```
Mieke Massink schrieb:
> A general approach to defining behavioural relations over
> processes as the maximal pre--congruences induced by basic observables
> can be summarised as follows:
> Given a language equipped with a reduction relation:
> 1.  Define a set of observables (values, normal forms, \ldots)
>       to which a program can evaluate by means  of successive reductions.
> 2.  Define a basic preorder over terms by stating that
>       a term is less defined than  another if it exhibits
>       a smaller  set of basic observables.
> 3. Consider the largest pre--congruence over the
>    language induced by the basic preorder.
>
> In the non-probabilistic setting this very general approach has lead to
> alternative formalisations of well-known pre-congruences such as the
> testing pre-congruence of De Nicola and Hennessy, but has also lead
> to new interesting pre-congruences such as the safe-must pre-order.
> Essentially, in this approach basic observables
> are used to provide information about the initial communication
> capabilities of processes and the preorders were obtained by
> observing processes within all possible contexts.
>
> Given the success in the non-probabilistic setting, we are currently
> adapting the approach to study models of probabilistic concurrent systems
> and their equivalences. The basic observables that we will use for our
> probabilistic setting will instead provide information on the probability
> that a process communicates along specific channels (the probability of
> observing a specific action depends on  the sequence of non-deterministic
> choices between several outgoing transitions).
>
> For the approach outlined above, the choice of the language and the choice
> of the basic observables are crucial ones. As base language for defining
> context and processes, we so far considered a simple variant of CCS,
> the probabilistic process calculus PCCS as introduced by
> Baier with an operational semantics based on the
> probabilistic automaton model. PCCS is a probabilistic extension of a
> significant subset of CCS.
>
> So far, we defined a probabilistic basic observation pre-order over
> PCCS terms and we showed that the induced congruence coincides with a
> probabilistic extension of the weak may testing preorder of De Nicola
> and Hennessy. Of course, there are many more open questions:
>
> 1) Can the approach be extended to probabilistic must testing as well?
> 2) What would be the most suitable base language to base the theory on?
> 3) What are in fact the most suitable and realistic probabilistic
> basic observables? I.e. what can be really observed of a probabilistic
> system in practice?
> 4) Can similar approaches be used also in a stochastic setting?
>
>

Maria Mateescu schrieb:
>
> 1. Are there any interesting properties that can not be expressed in PCTL?
>
> 2. Given a labelled Markov chain $A$, with 3 states, which would be the
> labelled Markov chain $B$, with only 2 states, such that $A$ and $B$ are
> as "close" as possible from an observable point of view.
> By a labelled MC we refer to a MC where the states are labelled with
> obervables.


Joel Ouaknine schrieb:
> 1) The Skolem Problem.
>
> An instance of the Skolem problem is the following: given a linear recurrence
> relation (of the form a_{n+k} = f(a_{n}, a_{n+1}, ..., a_{n+k-1}), where f is
a
> linear function) together with initial values a_0, a_1, ..., a_{k-1},
```

determine
> if the sequence <a_i> ever hits 0.
>
> It is not known in general if this problem is decidable. (It is decidable for
> recurrence depths of up to 5, but even this is highly non-trivial.)
>
> This (the question of decidability) is a long-standing problem, dating back
> possibly 70 years. See Vesa Halava, Tero Harju, Mika Hirvensalo and Juhani
> Karhumki, "Skolem's Problem - On the Border between Decidability and
> Undecidability", Technical Report 683, TUCS, Apr 2005, for a comprehensive
> survey.
>
> This problem is very interesting in its own right, and has several connections

> to various parts of mathematics (e.g. number theory, dynamical systems) and
> computer science (e.g. termination of linear programs). There are also strong
> connections to probabilistic systems; for example, an open model-checking
> problem raised by D. Beauquier, A. Rabinovich and A. Slissenko in "A logic of
> probability with decidable model-checking", CSL 2002, reduces to a special
case
> of the Skolem problem.
>
> ------------------------
>
> 2) The Probabilistic Contextual Equivalence Problem.
>
> Much work has been done in modelling and model checking probabilistic programs

> by deriving and analysing associated Markov chains obtained straightforwardly
> through an operational semantics, or more generally through a probabilistic
> state-transformer semantics. Certain problems, however, are more suitable to
be
> handled via equivalence checking than via model checking; a key example is
that
> of anonymity, in which most interpretations in the literature require an
> observer not to be able to distinguish between several versions of a given
> process.
>
> The (meta) problem we ask is this: given a probabilistic programming language,

> an instance of the contextual equivalence problem for this language consists
of
> two programs P1 and P2 potentially having free identifiers. The question is:
> are P1 and P2 are contextually equivalent? In other words is it the case that
> for all programming contexts C[-] (that have a 'hole' in it), C[P1] and C[P2]
> are observationally undistinguishable?
>
> We believe the probabilistic contextual equivalence problem (in particular,
its
> decidability) is potentially very interesting for a range of probabilistic
> programming languages.
>
> ------------------------
>
> 3) Minimisation of Probabilistic Automata.
>
> Given a (Rabin) probabilistic automaton A, are there good notions of the
> minimal automaton A' that is equivalent to A? If so, is it computable? What of

> the complexity of this procedure? And if the latter is high, are there
> relatively cheap `good approximations'?
>
> ------------------------
>
> 4) Approximation of Probabilistic Automata.
>
> Define good distance metrics between (Rabin) probabilistic automata. Are these

> computable/decidable? Do relatively cheap `good approximations' to such
metrics

