Diario de Desarrollo

Gustavo Rivas-Gervilla

21 de marzo de 2020

Índice

Bitácora

20/03/2020

En el día de hoy vamos a trabajar en el objetivo O1, hemos de tener en cuenta que un retículo se puede almacenar en forma de grafo, ya que tenemos distintos nodos (los conceptos formales del contexto) que están relacionados entre sí en una jerarquía padre-hijo (que en nuestro caso vendrá dada por la relación de inclusión que se da entre conceptos formales). Así, en la Figura 1 tenemos un retículos en el que los conceptos han sido representados por medio de nodos en un grafo, donde podemos observar las distintas relaciones de inclusión que se dan entre los conceptos formales.

Esto da lugar al requisito r1 que es el primero que debemos afrontar de cara a desarrollar la base sobre la que se sustentará nuestra aplicación.

Así vamos a tener por un lado la clase `Lattice` que se encargará de soportar toda la estructura del retículo, y la clase `Node` que representará a cada uno de los nodos que conforman el retículo, es decir, cada uno de los nodos que podemos ver en la Figura 1.

Además, queremos tener un acceso rápido a los conceptos de nuestro retículo, con lo cual lo que vamos a emplear será una tabla hash con la cual tener indexados nuestros conceptos formales, por medio de su *extent*, como se ilustra

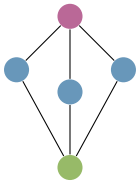

Figura 1: Ejemplo de un retículo visto como un grafo donde podemos ver tres *niveles* distintos de conceptos formales.

en la Figura 2. Y será cada `Node` el que se encargue de almancenar la información de quiénes son sus padres y sus hijos.

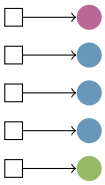

Figura 2: La clase `Lattice` contendrá una tabla hash en la que indexará los conceptos formales del retículo en base a su *extent*.

Hay que tener en cuenta que no podemos usar el número que en binario representa el *extent* visto como un `bitset` ya que, para por ejemplo un contexto en el que consideremos 100 objetos, el número 2^{100} no puede ser almanceando en un `int`. Y tomar otra aproximación como `long long int` no creo que sea una solución adecuada, ya que seguiríamos estando limitados.

Sería iteresante implementar un `Lattice::iterator` el cual recorriese el retículo, no solo por un orden usual para recorrer grafos, como puede ser el preorden, sino también un orden marcado por una medida de bondad de conceptos formales. Ya vimos en el TFM que existían varias *interestingness measures* para conceptos formales. También puede ser una medida que venga definida por el usuario, en base a sus preferencias para construir una descripción de la escena.

Con esto vamos a comenzar a trabajar en el issue #1.

21/03/2020

Hoy seguimos trabajando en el issue #1, dentro del objetivo O1, para ello vamos a proseguir con el trabajo sobre las clases `Lattice` y `Node`. Hay que tener algo en cuenta y es que ahora mismo estamos trabajando con un tamaño máximo para los `bitsets`, y por tanto para los contextos considerados `const int TMAX = 100;`, es decir, sólo podemos considerar contextos de una dimensión 100×100.

Evidentemente podemos aumentar el tamaño de `TMAX` pero esto tiene un efecto sobre el tiempo de ejecución del programa, y no tiene que ver únicamente con el tamaño del contexto en sí. Si trabajamos con un contexto 100×100, pero empleando un `bitset<200>` veremos cómo el tiempo de ejecución de la aplicación aumenta. Es decir, el tamaño de las operaciones con `bitsets` de mayor tamaño es más costosa que para `bitsets` más pequeños. Con lo cual emplear el tamaño adecuado de `bitset` es importante. Esto da lugar al issue #2.

Lo que hemos hecho para implementar la tabla hash es usar un `std::unordered_map` que como clave tendrá un `string`, aprovechando el método `bitset::to_string`.

A. Objetivos y Análisis de Requisitos

A.1. Objetivos

O1 Obtener el retículo de conceptos correspondiente a un contexto formal.

O2 Definir distintas medidas de bondad para las *referring expressions* obtenidas.

O3 Refinamiento mediante propiedades contextuales.

A.2. Análisis de Requisitos

r1 Permitir trabajar con la estructura de grafo que representa el retículo de conceptos formales de un contexto formal. Relacionado con el objetivo O1.

A.3. Diagramas de Casos de Uso

B. Diagrama de Clases