Plan de formalización del álgebra de Kleene concurrente en Isabelle/HOL

November 11, 2023

Introducción

Las álgebras de Kleene son álgebras utilizadas para modelar la ejecución de un programa que tienen aplicaciones a la verificación formal.

Como los objetos matemáticos que representan a los programas en este y otros formalismos relacionados tienen varios modelos, es posible interpretar los elementos del álgebra usando un modelo concreto para resolver problemas formulados en el *setting* general explotando teoremas, resultados y herramientas ya disponibles para una interpretación concreta.

Por ejemplo, los programas secuenciales pueden modelarse con cualquier monoide, como la concatenación de cadenas, la multiplicación de matrices, o la composicion de relaciones binarias, y eso permite decidir la teoría ecuacional del álgebra de Kleene de manera eficiente usando autómatas [2], o formular algoritmos matriciales sencillos de *model checking* para lógicas modales dinámicas expresivas, y obtener y demostrar su complejidad computacional apelando solamente a la complejidad de la multiplicación de matrices [4].

En particular, el álgebra de Kleene concurrente extiende el álgebra de Kleene 'cruda' con una operación para la composición concurrente de programas, y sus modelos o bien relajan la condición de que los eventos en una traza de ejecución estén totalmente ordenados (i.e. considerando las trazas como multiconjuntos parcialmente ordenados) o permiten intercalar 'al azar' los eventos de dos procesos separados en una sola traza de ejecución (i.e. interpretando los procesos como *shuffle languages*).

Estos dos modelos tienen diferencias relevantes en las aplicaciones de verificación. Por ejemplo, si se opta por interpretar un proceso con un *pomset*, se puede modelar la interacción entre componentes de un sistema que ejecutan acciones que pueden solaparse temporalmente, pero si se modela un proceso con

un *shuffle language* se excluye la posibilidad de representar de forma directa eventos no atómicos.

El objetivo de este proyecto de formalización será definir en Isabelle/HOL el álgebra de Kleene concurrente, definir los *shuffle languages*, y probar que son modelos del álgebra.

La formalización imitará la estructura de las formalizaciones existentes del álgebra de Kleene con tests [1], y usará las definiciones de [3].

Plan de trabajo

En términos breves, el plan de este proyecto es:

- Definir las estructuras algebraicas sobre las que descansa la definición del álgebra de Kleene concurrente (i.e. semianillos y *quantales*)
- Definir el álgebras de Kleene concurrente como una *typeclass*
- Probar que los *shuffle languages* son un modelo concreto del álgebra instanciando esa clase
- Si queda tiempo, definir los pomsets (o un subonjunto de sus operaciones asociadas) y probar que son instancias de la clase

Como la formalización de Armstrong, Struth y Weber ya define los semianillos y prueba varios lemas comunes, vamos a importar y extender sus definiciones para definir los *quantales*.

Álgebra de Kleene concurrente

Un *quantale* es un semianillo con adición idempotente, equipado con un órden definido por la equivalencia $a \leq b \iff a + b = b$ (que se denomina 'órden natural' y se denota con $\leq$), y que cumple además que los elementos de S forman un reticulado completo bajo $\leq$ (i.e. todo subconjunto de S tiene definido un supremo respecto del orden natural), y la multiplicación distribuye para el supremado de subconjuntos arbitrarios de S.

Un álgebra de Kleene concurrente es una estructura $\langle S, +, 0, *, ;, 1\rangle$ tal que $\langle S, +, 0, *, 1\rangle$ y $\langle S, +, 0, ;, 1\rangle$ son *quantales* relacionados por la ley de intercambio $(a * b); (c * d) \leq (b; c) * (a; d)$.

La idea es que el operador $*$ representa la composición concurrente, y el operador ; representa la composición secuencial, y que, bajo la interpretación concreta de los elementos del álgebra como conjuntos de trazas de ejecución o

conjuntos de cadenas, la suma es la unión de conjuntos y el órden natural es la inclusión.

La ley de intercambio es una versión más débil (una inecuación y no una ecuación) de la propiedad distributiva para la composición secuencial y concurrente.

Shuffle Languages

Los *shuffle languages* son lenguajes regulares para los que se define adicionalmente un operador de intercalado, que puede caracterizarse con una función recursiva:

$$\diamond : \Sigma^* \to \Sigma^* \to \mathcal{P}(\Sigma^*)$$
$$s \diamond \epsilon = s$$
$$\epsilon \diamond t = t$$
$$as \diamond bt = a(s \diamond bt) \cup b(as \diamond t)$$

La unión, la clausura de Kleene y la concatenación de lenguajes están definidos 'igual que siempre', y la operación de intercalado se puede generalizar a lenguajes de intercalado. La definición habitual del intercalado de lenguajes es $L_1 \diamond L_2 = \{s \diamond t : s \in L_1 \wedge t \in L_2\}$.

Introducir un operador de intercalado permite representar la concurrencia como el intercalado de eventos de dos procesos que se desarrollan de manera concurrente (como sucede con la ejecución de dos programas que ejecutan instrucciones de código de máquina en un mismo procesador).

References

[1] Alasdair Armstrong, Georg Struth, and Tjark Weber. "Program Analysis and Verification Based on Kleene Algebra in Isabelle/HOL". In: (2013). Ed. by Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, pp. 197–212.

[2] Thomas Braibant and Damien Pous. "An Efficient Coq Tactic for Deciding Kleene Algebras". In: (2010). Ed. by Matt Kaufmann and Lawrence C. Paulson, pp. 163–178.

[3] C. A. R. Tony Hoare et al. "Concurrent Kleene Algebra". In: (2009). Ed. by Mario Bravetti and Gianluigi Zavattaro, pp. 399–414.

[4] Martin Lange. "Model checking propositional dynamic logic with all extras". In: *Journal of Applied Logic* 4.1 (2006), pp. 39–49. ISSN: 1570-8683. DOI: https://doi.org/10.1016/j.jal.2005.08.002. URL: https://www.sciencedirect.com/science/article/pii/S1570868305000637.