Proyecto de
Ingeniería del Software II
2019

Proyecto:

- Investigar una herramienta de análisis y verificación
- Elegir un caso de estudio y realizarlo
- Preparar un informe preliminar según las pautas dadas a continuación
- Realizar la revisión anónima de algún trabajo de otro grupo
  - Esta tarea es individual aunque pueden ayudarse dentro del grupo
- Realizar una presentación en clase de la investigación reportada
- Presentar el informe final incorporando las sugerencias de los revisores

Proyecto:

- Investigar una herramienta de análisis y verificación
- Elegir un caso de e
- Preparar un inform
continuación
- Realizar la revisión
  - Esta tarea es individ
- Realizar una presen
reportada
- Presentar el inform
los revisores

Herramientas posibles:

TLA+ Proof System	http://tla.msr-inria.inria.fr/tlaps/
xSAP	https://es-static.fbk.eu/tools/xsap/
CBMC	http://www.cprover.org/cbmc/
BAP	http://bap.ece.cmu.edu
mCRL2	http://www.mcrl2.org
Event-B/Rodin	http://www.event-b.org
Modest Toolset	http://www.modestchecker.net
Prism	http://www.prismmodelchecker.org
Storm	http://www.stormchecker.org
Uppaal	http://uppaal.org
EasyCrypt	https://www.easycrypt.info
JPF	http://javapathfinder.sourceforge.net

Informe

1. Contexto de creación de la herramienta
2. Objetivo de la herramienta
3. Descripción de la herramienta del lado del usuario
4. Aspectos técnicos de la herramienta
5. Casos de estudio (exitosos o no) de la herramienta
6. Comparación con otras herramientas
7. Caso de estudio elegido
8. Conclusiones particulares

Informe

1. Con
2. Objetiv
3. Descrip
4. Aspect
5. Casos c
6. Compar
7. Caso de
8. Conclusiones particulares

Contexto de creación de la herramienta

- ¿Se creo en un contexto académico, industrial?
- ¿Dónde?
- ¿Quienes la desarrollaron?
- ¿En que estado se encuentra?
    - prototipo, prueba de concepto, uso académico, uso industrial
- ¿Quiénes la utilizan?
    - academia independientemente, academia bajo contratos, industria, los mismos desarrolladores, otros usuarios aparte de los desarrolladores)
- ...

Informe

1. Contexto de creación de la herramienta
2. Obj
3. Descrip
4. Aspect
5. Casos d
6. Compar
7. Caso de estudio elegido
8. Conclusiones particulares

Objetivo de la herramienta

- ¿A qué tipos de problema apunta la herramienta?
  - protocolos, tiempo real, dependibilidad, desempeño, seguridad,...
- ¿En cual etapa del proceso de desarrollo es útil la herramienta?
  - especificación, arquitectura, diseño, implementación, testing, mantenimiento,...
- ...

Informe

1. Conte
2. Objet
3. Desc
4. Aspec
5. Casos
6. Comp
7. Caso
8. Concl

Descripción de la herramienta del lado del usuario

- ¿Tiene interfaz gráfica?
- ¿Trabaja sobre línea de comando?
- ¿Como es el lenguaje de especificación de modelos?
- ¿y el de propiedades?
  - LTL, CTL, asercional,...
- ¿Que tipo de análisis permite permite?
  - deadlock, progreso, invariantes, control de algún tipo de fairness,...
- ¿Trabaja directamente sobre código? ¿Cuál lenguaje?
- ¿Es necesario anotar el modelo/programa?
- ¿Reporta contraejemplos?
- ¿Permite simulación?
- ¿Cómo visualiza?
- ...

Informe

1. Contex
2. Objeti
3. Descri
4. Aspe
5. Casos
6. Compa
7. Caso d
8. Conclu

Aspectos técnicos de la herramienta

- ¿Cómo representa el espacio de estado?
  - simbólico, explícito,...
- ¿Que estructura de datos utiliza?
  - BDD, MTBDD, Bitstate hashing, zonas...
- ¿Usa técnicas de reducción?
  - simetría, reducción de orden parcial,...
- ¿Usa técnicas de abstracción/refinamiento? ¿Cuáles?
- ¿Considera todo el espacio de estado? ¿una parte de éste?
- ¿La respuesta es correcta? ¿es exhaustiva? ¿es aproximada?
- ¿Tiene falsos positivos/negativos?
- ...

Informe

1. Contexto de creación de la herramienta
2. Objetiv
3. Descrip
4. Aspect
5. Caso
6. Compar
7. Caso de
8. Conclus

Casos de estudio (exitosos o no) de la herramienta

- ¿Para que tipos de sistemas se utilizó la herramientas?
  - hardware, protocolos de comunicación, sistemas embebidos, sistemas espaciales, control industrial, aviónica,...
- Enumerar algunos casos de estudios concretos
- Reportar resultados concretos del desempeño de la herramienta
  - tamaño de los casos de estudios, uso de memoria, tiempos de ejecución,...
- ...

Informe

1. Contexto de creación de la herramienta
2. Objetivo de la herramienta
3. Descrip
4. Aspect
5. Casos c
6. Comp
7. Caso de
8. Conclus

Comparación con otras herramientas

- En la medida de lo posible reportar comparación con otras herramientas.
- Buscar en los artículos
  - usualmente tienen secciones de trabajos relacionados o lo mencionan en la introducción/conclusión
- Usar su propio conocimiento (al menos conocen LTSA)
- ...

Informe

1. Contexto de creación de la herramienta
2. Objetivo de la herramienta
3. Descripción de la herramienta del lado del usuario
4. Aspect
5. Casos c
6. Compar
7. Caso
8. Conclus

Caso de estudio elegido

- Presentar el caso de estudio: Qué es y a dónde se usa
- Explicar cómo funciona
- Reportar las propiedades a verificar
- ¿Reportó algún error? en tal caso, mostrar contraejemplo y explicar como se soluciona (si es que es posible)
- ...

