro konstrukci přesného matematického důkazu bezchybnosti systému. Ten může být proveden ručně, což je velmi pracné a náchylné k chybám, nebo proveden s pomocí nějakého softwarového nástroje. Ten druhý přístup se nazývá *počítačem podpořená verifikace* a je obvykle efektivnější. Problémem ale je, že tento proces nemůže být plně automatický v celé obecnosti, protože mnoho problémů týkajících se chování počítačových programů je nerozhodnutelných. Např. i tak jednoduchá otázka, jestli se program nakonec zastaví, je známý nerozhodnutelný problém (zvaný problém zastavení nebo Halting problem).

Obecně existují tři hlavní přístupy k verifikaci, které umožňují zajistit bezchybnost pro všechna možná chování systému:

- Dokazování vět – Theorem proving
- Ověřování ekvivalencí – Equivalence checking
- Ověřování modelů – Model checking

Theorem proving je založen na konstrukci formálního důkazu bezchybnosti systému. Při této konstrukci může pomáhat softwarový nástroj zvaný *theorem prover*. Tyto nástroje vyžadují vedení uživatelem při provádění klíčových kroků důkazů, samy většinou jen pomáhají s nějakými teoreticky jednoduchými, ale pro člověka pracnými kroky. Hlavní odpovědnost za vedení důkazu je ale na uživateli, což po uživatelích vyžaduje velké znalosti, schopnosti a zkušenosti.

Model checking i equivalence checking mohou být plně automatizované a nevyžadují velkou interakci s uživatelem. Ale tyto metody nemohou být použity na libovolné systémy kvůli nerozhodnutelnosti ověřovaných vlastností. Proto se obvykle neověřují celé počítačové programy, ale modely těchto systémů

vytvořené v nějakém formalismu, který nemá plnou vyjadřovací sílu turingových strojů. Tyto techniky jsou založeny na teorii automatů a formálních jazyků, protože tato teorie nabízí prostředky pro konečný popis nekonečných jazyků a mnoho vlastností jazyků je rozhodnutelných.

V případě *equivalence checking* otázka zní, jestli dva systémy (nebo jejich popisy) jsou v nějakém smyslu ekvivalentní. Obvykle se porovnává, jestli specifikace a implementace mají stejné (nebo ekvivalentní) chování.

Model checking je založen na tom, že máme jen jeden systém (nebo jeho popis) a nějakou požadovanou vlastnost vyjádřenou ve formě formule nějaké (temporální) logiky a cílem algoritmů a nástrojů je ověřit, že systém splňuje danou formuli, tedy má požadovanou vlastnost. V praxi je ekvivalence checking používanější v oblasti tvorby hardwarových obvodů a čipů a model checking se častěji užívá pro verifikaci softwarových systémů. Další informace o model checking a temporálních logikách je možno nalézt v [2, 3, 17, 1].

Systémy mohou být pro potřeby ověřování modelů nebo ekvivalencí popsány mnoha různými způsoby a formalismy. Modely s větší vyjadřovací schopností nemohou být verifikovány automaticky a příliš omezené modely zase nemohou popsat mnoho aspektů skutečných systémů. Také vlastnosti pro model checking mohou být vyjádřeny v mnoha variantách logik a existuje i mnoho možných ekvivalencí pro equivalence checking. Kombinace těchto možností tvoří velké množství verifikačních problémů. Výzkum se, mimo jiné, zaměřuje na rozhodnutelnost a složitost těchto problémů.

Jedna z oblastí výzkumu se zabývá otázkami, které problémy z oblasti ověřování modelů a ekvivalencí jsou rozhodnutelné, a kde přesně leží hranice mezi rozhodnutelnými a nerozhodnutelnými problémy. Další důležitou otázkou je, jaká je přesná výpočetní (časová nebo prostorová) složitost rozhodnutelných verifikačních problémů. Některé verifikační problémy mohou teoreticky být řešeny automaticky počítačem, ale prakticky je to možné jen pro malé instance právě díky příliš velké výpočetní složitosti. Jedním ze známých jevů, který komplikuje návrh verifikačních nástrojů, je tzv. *stavová exploze*. Tento problém se vyskytuje, když několik komponent se zvládnutelně malým stavovým prostorem tvoří systém, který jako celek může mít stavový prostor exponenciálně velký vzhledem k velikosti komponent. Tento problém je v některých případech nevyhnutelný, ale také existují techniky, které v jiných případech stavové explozi zabrání nebo si s ní alespoň do nějaké míry poradí.

