Programación Declarativa Avanzada

Introducción

Juan Rodríguez Hortalá

Acerca de la asignatura

Juan Rodríguez Hortalá

juanrh@fdi.ucm.es

despacho 220

tutorías: L,M,J,V de 14:00 a 16:00

material docente y avisos a través del

Campus Virtual

Programación Declarativa

Programación Declarativa
vs
Programación Imperativa

Programación imperativa: C/C++, Java, ...

- Cómo: el programa especifica paso a paso como llevar a cabo una tarea
- Una receta de cocina paso a paso
- El uso de frameworks o librerías aumenta el grado de abstracción: oculta detalles del cómo

Programación declarativa: Haskell (PF), Prolog (PL),...

- Qué: en el programa se especifica qué resultado queremos obtener, sin dar demasiados detalles acerca de cómo calcularlo
- Ir a una buena pastelería y contarle al pastelero el tipo de tartas que te gustan: él elegirá los ingredientes y la cocinará para ti, basándose en las reglas que le diste
- Elevando el nivel de abstracción desde las propias construcciones del lenguaje

Características declarativas o imperativas en cada lenguaje de programación

Organización de la asignatura

Contenidos

Continuación de Programación Lógica y Programación Funcional

Veremos tres variantes más del paradigma declarativo

- Programación Lógica con Restricciones (CLP): Sicstus Prolog CLP(FD) y CLP(R)
- Programación Lógico-Funcional: el lenguaje Curry
- Programación Funcional Concurrente: el lenguaje Erlang

Programación Lógica con Restricciones

CLP extensión de la programación lógica con nuevas primitivas para expresar relaciones entre elementos de cierto dominio = restricciones

```
| ?- {X > 0, X =< 20, Y >0, 2*Y =< X}, maximize(Y) .
X = 20.0,
Y = 10.0 ? ;

| ?- X #> 0, X#<3, labeling([], [X]) .
X = 1 ? ;
X = 2 ? ;
no
```

Aplicación típica: problemas de asignación de recursos y optimización

implementación en el sistema Sicstus Prolog
http://www.sics.se/isl/sicstuswww/site/index.html

Programación Lógico-Funcional

PLF extensión de Haskell (programación funcional perezosa y pura) con

- Variables libres y unificación: estrechamiento
- Funciones indeterministas
- Restricciones

```
sublist xs ys | us ++ xs ++ vs =:= ys = True
   where us, vs free

Sublist> sublist xs [1,2] where xs free
Result: True
Bindings: xs=[]
More solutions? [Y(es)/n(o)/a(ll)]
Result: True
Bindings: xs=[1]
Result: True
Bindings:
xs=[1,2]
...
```

Aplicación a problemas de búsqueda

el lenguaje Curry y el sistema PAKCS

http://www.informatik.uni-kiel.de/~pakcs/

Programación Funcional Concurrente

Erlang un lenguaje funcional impaciente, impuro y con tipado dinámico diseñado para construir sistemas soft real-time masivamente escalables y con requerimientos de alta disponibilidad

- Soporte integrado para la programación concurrente, distribuida y tolerante a fallos

- Concurrencia basada en el “actor model”

- Desarrollado por Ericsson y utilizado por empresas como Ericsson, Amazon, Facebook, T-Mobile, Telia, Nortel y proyectos como la Apache Software Foundation y Ubuntu

Programación Funcional Concurrente

```
go() ->
     Pid2 = spawn(echo, loop, []),
     Pid2 ! {self(), hello},
     receive
          {Pid2, Msg} ->
               io:format("Main process received '~w'~n",
                         [Msg])
     end,
     Pid2 ! stop.
```

```
loop() ->
     receive
          {From, Msg} ->
               From ! {self(), Msg},
               loop();
          stop ->
               io:format("Echo server
stopped~n"),
               ok
     end.
```

```
9> echo:go() .
Main process received 'hello'
Echo server stopped
stop
```

Desarrollado por Ericsson para su aplicación a telecomunicaciones, también utilizado en banca, comercio electrónico, bases de datos distribuidas, ...

el lenguaje Erlang
http://www.erlang.org/

Evaluación

Tres prácticas: CLP, Curry y Erlang

- Pequeño programa
- Grupos de dos
- Entrega en el Campus Virtual
- Presentación en el despacho