Informe

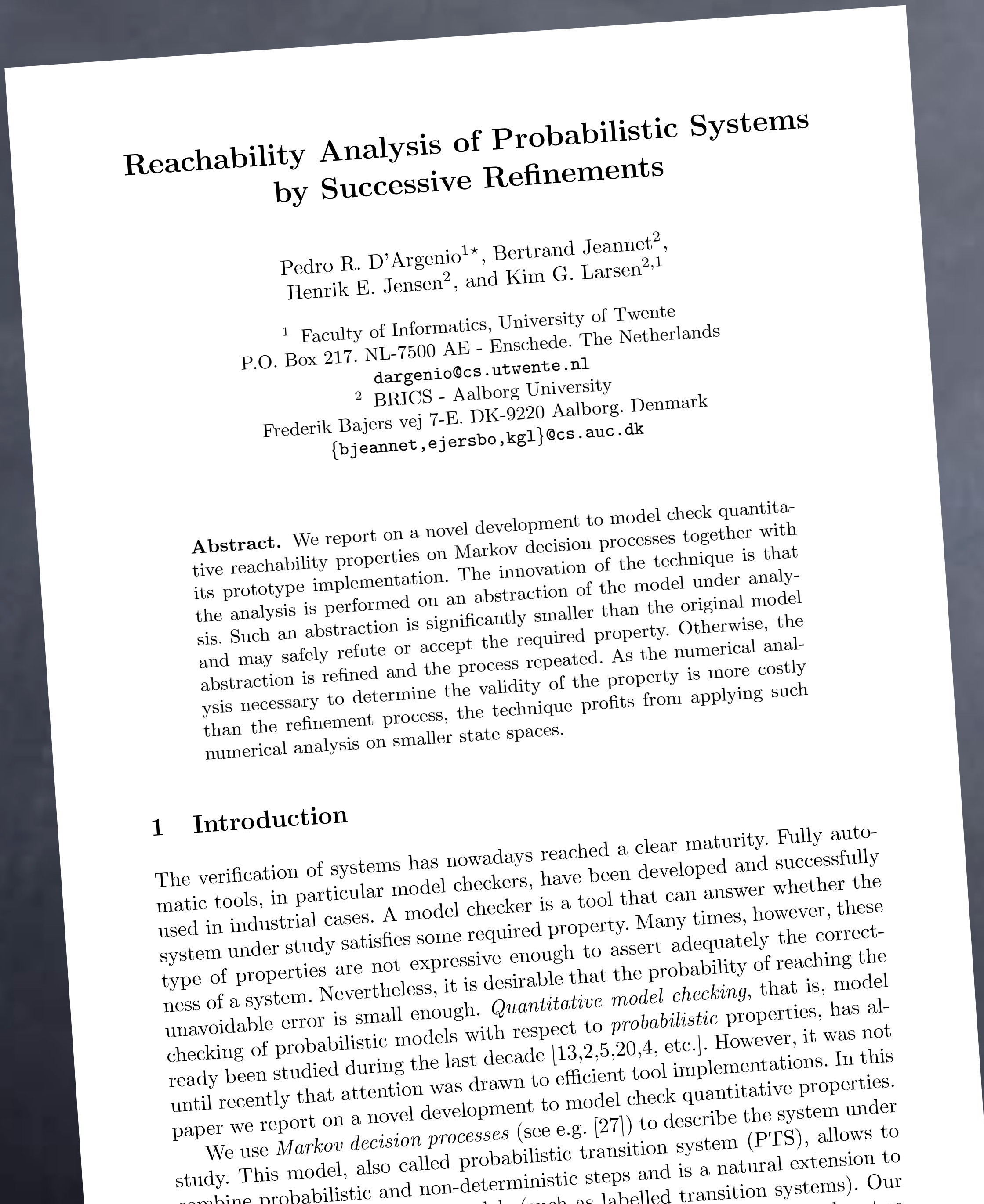
1. Contexto de creación de la herramienta
2. Objetivo de la herramienta
3. Descrip
4. Aspect
5. Casos d
6. Compar
7. Caso de
8. Concl

Conclusiones particulares

Reportar opinión propia de la herramienta en diversos aspectos, por ejemplo:

- complejidad de instalación
- complejidad en la comprensión de la herramienta
- "amigabilidad"
- usabilidad
- versatilidad
- eficiencia
- ...

Informe



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1*], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
dargenio@cs.utwente.nl
[2] BRICS - Aalborg University
Frederik Bajers vej 7-E. DK-9220 Aalborg. Denmark
{bjeannet,ejersbo,kgl}@cs.auc.dk

Abstract. We report on a novel development to model check quantitative reachability properties on Markov decision processes together with its prototype implementation. The innovation of the technique is that the analysis is performed on an abstraction of the model under analysis. Such an abstraction is significantly smaller than the original model and may safely refute or accept the required property. Otherwise, the abstraction is refined and the process repeated. As the numerical analysis necessary to determine the validity of the property is more costly than the refinement process, the technique profits from applying such numerical analysis on smaller state spaces.