V dizertační práci se soustředím na dva, docela rozdílné, typy problémů. V první části se zabývám složitostí některých problémů z oblasti equivalence

checking a prezentuji nějaké výsledky, které jsem na toto téma publikoval. V druhé části se zabývám ověřováním modelů. Konkrétně jde o možnosti použití verifikačního (softwarového) nástroje Uppaal na ověřování bezchybnosti real-time databázových systémů. Některé prezentované výsledky byly dosaženy spoluprací s Petrem Jančarem a Zdeňkem Sawou a společně s nimi jako spoluautory i publikovány.

Cíle práce

Dizertační práce má dva hlavní cíle. Prvním je přispět nějakými novými výsledky v oblasti ověřování ekvivalencí. Otázkám rozhodnutelnosti a složitosti rozhodování různých ekvivalencí mezi různými typy systémů se věnuje ve výzkumu poměrně hodně pozornosti. Stále ale existuje mnoho otevřených problémů, mnoho horních odhadů složitosti je zbytečně hodně nadsazených apod. Cílem práce tedy je zpřesnit horní složitostní odhady vybraných problémů souvisejících s bisimulační ekvivalencí na základních paralelních procesech.

Druhý cíl je něco jako případová studie využití verifikačního nástroje Uppaal na protokoly pro řízení souběžného přístupu k datům v real-time databázových systémech. Před započetím mé práce na tomto tématu bylo publikováno jen pár skromných pokusů o využití metod formální verifikace na real-time databáze. Existující verifikační nástroje nemají přímou podporu pro databázové systémy. Cílem práce je proto ukázat, že i tak je možno verifikační nástroje použít a ověřit některé požadované vlastnosti databázových systémů. Protože část zabývající se řízením souběžného přístupu k datům patří mezi nejdůležitější, byly protokoly používané pro řízení souběžného přístupu zvoleny pro ukázku možností verifikačního nástroje. Z nástrojů byl zvolen Uppaal, protože má podporu pro čas a real-time systémy.

Většina lidí zabývajících se oblastí real-time databází nemá zkušenosti s verifikačními nástroji a nezná jejich možnosti. Proto při konzultacích se mnou nebyli schopni určit rozumné vlastnosti, které by potřebovali ověřit a opravdu ověřitelné verifikačními nástroji byly, i když projevili o spojení verifikace a databází zájem. Proto bylo mým cílem nejprve prozkoumat možnosti modelování různých verzí protokolů pro souběžný přístup k datům, navrhnout různá zjednodušení a abstrakce, aby modely nebyly příliš složité (a tedy nezvládnutelné pro verifikaci v Uppaalu), ale přesto podchycovaly důležité vlastnosti. To by mělo ukázat cestu autorům nových protokolů pro ověření jejich správnosti. A hlavně na těchto modelech a jednoduchých ověřených

dotazech bude možno v dalších fázích výzkumu databázistům demonstrovat schopnosti a možnosti verifikačních nástrojů a dále pracovat na ověřování vlastnosti, které jsou opravdu z praxe důležité.

Přehled dosažených výsledků

Následující podsekce stručně popisují hlavní výsledky prezentované v práci.

Složitost vybraných problémů týkajících se bisimulační ekvivalence a BPP

Systémy při ověřování ekvivalencí mohou být popsány mnoha různými způsoby. Některé možné formalismy pro popis verifikovaných systémů byly seřazeny do tzv. (α, β)-PRS hierarchie ([14]). V práci se zabývám hlavně třemi nejjednoduššími třídami ze spodní části této hierarchie - konečně stavovými systémy, základními paralelními procesy a základními procesními algebrami.