> exist?
>
> -----------------------
>
> 5) Randomised Algorithms in Verification.
>
> Verification has traditionally relied almost exclusively on deterministic (as
> opposed to probabilistic) algorithms. Is there scope for randomised algorithms

> in verification?
>
> -----------------------
>
> 6) Decidability via Probability.
>
> The theory of infinite-state systems abounds with undecidable problems. There
> are several instances, however, in which a `probabilistic' version of an
> infinite-state problem becomes decidable -- see, e.g.,
>
> a) Eugene Asarin, Pieter Collins: Noisy Turing Machines. ICALP 2005:
1031-1042,
>
> b) P. A. Abdulla, N. Bertrand, A. Rabinovich and Ph. Schnoebelen. Verification

> of Probabilistic Systems with Faulty Communication. Information and
Computation
> 202(2), pages 141-165, 2005.
>
> c) C. Baier, N. Bertrand and Ph. Schnoebelen. Verifying nondeterministic
> probabilistic channel systems against omega-regular linear-time properties.
> ACM Transactions on Computational Logic 9(1), 2007. To appear.
>
> Are there other interesting classes of such instances, or general underlying
> principles at work? More speculatively: given an undecidable problem, are
there
> systematic ways to obtain a decidable `probabilised' version of this problem?
>

Sergio Giro schrieb:
> Abstracts:
>
> A formalism being suitable to restrict interleavings
>
> Distributed schedulers have been introduced in order to develop
> techniques for compositional reasoning for Markov Decision Processes.
> In the formalisms in which distributed schedulers have been introduced
> so far, there are no nondeterministic choices concerning the
> interleaving explicitly (for instance, in the Switched Probabilistic
> I/O Automata, the next to component to perform an output is decided
> using a token-based
> mechanism). Distributed schedulers are used to restrict the behaviours
> of the MDPs in order to discard unrealistic behaviours. The challenge
> is to develop a formalism having explicit interleaving choices, being
> as simple and powerful as possible, in order to restrict behaviours
> arising from unrealistic interleavings.
>
> Randomization in restricted interleavings
> Distributed schedulers have been introduced in order to develop
> techniques for compositional reasoning for Markov Decision Processes.
> In the formalisms in which distributed schedulers have been introduced
> so far, there are no nondeterministic choices concerning the
> interleaving explicitly (for instance, in the Switched Probabilistic
> I/O Automata, the next to component to perform an output is decided
> using a token-based mechanism). Distributed schedulers are used to
> restrict the behaviours of the MDPs in order to discard unrealistic
> behaviours. If there are nondeterministic choices concerning
> interleaving, schedulers in which interleavings are also restricted
> can be defined. We called this schedulers "Strongly distributed
> schedulers" (SDS). An interesting property to prove about SDS is
> whether randomization gives extra power to SDS or not. The proof for

> distributed schedulers cannot be extended to the strongly distributed case. After explaining intuitively the proof for distributed schedulers, and why this proof cannot be extended, we will look for intuitions for the proof for SDS. A preliminar draft illustrating the problem can be found at
> www.famaf.unc.edu.ar/~sgiro/VOSSpuzzle.pdf

Dominik Wojtczak schrieb:

> Recursive Markov Chains are a natural abstract model of probabilistic procedural programs and other systems involving recursion and probability. They are formally equivalent to probabilistic Pushdown Systems and they define a class of infinite-state Markov chains.
>
> Informally, an RMC consists of several component Markov Chains that can call each other recursively. Each component consists of nodes and boxes with possible probabilistic transitions between them. Each box is mapped to a specific component so that every time we reach an entry of this box, we jump to the corresponding entry of the component it is mapped to. When/if we finally reach an exit node of that component, we will jump back to a respective exit of the box that we have entered this component from. This process models, in an obvious way, function invocation in a probabilistic procedural program. Every potential function call is represented by a box. Entry nodes represent parameter values passed to the function, while exit nodes represent returned values. Nodes within a component represent control states inside the function.
>
> We can extend RMCs to a model where some of the nodes are controlled by a player. Whenever we reach such a node it is the player who chooses which of the possible transitions from that node the process will take. This controlled version of RMCs is called Recursive Markov Decision Processes(RMDP) and it allows us to model nondeterministic and interactive behavior. Unfortunately most of the interesting problems, such as computing the probability of termination at a given exit, for general RMDPs was proved to be undecidable by Etessami and Yannakakis (ICALP'05). Nevertheless if we restrict our model to the case where all of the components are allowed to have just one exit (1-exit RMDPs) this problem becomes efficiently computable.
>
> Among many interesting questions one can ask for 1-exit RMDPs is a question of computing the maximum probability of reaching a given node in any context. It was shown in Brazdil, Kucera, et. al. (CONCUR'06), building on 1-RMDP qualitative termination algorithm devised by Etessami and Yannakakis (STACS'06), that we can decide in P-time if there exists a strategy for the player such that the probability of reaching that node is 1. However there are examples of 1-RMDPs such that no optimal strategy exists, although for any probability p less than 1 it is possible to construct a strategy for which the probability of reaching that node is higher that p.
>
> The quantitative version of this problem is still not known to be decidable. Namely we do not know any algorithm that could decide whether there exists a strategy under which the probability of reaching a given node is higher than a given p \in (0,1). This is the case although we have both numerical and decision algorithms for the termination problem. If the reachability question is decidable what is its complexity? Also we do not know any algorithm for deciding whether we can achieve a probability arbitrary close to 1.