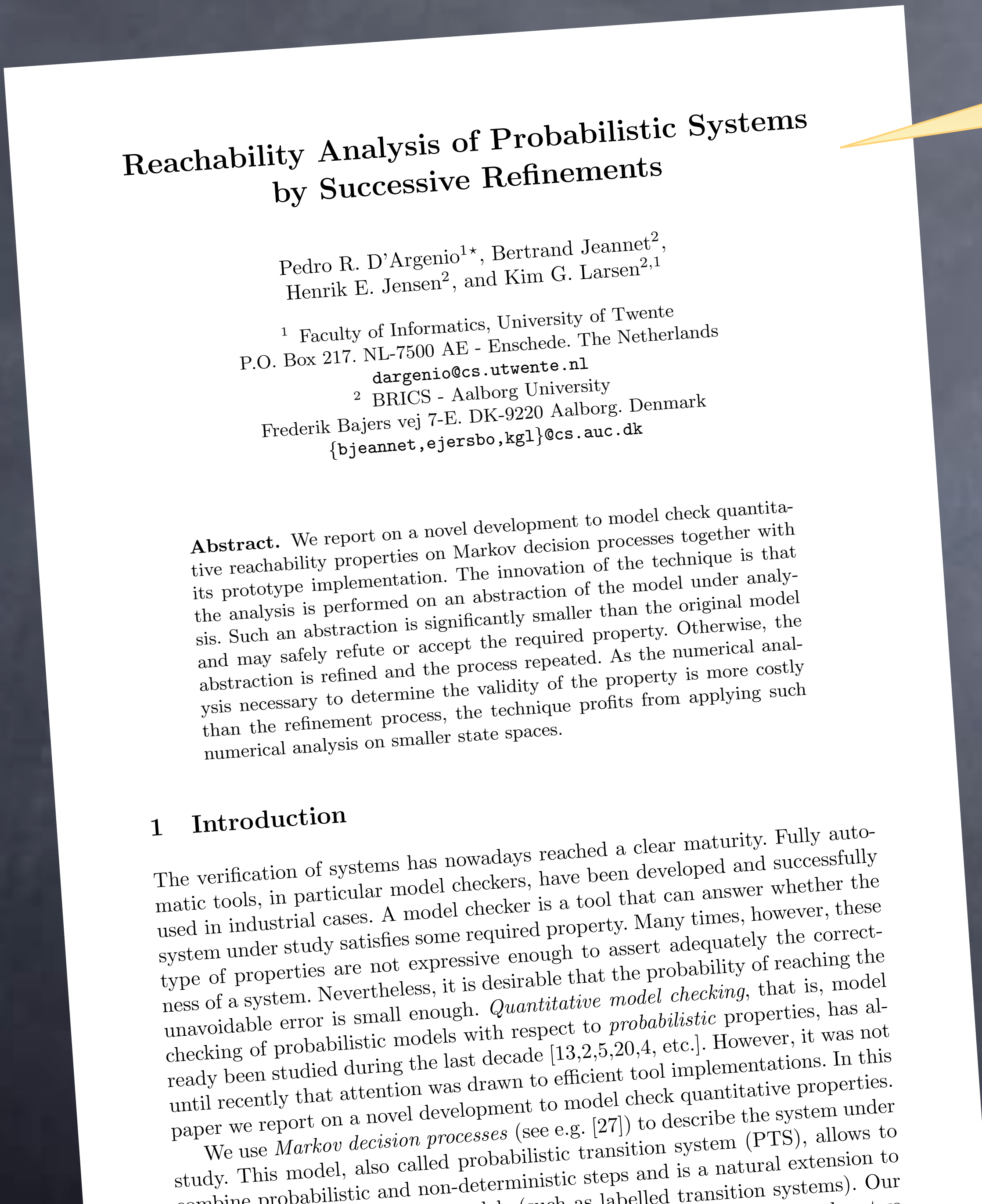
1 Introduction

The verification of systems has nowadays reached a clear maturity. Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to ... (such as labelled transition systems). Our

Informe

Título



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1⋆], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
dargenio@cs.utwente.nl

[2] BRICS - Aalborg University
Frederik Bajers vej 7-E. DK-9220 Aalborg. Denmark
{bjeannet,ejersbo,kgl}@cs.auc.dk

Abstract. We report on a novel development to model check quantitative reachability properties on Markov decision processes together with its prototype implementation. The innovation of the technique is that the analysis is performed on an abstraction of the model under analysis. Such an abstraction is significantly smaller than the original model and may safely refute or accept the required property. Otherwise, the abstraction is refined and the process repeated. As the numerical analysis necessary to determine the validity of the property is more costly than the refinement process, the technique profits from applying such numerical analysis on smaller state spaces.

1 Introduction

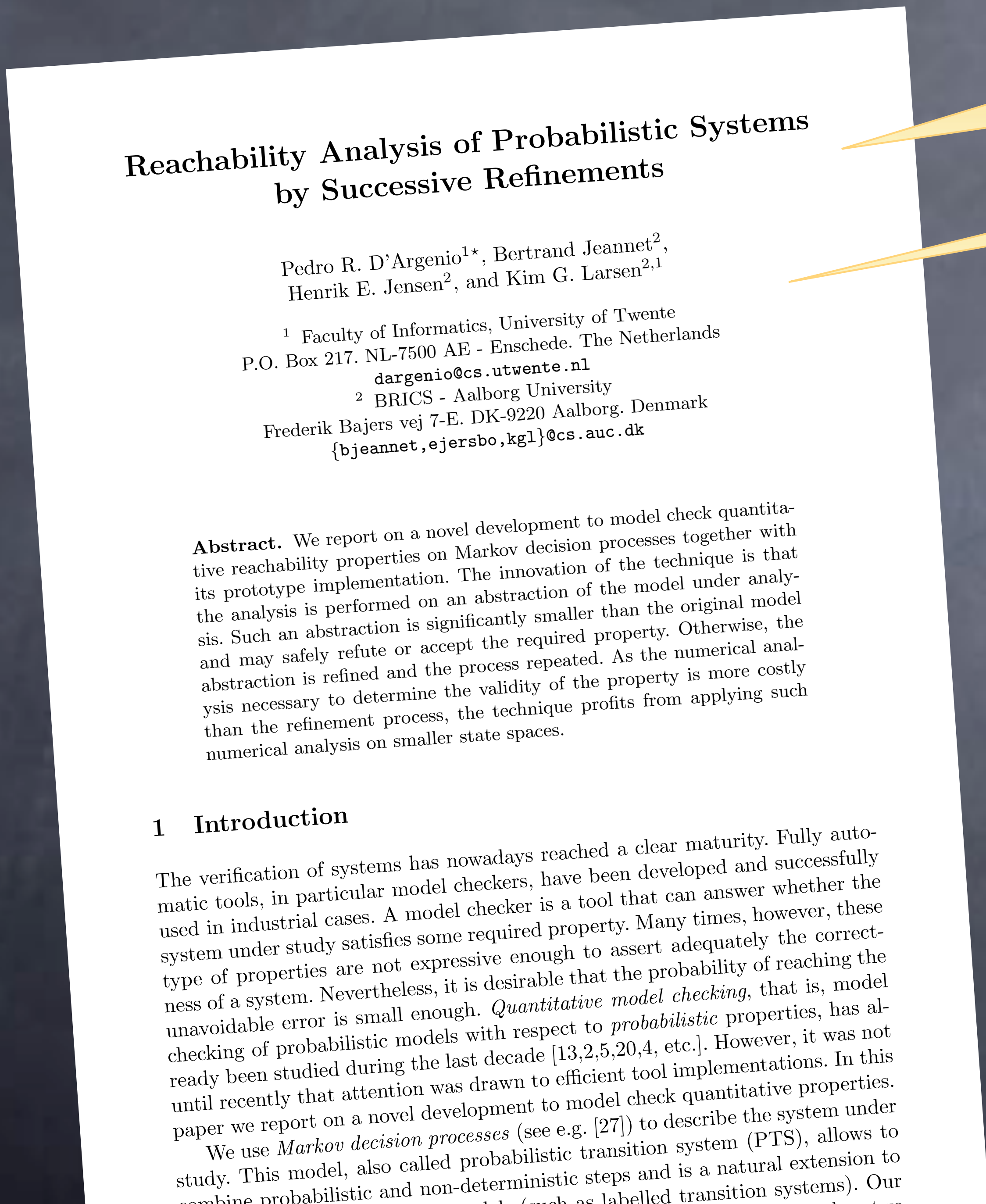
The verification of systems has nowadays reached a clear maturity. Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to

Informe

Título

Autores y afiliación



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1⋆], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
`dargenio@cs.utwente.nl`
[2] BRICS - Aalborg University
Frederik Bajers vej 7-E. DK-9220 Aalborg. Denmark
`{bjeannet,ejersbo,kgl}@cs.auc.dk`

Abstract. We report on a novel development to model check quantitative reachability properties on Markov decision processes together with its prototype implementation. The innovation of the technique is that the analysis is performed on an abstraction of the model under analysis. Such an abstraction is significantly smaller than the original model and may safely refute or accept the required property. Otherwise, the abstraction is refined and the process repeated. As the numerical analysis necessary to determine the validity of the property is more costly than the refinement process, the technique profits from applying such numerical analysis on smaller state spaces.

