Un Enfoque Genérico y Cooperativo para la Resolución de Restricciones de Intervalo*

Antonio J. Fernández

Dpto. Lenguajes y Ciencias de la Computación,
E.T.S.I.I., Universidad de Málaga,
29071 Málaga, Spain,
afdez@lcc.uma.es

Palabras clave: Satisfacción de restricciones, programación lógica, retículo, intervalo.

1. Introducción

La evidencia del éxito del paradigma de la *programación lógica con restricciones* (CLP) [3, 11] se encuentra en el número creciente de sistemas para CLP que se emplean, hoy día, en aplicaciones reales [14]. Principalmente existen dos razones para este éxito: la primera es que CLP extiende el paradigma de la programación lógica permitiendo soluciones más declarativas y legibles y, la segunda, es que proporciona propagación de restricciones sobre dominios específicos, con lo cual se generan implementaciones eficientes de los procedimientos operacionales.

El componente esencial del esquema de CLP es que puede ser parametrizado por diferentes dominios de computación los cuales determinan diferentes instancias del esquema tales como CLP(Interval(*Integer*)) (i.e., CLP sobre los intervalos de enteros) [1], CLP($\Re$) (i.e., CLP sobre los reales) [12], CLP(Sets) (i.e., CLP sobre los conjuntos finitos) [10] o CLP(*Bool*) (i.e., CLP sobre los booleanos) [2]. El tipo del dominio determina la naturaleza de las restricciones y el resolutor usado para resolverlas. En particular, la cardinalidad del dominio determina el procedimiento de resolución de tal forma que los actuales sistemas de CLP tienen métodos diferentes para el dominio finito/discreto y para los dominios infinitos/continuos.

Sin embargo, en la práctica los problemas no son específicos a un dominio y su formulación tiene que ser artificialmente adaptada a un resolutor específico (i.e., a un dominio específico). Además, la mayoría de los resolutores de restricción son *opacos* y su control depende exclusivamente del sistema. Estas restricciones proporcionan implementaciones muy eficientes para aplicaciones prácticas pero carecen de la flexibilidad para ser usadas en aplicaciones para las cuales no fueron creadas. Este problema es solucionado por las llamadas *restricciones transparentes* [1, 9] las cuales pueden ser definidas por el usuario para aplicaciones específicas.

Además, a pesar de que la mayoría de los resolutores se basan en extensiones de las técnicas de consistencia descritas en [13], el razonamiento resolutivo difiere para cada dominio de computación. En general hay dos formas básicas de razonamiento: el de dominio y el de intervalo. El razonamiento de intervalo (que considera únicamente los límites del dominio) es más eficiente que

*Esta tesis fue dirigida por la Dra. Patricia M. Hill perteneciente al departamento de School of Computing de la Universidad de Leeds en Inglaterra, y tutorizada por el Dr. José María Troya Linero, perteneciente al mismo dpto. que el tesinando. La tesis fue presentada el 25 de Enero de 2002, obteniendo la calificación de Sobresaliente Cum Laude.

el razonamiento de dominio (que razona sobre todos los valores del dominio). Como consecuencia, si requerimos eficiencia, el razonamiento de intervalo parece ser el más adecuado.

Esta tesis unifica todos los puntos anteriormente citados y propone un esquema genérico y cooperativo para CLP(Interval($\mathcal{X}$)) donde $\mathcal{X}$ es cualquier dominio de computación con estructura de retículo. El esquema, que está basado en la teoría de retículos, es un enfoque general para la satisfacción y optimización de restricciones de intervalo así como para la cooperación de resolutores de intervalo definidos sobre dominios de computación con estructura de retículos, independientemente de la cardinalidad de éstos. Se proporciona pues un esquema novedoso para la implementación de un resolutor único que resuelve los muchos problemas de los resolutores actuales pues combina generalidad, transparencia, cooperación y potencial eficiencia.

El resultado obtenido asegura un enfoque novedoso totalmente genérico y transparente sobre el cual las restricciones, los dominios de computación y los mecanismos de propagación y cooperación, definidos entre las variables restringidas, pueden ser fácilmente especificados a nivel del usuario y aplicados sobre la resolución de problemas reales heterogéneos que no tienen solución con las propuestas actuales.

2. Contribuciones

Las principales aportaciones de la tesis podrían resumirse en los siguientes cuatro puntos.

La tesis presenta una comparativa global sobre la eficiencia y algunos aspectos de la expresividad de ocho de los sistemas existentes más competitivos para la satisfacción de restricciones. Esta comparativa, realizada sobre el dominio finito y el dominio booleano, muestra diferencias principales entre los sistemas de restricciones existentes. Hasta la fecha, ésta constituye la comparativa más extensa y la que abarca más sistemas distintos publicada en la comunidad de las restricicones.