Intuitivně si můžeme tyto třídy představovat následovně (přesné definice jsou uvedeny v dizertační práci). Konečně stavové systémy (Finite state systems – FS) jsou, jak název napovídá, systémy s (explicitně nebo implicitně) daným konečným počtem stavů. Stav systému může být změněn pomocí nějakého přechodu. Základní procesní algebry (Basic process algebra – BPA) mohou modelovat jednoduché sekvenční systémy s rekurzivním voláním procedur. Stav BPA systému je dán jako obsah zásobníku a chování je dáno konečnou množinou pravidel popisujících, jak se může změnit symbol na vrcholu zásobníku. Základní paralelní procesy (Basic parallel processes – BPP) umožňují modelovat jednoduchý paralelismus bez možnosti komunikace mezi paralelními komponentami. Stav BPP systému je multimnožina symbolů a chování je dáno množinou pravidel popisujících, jak jeden prvek této multimnožiny může být vyměněn za jiné prvky. Často jsou přechody mezi stavy (ve všech uvedených třídách) doprovázeny nějakou navenek viditelnou akcí z předem dané konečné množiny akcí. Systémy, kde přechody nemají přiřazeny akce, nebo umožňují i navenek neviditelnou akci, se v práci nezabývám.

Systémy spadající do všech tří uvedených tříd mohou být normované. To znamená, že z každého stavu je možné dosáhnout nějakou konečnou posloupností kroků speciální "prázdný stav", z nějž již není možné provedení žádného dalšího kroku.

Existuje mnoho ekvivalencí vhodných pro ekvivalence checking. V práci se zaměřuji na jednu z nejznámějších a nejdůležitějších z nich - bisimulační ekvivalenci. Tuto ekvivalenci je možné neformálně chápat takto: kdykoliv jeden ze dvou ekvivalentních stavů umožňuje použitím nějakého pravidla s nějakou akcí změnit stav, musí být druhý z těchto ekvivalentních systémů být schopen také podle nějakého pravidla se stejnou akcí změnit stav tak, aby oba cílové stavy byly opět spolu ekvivalentní. Přehled známých výsledků týkajících se rozhodování bisimulační ekvivalence na třídách (α, β)-PRS hierarchie, spolu s odkazy na příslušné publikace, je možné najít v [16]. Tato přehledová publikace je dostupná i online, kde je aktualizována o nové výsledky (např. i o některé prezentované v této dizertační práci).

V práci uvádím některé vlastní (případně se spoluautory) dosažené výsledky týkající se BPP, BPA a FS a bisimulační ekvivalence. Nejprve jde o algoritmus rozhodující bisimulační ekvivalenci mezi jedním konečně stavovým systémem a jedním BPP systémem. Analýza výpočetní složitosti tohoto algoritmu vedla k hornímu časovému odhadu $O(n^4)$. Tento výsledek byl publikován v [13].

Dále popisuji algoritmus rozhodující bisimulační ekvivalenci mezi dvěma normovanými BPP systémy. Čas běhu tohoto algoritmu je shora odhadnut $O(n^3)$. Toto bylo publikováno v [6]. Pro tento problém již byl znám polynomiální algortitmus založený na jiných myšlenkách ([4]), ale nebyla provedena přesná analýza pro stanovení stupně polynomu, a po jeho prozkoumání to nevypadá, že by odhad $O(n^3)$ taky umožňoval (bez nějakých podstatných vylepšení). Oba v práci uvedené algoritmy se inspirovaly technikou založenou na tzv. DD-funkcích zavedenou poprvé v [5] v algoritmu pro rozhodování bisimulační ekvivalence mezi dvěma BPP systémy. Základním principem je, že se postupným zjemňováním provádí rozklad množiny přechodových pravidel mezi stavy (v našem případě jde o přechody v Petriho síti), a problém ekvivalence dvou stavů se převede na problém, jestli dvě speciálně vytvořená pravidla jsou ve stejné třídě finálního rozkladu.

Třetím prezentovaným výsledkem v této oblasti (publikovaným v [9]) je algoritmus rozhodující pro BPP systém, jestli existuje nějaký s ním bisimulačně ekvivalentní konečně stavový systém. Tento problém se nazývá regularita BPP a byla známa jeho PSPACE obtížnost ([15]). V práci uvedený algoritmus pracuje v polynomiálním prostoru, čímž dostáváme PSPACE úplnost daného problému.

