

Proyecto #2

Representación del conocimiento

IC-6200 Inteligencia Artificial
Ingeniería en Computación
Instituto Tecnológico de Costa Rica
Prof. Diego Munguía

El objetivo es desarrollar un sistema experto en el área de la ingeniería de software, específicamente en diseño de software. Se construirá un programa en Prolog que analizará código fuente Java y determinará si el código cumple con algún patrón de diseño.

Se trabajará con un subconjunto de Java que incluye lo siguiente:

- Declaración de clases e interfaces con todos los posibles modificadores.
- Declaración de métodos con todos los posibles modificadores.
- Los tipos de datos soportados son `void` , `String` y cualquier clase.
- El cuerpo de un método permite declaraciones locales, asignaciones, llamadas a métodos, el operador `new` y los keywords `return` , `super` y `this` . Los métodos de tipo `void` pueden tener cuerpo vacío. No es necesario soportar expresiones más allá de las llamadas a métodos y literales `String` , tampoco es necesario soportar ningún tipo de estructura de control.

Debe realizar las siguientes tareas:

1. Construir un parser que lea código fuente Java y produzca una base de conocimiento (una serie de hechos/relaciones) en Prolog describiendo el programa en cuestión.
2. Construir un programa en Prolog que contenga las reglas que describen los patrones de diseño.

El analizador utilizará los hechos producidos por el parser de Java y las reglas de reconocimiento de patrones de diseño para determinar si el código analizado cumple con uno o más patrones de diseño.

Para construir el parser de Java puede utilizar `antlr` y basarse en alguna gramática ya existente, por ejemplo: <https://github.com/antlr/grammars-v4/blob/master/java/Java.g4> (<https://github.com/antlr/grammars-v4/blob/master/java/Java.g4>). Puede suponer que los programas a analizar siempre serán correctos en todo sentido -léxico, sintáctico y semántico-, no es necesario programar ningún tipo de manejo de errores.

Con respecto a los patrones que el analizador debe identificar, puede escoger cuatro de la siguiente lista:

- Abstract factory
- Builder
- Factory method
- Prototype
- Adaptor
- Composite
- Command
- Null object
- Visitor

Puede integrar el parser en Java con el programa en Prolog utilizando JPL (<https://github.com/dmunquiatec/ic4700/tree/master/JPL> (<https://github.com/dmunquiatec/ic4700/tree/master/JPL>)), o también puede hacerlo a mano, es decir el parser en Java produce como respuesta una hilera en pantalla que contiene el código Prolog de la base de conocimiento, copiarlo manualmente a un archivo junto con las reglas para reconocer los patrones de diseño y ejecutarlo desde Prolog. El primer acercamiento es ideal pero no necesario.

La interfaz esperada será la siguiente:

```
?- patron(X).  
X = builder.  
X = visitor.
```

```
?- patron(X).  
false.
```

El proyecto debe ser trabajado en parejas, fecha de entrega miércoles 18 de mayo 2016 a más tardar a las 11:59 pm. El proyecto será trabajado en un repositorio privado de gitlab.

