Programación en lógica rebatible: una semántica declarativa

Laura A. Cecchi
Departamento de Informática y Estadística
UNIVERSIDAD NACIONAL DEL COMAHUE
e-mail: lcecchi@uncoma.edu.ar

PALABRAS CLAVE: Extensiones de la Programación en Lógica. Semántica Declarativa de Extensiones de la Programación en Lógica. Semántica de Juegos. Sistemas Argumentativos.

Introducción

La Programación en Lógica Rebatible [GS99, GSC98, Gar00] (de ahora en más P.L.R.) es una extensión de la Programación en Lógica (P.L.) con una nueva clase de reglas, las reglas rebatibles. Estas reglas permiten representar conocimiento tentativo, aumentando, de este modo, el poder expresivo de la P.L..

El razonamiento no monotónico basado en el análisis dialéctico constituye la semántica operacional de la P.L.R.. Para verificar si una consulta es consecuencia de un programa lógico rebatible, este formalismo utiliza un análisis dialéctico de argumentos y contraargumentos. Así, una consulta q tendrá éxito si existe un argumento $\mathscr{A}$ de q que lo justifique, *i.e.*, no existen contraargumentos que derroten a $\mathscr{A}$. Ya que los derrotadores son también argumentos podrían existir derrotadores para éstos últimos y así sucesivamente.

En los últimos años, la semántica operacional de la P.L.R. ha sido estudiada desde un punto de vista declarativo [Dun95, KT99, JV99, CDSS00], con el objeto de determinar el significado preciso de un programa lógico sin recurrir al control del sistema. El propósito principal de dicho estudio es ayudar al programador a especificar el conocimiento y razonar a partir de él independientemente de cualquier implementación. Por otra parte, la definición de una semántica declarativa ayudará a caracterizar el comportamiento de los programas lógicos rebatibles como sistema de razonamiento no monótono, a través del conjunto de sus consecuencias y compararlo [Dix95a, Dix95b, Dix95c, CDSS00] con otros sistemas de razonamiento, mostrando ventajas y desventajas.

En [CS99,CS00a,CS00b], se introdujo una semántica declarativa trivaluada $\mathcal{GS}$ basada en juegos que permite modelar la semántica operacional de la P.L.R., en donde el criterio para decidir entre argumentos contradictorios no permite elementos incomparables. Aunque dicha semántica modela el análisis dialéctico, la noción de argumento quedó indefinida. En otras palabras, se asume que el conjunto de argumentos para un literal es dado por algún oráculo. El hecho de considerar a los argumentos como entidades abstractas cuyos roles están determinados por alguna relación de *ataque* entre argumentos, es común a la mayoría de las semánticas declarativas ya existentes [Dun95].

Esto motivó una caracterización declarativa de la definición procedural de argumento. El estudio presentado en [CS00c] está basado en un concepto introducido, en primer instancia por Tarski para la lógica clásica y, luego adaptado por Lifschitz para programas lógicos básicos.

El objetivo de este trabajo es presentar la estructura final de la semántica declarativa trivaluada $\mathcal{GS}$ a través de la que se determina el conjunto de las consecuencias de un programa lógico rebatible donde los argumentos también son analizados de una forma declarativa.

Semántica declarativa

La semántica basada en juegos ha sido utilizada para dar una construcción independiente de la sintaxis de una variedad de lenguajes de programación. Existen dos enfoques *HO-games* [HO94] y *AJM-games* [Abr97, AM97]. Este último, que fue introducido con el objeto de modelar a la interacción entre dos participantes (el Sistema y el Ambiente) como un juego, es el fundamento de la

semántica $\mathcal{GS}$ definida para la P.L.R. Esta decisión tiene su motivación en la similitud entre el árbol que especifica al conjunto de todas las posibles movidas legales en un juego y el árbol dialéctico que se debe realizar cada vez que queremos verificar si un literal pertenece a las consecuencias de un programa.

Sea $\mathcal{P}$ un programa lógico rebatible, cuyo lenguaje es el conjunto de todos los literales fijos que se puedan formar con la signatura de $\mathcal{P}$, y que notaremos *Lit*. Con el objeto de determinar la semántica declarativa deberemos primero encontrar el conjunto de todos los posibles argumentos para todos los literales de *Lit*, a través de las consecuencias estrictas de $\mathcal{P}$ [CS00c]. Dados dos argumentos podremos determinar declarativamente si son contraargumentos considerando si la consecuencia estricta es inconsistente, *i.e.*, si genera el lenguaje completo.

Así, dado un literal *q* construiremos la familia de todos los juegos para *q*. En esta familia no habrá dos juegos que comiencen con el mismo argumento que soporte a *q*. Las posibles movidas de estos juegos serán los argumentos hallados anteriormente y una jugada involucrará que el participante que tiene el turno juegue un contraargumento de la jugada anterior.

El conjunto de las consecuencias de $\mathcal{P}$ estará compuesto por aquellos literales en cuya familia de juegos existe al menos un juego ganado [CS00a] por el proponente del argumento inicial a favor de *q*.