Poslední v práci uvedený výsledek z oblasti ověřování ekvivalencí je algoritmus rozhodující bisimulační ekvivalenci mezi normovaným BPA procesem

a normovaným BPP procesem. Pro tento problém byl dříve znám exponenciální algoritmus ([18]), ale my jsme v [8, 7] představili polynomiální algoritmus, jehož podrobnější analýza výpočetní složitosti vedla k odhadu $O(n^7)$. Prvním krokem našeho algoritmu je převedení normovaného BPP systému do speciálního tvaru, kde bisimulační ekvivalence odpovídá identitě mezi stavy. Tento krok je proveden v čase $O(n^3)$ a je využitelný i samostatně (např. jako podprocedura v jiných algoritmech pracujících s normovanými BPP). Dále ověřujeme v čase $O(n^3)$, jestli k danému normovanému BPP vůbec existuje nějaké ekvivalentní BPA. V záporném případě náš algoritmus končí, protože konkrétní BPA systém jistě nemůže být ekvivalentní s daným BPP systémem, když s tímto BPP není ekvivalentní žádný BPA systém. Pokud nějaký ekvivalentní BPA systém k danému BPP existuje, tak jej zkusíme sestrojit. Obecně takto sestrojený BPA systém může být exponenciálně velký vzhledem k BPP, ke kterému se konstruuje. Ale hlavní částí našeho výsledku je důkaz, že pokud je sestrojovaný BPA větší než $4n^2$ (kde n je součet velikostí BPA a BPP systému), tak systémy dané v instanci problému nejsou ekvivalentní. Pokud se BPA podaří úspěšně sestrojit, je posledním krokem ověření ekvivalence mezi BPA z instance a sestrojeným BPA systémem. Toto by mohlo být provedeno nějakým v literatuře dříve publikovaným algoritmem. Pro dosažení meze $O(n^7)$ ale navrhujeme vlastní algoritmus, který využívá toho, že sestrojený BPA systém se blíží konečně stavovému systému. Vedlejším produktem tohoto algoritmu je také algoritmus rozhodující bisimulační ekvivalenci mezi normovaným BPA systémem a konečně stavovým systémem v čase $O(n^4)$.

Modelování a verifikace real-time databázových systémů v Uppaalu

Real-time databázové systémy jsou založeny na technikách a algoritmech známých z klasických databázových systémů. Navíc k efektivnímu ukládání dat, vyhodnocování databázových dotazů apod. ale poskytují jisté meze na dobu reakce na požadavky, což je velmi důležité v real-time prostředí. K tomuto účelu využívají priority, lhůty (deadline) a další mechanismy.

Řízení souběžného přístupu k datům je jednou z nejdůležitějších částí databázových systému, které umožňují souběžný přístup více transakcí k datům. V real-time databázích se používají modifikované protokoly pro řízení souběžného přístupu známé z klasických databází. Modifikace jsou potřebné, aby protokoly dodržovaly priority a transakce byly dokončovány v určených lhůtách.

Snažil jsem se využít model checking k verifikaci některých důležitých vlastností těchto protokolů pro souběžný přístup používaných v real-time databázích. Využil jsem k tomu existující, zdarma dostupný, softwarový verifikační nástroj Uppaal, který je vyvíjen ve spolupráci dvou univerzit – v Uppsale a v Aalborgu. Modely pro tento nástroj se popisují formou sítě časovaných automatů. Vlastnosti se poté popisují ve formě formulí temporální logiky a Uppaal je umí automaticky na modelech ověřovat. Hlavní důraz mé práce byl kladen na možnosti vytváření modelů protokolů, verifikované vlastnosti jsou spíše pro ilustraci.

Nejpoužívanější protokoly pro řízení souběžného přístupu k datům je možno rozdělit do dvou skupin – na pesimistické a optimistické. Pesimistické protokoly jsou založeny na dvoufázovém zamykání dat - transakce nejprve musí získat zámky na všechny záznamy, se kterými bude pracovat, potom provede všechny akce a nakonec všechny zámky uvolní. V základní verzi hrozí uváznutí, proto je navrženo rozšíření, kdy transakce čekající na přidělení zámku příliš dlouho je restartována. Pro respektování priorit transakcí byla navržena verze High priority, kdy transakce žádající obsazený zámek může na základě vyšší priority způsobit restart transakce držící tento zámek. Optimistické protokoly pracují tak, že změny do databáze jsou natrvalo uloženy až po proběhnutí tzv. validační fáze, kde se ověří, jestli nedošlo ke konfliktu. Opět existuje několik verzí, např. Broadcast commit, kde validující transakce je vždy dokončena a všechny konfliktní transakce jsou restartovány, nebo Sacrifice, kdy je z konfliktních transakcí restartována ta s nižší prioritou.

