Propuesta de Tesis

para obtener el título de Licenciado en Ciencias de la Computación

Carlos Gustavo LOPEZ POMBO
Libreta universitaria: 26/95 - **e-mail**: clpombo@dc.uba.ar

Director: Dr. Victor Adrián BRABERMAN

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Resumen

Los sistemas de tiempo real requieren atención particular debido a que usualmente llevan a cabo tareas críticas que pueden representar pérdidas, ya sea en términos de vidas o grandes sumas de dinero.

Los automatas temporizados, presentados en [3], son un formalismo conocido y aceptado para el modelado de sistemas donde el tiempo tiene un papel preponderante y por lo tanto se convienten en el marco de especificación sobre el que se plantea el proceso de verificación de propiedades.

Una ventaja que los autómatas temporizados poseen es que se han desarrollado herramientas (por ejemplo [7], [6] y [9]) que posibilitan la verificacin automtica propiedades sobre este formalismo. Estas técnicas han sido utilizadas con éxito en casos de estudio reales.

Una de las propiedades más utilizada es la que establece la posibilidad de que un conjunto de estados que satisfacen una propiedad dada sea alcanzable desde otro conjunto de estados. Esta propiedad es llamada *alcanzabilidad*, y es verificada a través de un algoritmo que implementa un método de punto fijo.

Las modificaciones sobre este algoritmo surgirán de propiedades intrínsecas de las operaciones que realiza el algoritmo que verifica alcanzabilidad implementado en el *Model Checker KRONOS*, ([7]), durante el proceso de cómputo y el análisis de la topología del grafo subyacente en el autómata temporizado.

1 Introducción y estado del arte

La importancia de los sistemas de tiempo real es que, en general, las tareas que estos llevan a cabo son críticas, ejemplos de esto son los desarrollos en la industria electrónica y aeroespacial, que pueden representar pérdidas, ya sea en términos de vidas o grandes sumas de dinero.

Los automatas temporizados, presentados en [3], son un formalismo conocido y aceptado para el modelado de sistemas donde el tiempo tiene un papel preponderante y por lo tanto se convienten en el marco de especificación sobre el que se plantea el proceso de verificación de propiedades TCTL, una extensión temporizada de la lógica CTL, que fue presentada en [2] con el objetivo de encontrar un método "Model Checking" para sistemas de tiempo real.

Una ventaja que los autómatas temporizados poseen por sobre otros formalismos es que se han desarrollado herramientas (por ejemplo [7], [6] y [9]); que posibilitan la verificacin automtica propiedades sobre este formalismo. Estas técnicas han sido utilizadas con éxito en casos de estudio reales que requerían del uso de este tipo de técnicas.

Una técnica comunmente utilizada para implementar los algoritmos que resuelven el problema de verificar una propiedad sobre un autómata temporizado es la búsqueda del menor punto fijo de un funcional monótono, como se explica en [13] y [12].

Una de las propiedades más utilizada es la que establece la posibilidad de que un conjunto de estados que satisfacen una propiedad dada sea alcanzable desde otro conjunto de estados. Esta propiedad es llamada *alcanzabilidad*, y es verificada a través de un algoritmo que implementa un método de punto fijo sobre un reticulado formado por los conjuntos de estados el los que un sistema de tiempo real puede encontrarse.

En este trabajo se analizará la posibilidad de optimizar el algoritmo que evalúa alcanzabilidad implementado en el *Model Checker KRONOS*, ([7]); las modificaciones surgirán de propiedades intrínsecas de las operaciones que realiza el algoritmo este algoritmo durante el proceso de cómputo y el análisis de la topología del grafo subyacente en el autómata temporizado.

Lamentablemente el proceso de verificacin es muy costoso. Se ha probado que la verificación es un problema $P - SPACE$ que depende del tamaño de la entrada, es decir, depende de la cantidad de relojes y el valor de las constantes que se utilizan en las condiciones del autómata. Para enfrentar este problema se han desarollado varios métodos de reducción de relojes [14] y otros que apuntan a la reduccin del espacio de estados, por ejemplo [15], [16], [17]. Aún en esta situación, es importante tener algoritmos de verificación de propiedades TCTL tan eficientes como sea posible, en particular, aquel que

verifica propiedades de alcanzabilidad.

2 Propuesta de tesis

La propuesta de tesis está basada en la obtención de resultados teóricos y prácticos que permitan justificar la elaboración e implementación de un método de verificación de propiedades de alcanzabilidad sobre autómatas temporizados basado en el esquema que se halla implementado en el *KRONOS*, agregando a este distintas capacidades, entre las que se puede contar la de realizar algún análisis sobre la topología del grafo subyacente que suministre la información necesaria al momento de verificar una propiedad como las mencionadas anteriormente.

Además de la incorporación de las capacidades mencionadas anteriormente, es posible que surjan modificaciones a dicho algoritmo que provengan del estudio de las propiedades de las operaciones que este algoritmo realiza a lo largo del proceso de verificación. Para esto será de suma importancia un estudio profundo de la implementación existente del *Model Checker KRONOS* que permita establecer la factibilidad de las modificaciones que se propongan.