Gianfranco Ciardo schrieb:

> When asked "What is today's most important research challenge in probabilistic verification?", my answer, probably biased by my own research interests, is "Approximations!".
>
> Let me explain what I mean by that.
>
> Clearly, in the strictly logical verification arena, enormous real-life systems are being modeled and analyzed using techniques such as partial

> order reductions, symbolic representations, and abstractions. The result is that important systems of practical interest are being tackled, even when their size (measured in number of reachable states in their "exact" representation) is much larger than the number of atoms in the universe, or even infinite.
>
> When we turn to probabilistic verification, however, the situation is much worse. Even for the fairly simple class of models having an underlying continuous-time Markov chain (CTMC), we can always define the model using a compact high-level formalism, and we can often generate and store the underlying CTMC using some symbolic representation (such as MTBDDs, EVBDDs, or matrix diagrams). However, when we try to solve the CTMC to study its transient or steady-state behavior, we are unable to go much beyond 1G states, and even that requires a computer with very large amounts of RAM and exceedingly long runtimes.
>
> This is because, unlike the other steps of the analysis, the "exact" numerical solution requires a vector (or more) of size equal to the number of states in the state space. Exploitation of symmetries can sometimes ease the burden, but only in the fairly rare case when the CTMC is lumpable, while Courtois-style aggregation techniques can help convergence and improve numerical stability, but, again, require a fairly strong structure in the CTMC.
>
> What is needed, at least as long as we strive to remain in the CTMC domain, is not only (1) a set of general approximation techniques that can be applied to the large underlying CTMCs arising in probabilistic verification (we have done some substantial work in this direction already), but, also, (2) a general theory of how to carry on probabilistic verification when the results from the numerical computation are known to be approximate.
>
> Indeed, once could argue that such general theory is needed in any case, since, even when we perform an "exact" numerical solution, the iterative methods employed are known to provide only a (hopefully good) approximation of the desired exact probability vector. Of course, however, the exact and approximate numerical solution fundamentally differ, at least in theory, in that the former converges to the exact probability vector, given "enough" time, while the latter converges to a vector which is, at best, "close" to the exact vector.
>
> -- Gianfranco Ciardo
>
> PS
> An even more desirable goal than "approximations" is "bounds". Unfortunately, for bounds, problem (2), how to use bounds in probabilistic verification, becomes easier, but very few practical results are known for (1), how to obtain bounds in a general setting.

Dave Parker schrieb:

> Sorry for the belated response. My suggestion for a challenging foundational question concerns verification of MDPs and, more specifically, the types of adversaries (/schedulers/policies/etc.) that are used. The basic algorithms for say PCTL model checking on MDPs compute best-/worst-case behaviour over all adversaries. Often, this is too strong and in practice you would want to restrict the power (visibility) of the adversary (an obvious example is in the domain of security protocols). There is a '99 de Alfaro paper which discusses partial-information adversaries and there is some more recent work by e.g. Nancy Lynch and co-authors. But I would say this remains an important area with useful contributions to be made.

Lucia Cloth schrieb:

> Boudewijn proposes to have a look at the Collatz Conjecture and to tackle it with the methods and tools we have. What *I* really would like to know is whether it is easier to compute the performability distribution for an MRM in steady-state than for an arbitrary initial distribution. However, I guess this question does not qualify as a 'theory puzzle' ;-).

>

Marielle Stoelinga schrieb:
> 1. There are various open issues in the area of Probabilistic Automata (PAs).
> * Weak probabilistic forward simularity is a refinement relation that
> relates states to distributions over states, but there is no symmetric
> variant (corresponding to bisimilarity) of it. * Also decidability of
> weak probabilistic forward simulation and of trace distribution
> inclusion (and equivalence) is still open.
> * The quest for a compositional trace distribution-like relation has
> not ended yet
> (dispite significant contributions in this direction.)
>
> 2. IMCs + PA: Many applications ask for a model that combines features
> from IMCs and PAs. A comprehensive theory is of utmost importance in
> order for those applications to be analysed.
>
> 3. Stochastic Interface Automata. Interfaces are a framework that
> allow one to propagate constraints on individual components to
> constraints on a system. Interface theories have been proposed for
> asynchronous, synchronous, reward-based and timed systems, but not for
> stochastic systems.
>
> 4. Compositional quantitative reasoning.
> [dAFS04] proposes trace distance/simulation distance as a quantitative
> analogon of trace inclusion and similarity. These distances are not
> compositional: it needs not be true that
> dist(P,Q) <= dist(P||R,Q||R). Some progress has been made in
> [CdAMS05], but a definitive answer is not given.

Anne Remke schrieb:
> Ich beschaeftige mich ja (noch immer) mit der 'Bottleneck Analyse in
> IEEE 802.11 Ad Hoc Networks'. Ueber die DCF wird ja in diesem Standard
> geregelt, dass jede Station, die ein Packet verschicken moechte einen
> Backoff zieht und die Station mit dem niedrigsten Backoff in jeder Runde
> sein Packet verschicken darf. Stationen die nicht senden durften,
> behalten den einmal gezogenen Backoff und zaehlen halt so lange mit
> runter, bis sie dran sind. Die Station, die senden durfte, zieht fuer
> die naechste Runde einen neuen Backoff.
>
> Die Tatsache, dass Stationen sich ihre Backoffs merken, ist sehr
> schwierig zu modelieren oder in eine einfache Formel zu packen. Bisher
> konnte ich solche Formeln nur aufstellen unter der Annahme, dass in
> jeder Runde jeder einen neuen Backoff zieht...
>
> Also waere mein Vorschlag fuer ein Puzzle, diesen Backoff Mechanismus
> (im Grunde ein einfaches Wuerfelspiel mit Gedaechtniss) zum Modelieren
> frei zu geben. Hinterher koennte man dann ja an einigen Parametern
> drehen und beobachten wie sich die Gewinnwahrscheinlichkeiten fuer die
> einzelenen Station veraendern.