Modely jsou navrženy ze dvou různých pohledů. Nejprve jsem se snažil namodelovat experimentální databázový systém V4DB s jeho hlavními částmi, přičemž část věnovaná souběžnému přístupu byla namodelovaná podrobněji a ostatní části byly mnohem více zjednodušené. Namodelována byla verze s využitím pesimistického protokolu založeného na dvoufázovém zamykání dat a s využitím optimistického protokolu Sacrifice. Tyto modely byly publikovány v [10].

Druhý přístup je založen na tom, že se modelují protokoly samotné. Tím se modely zjednoduší a Uppaal dokáže ověřit více různých (i složitějších) vlastností. Při tomto přístupu byly namodelovány 3 verze pesimistického protokolu založeného na dvoufázovém zamykání dat (základní, doplněná o limity na délku čekání na zámek a verze High priority) a dvě varianty optimistických protokolů – Broadcast commit a Sacrifice. Tento přístup byl publikován v [11] a [12].

Prezentované modely samozřejmě nejsou jediné možné a ani není možné říct, že by byly v nějakém ohledu nejlepší. Ale jsou v nich představeny některé myšlenky a možné způsoby abstrakce, které mohou pomoci při modelování jiných částí real-time databází, při modelování a verifikaci nově navržených protokolů pro řízení souběžného přístupu k datům apod.

Dodatek A

Seznam publikací

Následující seznam obsahuje recenzované publikace autora v chronologickém pořadí.

- Jančar, P., Kot, M., *Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$*, in: *Proceedings of the Third International Workshop on Automated Verification of Infinite-State Systems - AVIS2004, Barcelona, 2004* [6]
- Kot, M., *Some Problems Related to Bisimilarity on BPP*, in: *Proceedings of Movep'04, Universit Libre de Bruxelles, Brussels, 2004, p. 96-102*
- Kot, M., Sawa, Z., *Bisimulation equivalence of a BPP and a finite-state system can be decided in polynomial time*, in: *Electronic Notes in Theoretical Computer Science, 2005, vol. 138, issue 3 (Proceedings of the 6th International Workshop on Verification of Infinite-State Systems –INFINITY 2004), p. 49-60, ISSN 1571-0661* [13]
- Kot, M., *Complexity of some Bisimilarity Problems between BPP and BPA or Finite-State System*, in: *Proceedings of Movep'06, University of Bordeaux-1, Bordeaux, 2006, p. 318-323*
- Kot, M., *Notes on Modeling of Real-Time Database System V4DB in Verification Tool Uppaal*, in: *MEMICS proceedings (MEMICS 2007 - Third Doctoral Workshop on Mathematical and Engineering Methods in Computer Science), Brno, 2007, p. 82-89, ISBN 978-80-7355-077-6* [10]

- Jančar, P., Kot, M., Sawa, Z., *Normed BPA vs. Normed BPP Revisited*, in: *Proceedings of CONCUR 2008 - Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, Lecture Notes in Computer Science 5201, Springer, 2008, p. 434-446, ISBN 978-3-540-85360-2, ISSN 0302-9743 (Print), ISSN 1611-3349 (Online)*, Best paper award [8]

- Kot, M., *Modeling Real-Time Database Concurrency Control Protocol Two-Phase-Locking in Uppaal*, in: *Proceedings of the International Multiconference on Computer Science and Information Technology, Volume 3 (2008), IEEE Computer Society Press, 2008, p. 673-678, ISBN 978-83-60810-14-9, ISSN 1896-7094* [11]

- Kot, M., *Modeling selected real-time database concurrency control protocols in Uppaal*, in: *Innovations in Systems and Software Engineering, Volume 5, Number 2, June 2009, Springer, London, p. 129-138, ISSN 1614-5046 (Print), ISSN 1614-5054 (Online)* [12]