Una segunda contribución más teórica consiste en el desarrollo formal de un modelo genérico para CLP(Interval($\mathcal{X}$)). Para definir este modelo se ha descrito el proceso global de resolución de restricciones de intervalo sobre cualquier retículo, separando claramente los procesos de propagación y división (ramificación) de intervalos. Una de las ventajas es que la monotonía de las restricciones está implícitamente definida en la teoría. Además, se declaran un conjunto de propiedades interesantes que, bajo ciertas condiciones, son satisfechas por cualquier instancia del modelo. Más aún, se muestra que muchos resolutores de restricciones actualmente existentes son instancias de este modelo y satisfacen estas condiciones. Además, se proporcionan indicaciones sobre cómo extender el sistema mediante la especificación de otras instancias interesantes y novedosas.

Una tercera contribución consiste en la extensión del esquema de CLP(Interval($\mathcal{X}$)) para posibilitar la cooperación de resolutores de manera que la información puede fluir entre diferentes dominios de computación, bien sean predefinidos o bien el resultado de la combinación de dominios ya existentes mediante operadores de combinación de retículos. Este enfoque abre nuevas perspectivas en el área de la cooperación de resolutores.

La cuarta contribución consiste en la implementación de un prototipo basado en el marco teórico propuesto y donde se demuestra que un único sistema, basado en el esquema para CLP(Interval($\mathcal{X}$)), puede proporcionar soporte para la satisfacción y la optimización de restricciones así como para la cooperación de resolutores sobre múltiples dominios de computación. Este prototipo sigue un novedoso enfoque transparente sujeto a una doble perspectiva ya que el usuario puede definir no sólo nuevas restricciones y su mecanismo de propagación, sino también nuevos dominios sobre los cuales nuevas restricciones pueden ser resueltas así como el mecanismo de cooperación entre todos los dominios de computación (ya sean definidos por el usuario o predefinidos por el sistema).

3. Estructura

El trabajo presentado se desarrolla en ocho capítulos distribuidos en cuatro partes.

La primera parte, *el marco introductorio*, se compone de dos capítulos. El primer capítulo expone tanto los objetivos planteados como las motivaciones que alentaron el trabajo, y lo hace describiendo las limitaciones actuales de los principales sistemas existentes de CLP y mostrando cómo éstas son resueltas en el esquema que se define. El segundo capítulo proporciona una reseña histórica sobre los fundamentos de CLP y los últimos avances en el campo.

La segunda parte, *el marco comparativo*, describe un exhaustivo trabajo de comparación sobre un conjunto de sistemas CLP sujetos a diferentes enfoques y distintos mecanismos de resolución. Esta parte permite elegir el modelo de resolución más adecuado sobre el cual definir el marco teórico.

La tercera parte, *el marco teórico*, constituye el grueso de la tesis y presenta la formalización teórica. Esta parte comienza definiendo en el Capitulo 4, un modelo genérico para la propagación de restricciones de intervalo sobre dominios con estructura de retículo y demuestra propiedades interesantes del mismo (e.g., corrección y terminación). El Capítulo 5 extiende este modelo para admitir la cooperación de resolutores y el Capítulo 6 completa el marco teórico con un nuevo esquema genérico para la ramificación de restricciones de intervalo. Este nuevo esquema añade completitud al modelo original tanto en la resolución clásica como en la optimización de los *problemas de satisfacción de restricciones* (CSPs).

La cuarta parte, *el marco práctico*, consiste en el capítulo 7 que presenta una implementación prototipo de todo el marco teórico, y además desarrolla y resuelve complejos CSPs mostrando la viabilidad del marco teórico así como su flexibilidad y declaratividad.

Finalmente, el capítulo 8 resume los resultados obtenidos y presenta diversas líneas de investigación futura.

4. Resumen y Conclusiones

En esta tesis, hemos propuesto un esquema para la resolución de restricciones de intervalo que combina tres características deseables: (1) generalidad real, (2) cooperativismo de los resolutores y (3) transparencia total en la definición de restricciones, dominios y el mecanismo de propagación. Además vaticina una cuarta propiedad importante: eficiencia en la resolución de las restricciones. El esquema ha sido desarrollado en cuatro etapas.

Primeramente, se han comparado diferentes enfoques transparentes para la resolución de restricciones. Como resultado de esta comparación, se ha elegido el modelo de *indexicals* [1], definido para el dominio finito (FD), como la base sobre la cual construir los fundamentos teóricos del resolutor. La elección se justificó en el hecho de que este modelo muestra un rendimiento alto y una flexibilidad elevada en la formulación de CSPs.

Después se ha generalizado el modelo de indexicals desde el FD a cualquier dominio con estructura de retículo. El resultado es un resolutor genérico para la propagación de restricciones de intervalo sobre cualquier retículo, independiente de la cardinalidad y naturaleza de éste. Se ha mostrado que el esquema es útil para todos los dominios clásicos y que, además, es válido para nuevos dominios que son generados a partir de constructores y combinadores de retículos.

Posteriormente se ha extendido el modelo dando paso a una novedosa forma de cooperación en la que se permite que la información fluya entre los diferentes dominios soportados por el sistema (i.e., cualquier retículo). El nuevo esquema es transparente, colaborativo y genérico y, por tanto, puede usarse tanto para la resolución clásica de CSPs como para la resolución parcial de los mismos. El modelo se ha completado definiendo un esquema genérico para la ramificación de restricciones de intervalo. Este esquema de bifurcación está parametrizado en varios procedimientos que son parcialmente especificados y cuya implementación queda nuevamente en manos del usuario de forma transparente.

