Sistemas de producción en CLIPS. Control de ejecución

Virginia Francisco Rubén Fuentes Alberto Díaz Carmen Fernández





El control de ejecución en CLIPS

- Control de ejecución:
 - En CLIPS las reglas deben actuar de manera oportuna cada vez que son aplicables
 - Sin embargo, a veces es necesario conseguir cierta secuencialidad en la ejecución
- Existen distintos métodos para controlar la ejecución:
 - Hechos de control
 - Establecer prioridades
 - Módulos

Fases y Hechos de control

- Control empotrado en las reglas
 - Ejemplo Juego de los palillos (sticks.clp): Hecho que indica el turno en un juego entre el computador y un usuario
 - Elección del jugador que empieza, Elección del número de piezas
 - Turno del humano, Turno de la computadora
 - Hechos de control:
 - (fase elige-jugador), (fase -elige-numero-de-piezas)
 - (turno h), (turno c)
 - Desventaja:
 - Se mezcla el conocimiento del dominio con la estructura de control
 Mantenimiento y desarrollo más costoso.
 - Dificultad para precisar la conclusión de una fase

Control con prioridades

- Prioridad para cambiar de fase
 - No recomendable
- Prioridad para reglas especiales
 - Regla inicial
 - Prioridad alta
 - Regla por defecto si no se aplica ninguna de las otras
 - Prioridad baja
- Ejemplo auto.clp
 - Uso de hecho con not para control de activación de reglas

Diseñar las Bases de Reglas: Módulos

- Cuando un problema es grande conviene
 - Dividirlo en subproblemas: cada uno es un módulo
- Un Módulo es un espacio de nombres
 - Base de reglas dividida en bloques o módulos: Mejor diseño
 - Ejemplo: módulo de averías mecánicas y módulo de averías eléctricas
 - Más eficiente la selección de reglas y el razonamiento
 - Ejemplo
 - (defmodule etapaCinco)
 - (defrule etapaCinco::PasoUno ...)
 - (focus etapaCinco) ... (run)
 - La ejecución en Clips se hace por módulos
 - por defecto "module MAIN"
- Ejemplo: taxadvisor.clp

Módulos

- Los módulos representan diferentes estados en la resolución del problema, y aíslan los hechos y reglas.
 - (defmodule etapaCinco)
- Deberemos especificar a qué módulo pertenecen las construcciones con <módulo>::

Módulos (auto-focus)

```
(defrule QUESTIONS::ask-question-by-id
  "Given the identifier of a question, ask it and assert the answer"
  (declare (auto-focus TRUE))
  (question (ident ?id) (text ?text) (type ?type))
  (not (answer (ident ?id)))
  ?ask <- (ask (ident ?id))
  =>
  (bind ?answer (ask-user ?text ?type))
  (assert (answer (ident ?id) (text ?answer)))
  (retract ?ask))
```

- Tutorial Jess: Chapter 5
 (disponible en Campus Virtual)
- Asesor de formularios de impuestos
- Pregunta a un usuario sobre su situación financiera y le recomienda cual es el formulario de impuestos adecuado
- Etapas
 - Bienvenida
 - Preguntar al usuario
 - Decidir qué formularios son adecuados
 - Notificar al usuario la recomendación
- Módulos
 - INTERVIEW, RECOMMEND, REPORT
 - QUESTIONS: Otro módulo para las preguntas
 - Si tengo una pregunta sin respuesta, haz la pregunta
 - Auto-focus TRUE

- Base de conocimiento
 - Tipos de formularios
- Hechos
 - deftemplate user (slot income (default 0)) (slot dependents (default 0)))
 - (deftemplate question (slot text) (slot type) (slot ident))
 - (deftemplate answer (slot ident) (slot text))
 - (deftemplate recommendation (slot form) (slot explanation))

- Base de conocimiento
 - Tipos de formularios
- Hechos
 - Todos en MAIN porque se usan en todos los módulos
 - deftemplate user (slot income (default 0)) (slot dependents (default 0)))
 - (deftemplate question (slot text) (slot type) (slot ident))
 - (deftemplate answer (slot ident) (slot text))
 - (deftemplate recommendation (slot form) (slot explanation))
 - Deffacts con las preguntas
 - (deffacts question-data "The questions the system can ask." (question (ident income) (type number)

```
(text "What was your annual income?"))
(question (ident interest) (type yes-no)
(text "Did you earn more than $400 of taxable interest?"))
```

Módulo interview

- Determina qué preguntas hay que hacer, añadiendo los hechos ask correspondientes.
- Hay reglas que no tienen parte izquierda, se ejecutan siempre.
- Otras si tienen porque dependen de respuestas previas

Ventajas del uso de módulos

- Permiten dividir la base de conocimiento
- Se puede controlar qué hechos son visibles en cada módulo
- Se puede controlar sin utilizar prioridades ni hechos de control
 - Es posible salir de una fase, y volver a la fase posteriormente para ejecutar las instancias que todavía quedan activas en la agenda.

Materiales de referencia

Friedman-Hill, Ernest Jess in action [Recurso electrónico]: rule-based systems in Java Ernest Friedman-Hill. Publicación Greenwich, Conn.: Manning, c2003 http://cisne.sim.ucm.es/record=b2548266~S6*spi

Strauss, Martin
Jess. The Java Expert System Shell. Abril 2007
(tutorial disponible en el Campus Virtual que incluye una discusión en profundidad del ejemplo del taxadvisor)

Materiales de referencia

- Otros enlaces interesantes
 - https://www.cs.us.es/cursos/ia2-2004/temas/tema-01.pdf
 - http://www.lsi.upc.es/~bejar/ia/transpas/teoria/4-SBC3desarrollo-SBC.pdf