Kostas Chatzikokolakis schrieb:
> This issue arises when using formalisms which express both
> nondeterministic and probabilistic behavior. In such formalisms
> (process calculi, automata, ...) it is customary to introduce the
> notion of scheduler to resolve the nondeterminism. It has been
> observed that for certain applications, notably those in security, the
> scheduler needs to be restricted so not to reveal the outcome of the
> protocol's random choices, or otherwise the model of adversary would
> be too strong even for "obviously correct" protocols. There is some work
> on models allowing to express restrictions on the power of
> the scheduler, but so far the problem has not been addressed in its
> generality and, to my knowledge, probabilistic model checking tools do
> not provide a way to state such restrictions. Considering the
> theoretical and practical interest of this issue I think it will be the
> topic of more research in the near future.
>

Bernd Becker und Ralf Wimmer schrieb:

> Formal verification and thus also probabilistic verification strives for
> formal PROOFS of correctness, properties, ... On the other hand, programs
> (and also programs for formal verification) contain bugs and may produce
> unreliable results.
>
> Acitivities to circumvent these problems in classical domains include
> "(I)LP and numerical stability", "Certificates in SAT and QBF". First
> results show that also in "Probabilistic Verification" these concerns might
> be of interest:
>
> How to balance efficiency versus reliability in probabilistic verification?
> How to efficiently compute certificates?
>

Martin Neuhäußer schrieb:

> as you know I am working in the field of continuous-time Markov
> decision processes. Hence, in my opinion, the most striking puzzle
> is to overcome the uniformization problems in model checking
> (non-uniform) CTMDPs. Your timed-reachability paper would certainly
> provide a good basis for a discussion in that respect.
>
> Another, maybe less ambitious topic would be to think about how to
> extend the exiting timed reachability algorithm for uniform CTMDPs
> to CSL model checking (and here, especially to timed until formulas
> with arbitrary time bounds [t,t'] where t > 0).

Markus Siegle schrieb:

> Es wäre aus meiner Sicht interessant, die gewinnbringende Anwendung von
> Bisimulation nochmals zu diskutieren (ich weiß, das ist nicht neu).
> Aspekte dabei: Information aus dem high-level Modell (Symmetrien,
> Regularität, andere (welche?) strukturelle Eigenschaften) so weit
> verfügbar ausnutzen. Sind auch Ansätze, die von einem flachen
> Zustandsraum ausgehen, tatsächlich praktikabel? Implementierung auf
> expliziten oder symbolischen Datenstrukturen? Was kostet die
> Bisimulation? Wie geht man vor bei "near lumpability"? Approximative
> Bisimulation, Schranken?
>
> Vielleicht auch für einige interessant: (Wie) kann man die Manipulation
> von Decision Diagrams effizient parallelisieren?
>

Annabelle McIver schrieb:

> My foundational "puzzle" is to do with the interaction of
> nondeterminism and probability in a context where the
> nondeterministic choice cannot depend on the result of a
> previous or (possibly simultaneous) probabilistic choice.
> Although there are existing formalisms which consider this
> problem, none that I know of treat the problem of refinement
> or data abstraction.
>
> This is particularly an issue if probability is used in
> modelling security applications. What I'm really after is a
> practical reasoning tool rather than say a model
> checking technique, although of course the two
> approaches are related. For practicality the semantics/logic
> must be as simple as reasonably possible!

Hichem Boudali schrieb:

> It is believed that if any of the fundamental physical constants (e.g.
> speed of light in vacuum, mass of electron, etc) is slightly (we are
> speaking of very small percentage change) changed then the story of
> our universe would have been radically differently, and most probably
> such things as stars and planets (and US!!) would not exist.
>
> Is it possible to find a "probability value" for the absence of human beings

> given the change by say for example 0.00001 percent of the elementary charge
> e (=1.602176462 x 10-19 C); and if so how would one go about
> finding this probably by involving astrophysicist, biologist, computer
> scientist, etc.

Reza Pulungan schrieb:
> The minimal representations of phase-type distributions
>
> Transient analysis of Markov chains plays an important role in
> probabilistic verification. Many measures of interest can be inferred
> from this analysis. In CSL model checking, for instance, the computation
> of the probability distribution of the time required to arrive at some
> goal states can be carried out through transient analysis by first
> making all the goal states absorbing. If all the goal states are lumped
> into a single state, the probability distribution is called a phase-type
> distribution.
>
> Such absorbing Markov chains are called the representations of the
> phase-type distributions. It was known that these representations are
> not unique: distinct absorbing Markov chains may represent the same
> distribution. Therefore, the problem of identifying and finding the
> smallest representations, namely the representations having the smallest
> size of state spaces, is interesting and important. This problem remains
> one of the most interesting theoretical research in the field of
> phase-type distributions.