- Jančar, P., Kot, M., Sawa, Z., *Complexity of Deciding Bisimilarity between Normed BPA and Normed BPP*, in: *Information and Computation, Elsevier, 28 p., ISSN 0890-5401, to appear* [7]

Některé výsledky byly prezentovány také na studentském workshopu Wofex:

- Kot, M., Onderek, O., *Two known algorithms for checking bisimilarity of normed BPPs*, in: *Sborník semináře Wofex 2003, Ostrava, 2003*

- Kot, M., *Complexity of deciding bisimilarity of nBPP and bisimilarity of BPP with finite-state system*, in: *Proceedings of the 2nd annual workshop WOFEX 2004, Ostrava, 2004, p. 310-315, ISBN 80-248-0596-0*

- Kot, M., *Regularity of BPP is PSPACE-complete*, in: *Proceedings of the 3nd annual workshop WOFEX 2005, Ostrava, 2005, p. 393-398, ISBN 80-248-0866-8* [9]

Literatura

[1] Bérard, B., M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen and P. McKenzie, "Systems and Software Verification: Model-Checking Techniques and Tools," Springer, 2001.

[2] Clarke, E. M., O. Grumberg and D. A. Peled, "Model Checking," The MIT Press, 1999.

[3] Emerson, E. A., *Temporal and modal logic*, Handbook of Theoretical Computer Science **B** (1991), pp. 995–1072.

[4] Hirshfeld, Y., M. Jerrum and F. Moller, *A polynomial-time algorithm for deciding bisimulation equivalence of normed basic parallel processes*, Mathematical Structures in Computer Science **6** (1996), pp. 251–259.

[5] Jančar, P., *Strong bisimilarity on basic parallel processes is PSPACE-complete*, in: *Proc. 18th LiCS* (2003), pp. 218–227.

[6] Jančar, P. and M. Kot, *Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$*, in: *Proceedings of the Third International Workshop on Automated Verification of Infinite-State Systems – AVIS 2004*, 2004, p. 9.

[7] Jančar, P., M. Kot and Z. Sawa, *Complexity of deciding bisimilarity between normed BPA and normed BPP*, Information and Computation, Elsevier (to appear).

[8] Jančar, P., M. Kot and Z. Sawa, *Normed BPA vs. normed BPP revisited*, in: *Proceedings of CONCUR 2008*, Lecture Notes in Computer Science **5201** (2008), pp. 434–446.

[9] Kot, M., *Regularity of BPP is PSPACE-complete*, in: *Proceedings of the 3rd annual workshop WOFEX 2005* (2005), pp. 393–398.

[10] Kot, M., *Notes on modeling of real-time database system V4DB in verification tool Uppaal*, in: *MEMICS proceedings (MEMICS 2007 - Third Doctoral Workshop on Mathematical and Engineering Methods in Computer Science)* (2007), pp. 82–89.

[11] Kot, M., *Modeling real-time database concurrency control protocol two-phase-locking in Uppaal*, in: *Proceedings of the International Multiconference on Computer Science and Information Technology* (2008), pp. 673–678.

[12] Kot, M., *Modeling selected real-time database concurrency control protocols in Uppaal*, Innovations in Systems and Software Engineering **5** (2009), pp. 129–138.

[13] Kot, M. and Z. Sawa, *Bisimulation equivalence of a BPP and a finite-state system can be decided in polynomial time*, Electronic Notes in Theoretical Computer Science **138** (2005), pp. 49–60, proceedings of Infinity 2004).

[14] Mayr, R., *Process rewrite systems*, Information and Computation **156** (2000), pp. 264–286.

[15] Srba, J., *Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard*, in: *Proc. STACS'02*, LNCS **2285** (2002), pp. 535–546.

[16] Srba, J., *Roadmap of infinite results*, in: *Current Trends In Theoretical Computer Science, The Challenge of the New Century, Volume 2: Formal Models and Semantics*, World Scientific Publishing Co., 2004 pp. 337–350, (See an updated version at http://www.brics.dk/∼srba/roadmap/).

[17] Stirling, C., *Modal and temporal logics*, Handbook of Logic in Computer Science **2** (1992), pp. 477–563.

[18] Černá, I., M. Křetínský and A. Kučera, *Comparing expressibility of normed BPA and normed BPP processes*, Acta Informatica **36** (1999), pp. 233–256.