



UNIVERSIDAD DE ALMERÍA

Ingeniería de Sistemas Basados en Conocimiento

Sistemas de reglas de producción

Ingeniero en Informática

Departamento de Informática



Agenda

- Introducción
- Elementos de un sistema de producción
- El motor de inferencias
 - Ciclo de ejecución
- Tipos de razonamiento
 - Razonamiento guiado por los datos
 - Razonamiento guiado por los objetivos
- Las reglas como lenguaje de programación
- Las reglas como parte de aplicaciones generales



Introducción

- Lógica de predicados \rightarrow Mecanismo para describir procedimientos
- Describir un problema como un proceso de razonamiento
- Los predicados descomponen el problema en problemas más sencillos e indican orden total o parcial
- Restricción \rightarrow sólo fórmulas lógicas cuantificadas universalmente que se pueden transformar en cláusulas disyuntivas (cláusulas de Horn)
- A este tipo de fórmulas las denominaremos reglas de producción
 - Fórmulas condicionales en las que puede hacer cualquier número de átomos en el antecedente y sólo un átomo en el consecuente
 - Ejemplo: $(mujer(A) \wedge padre(B, A)) \rightarrow hija(A, B)$ — Regla de producción
 $\neg mujer(A) \vee \neg padre(B, A) \vee hija(A, B)$ — Cláusula de Horn



Elementos de un sistema de producción

- Conjunto de reglas a partir del cual podemos obtener una solución mediante un proceso de demostración
- Lenguaje declarativo
 - No se indica explícitamente el algoritmo a utilizar
 - Se indican las condiciones que la solución debe cumplir
- Un problema quedará definido por:
 - Base de hechos: Planteamiento del problema. Predicados primitivos, funciones, relaciones y constantes definidas
 - Base de conocimiento (o de reglas): Reglas que describen los mecanismos de razonamiento que permiten resolver problemas
 - Motor de inferencia: Ejecuta el formalismo y obtiene la cadena de razonamiento que soluciona el problema



El motor de inferencias

- Funciones:
 - Deducir nuevos hechos
 - Ejecutar acciones para resolver el problema (objetivo)
 - A partir de un conjunto inicial de hechos y eventualmente a través de una interacción con el usuario
- Elementos de los que se compone:
 - Intérprete de reglas: debe ser capaz de ejecutar las reglas. Interpretar el antecedente, buscar las instancias para que el antecedente se cumpla y generar la deducción (consecuente). Generar el **conjunto de conflicto** (con todas las reglas que se pueden utilizar en un instante)
 - Estrategia de control: elige la regla u objetivo que debe resolverse. Se basa en una **estrategia de resolución de conflictos**



El motor de inferencias — Ciclo de ejecución

- Fases que definen su funcionamiento:
 - Detección (filtro): **Reglas pertinentes**
 - Obtención del conjunto de instanciaciones de reglas.
Formación del conjunto de conflicto
 - Selección: **¿Qué regla?**
 - Resolución de conflictos: selección de la instanciación a aplicar
 - Aplicación: **Acción**
 - Aplicación de la regla



El motor de inferencias — Ciclo de ejecución (Detección)

- Construcción del conjunto de instanciaciones de reglas (conjunto de conflicto)
- El intérprete de reglas realiza los cálculos e instanciaciones necesarias que son posibles en cada estado de resolución del problema (*matching*)
- Una regla puede instanciarse más de una vez, caso de existir variables



El motor de inferencias — Ciclo de ejecución (Selección)

- Selección de la mejor instanciación
- Las reglas instanciadas son seleccionadas para aplicarse dependiendo de la estrategia de control (estrategia de resolución de conflictos)
 - estrategia fija
 - estrategia dinámica prefijada
 - estrategia guiada por meta-reglas
- Criterios aplicables:
 - primera regla por orden en la Base de Conocimiento
 - la regla más/ menos utilizada
 - la regla más específica/la más general
 - la regla que tenga el grado de certeza más alto
 - la instanciación que satisfaga los hechos: más prioritarios, más antiguos (instanciación más antigua), más nuevos (instanciación más reciente), ...
- Posible combinación de varios criterios



El motor de inferencias — Ciclo de ejecución (Aplicación)

- Ejecución de la regla
 - Modificación de la base de hechos
 - Nuevos cálculos, nuevas acciones, preguntas al usuario
 - Nuevos subobjetivos
- Propagación de las instanciaciones
- Propagación del grado de certeza
- El proceso de deducción acaba cuando:
 - se encuentra la conclusión (el objetivo) buscado → éxito
 - no queda ninguna regla aplicable → éxito? / fracaso?