Bjoern Wachter schrieb:
> Relating meta-approaches to the abstraction of probabilistic models
>
> There are at least 3 systematic "meta"-approaches to
> the abstraction of probabilistic models (the term meta is justified
> because different instances of each respective approach may vary
> siginficantly):
>
> 1) region-based abstractions as in
>   -> magnifying lens
>   -> game-based abstraction
>   -> predicate abstraction
>   they are based on partitioning the state space
>   into regions and computing a safe abstraction of the transition relation
> over the regions.
>
> 2) the abstract interpretation-based approach of Monniaux
>   is an extension of abstract interpretation (AI) to the probabilistic
> world.
>   As common in AI, the instantiation consists in the choice of an abstract
> domain.
>
> 3) "probabilistic refinement of action systems" /
>   While (1) and (2) are state-based, this approach is based on expectation
> transformers
>   Quantitative Refinement *and* Model Checking for the Analysis of
> Probabilistic Systems (A.K. McIver)
>
> How do these 3 meta-approaches relate in terms of precision, expressiveness,
> potential for automation?
> Is there a formal or conceptual relationship between them?
> I would expect (3) to be an abstraction of (1).
>

Wan Fokkink und Rena Bakshi schrieb:
> 1) What is a good notion of weak/branching bisimilarity for Markov
> Decision Processes?
>
> 2) How can probabilistic process algebra be equipped with abstract data
> types?
>
> Fot the latter point, we have for instance struggled with the fact that

> for lists of length n, one cannot easily take an alternative composition
> over the sublists of length k<n in say PRISM.
>

Nicolas COSTE schrieb:
> One assumption has to be done before using probabilistic methods: the
> studied system has to present a stochastic behavior if we want to have
> accurate results. This assumption is more or less difficult to justify
> according to the system studied. for example, Poisson arrivals in a
> queueing system is the well-known assumption, but the accuracy of the
> results found from probabilistic methods depends on the validity of this
> assumption. Generally this assumption is well adapted but can be wrong
> for "critical" cases or critical behaviors.
>
> However, I think it is often possible to define the error done
> concerning this assumption (for example : the arrivals in a queueing
> system are poissonnian in 90% of the cases so, the arrival law is
> approached by the Poisson law and the error between the real law and the
> Poisson law may be estimated. On the same idea we can see the problem
> encountered for the xSTream architecture: a constant delay is
> approximated by an erlang distribution, and we are able to estimate the
> error between the distribution of a constant delay and the erlang
> distribution)
>
> So, knowing that any arbitrary law can be approached by a phase-type
> distribution (which fits well for probabilistic verification), and
> knowing the error done between a real law present in the studied system
> and its phase-type approximation, a great result would be the knowledge
> of the error done on the results found from probabilistic verification
> methods.
>
> I think this question of errors due to approximations is important in
> the area of stochastic model checking. Indeed, model checking is
> generally based on a complete exploration of the state space, and the
> evaluation of the impact of the different approximation errors could be
> used to size (and to limit) the phase-type approximations (and
> consequently to limit the size of the system model). Otherwise, the only
> thing we can say is that the improvement of the phase-type
> approximations (generally increasing the number of states of the phase
> type distributions) implies that the results found are more accurate,
> but we can not evaluate the link between the gain of accuracy and the
> way the phase-type distribution "get fatter". Unfortunately, using more
> and more accurate approximations is a problem for model checking due to
> size explosion of the system.
>
> I don't know if this problem can be seen as a foundational problem
> because it may be unsolvable. But I think that this problem may fit with
> the definition of the 'theoretical puzzles' because it seems to be a
> very challenging problem :-)
> If there is a solution, it may be a great improvement for probabilistic
> verification and more generally for performance evaluation.

crouzen@alan.cs.uni-sb.de schrieb:
> I'm afraid the most interesting and most challenging question in
> probabilistic verification is an old one: "What to do about the state
> space explosion?" This question is as old as computer science itself and
> many solutions have been proposed. A common way of dealing with the stace
> space explosion is to avoid generating the full state space (for instance
> in assume-guarantee reasoning). In probabilistic verification such
> techniques generally do not work. The foundational methods of solving
> probabilistic models (e.g. transient and steady-state analysis) require
> that we know the entire state space.
>
> There are of course different approaches to tackling this problem. I list
> here the ones I can easily think of:
> - Modularize the solution techniques in the spirit of assume-guarantee
> reasoning,
> - Optimize model generation to avoid large intermediary state spaces

> appearing during the generation, and
> - Use approximation techniques.
>
> Lastly I believe it is important to realize the fundamental difference
> between probabilistic models and non-probabilistic models.
> Non-probabilistic models often model the allowed or desired behavior of a
> system. As such these models are often very restrictive. Out of all
> possible event-chains only a few are allowed. When "X" happens the system
> must do "Y".
>
> On the other hand probabilistic models often model the unpredictable
> nature of a system. A lot of the possible event-chains are allowed and we
> are in fact interested in the probability of some very unprobable
> event-chain happening. When "X" happens "Y1" may happen with probability
> "p1", "Y2" with probability "p2", et cetera. In short non-probabilistic
> models are often very "narrow" where probabilistic models are very "wide".
> In my opinion this difference is important and should be kept in mind
> when dealing with the state space explosion in probabilistic verification.
>
> To put it briefly: in probabilistic verification we need a new answer to
> an old question: "What about the state space explosion?"

Jaco van de Pol schrieb:
> 1. is there a polynomial algorithm for parity games cq. mu-calculus
> model checking.
> (I have "small progress on the strategy improvement algorithm" for
> parity games of
>   Jurdzinski, of which polynomiality is open)
>
> 2. Basically, time can be treated as data. So timed verification can
> be reduced to
> functional verification. This holds, despite the obvious fact that
> specialized algorithms
> can be useful for a more effective study of timed systems.
> Question: can probabilities be treated as data as well?
> Answer: I have no idea.
>

Tingting Han schrieb:
> 1) For continuous-state systems, exponential or generalized distributed
> delays, with or without non-determinism, how to
>   a) give a formal definition of the model?
>   b) minimize the state-space by some equivalences?
>   c) do the parallel compositions?
>   d) extend logics to specify certain properties?
>   d) do model checking?
>
> 2) For probabilistic automata, we have already known that the trace
> distribution precongruence is a branching relation that is compositional.
> Does there exist a real trace-like (linear) congruence for probabilistic
> automata that is compositional ?

