

Computación Bio-inspirada

David Orellana Martín

Noviembre de 2023

Reinas de la Colina, o *Queens of the Hill*, es un juego inspirado en el juego de programación *Core Wars* ¹, en el que distintos competidores incluyen su programa en una memoria que tiene un conjunto de instrucciones específico. El objetivo de cada competidor es que su programa sea el último rival vivo. Para ello, se tienen que valer de la capacidad de las instrucciones para modificar el contenido de otras posiciones de memoria y, así, eliminar instrucciones del enemigo o, incluso, de hacer *evolucionar* sus propias instrucciones.

Queens of the Hill captura esta idea de **competición** al campo de la *computación con membranas*, en cuyo caso cada participante incluye su sistema P (llamado en este caso valquiria) en un supersistema de membranas que contendrá todas las valquirias. En [1] ² está definida la sintaxis y semántica de las valquirias, que como se puede observar, no son más que sistemas P de transición con algunos “ingredientes extra”. En particular, la sintaxis es prácticamente igual a estos, aunque con algunas particularidades.

Por ejemplo, todos los conjuntos de reglas R_i (correspondiente a la membrana etiquetada por i) contienen un conjunto de reglas especial R_k^δ , definido de la siguiente manera:

$$R_k^\delta = \{\delta_t \rightarrow \delta_{t-1} \mid 2 \leq t \leq k\} \cup \{\delta_1 \rightarrow \delta\} \cup \{\delta_t \bar{\delta}_t \rightarrow \lambda \mid 1 \leq t \leq k\},$$

teniendo la regla $\delta_t \bar{\delta}_t \rightarrow \lambda$ prioridad sobre la regla $\delta_t \rightarrow \delta_{t-1}$ y $\delta_1 \rightarrow \delta$. Además, no existe prioridad de reglas en las reglas de $R_i \setminus R_k^\delta$. Si en una membrana aparece un objeto δ en un paso de computación, la membrana será disuelta al finalizar dicho paso de computación, teniendo los mismos efectos que tiene la disolución en los sistemas P de transición.

Un torneo es una competición entre m participantes, que lucharán para ver qué valquiria sobrevive al final de la computación (se establece un límite de N pasos para una computación). En realidad, debido al carácter no determinista de estos sistemas, parece razonable realizar varias computaciones y realizar una media para obtener una puntuación que tenga sentido (y no sea cuestión de suerte de una simple computación). Supongamos que tenemos los sistemas Π_1, \dots, Π_m como los participantes de un torneo. Dichos participantes comparten el alfabeto de trabajo. Para realizar una computación del supersistema, debemos introducir estas valquirias en una membrana externa a ellos (etiquetada por 0), de tal manera que la membrana piel de cada una sea hija de esta membrana. Además, para que las etiquetas de las membranas de distintas valquirias no sean iguales, las cambiamos de tal manera que la membrana j de la valquiria Π_i quedará reetiquetada como (i, j) . Todas las valquirias pueden enviar objetos a esa membrana 0 que tendrá disponible el conjunto de reglas

$$\{a \rightarrow (a, in) \mid a \in O\} \cup R_k^\delta,$$

que enviará los objetos que se envíen al entorno a cualquier membrana hija de dicha membrana 0 (sí, incluso los objetos δ_t , aunque estos también pueden evolucionar mediante la ejecución de las reglas R_k^δ).

En el artículo citado se pueden ver varios ejemplos de valquirias con diversos comportamientos interesantes. En estos ejercicios propondremos algunos otros sistemas que puedan resultar relevantes para estudio y, dado el caso, para un torneo.

¹<https://www.corewars.org/>

²El artículo se puede descargar en http://www.gcn.us.es/files/19bwmc/041_Queens.pdf.

1. Supongamos que somos participantes en un torneo, y nuestro sistema Π_1 recibe, en cada paso de computación, c objetos δ_3 . ¿Cuál es la valquiria mínima capaz de sobrevivir a este ataque?
2. Ahora supongamos que seguimos las reglas descritas arriba. Supongamos que somos dos competidores, Π_1 y Π_2 , y Π_2 envía al entorno c objetos δ_3 . Si, en lugar de usar no determinismo, la aplicación entre las distintas reglas que compiten por el mismo objeto fueran sucesos equiprobables, ¿qué probabilidad habría de que al menos un objeto $\delta_t (1 \leq t \leq 3)$ llegase a Π_1 .
3. Nuestro sistema Π_1 es capaz de enviar objetos δ_t al exterior, pero por el no determinismo, estos objetos pueden volver a nuestra valquiria, poniéndola en peligro. ¿Existe alguna manera eficaz de defendernos de nuestros propios ataques?
4. ¡Pasamos al ataque! Es el momento de enviar algunos objetos δ . Supongamos que nuestra valquiria es Π_1 , y hay otros $m - 1$ competidores. Sin preocuparnos de las defensas, y pudiendo usar el número de membranas que se desee, crear un sistema que pueda atacar a los demás de la mejor manera posible. Se intentará minimizar el número de objetos en la *parte derecha de la regla*.
5. Ahora, todo junto. Vamos a intentar diseñar un sistema mínimo capaz de atacar y a la vez defenderse de los ataques de $m - 1$ contrincantes. Se valorará la minimización, de nuevo, de los objetos en la *parte derecha de la regla*.
6. Una sola regla para gobernarlos a todos. Diseñar una valquiria con una sola membrana y una sola regla, sin escatimar en los objetos en la *parte derecha de la regla*, capaz de destruir a los contrincantes mientras se defiende de ellos.

Referencias

- [1] Artiom Alhazov, Sergiu Ivanov, and David Orellana-Martín. Queens of the hill. In David Orellana-Martín, Agustín Riscos-Núñez, and Gheorghe Păun, editors, *Proceedings of the Nineteenth Brainstorming Week on Membrane Computing*, pages 21–32. Artes Gráficas Moreno, S.L., February 2023.