Tipos de razonamiento

- Razonamiento guiado por los datos
 - deductivos, progresivos, razonamiento hacia adelante, *forward chaining* (FC)
 - evidencias, síntomas, datos → conclusiones, hipótesis
- Razonamiento guiado por los objetivos
 - inductivos, regresivos, razonamiento hacia atrás, *backward chaining* (BC),
 - conclusiones, hipótesis → datos, evidencias, síntomas
- Mixtos, encadenamiento híbrido



Razonamiento guiado por los datos

- Basado en modus ponens: $A, A \rightarrow B \vdash B$
- La base de hechos (BH) se inicializa con los hechos que describen el problema

Procedimiento: Razonamiento Hacia Adelante

Entrada: Base de hechos, Base de reglas, Objetivos

Alternativas \leftarrow cierto

mientras $\exists o(o \in \text{Objetivos} \wedge o \notin \text{Base_de_hechos}) \wedge \text{Alternativas}$ **hacer**

Conjunto_Conflicto \leftarrow Interprete.Antecedentes_satisfactibles(Base_de_hechos,
Base_de_reglas)

si *Conjunto_Conflicto* $\neq \emptyset$ **entonces**

Regla \leftarrow Estrategia_Control.Resolucion_Conflictos(Conjunto_Conflicto)

Interprete.Aplicar(Base_de_hechos, Regla)

sino

Alternativas \leftarrow falso



Razonamiento guiado por los datos

- Problemas:
 - La búsqueda no está localizada en el objetivo
 - Explosión combinatoria, deducimos hechos no relacionados con la solución
- Ventajas:
 - Deducción intuitiva
 - Facilita la formalización del conocimiento al hacer un uso natural del mismo
 - Puede ser usado de manera exploratoria



Razonamiento guiado por los objetivos

- Método Inductivo. A partir de la hipótesis inicial se reconstruye la cadena de razonamiento en orden inverso hasta los hechos
- Cada paso implica nuevos subobjetivos: hipótesis a validar

Procedimiento: Razonamiento Hacia Atrás

Entrada: Base de hechos, Base de reglas, Objetivos

Alternativas \leftarrow cierto

mientras $Objetivos \neq \emptyset \wedge Alternativas$ **hacer**

Objetivo \leftarrow Estrategia_Control.Escoger_Objeto(Objetivos)

Objetivos.Quitar(Objetivo)

Conjunto_Conflicto \leftarrow Interprete.Consecuentes_satisfactibles(Objetivo, Base_de_reglas)

si $Conjunto_Conflicto \neq \emptyset$ **entonces**

Regla \leftarrow Estrategia_Control.Resolucion_Conflictos(Conjunto_Conflicto)

Objetivos.Añadir(Regla.Extraer_antecedente_como_objetivos())

sino

└─ Alternativas \leftarrow falso



Razonamiento guiado por los objetivos

- Problemas:
 - Hemos de conocer la solución del problema a priori
- Ventajas:
 - Sólo se considera lo necesario para la resolución del problema
 - La resolución se plantea como una descomposición en subproblemas
 - El proceso de resolución consiste en la exploración de un árbol Y/O



Encadenamiento híbrido

- Partes de la cadena de razonamiento que conduce de los hechos a los objetivos se construyen deductivamente y otras inductivamente
- Exploración bidireccional
- El cambio de estrategia suele llevarse a cabo a través de meta-reglas
- Se evita la explosión combinatoria del razonamiento deductivo
- Mejora la eficiencia del razonamiento inductivo cuando no existen objetivos claros



Las reglas como lenguaje de programación

- El uso de reglas como mecanismo de programación está muy extendido
- Lenguajes de programación dentro de los paradigmas de:
 - Programación declarativa
 - Programación lógica
- Como mecanismo de transformación, compilación, traducción, ...
 - Compiladores de lenguajes (LEX, YACC)
 - En la Web: XLTS (Extensible Stylesheet Language Transformations)
 - Transformaciones modelo-a-modelo M2M (ATL, etc.)
 - Automatización de tareas: Make, ANT, etc.
- El lenguaje de programación por excelencia es PROLOG
 - Mecanismo de razonamiento hacia atrás
 - Descomposición del problema mediante reglas y hechos
- Lenguaje con mecanismo de razonamiento hacia adelante → CLIPS



Las reglas como parte de aplicaciones generales

- Como representación de reglas de negocio en aplicaciones
 - Motores de inferencia como parte del desarrollo de aplicaciones:
Reglas interpretadas en lugar de código
 - Muchas herramientas de desarrollo comercial las incluyen: SAP, IBM, Oracle, Microsoft, etc.
 - Forma parte de la filosofía de paradigmas de desarrollo de aplicaciones (Service Oriented Architectures)



UNIVERSIDAD DE ALMERÍA