Kai Lampka schrieb:

> Given symbolic CTMC generation techniques, one of the biggest practical
> problems in the field of stochastic verification by the means of CTMCs
> in my opinion is the computation of individual state probabilities. Thus
> a reduction in the number of states would not only decrease the memory
> overhead imposed by extremely large solution vectors, but also in the
> run-time per matrix-vector multiplication. In this respect a reward
> function driven CTMC reduction algorithm would be something very useful.
> (a) But how does one aggregate the CTMC, especially if the equivalence
> classes with respect to the reward function contain not bi-similar

> states of the original CTMC and still guarantees correct results? (b) If
> the solution computed on the reward-dependent somehow aggregated CTMC is
> not correct, can one compute bounds on the error, which often would be
> satisfactory, when it comes to the computation of the availability of a
> system.
>
> Usefull literature as starting point in this context would be:
>
> Graham Horton, Scott T. Leutenegger
>
<http://www.informatik.uni-trier.de/%7Eley/db/indices/a-tree/l/Leutenegger:Scott_T=.html>:
> A Multi-Level Solution Algorithm for Steady-State Markov Chains.
> SIGMETRICS 1994
>
<http://www.informatik.uni-trier.de/%7Eley/db/conf/sigmetrics/sigmetrics94.html#HortonL94>:
> 191-200
>
> *and
>
> Bazan, Peter ; German, Reinhard
>
<http://univis.uni-erlangen.de/prg?show=info&key=821/persons/2007w:tech/IMMD/IMMD7/german>*:
> Approximate Analysis of Stochastic Models by Self-Correcting Aggregation .
> In: *Ciardo, G. ; D'Argenio, P. ; Miner, A. ; Rubino, G.* (Hrsg.) :
> */Proc. 2nd. Int. Conf. on the Quantitative Evaluation of Systems 2005/*
> /(QEST 2005 Torino, Italy 19-22 September)./
> 2005, S. 134-143. - ISBN 0-7695-2427-3

Lijun Zhang schrieb:
> * Logical characterisation of (bi-)simulations on image-infinite systems. In
> [NK07], logical characterisation (using CSL) of bisimulation for finite
> CTMDPs is presented. The characterisation is sound, but not complete. In
> [PS07], strong and weak bisimulation are characterised for image-finite PAs.
> We have studied [submitted] the logical characterisation of simulation for
> image-finite PAs, and their continuous-time extensions. It is interesting to
> study the logical characterisation of (bi-)simulations over image-infinite
> systems
>
> * Deciding weak (probabilistic) simulation for PAs. In [ZH07], we presented
> algorithm for deciding strong (probabilistic) simulation for PAs and their
> continuous-time variant. The algorithm can not be extended to weak
> (probabilistic) simulation in an obvious way: as infinite many different
> distributions can be arrived via internal actions. Perhaps we can again
> exploit the power of parametric maximum flow algorithm to compute the weak
> (probabilistic) simulation for PAs.
>

Verena Wolf schrieb:
> Time-dependent schedulers (depending on the time already elasped and
> on the sequence of visited states) and schedulers that may decide to
> "wait" with their decision for a certain amount of time (add an extra
> delay to the residence time of a state).
> How can we find a scheduler that maximizes the probability to reach a
> certain set of states within time interval [a,b]?
>
> There is another problem which is probably not interesting enough ...
>
> It is still an open problem if trace equivalence on IMCs defined using
> randomized stationary schedulers is a subset of trace equivalence
> defined using randomized history-dependent schedulers.
>
> There is also some stuff in the area of probabilistic testing, but at
> the moment I am not engough into that to suggest something...

Gethin Norman schrieb:
> This is quite a difficult question to answer as there is not just one

```
> foundational question or  one answer/solution, but I suppose what
> appears to be the most challenging question is how to extend
> compositional model checking (like for example assume guarantee
> reasoning). The most interesting/difficult part comes from the
> quantitative aspect, for example can one combine quantitative results
> about two subsystems to obtain a quantitative result about the complete
> system. (There has been some preliminary work in this area but there is
> still are long way to go and a lot of different ways to go)

Daniel Klink schrieb:
> "how to (efficiently) model check CTMDPs"...
>
> to begin with, probably
>
> "Efficient computation of time-bounded reachability probs in /arbitrary/
> CTMDPs"
>
> would be puzzling enough... sounds familiar? ;)
>
>
> okay... just some initial ideas on that topic... (I hope the notations
> are clear... should be similar to the ones in "Efficient computation ...
> in uniform CTMDPs")
>
>> applying uniformization as for CTMCs in a naive way does not do the
> trick...
>> consider the following example:
>>
>> /s_l <--a,1-- s --b,2--> s_r
>> /
>> adding edge /s --a,1--> s/ allows for path /s --a--> s --b--> s_r/ and
> as consequence
>> Psi = /in(s) => P_>0(in(s) U in(s_l)) & P_>0(in(s) U in(s_r)) /
>>
>> would be satisfied in the uniformized CTMDP but not in the original one!
>
>> to fix uniformization for this example it would suffice to make a copy
> of /s/, say /s_a
> /> where only outgoing /a/-transitions of /s/ are copied... instead of s
> --a,1--> s the
>> following two transitions can be added to make the CTMDP uniform
> (keeping it
>> weakly bisimilar):
>>
>>    /s --a,1--> s_a/   and   /s_a --a,1--> s_a/
>>
>> Once a is chosen in s, b is not an option anymore... still, this is
> not a solution for
>> arbitrary CTMDPs as uniformization allows to /guess /the time a system
> has been
>> running fairly well... this implies that the set of HD schedulers on
> the uniformized
>> CTMDP is not coinciding with the set of HD schedulers on the original
> one, nor
>> with the set of THD schedulers on the original one...
```