2.1 Plan de trabajo

2.1.1 Estudio del problema a resolver

La elaboración de la tesis constará primeramente por el estudio del estado del arte, lo que parcialmente se encuentra desarrollado en la bibliografía básica citada anteriormente, para luego pasar al estudio de la problemática puntual que se desea resolver, que en este caso implicará, no solo la lectura y estudio de bibliografía más específica sobre el tema en cuestión, sino también el análisis de la implementación existente del algoritmo que evalúa alcanzabilidad presente en *KRONOS*.

Al final de estas etapas debería ser posible tener un análisis de factibilidad que determine la posibilidad de implementar las modificaciones que surjan de dicho estudio.

2.1.2 Desarrollo de una solución

Una vez establecida la posibilidad de implementar las modificaciones que se planteen se procederá al desarrollo de los resultados teóricos que susten-

ten y justifiquen la implementación de dichas modificaciones como parte de *KRONOS*; para posteriormente pasar a desarrollo de dicha implementación.

Es de suma importancia que previo a la implementación de las modificaciones que se propongan, se supere una etapa en la que se plantee el diseño de las estructuras de datos que serán necesarias para poder incorporar dichas modificaciones; esta etapa es de suma importancia debido a la relación que tiene la elección de estas estructuras con el costo computacional del algoritmo que se desea optimizar.

2.1.3 Estudio empírico de los resultados obtenidos

Luego de la implementación de las modificaciones que se propongan, será de suma importancia la elaboración de casos de estudio que permitan obtener una visión tan clara como sea posible de los resultados, como una comparación entre el desempeño de la implementación original del algoritmo, contrastandolo con el desempeño de la implementación propuesta en la tesis.

Referencias

[1] R. Alur, C. Courcoubetis, D. Dill. "Model Checking for Real-Time systems". In Proc. of 5th Symp. on Logic in Computer Science, pages 414-425. IEEE Computer Society Press, 1990. See also "Model-checking in dense real-time". In Information and Computation, 126:183-235, 1993.

[2] R. Alur, C. Courcoubetis and D. Dill. "Model-Checking in Dense Real-Time". Information and Computation 104, Nr. 1, pages 2–34, 1993.

[3] R. Alur and D. Dill. "Automata for Modeling Real-Time Systems". In Proc. of 17th ICALP, pages 322-335. Lecture Notes in Computer Science 443, Springer-Verlag, 1990. See also "A theory of timed automata". In Theoretical Computer Science 126:183-235, 1994.

[4] R. Alur and D.L. Dill. "A theory of timed automata". Theoretical Computer Science 126, pages 183–235, 1994.

[5] R. Alur and T. A. Henzinger. "Logics and Models of Real-Time: a survey". In Proc. of REX Workshop on Real-time: Theory in Practice. Lecture Notes in Computer Science 600, Springer-Verlag, 1991.

[6] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson and W. Yi. "UPPAAL - A Tool Suited for the Automatic Verification of Real-Time Systems". In Proc. of Hybrid Systems III, pages 232-243. Lecture Notes in Computer Science 1066, Springer-Verlag, 1996.

[7] C. Daws, A. Olivero, S. Tripakis and S. Yovine. "The Tool KRONOS". In Proc. of Hybrid Systems III, pages 494-510. Lecture Notes in Computer Science 1687, Springer-Verlag, 1996.

[8] E. A. Emerson and E. Clarke. "Design and Synthesis of Synchronization Skeletons using Branching-Time Temporal Logics". In Proc. of Workshop of Logics of Programs. Lecture Notes in Computer Science 131, Springer-Verlag, 1981.

[9] T. A. Henzinger, P. H. Ho and H. Wong-Toi. "A User Guide to HyTec". In Proc. of Intl Workshop on Tools and Algorithms for the Construction and Analysis of Systems, pages 41-71. Lecture Notes in Computer Science 1019, Springer-Verlag, 1995.

[10] T. A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine. "Symbolic Model Checking for Real-Time Systems". In Proc. of 7th Symp. on Logic in

Computer Science, pages 394-406. IEEE Computer Society Press, 1992. See also Information and Computation, 111(2):193-244, 1994.

[11] S. Yovine. "Model Checking Timed Automata". In Proc. of Workshop of Logics of Programs. Lecture Notes in Computer Science 1494, Springer-Verlag, 1998.

[12] A. Olivero. "Modélisation et Analyse de Systèmes Temporisés et Hybrides". Institut National Politechnique de Grenoble. These pour obtenir le grade de Docteur es Infotmatique. 1994.

[13] S. Yovine. "Méthodes et Outils pour la Vèrification Symbolique de Systémes Temporisés". Institut National Politechnique de Grenoble. These pour obtenir le grade de Docteur es Infotmatique. 1993.

[14] C.Daws and S.Yovine. "Reducing the Number of clock Variables of Timed Automata". Proc. IEEE Real-Time Systems Symposium '96, IEEE Computer Soc. Press, Los Alamitos, Calif., 1996.

[15] S.Tripakis ans S.Yovine. "Analysis of timed systems based on time-abstracting bisimulation"

[16] R.P.Kurshan S.Tasiran, R.Alur and R.K.Brayton. "Verifying Abstractions of Timed Systems". In Proc. of the 7th Int'l. Conf. on Concurrency Theory (CONCUR 1996), Lecture Notes in Computer Science 1119, Springer Verlag, 1996.

[17] D.Clarke, O.Grumberg and D.Long. "Model Checking and Abstraction". Proc., Principles of Programming Languages (POPL), 1994.