1 Introduction

The verification of systems has nowadays reached a clear maturity. Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

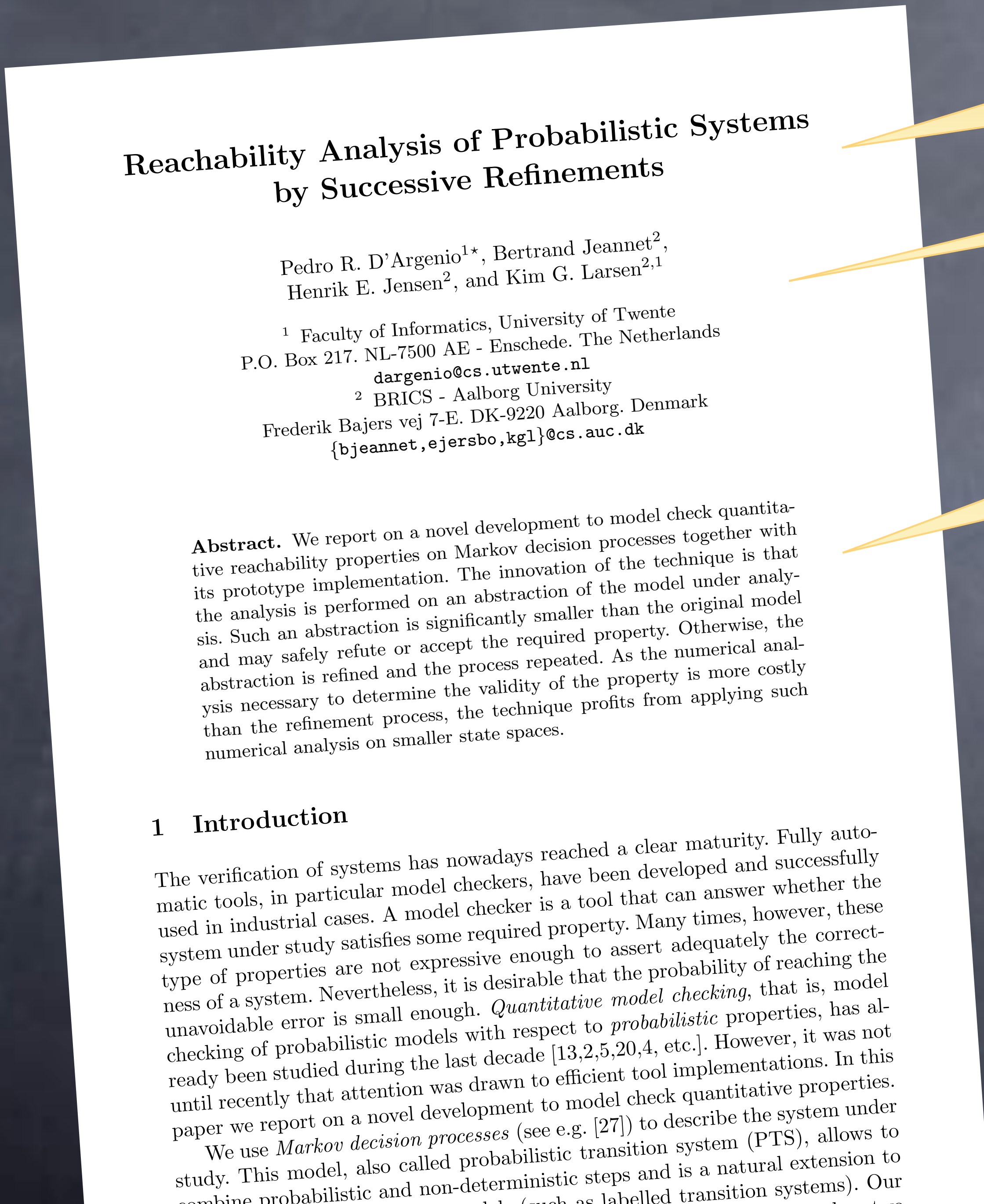
We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to ... (such as labelled transition systems). Our

Informe

Título

Autores y afiliación

Resumen: describiendo brevemente el contenido y contribución del artículo



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1*], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
dargenio@cs.utwente.nl

[2] BRICS - Aalborg University
Frederik Bajers vej 7-E. DK-9220 Aalborg. Denmark
{bjeannet,ejersbo,kgl}@cs.auc.dk

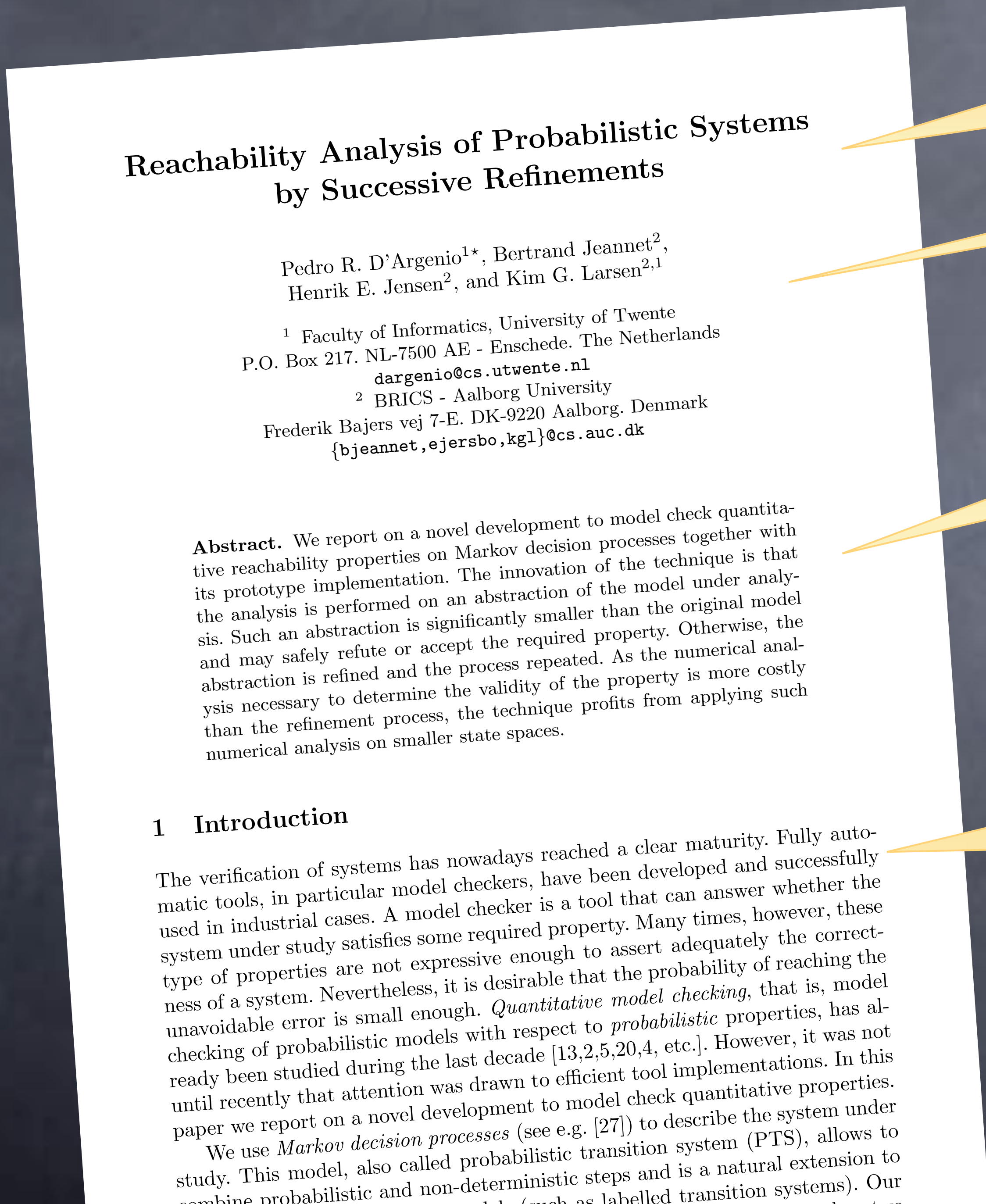
Abstract. We report on a novel development to model check quantitative reachability properties on Markov decision processes together with its prototype implementation. The innovation of the technique is that the analysis is performed on an abstraction of the model under analysis. Such an abstraction is significantly smaller than the original model and may safely refute or accept the required property. Otherwise, the abstraction is refined and the process repeated. As the numerical analysis necessary to determine the validity of the property is more costly than the refinement process, the technique profits from applying such numerical analysis on smaller state spaces.