Properties on $\mathrm{P}^2\mathrm{TA}$[*]

Jasper Berendsen David Jansen

November 6, 2007

Recently, the use of real-time model checkers for scheduling synthesis has become *en vogue* [HLP01, Feh99]. The basic idea here is to model all resources as well as all individual tasks together with their (hard) deadlines as timed automata. The question whether there exists a schedule that meets all requirements (such as order of tasks, timing aspects and deadlines) can be formulated as timed reachability question and be tackled with model checkers such as Uppaal.

Scheduling synthesis has been the major motivation to enrich timed automata with prices [LBB$^+$01, BFH$^+$01, ALTP01]. Such prices can be interpreted as bonus, gain, or dually, as cost. Price rates attached to locations indicate the increase of price per time unit, whereas prices attached to edges indicate instantaneous costs. (This is similar to state and impulse rewards, respectively, in Markov reward models [Tij03].) The problem of minimal cost reachability on priced timed automata (also called weighted timed automata) has been shown to be decidable [BFH$^+$01, ALTP01]. The symbolic algorithms are based on priced extensions of the symbolic data structures used for timed automata, such as regions and zones. When interpreting prices as resource costs, these timed automata can be used to obtain minimal cost schedules. In combination with the use of heuristics, scheduling synthesis with (priced) timed automata can often handle larger problem instances than with standard approaches using, e. g., mixed-integer linear programming [NW88].

An important restriction, however, of these approaches is that resources are typically considered to be fully reliable. That is to say, resources are assumed never to break down and (e. g., in case of production machines) never to produce imperfect output. In order to handle situations where things may fail, we introduce *priced probabilistic timed automata* ($\mathrm{P}^2\mathrm{TA}$, for short), which are a probabilistic extension of LPTA (linear priced timed automata) [BFH$^+$01]. $\mathrm{P}^2\mathrm{TA}$ are an orthogonal extension of LPTA, as well as PTA (probabilistic timed automata [KNSS02].) When prices are omitted, probabilistic timed automata are obtained, whereas the deletion of probabilities yields priced timed automata.

Definitions

A *zone* is a conjunction of inequalities where the value of a single clock or the difference between two clocks is compared to an integer[1]. Formally, for the set

[*] Similar problems to the ones described in this text were originally formulated by Joost-Pieter Katoen

[1] The definition of zones is one of the major results for timed automata [AD94].

$\mathbb{X}$ of clocks the set *Zones*$(\mathbb{X})$ of zones Z is defined by the grammar:

$$Z ::= x \bowtie b \mid x - y \bowtie b \mid Z \wedge Z \mid \mathit{true} \qquad \text{where } x, y \in \mathbb{X}, b \in \mathbb{Z}, \bowtie \in \{\le, \ge\}$$

Definition 1 [2] *A* priced probabilistic timed automaton *(P^{2}TA) is a tuple* $(L, l_{init}, \mathbb{X}, pE, \dot{\$})$*, where:*

- L *– finite set of locations;*
- $l_{init} \in L$ *– the initial location;*
- $\mathbb{X}$ *– finite set of clocks;*
- $pE \subseteq L \times \mathit{Zones}(\mathbb{X}) \times 2^{\mathbb{X}} \times \mathrm{Dist}(L)$ *– probabilistic edges;*
- $\dot{\$} : L \to \mathbb{N}$ *– function assigning a price rate to each location.*

For probabilistic edge $(l, g, r, p) \in pE$, l denotes the source location, g the guard, r the set of clocks to be reset, and p a distribution on destination locations. The set E_W of edges of a P^{2}TA W is defined as follows: $(l, g, r, p, l') \in E_W$ if $(l, g, r, p) \in pE$ and $p(l') > 0$.

- $(L, l_{init}, \mathbb{X}, E_W, \dot{\$})$ is an LPTA.
- $(L, l_{init}, \mathbb{X}, pE)$ is a PTA.
- $(L, l_{init}, \mathbb{X}, E_W)$ is a timed automaton.

The intuitive semantics of P^{2}TA is as follows. Each P^{2}TA is mapped onto a (typically infinite-state) transition system. States in these transition systems consist of a location, a *clock valuation* assigning a value to each clock, and the accumulated cost. Execution starts in the initial location with all clocks and the accumulated cost equal to zero. Time may pass in a location, as a result, all clocks increment by the same value. The cost of delaying is determined by the price rate of the location: residing d time units in location l incurs the cost $\dot{\$}(l) \cdot d$. To accommodate for the probabilistic branching, MDPs are used as semantics. A probabilistic edge that emanates from location l may be taken when the state of the system is in l, and the guard is satisfied. On taking a probabilistic edge, the reset set determines which clocks are reset, the destination location is chosen probabilistically according to the distribution of the edge. No time elapses when taking a probabilistic edge.