Dado que el valor de verdad de cualquier literal en *Lit* depende del programa completo, no podremos asegurar que si *q* es una consecuencia de $\mathcal{P}$, la negación fuerte de *q*, *~q*, sea falso bajo esta semántica. Por lo tanto, consideraremos dos casos más. Un literal *q* tendrá valor de verdad *falso* si no existe en la familia ningún juego ganado por el proponente y *q* tendrá valor de verdad *desconocido* si la familia de juegos es vacía.

La semántica $\mathcal{GS}$ queda definida como $\langle V, F \rangle$ donde *V* contiene las consecuencias de $\mathcal{P}$, i.e., los literales cuyo valor de verdad es *verdadero* , *F* contiene aquellos literales que son *falsos* y $Lit - (V \cup F)$ es el conjunto de literales cuyo valor de verdad es *desconocido*.

Se ha probado [CS00b] que la semántica $\mathcal{GS}$ es sensata y completa con respecto a la semántica operacional definida para la programación en lógica rebatible, siempre y cuando la relación de preferencia entre contraargumentos sea total.

Conclusiones y Trabajos Futuros

Se ha presentado la estructura final de la semántica $\mathcal{GS} = \langle V, F \rangle$, en donde cada argumento involucrado en el juego es determinado en forma declarativa, prescindiendo de la relación estática de ataque. Las consecuencias de un programa lógico rebatible obtenidas a través del análisis dialéctico coincide con el conjunto *V* de la semántica definida.

Este resultado parcial motiva el siguiente paso en la investigación que involucra la eliminación de la restricción sobre el criterio de decisión entre argumentos. En particular, es nuestro interés estudiar el criterio de preferencia entre argumentos basados en la especificidad.

La semántica $\mathcal{GS}$ se adecua al sistema rebatible estudiado [Gar00]. Entre nuestros trabajos futuros se encuentra también, la posibilidad de adaptar dicha semántica a otros sistemas, como el propuesto en [AM98].

Referencias

[Abr97] Samson Abramsky. Semantics of Interacion. In A. Pitts. and P. Dibyer, editors, Semantics and Logic Computation. Cambridge, 1997.

[AM97] Samson Abramsky and Guy McCusker. Game Semantics. In *Proceedings of Marktoberdorf'97 - Summer School*, 1997.

[CDSS00] C. Chesñevar, J. Dix, F. Stolzenburg and G. Simari. Relating defeasible and normal programming through transformation properties. In *Proceedings of CACiC'2000*. Ushuaia, 2000.

[CS99] Laura A. Cecchi and Guillermo R. Simari. Game-based approach for modeling dialectical analysis. In *Proceedings of CACiC'99*.Tandil, 1999.

[CS00a] Laura A. Cecchi and Guillermo R. Simari. Una semántica declarativa basada en juegos para la programación en lógica rebatible básica. In *Proceedings of ICIE 2000*, 2000.

[CS00b] Laura A. Cecchi and Guillermo R. Simari. Análisis de la semántica declarativa trivaluada GS para la programación en lógica rebatible básica. Resumen. WICC *2000 – ATIA*, La Plata, 2000.

[CS00c] Laura A. Cecchi and Guillermo R. Simari. Sobre la Relación entre la Definición Declarativa y Procedural de Argumento. In *Proceedings of CACiC 2000*, Ushuaia, 2000.

[Dix95a] Jürgen Dix. A classification theory of semantics of normal logic programs : I. Strong properties *Fundamenta Informaticae*, XXII(3) :227-255, 1995.

[Dix95b] Jürgen Dix. A classification theory of semantics of normal logic programs : II. Weak properties *Fundamenta Informaticae*, XXII(3) :257-288, 1995.

[Dix95c] Jürgen Dix. Semantics of logic programs : Their intuitions and formal properties. An overview. In André Fuhrmann and Hans Rott, editors, *Logic, Action and Information*, páginas 227-312. Gruyter, 1995.

[Dun95] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77 :321-357,1995.

[GS99] A. García and G.R. Simari. Strong and Default Negation in Defeasible Logic Programming. In *4th Duth/German Workshop on Nonmonotonic Reasoning Techniques and their applications*, Amsterdam, 25-27, Marzo 1999.

[GSC98] A. García , G.R. Simari and Carlos Chesñevar. An Argumentative Framework for Reasoning with Inconsistent and Incomplete Information. In *ECAI 98, Proceedings of Workshop on Practical Reasoning and Rationality, 13th European Conference on Artificial Intelligence*, Brighton, England, *1998.*

[Gar00] A. García. Programación en lógica rebatible : Lenguaje, Semántica operacional y Paralelismo, Tesis Doctoral, 2000.

[HO94] M.Hyland and L. Ong. On full abstraction for PCF:I,II,III. Technical report, Draft disponible a través de ftp en theory.doc.ic.ac.uk, 1994.

[JV99] H. Jakobovits and D. Vermeir. Dialectic semantics for argumentation frameworks. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, Oslo, Norway, 1999.

[KT99] A. Kakas And F. Toni Computing argumentation in logic programming. *Journal of Logic and Computation,* 9(4):515-562, 1999.