1 Introduction

The verification of systems has nowadays reached a clear maturity. Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to ... (such as labelled transition systems). Our

Informe



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1*], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
dargenio@cs.utwente.nl
[2] BRICS - Aalborg University
Frederik Bajers vej 7-E. DK-9220 Aalborg. Denmark
{bjeannet,ejersbo,kgl}@cs.auc.dk

Abstract. We report on a novel development to model check quantitative reachability properties on Markov decision processes together with its prototype implementation. The innovation of the technique is that the analysis is performed on an abstraction of the model under analysis. Such an abstraction is significantly smaller than the original model and may safely refute or accept the required property. Otherwise, the abstraction is refined and the process repeated. As the numerical analysis necessary to determine the validity of the property is more costly than the refinement process, the technique profits from applying such numerical analysis on smaller state spaces.

1 Introduction

The verification of systems has nowadays reached a clear maturity. Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to

Título

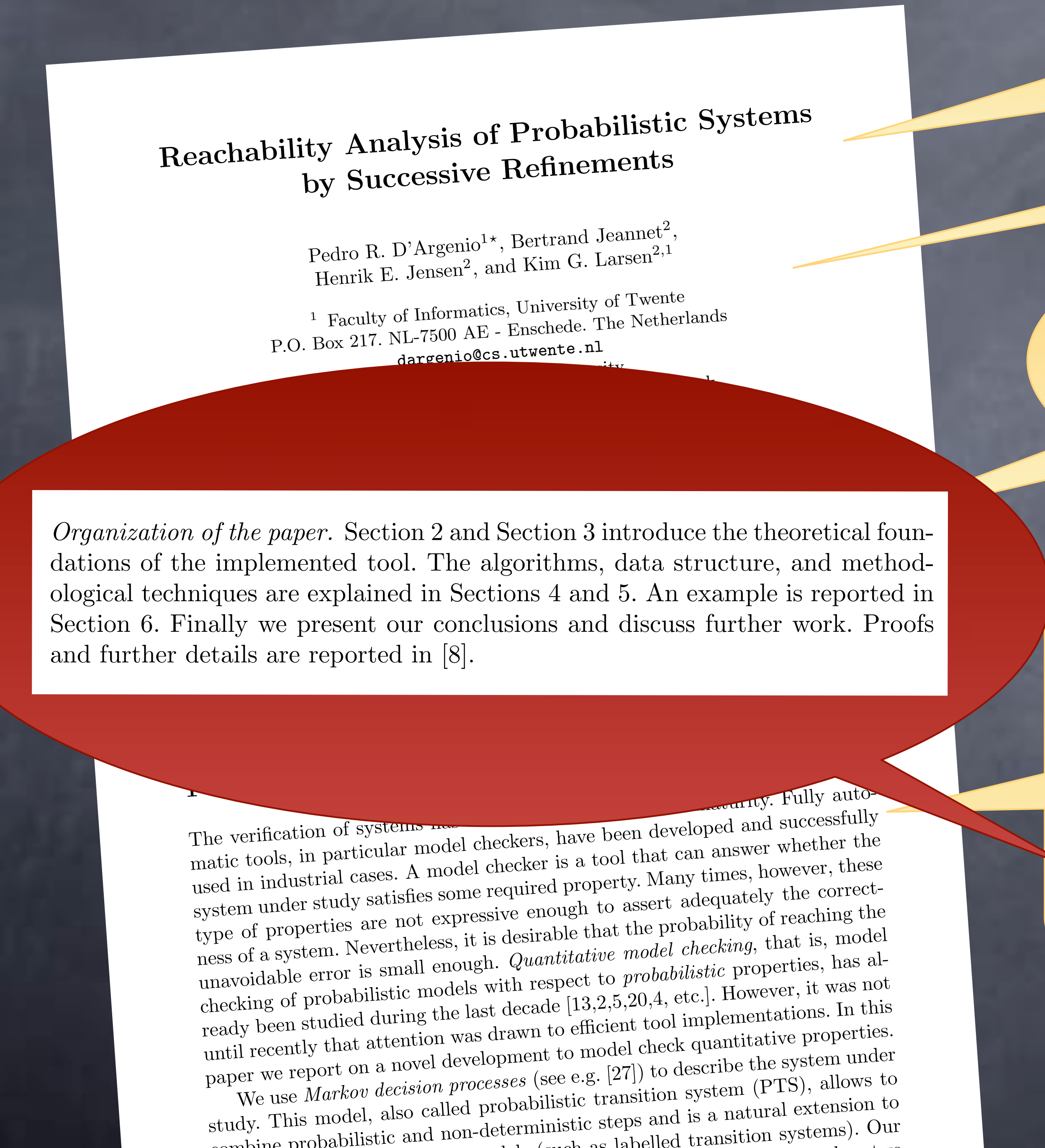
Autores y afiliación

Resumen:
describiendo brevemente el contenido y contribución del artículo

Introducción:

a. contextualizar la contrib.
b. motivación
c. contribución
d. descripción del contenido

Informe



Reachability Analysis of Probabilistic Systems by Successive Refinements

Pedro R. D'Argenio[1*], Bertrand Jeannet[2], Henrik E. Jensen[2], and Kim G. Larsen[2,1]

[1] Faculty of Informatics, University of Twente
P.O. Box 217. NL-7500 AE - Enschede. The Netherlands
dargenio@cs.utwente.nl

Organization of the paper. Section 2 and Section 3 introduce the theoretical foundations of the implemented tool. The algorithms, data structure, and methodological techniques are explained in Sections 4 and 5. An example is reported in Section 6. Finally we present our conclusions and discuss further work. Proofs and further details are reported in [8].

