

Awk para “text mining” del Plan de Estudios

Objetivos

- Este es un pequeño proyecto de minería de textos cuyo propósito es estudiar un lenguaje de propósito específico (awk) en el marco del curso. Dicho lenguaje usa un paradigma mixto (entre imperativo/basado en reglas) da muchas facilidades para la tarea de minería de texto como la que se explica a continuación.
- Ejercitar algunas nociones de compilación
- Ejercitar programación declarativa para visualizar grafos
- Dar una primera iteración en Prolog
- Como objetivo colateral este trabajo debe permitirle evaluar el trabajo de su grupo para refinar la constitución de los grupos de trabajo definitivos para los demás proyectos del curso.

Tareas

Desarrollar un pequeño “microcompilador” que tome el plan de estudios que se le entrega en formato textual y produzca dos resultados independientes:

1. Un grafo en formato graphviz como el que se le entrega de modelo que permita ver las dependencias entre los cursos de la malla curricular (ver caso de uso más adelante).
2. Un programa SWI-Prolog que permita hacer consultas a la malla (ver caso de uso esperado adelante)

Material de Apoyo

- Se adjunta base de proyecto esperado
- Se adjunta ejemplo de salidas esperadas a como fueron generadas por el profesor.
- Se dará una clase de explicación de todo el material

Evaluación y Valor

- Se revisará siguiendo guía que se comunicará con antelación.
- Vale 4 quices (2 quices cada parte).

Extras

Pueden obtenerse puntos extra si se añaden salidas y resultados adicionales a los pedidos. Consulte con el profesor sobre posibles opciones. Máximo un 15% sobre la nota, siempre que lo que es obligatorio funcione según lo pedido.

Forma de Trabajo

Grupos de 4 (no tiene que ser el grupo definitivo de proyectos del curso). Si son más debe ser autorizado por el profesor (sólo en caso justificado).

Entregables

- Un pequeño proyecto que desde una consola permita fácilmente producir y visualizar los resultados pedidos.

- El entregable debe seguir la estructura de proyecto del material que se le entrega.
- La forma de entrega del proyecto se comunicará oportunamente.
- Todo entregable debe llevar los nombres, id de los autores.

Fechas

Se indicará oportunamente. Tentativa: 20 de agosto

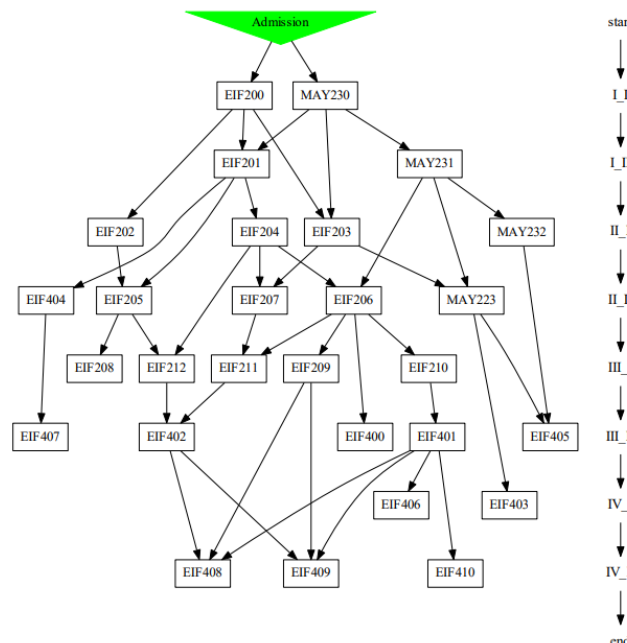
Salidas (Casos de uso mínimos)

El plan de estudios textual es `study_plan_input/study_plan.txt`. Ese archivo es “read-only” Ud. no puede modificarlo (lo contrario anula su proyecto).

El Shell `runall.sh` correría la compilación completa. Las salidas esperadas se indican en dicho archivo. La corrida desde Shell de Cygwin

```
una@DESKTOP-I135CQR /cygdrive/d/UNA/Paradigmas/
$ ./runall.sh
*** Generating Graphviz and Prolog for study p
*** Generating composed plan ***
*** Generating Graphviz ***
*** Generating Prolog ***
una@DESKTOP-I135CQR /cygdrive/d/UNA/Paradigmas/
$ |
```

Un modelo del grafo de dependencias a generar se encontraría en el directorio `gv_output`. Este grafo solo muestra nodos que sean EIF.



El caso de uso de `Prolog` debe permitir generar la lista de todos los cursos que son hojas (`all_leaves`) en ese grafo. Son cursos que no son requisitos de ninguno otro. La Interacción esperada (desde consola Windows) que calcula todas las 32 hojas del grafo es la siguiente:

```
GV:swipl src/queries.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- all_leaves(AL), length(AL, S).
AL = ['EstudiosGeneralesI', 'EstudiosGeneralesII', 'EstudiosGeneralesIII', 'EstudiosGeneralesIV', 'LIX412', 'EIF400', 'EIF405', 'EIF407', 'EIF406', 'OptativaI', 'OptativaII', 'EIF403', 'EIF408', 'EIF409', 'OptativaIII', 'OptativaIV', 'EIF410', 'EIF1000', 'EIF4200', 'EIF4210', 'EIF4220', 'EIF4240', 'EIF4260', 'EIF4270', 'EIF4280', 'EIF4310', 'EIF4330', 'EIF4350', 'EIG4160', 'EIF4300', 'EIG4170', 'EIG4180'],
S = 32.

2 ?-
```