Para demostrar la viabilidad del modelo propuesto, se definió un lenguaje para CLP(Interval($\mathcal{X}$)) basado en el marco teórico y, además, se implementó un prototipo para dicho lenguaje en el cual se resolvieron ejemplos reales que ilustraron la declaratividad del sistema.

El modelo para CLP(Interval($\mathcal{X}$)) soluciona muchas de las evidentes limitaciones de los actuales sistemas de CLP. Por un lado soluciona las desventajas obvias de un resolutor específico de dominio tales como la incapacidad de resolver restricciones definidas sobre otros dominios de computación lo cual restringe su contexto de aplicación al dominio para el cual fue diseñado; esta es una limitación muy importante ya que, en la práctica, los problemas no son específicos a un dominio de computación. Como consecuencia la formulación de un problema tiene que ser artificialmente adaptada para que pueda ser resuelta por un resolutor específico, perdiendo pues parte de la declaratividad de la solución. Además resuelve las carencias de los sistemas opacos (e.g., insuficiente flexibilidad en aplicaciones no convencionales ya que el problema tiene que ser codificado mediante restricciones predefinidas que son proporcionadas por el sistema) y de los sistemas transparentes (e.g., restricción a dominios predefinidos lo cual acota la flexibilidad ya que -como ya ha sido apuntado- en la práctica, los problemas son heterogéneos y, a menudo, su formulación natural usa dominios de computación que son diferentes a los dominios predefinidos por el sistema).

En la literatura encontramos enfoques cooperativos que ayudan a solventar tanto la carencia de declaratividad de las soluciones como el pobre rendimiento de los sistemas. Sin embargo la mayoría del trabajo realizado sobre este campo está enfocado en dominios predefinidos y no es flexible ni adaptable a otros dominios lo cual supone un problema de nuevo por la no-homogeneidad de los CSPs. Desde este punto de vista, el esquema para CLP(Interval($\mathcal{X}$)) supone una nueva perspectiva en este campo que soluciona parte de este problema.

Por otra parte, la principal crítica a los enfoques genéricos es que la eficiencia es una de las razones principales para escoger CLP y los resolutores genéricos raramente alcanzan la velocidad de los resolutores específicos. Sin embargo el modelo definido en la tesis se basa en un razonamiento de intervalo que es más eficiente que un razonamiento de dominio, con lo cual potencialmente (i.e., a falta de una demostración empírica) una implementación (i.e., no un prototipo) basada en el esquema debería ser competitiva.

Se han alcanzado todos los objetivos trazados y los resultados han sido divulgados en numerosos congresos (e.g., [4] y [5] por citar algunos) y revistas internacionales de primer nivel (e.g., [6, 7, 8]). Como trabajo futuro más inmediato se plantea la implementación completa del resolutor genérico y su posible integración en otros paradigmas declarativos tal como el lógico-funcional.

Referencias

[1] P. Codognet and D. Diaz. Compiling constraints in clp(FD). *The Journal of Logic Programming*, 27(3):185–226, 1996.

[2] P. Codognet and D. Diaz. A simple and efficient boolean solver for constraint logic programming. *The Journal of Automated Reasoning*, 17(1):97–129, 1996.

[3] J. Csontó and J. Paralič. A look at CLP: theory and application. *Applied Artificial Intelligence*, 11:59–69, 1997.

[4] A. J. Fernández and P.M. Hill. An impartial efficiency comparison of FD constraints systems. In *Principles and Practice of Constraint Programming*, number 1520 in LNCS, page 468. Springer-Verlag, 1998.

[5] A. J. Fernández and P.M. Hill. An interval lattice-based constraint solving framework for lattices. In *Symposium on Functional and Logic Programming*, number 1722 in LNCS, pages 194–208. Springer-Verlag, 1999.

[6] A. J. Fernández and P.M. Hill. A comparative study of eight constraint programming languages over the Boolean and finite domains. *Constraints*, 5(3):275–301, 2000.

[7] A. J. Fernández and P.M. Hill. An interval constraint system for lattice domains. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 26(1):1–46, 2004.

[8] A. J. Fernández and P.M. Hill. An interval constraint branching scheme for lattice domains. *The Journal of Universal Computer Science*, 2005. Accepted for publication. Selected papers presented in PROLE'2005.

[9] T. Frühwirth. Theory and practice of constraint handling rules. *The Journal of Logic Programming*, 37:95–138, 1998.

[10] C. Gervet. Interval propagation to reason about sets: definition and implementation of a practical language. *Constraints*, 1(3):191–244, 1997.

[11] J. Jaffar and M. Maher. Constraint logic programming: a survey. *The Journal of Logic Programming*, 19-20:503–581, 1994.

[12] J. Jaffar, S. Michaylov, P. Stuckey, and R. Yap. The CLP($\Re$) language and system. *ACM Transactions on Programming Languages and Systems*, 14(3):339–395, 1992.

[13] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[14] PACLP'2000. *The Practical Applications of Constraint Technology and Logic Programming*. Practical Application Company, Manchester, UK, April 2000.