... Fully automatic tools, in particular model checkers, have been developed and successfully used in industrial cases. A model checker is a tool that can answer whether the system under study satisfies some required property. Many times, however, these type of properties are not expressive enough to assert adequately the correctness of a system. Nevertheless, it is desirable that the probability of reaching the unavoidable error is small enough. *Quantitative model checking*, that is, model checking of probabilistic models with respect to *probabilistic* properties, has already been studied during the last decade [13,2,5,20,4, etc.]. However, it was not until recently that attention was drawn to efficient tool implementations. In this paper we report on a novel development to model check quantitative properties.

We use *Markov decision processes* (see e.g. [27]) to describe the system under study. This model, also called probabilistic transition system (PTS), allows to combine probabilistic and non-deterministic steps and is a natural extension to ... (such as labelled transition systems). Our

Título

Autores y afiliación

Resumen:
describiendo brevemente el contenido y contribución del artículo

Introducción:
a. contextualizar la contrib.
b. motivación
c. contribución
d. descripción del contenido

Informe

is fundamental as the method ... techniques.

We focus on a restricted set of reachability properties. They allow to specify that the probability to reach a particular final condition f from any state satisfying a given initial condition i is smaller (or greater) than a probability p. This type of properties is not so restrictive as it seems since we can always use checking automata to add additional constraints to the property.

The method we present is based on automatic abstraction and refinement techniques. The basic idea is to use abstraction to reduce the high cost of probabilistic analysis. The difficulty lies in finding the right abstraction level, depending on the property to prove. To address it, the method starts with a coarse abstraction of the system which is obtained by *partitioning* the state space, according to the property under study. The property is then checked on the obtained abstract model. The verdict may be inconclusive, that is, p happens to be between the calculated upper bound of the minimum and the lower bound of the maximum actual probabilities. In this case the previous abstraction is refined and the question posed again. The process is successively repeated until a satisfactory answer is given, or no further refinement is possible. To efficiently store the state space, perform abstractions and process the refinement steps, we use BDDs and MTBDDs (more precisely ADDs) [10,3]. The soundness of the method is asserted by considering a suitable probabilistic simulation [22,28] (which preserves the kind of property we consider), and by showing that abstraction by partitioning respects this simulation relation.

The contributions of this paper are first the definition of the probabilistic simulation relation that allows to prove the soundness of our method, and secondly, the design of efficient algorithms to abstract PTSs, to analyse and to refine them. Finally, experimental results shows the effectiveness of the method.

Related work. The partition refinement method we use on PTSs resorts to principles already applied to finite-state systems [6] and timed automata [1]. However our aim is not to generate a minimal model w.r.t. a bisimulation relation, but to steer the refinement process in order to prove as early as possible an intended (probabilistic) property.

The efficiency provided by MTBDDs to store and logically manipulate the state space made them also the choice of recent quantitative model checkers [14, 9]. However, if it comes to model analysis via numerical recipes like simplex or (iterative) solutions of equations systems, experience has shown that MTBDDs do not outperform classical data structures (such as sparse matrices) [3,18,9]. The main reason appears to be that any of these algorithms tend to require the storage of a distinct real number per actual state [16]. In our case, the use of MTBDDs is focus on the manipulation of probabilistic transition relations and its use in the abstraction techniques. After abstraction, the size of the problem submitted to numerical analysis becomes a significantly smaller issue.

Informe

Habitualmente, la sección siguiente a la introducción contiene conceptos preliminares y fija la nomenclatura y la notación.
Las siguientes secciones se abocan directamente a la contribución del artículo

is fundamental as the method ...
techniques.

We focus on a restricted set of reachability properties. They allow to specify that the probability to reach a particular final condition f from any state satisfying a given initial condition i is smaller (or greater) than a probability p. This type of properties is not so restrictive as it seems since we can always use checking automata to add additional constraints to the property.

The method we present is based on automatic abstraction and refinement techniques. The basic idea is to use abstraction to reduce the high cost of probabilistic analysis. The difficulty lies in finding the right abstra... level depending on the property to prove. To address it, the metho... abstraction of the system which ... obtained by *partitio*... according to the property under study. T... ...perty is t... tained abstract model. The verdict may be inconc... between the calculated upper bound of the minimum and ... maximum actual probabilities. In this case the previou... and the question posed again. The process is successively ... factory answer is given, or no further refinement is poss... the state space, perform abstractions and process the r... BDDs and MTBDDs (more precisely ADDs) [10,3]. The s... is asserted by considering a suitable probabilistic simula... serves the kind of property we consider), and by show... partitioning respects this simulation relation.

The contributions of this paper are first the defin... simulation relation that allows to prove the soundness... ondly, the design of efficient algorithms to abstract ... refine them. Finally, experimental results shows the effe...

Related work. The partition refinement method we use on PTSs resorts to principles already applied to finite-state systems [6] and timed automata [1]. However our aim is not to generate a minimal model w.r.t. a bisimulation relation, but to steer the refinement process in order to prove as early as possible an intended (probabilistic) property.

The efficiency provided by MTBDDs to store and logically manipulate the state space made them also the choice of recent quantitative model checkers [14, 9]. However, if it comes to model analysis via numerical recipes like simplex or (iterative) solutions of equations systems, experience has shown that MTBDDs do not outperform classical data structures (such as sparse matrices) [3,18,9]. The main reason appears to be that any of these algorithms tend to require the storage of a distinct real number per actual state [16]. In our case, the use of MTBDDs is focus on the manipulation of probabilistic transition relations and its use in the abstraction techniques. After abstraction, the size of the problem submitted to numerical analysis becomes a significantly smaller issue.

Informe

is fundamental as the method ...
techniques.

We focus on a restricted set of reachability properties. They allow to specify that the probability to reach a particular final condition f from any state satisfying a given initial condition i is smaller (or greater) than a probability p. This type of properties is not so restrictive as it seems since we can always use checking automata to add additional constraints to the property.

The method we present is based on automatic abstraction and refinement techniques. The basic idea is to use abstraction to reduce the high cost of probabilistic analysis. The difficulty lies in finding the right abstra... level, depending on the property to prove. To address it, the metho... abstraction of the system which is obtained by *partitio*... according to the property under study. ... tained abstract model. The verdict may be incon... between the calculated upper bound of the minimum and... maximum actual probabilities. In this case the previou... and the question posed again. The process is successively... factory answer is given, or no further refinement is poss... the state space, perform abstractions and process the r... BDDs and MTBDDs (more precisely ADDs) [10,3]. The s... is asserted by considering a suitable probabilistic simul... serves the kind of property we consider), and by show... partitioning respects this simulation relation.

