

Paralelización de Analizador de Niveles de Puzzle

Francisco J. A. Casas B.

13 de julio del 2017

Estrategia

Se busca poder estimar la **dificultad** de niveles de un puzzle, determinado por ciertas **reglas**, lo que se puede usar para generar niveles aleatorios que cumplan con **requisitos de dificultad**.

Se contruye un **grafo** con los estados alcanzables a partir del estado inicial, con el que se pueden obtener varias indicadores de la dificultad de un nivel, por ejemplo:

- Cantidad mínima de decisiones para ganar.
- Formas diferentes de ganar.
- Probabilidad de ganar realizando movimientos aleatorios.
- Si se deben mover todas las piezas para obtener la victoria más rápida.

Estrategia

Como es necesario saber cuál es la cantidad mínima de elecciones para alcanzar cierto estado, se realiza **búsqueda en amplitud**.

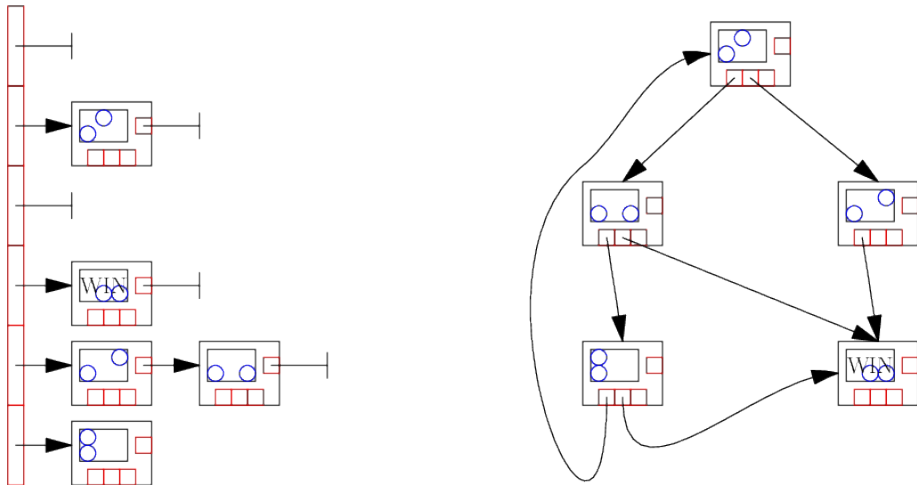
Si se alcanza un estado al que ya se llegó de otra manera, no se expande esa parte del árbol otra vez.

Sólo se almacenan los estados significativos, que resultan en una **decisión** o **victoria**, no los estados *de paso*.

Implementación

- Se tiene una **hash table** que almacena listas enlazadas de nodos, que sirve para comprobar si un estado existe ya o no.
- En cada nivel de profundidad se tiene una **cola** de todos los estados pendientes por revisar y se construye la **cola** del nivel de profundidad siguiente.

Implementación



Paralelización

Se requerían las siguientes sincronizaciones:

- Exclusión mutua al hacer pop en la **cola** actual.
(Un **mutex** para la cola actual).
- Exclusión mutua al hacer push en la **cola** siguiente.
(Un **mutex** para la cola siguiente y las propiedades del árbol).
- Exclusión mutua al comprobar si un estado existe en la **hash table** y agregarlo si no.
(Un **mutex** por cada lista enlazada de la **hash table**).
- Una **barrera** una vez se completa el nivel de profundidad, necesario para que todos los estados se registren en su nivel de profundidad menor.
(El thread principal crea los threads en cada nivel de profundidad y estos terminan cuando se vacía la **cola** actual, luego hace *join*).

Tiempos

Threads	Puzzle simple	Puzzle complejo
Sin MT	0.019	4.331
1	0.037	4.494
2	0.037	2.342
3	0.037	1.657
4	0.037	1.306
5	0.037	1.327
6	0.037	1.358
7	0.038	1.486
8	0.038	1.638
9	0.038	1.686
10	0.038	1.689
20	0.039	1.693
40	0.042	1.717
80	0.048	1.761