Definition 2 *[KNSW07] A* timed probabilistic system (TPS) *is a tuple* (S, TSteps)*, where TSteps has one extra label in comparison to a normal discrete time MDP:*

$$\mathit{TSteps} \subseteq S \times \mathbb{R}_{\ge 0} \times \mathrm{Dist}(S) \quad .$$

For $(s, d, \mu) \in \mathit{TSteps}$*, the real* d *denotes the* duration *that the system remains in state* s*. We write* $s \xrightarrow{d,\mu} s'$ *for a* transition *between states* s *and* s' *whenever there exists* $(s, \mu) \in \mathit{TSteps}$ *such that* $\mu(s') > 0$*. Transitions conform to the following rules:*

[2] In comparison to [BJK06] location invariants are removed, distributions are on destination location instead of on pairs of destination location and reset, and instantaneous costs on transitions are removed. Note however that the more elaborate model is easily encoded by the one presented here.

- *time determinism: if $d > 0$, then μ is a point distribution, and if $s \xrightarrow{d,\cdot} t$ and $s \xrightarrow{d,\cdot} t'$ then $t = t'$,*

- *Wang's axiom: $s \xrightarrow{d,\cdot} t$ iff for all $0 \leq d' \leq d$ there exists s' such that $s \xrightarrow{d',\cdot} s'$ and $s' \xrightarrow{d-d',\cdot} t$.*

A path ω in a TPS has the form: $\omega = s_0 \xrightarrow{d_0,\mu_0} s_1 \xrightarrow{d_1,\mu_1} s_2 \xrightarrow{d_2,\mu_2} \cdots$ with $(s_i, d_i, \mu_i) \in \mathit{TSteps}_i$ and $\mu_i(s_{i+1}) > 0$ for all $i \in \mathbb{N}$. The probability space corresponding to a TPS is defined analogously to the probability space of a MDP by means of adversaries (or schedulers) see [KNSW07]. Note that these adversaries have full control on the non-deterministic choices.

Definition 3 (P^{2}TA Semantics) *The semantics of P^{2}TA $(L, l_{init}, \mathbb{X}, pE, \dot{\$})$ is the TPS (S, TSteps), where*

$$S = \{(l, v, c) \mid l \in L \wedge v \in \mathbb{R}_{\geq 0}^{\mathbb{X}} \wedge c \in \mathbb{R}_{\geq 0}\}$$

A probabilistic transition $((l, v, c), d, \mu) \in \mathit{TSteps}$ if and only if one of the following conditions holds:

- *$d \geq 0$, and $\mu(l, v+d, c+\dot{\$}(l)d) = 1$, where $v+d$ denotes the clock valuation v with all values are increased by d;*

- *$d = 0$, and there exists $(l, g, r, p) \in pE$ such that $v \vDash g$, and for any $l' \in L$: $\mu(l', v[r := 0], c) = p(l')$, where $v[r := 0]$ denotes the clock valuation v with all clocks in r set to 0;*

Problems

For P^{2}TA a number of interesting problems can be formulated. In [BJK06] a semi-decidable algorithm for *cost bounded maximal reach probability* is presented. The algorithm computes the maximal probability (under all adversaries) of reaching some location within a cost bound. Here we present two interesting problems which to the best of the authors' knowledge are new.

Problem 1 *The* maximal expected cost *gives the maximal expected cost, under all possible adversaries, of reaching some location with some clock valuation. An adversary assigns a choice to all non-deterministic choices in a P^{2}TA. Therefore under some adversary the semantics of a P^{2}TA is fully probabilistic, and all paths have a certain cost and probability. By looking at all paths that can reach a certain location with some clock valuation, one can define the expected cost of reaching this location/valuation pair under the adversary.*

Problem 2 *The* minimal expected operational cost *gives, under all adversaries, the minimal expected cost* per time unit*. When fixing the adversary, all paths have a certain cost and duration. The operational cost of a path is its cost divided by its duration. The expected operational cost is then defined using the probabilities of the paths. Finally we are interested in the adversary that minimizes this value.*

References

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[ALTP01] Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control: ... HSCC*, volume 2034 of *LNCS*, pages 49–62, Berlin, 2001. Springer.

[BFH$^+$01] Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Larsen, Paul Pettersson, Judi Romijn, and Frits Vaandrager. Minimum-cost reachability for priced timed automata. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control: ... HSCC*, volume 2034 of *LNCS*, pages 147–161, Berlin, 2001. Springer.

[BJK06] Jasper Berendsen, David N. Jansen, and Joost-Pieter Katoen. Probably on time and within budget: On reachability in priced probabilistic timed automata. Technical Report TR CTIT 06-26, Centre for Telematics and Information Technology, University of Twente, June 2006.

[Feh99] Ansgar Fehnker. Scheduling a steel plant with timed automata. In *Sixth Int. Conf. on Real-Time Computing Systems and Applications: RTCSA*, pages 280–286, Los Alamitos, 1999. IEEE Computer Soc. Pr..

[HLP01] Thomas Hune, Kim G. Larsen, and Paul Petterson. Guided synthesis of control programs using Uppaal. *Nordic J. of Computing*, 8(1):43–64, 2001.

[KNSS02] Marta Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282(1):101–150, 2002.

[KNSW07] M. Kwiatkowska, G. Norman, J. Sproston, and F. Wang. Symbolic model checking for probabilistic timed automata. *Information and Computation*, 205(7):1027–1077, 2007.

[LBB$^+$01] Kim Larsen, Gerd Behrmann, Ed Brinksma, Ansgar Fehnker, Thomas Hune, Paul Pettersson, and Judi Romijn. As cheap as possible: efficient cost-optimal reachability for priced timed automata. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Computer Aided Verification*, volume 2102 of *LNCS*, pages 493–505, Berlin, 2001. Springer.

[NW88] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, 1988.

[Tij03] Henk C. Tijms. *A First Course in Stochastic Models.* Wiley, Chichester, 2003.