The contributions of this paper are first the defin... simulation relation that allows to prove the soundness... ondly, the design of efficient algorithms to abstract ... refine them. Finally, experimental results shows the effe...

Related work. The partition refinement method we use on PTSs resorts to principles already applied to finite-state systems [6] and timed automata [1]. However our aim is not to generate a minimal model w.r.t. a bisimulation relation, but to steer the refinement process in order to prove as early as possible an intended (probabilistic) property.

The efficiency provided by MTBDDs to store and logically manipulate the state space made them also the choice of recent quantitative model checkers [14, 9]. However, if it comes to model analysis via numerical recipes like simplex or (iterative) solutions of equations systems, experience has shown that MTBDDs do not outperform classical data structures (such as sparse matrices) [3,18,9]. The main reason appears to be that any of these algorithms tend to require the storage of a distinct real number per actual state [16]. In our case, the use of MTBDDs is focus on the manipulation of probabilistic transition relations and its use in the abstraction techniques. After abstraction, the size of the problem submitted to numerical analysis becomes a significantly smaller issue.

Habitualmente, la sección siguiente a la introducción contiene conceptos preliminares y fija la nomenclatura y la notación.
Las siguientes secciones se abocan directamente a la contribución del artículo

Trabajos relacionados:
usualmente antes de las conclusiones, a veces en la introducción

Informe

7 Concluding Remarks

In this article we introduced an efficient technique for quantitative model checking. The method relies on automatic abstraction of the original system. This allows to significantly reduce the size of the problem to which numerical analysis is applied in order to compute the quantitative factor of the property under study. Since the numerical analysis is the most costly part of the whole process this reduction is of high importance. This reduction is achieved first because bisimilar states are never distinguished, and secondly because using incremental abstraction refinement and confronting the analysis against a desired (or undesired) probability allows prompt answers on very compact spaces.

The execution time is currently not the best as the tool should be optimised. Table 3 reports the tool performance for a set of properties[1]. The current implementation performs numerical analysis using linear programming techniques under exact rational arithmetics. This method is very fast (compared to the painfully slow iterative methods) and it does not suffer of numerical unstability since numbers are represented in its exact form. Two remarks are in order. First, numerical analysis is applied in each refinement step, which is inefficient since a refinement step may add only few partitions with low chances of sensibly affecting the result of the previous iteration. Second, the already mentioned asymmetric convergence in which only the minimum or the maximum gradually converges to the actual value while the other does not until the last refinements.

It is in our near future plans to develop efficiency improvements. One of these improvements concerns the refinement strategy and the suitable alternation of refinement and analysis that should be used. Another improvement would be to take advantage of the fact that probabilities usually appears only in some part of the modelled system: failures do not appear everywhere!

On a long term agenda, we plan to use this incremental refinement technique to check probabilistic timed automata. Model checking of PTCTL properties on such model was proven decidable by resorting to their region graphs [25]. However, region graphs are known to be impractical. Our technique would allow to generate progressively a minimal *probabilistic* model, in the spirit of [1].

Acknowledgements. We thank Holger Hermanns and Joost-Pieter Katoen for fruitful discussions.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, D. Dill, and H. Wong-Toi. Minimization of timed transition systems. In R. Cleaveland, ed., *Procs. of CONCUR 92*, Stony Brook, NY, LNCS 630, pp. 340–354. Springer, 1992.
2. A. Aziz, V. Singhal, F. Balarin, R.K. Bryton, and A.L. Sangiovanni-Vincentelli. It usually works:the temporal logics of stochastic systems. In P. Wolper, ed., *Procs. of the 7th CAV*, Liège, LNCS 939, pp. 155–165. Springer, 1995.

Informe

7 Concluding Remarks

In this article we introduced an efficient technique for quantitative model checking. The method relies on automatic abstraction of the original system. This allows to significantly reduce the size of the problem to which numerical analysis is applied in order to compute the quantitative factor of the property under study. Since the numerical analysis is the most costly part of the whole process this reduction is of high importance. This reduction is achieved first because bisimilar states are never distinguished, and secondly because using incremental abstraction refinement and confronting the analysis against a desired (or undesired) probability allows prompt answers on very compact spaces.

The execution time is currently not the best as the tool should be optimised. Table 3 reports the tool performance for a set of properties[1]. The current implementation performs numerical analysis using linear programming techniques under exact rational arithmetics. This method is very fast (compared to the painfully slow iterative methods) and it does not suffer of numerical unstability since numbers are represented in its exact form. Two remarks are in order. First, numerical analysis is applied in each refinement step, which is inefficient since a refinement step may add only few partitions with low chances of sensibly affecting the result of the previous iteration. Second, the already mentioned asymmetric convergence in which only the minimum or the maximum gradually converges to the actual value while the other does not until the last refinements.

It is in our near future plans to develop efficiency improvements. One of these improvements concerns the refinement strategy and the suitable alternation of refinement and analysis that should be used. Another improvement would be to take advantage of the fact that probabilities usually appears only in some part of the modelled system: failures do not appear everywhere!

On a long term agenda, we plan to use this incremental refinement technique to check probabilistic timed automata. Model checking of PTCTL properties on such model was proven decidable by resorting to their region graphs [25]. However, region graphs are known to be impractical. Our technique would allow to generate progressively a minimal *probabilistic* model, in the spirit of [1].

Acknowledgements. We thank Holger Hermanns and Joost-Pieter Katoen for fruitful discussions.

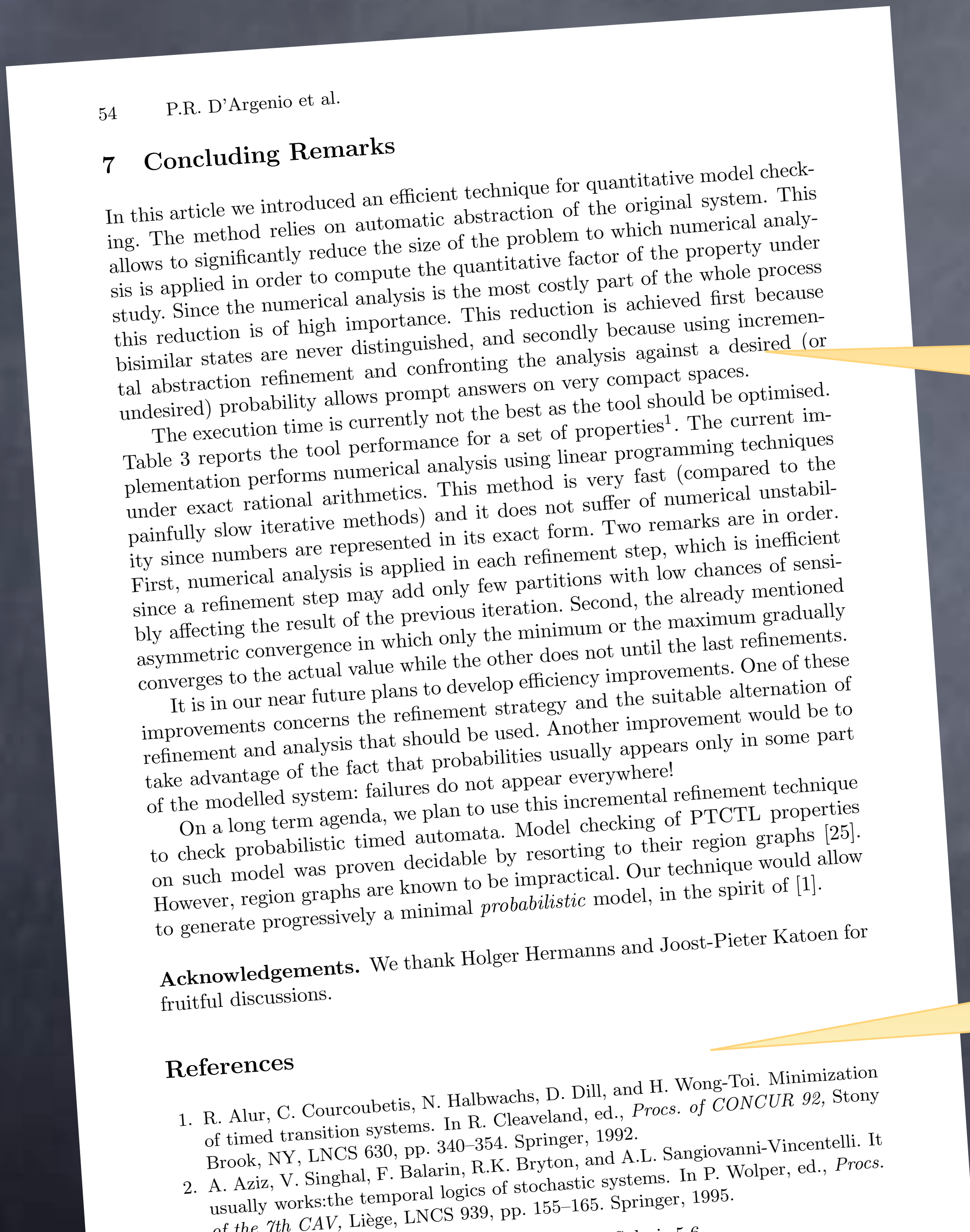
References

1. R. Alur, C. Courcoubetis, N. Halbwachs, D. Dill, and H. Wong-Toi. Minimization of timed transition systems. In R. Cleaveland, ed., *Procs. of CONCUR 92*, Stony Brook, NY, LNCS 630, pp. 340–354. Springer, 1992.
2. A. Aziz, V. Singhal, F. Balarin, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. It usually works:the temporal logics of stochastic systems. In P. Wolper, ed., *Procs. of the 7th CAV*, Liège, LNCS 939, pp. 155–165. Springer, 1995.

Conclusiones:

- comienza redondeando rapidamente lo que se trató en el artículo
- resalta contribuciones
- describe ventajas y desventajas
- describe posibles trabajos futuros

Informe



7 Concluding Remarks

In this article we introduced an efficient technique for quantitative model checking. The method relies on automatic abstraction of the original system. This allows to significantly reduce the size of the problem to which numerical analysis is applied in order to compute the quantitative factor of the property under study. Since the numerical analysis is the most costly part of the whole process this reduction is of high importance. This reduction is achieved first because bisimilar states are never distinguished, and secondly because using incremental abstraction refinement and confronting the analysis against a desired (or undesired) probability allows prompt answers on very compact spaces.

The execution time is currently not the best as the tool should be optimised. Table 3 reports the tool performance for a set of properties[1]. The current implementation performs numerical analysis using linear programming techniques under exact rational arithmetics. This method is very fast (compared to the painfully slow iterative methods) and it does not suffer of numerical unstability since numbers are represented in its exact form. Two remarks are in order. First, numerical analysis is applied in each refinement step, which is inefficient since a refinement step may add only few partitions with low chances of sensibly affecting the result of the previous iteration. Second, the already mentioned asymmetric convergence in which only the minimum or the maximum gradually converges to the actual value while the other does not until the last refinements.

It is in our near future plans to develop efficiency improvements. One of these improvements concerns the refinement strategy and the suitable alternation of refinement and analysis that should be used. Another improvement would be to take advantage of the fact that probabilities usually appears only in some part of the modelled system: failures do not appear everywhere!

On a long term agenda, we plan to use this incremental refinement technique to check probabilistic timed automata. Model checking of PTCTL properties on such model was proven decidable by resorting to their region graphs [25]. However, region graphs are known to be impractical. Our technique would allow to generate progressively a minimal *probabilistic* model, in the spirit of [1].

Acknowledgements. We thank Holger Hermanns and Joost-Pieter Katoen for fruitful discussions.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, D. Dill, and H. Wong-Toi. Minimization of timed transition systems. In R. Cleaveland, ed., *Procs. of CONCUR 92*, Stony Brook, NY, LNCS 630, pp. 340–354. Springer, 1992.
2. A. Aziz, V. Singhal, F. Balarin, R.K. Bryton, and A.L. Sangiovanni-Vincentelli. It usually works:the temporal logics of stochastic systems. In P. Wolper, ed., *Procs. of the 7th CAV*, Liège, LNCS 939, pp. 155–165. Springer, 1995.

Conclusiones:

- comienza redondeando rapidamente lo que se trató en el artículo
- resalta contribuciones
- describe ventajas y desventajas
- describe posibles trabajos futuros

Referencias bibliográficas:
¡Importante!

Informe

- 8 páginas máximo
- En LaTeX
  - Lo deberían poder bajar con un aptget, sino
    - http://www.ctan.org/starter o http://www.latex-project.org
  - Leer "The (Not So) Short Introduction to LaTeX2e"
    - Está en la documentación del paquete
    - http://ctan.dcc.uchile.cl/info/lshort/english/lshort.pdf
    - http://ctan.dcc.uchile.cl/info/lshort/spanish/lshort-a4.pdf
- Usar el formato LNCS:
  - ftp://ftp.springer.de/pub/tex/latex/llncs/latex2e/llncs2e.zip
- La mejor forma de aprender es copiando: usen el ejemplo que viene con el paquete de LNCS.

Informe

Algunas ayudas para la redacción:

- Evitar usar subsecciones
- Toda figura debe ser explicada en el texto principal
- Evitar largos "captions" en figuras y tablas
- Minimizar los listados a través de itemizaciones o enumeraciones
- Agrupar figuras usando "subigures" o usar "wrapfigure" para hacer un mejor uso del espacio
- La bibliografía listada debe ser apropiadamente citada en el cuerpo del paper