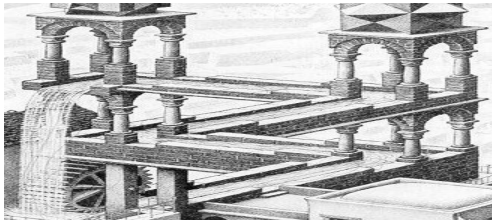


Expresividad de los Lenguajes de Consulta

23/Agosto/2019



Expresividad de los Lenguajes

- La expresividad (o el poder expresivo) de un lenguaje es la amplitud de ideas que pueden ser representadas y comunicadas en ese lenguaje.
- Cuanto más expresivo es un lenguaje, mayor es la cantidad y la variedad de ideas que pueden ser representadas.

Expresividad de los Lenguajes

- La expresividad (o el poder expresivo) de un lenguaje es la amplitud de ideas que pueden ser representadas y comunicadas en ese lenguaje.
- Cuanto más expresivo es un lenguaje, mayor es la cantidad y la variedad de ideas que pueden ser representadas.
- Desde el punto de vista de un lenguaje de consulta de bases de datos, equivale al conjunto de consultas que pueden hacersele a la base de datos por medio de ese lenguaje.

Expresividad de los Lenguajes

- La expresividad (o el poder expresivo) de un lenguaje es la amplitud de ideas que pueden ser representadas y comunicadas en ese lenguaje.
- Cuanto más expresivo es un lenguaje, mayor es la cantidad y la variedad de ideas que pueden ser representadas.
- Desde el punto de vista de un lenguaje de consulta de bases de datos, equivale al conjunto de consultas que pueden hacersele a la base de datos por medio de ese lenguaje.
- ¿Cuan expresivos son AR y CRT? ¿Qué relación existe entre ellos: puedo expresar el mismo conjunto de consultas? ¿Y respecto de la Lógica de Primer Orden (LPO)?

Expresividad de los Lenguajes de Consultas

- **Comparación CRT y LPO.** CRT es una especialización de la LPO, adaptada a bases de datos donde la única estructura de representación necesaria son las relaciones.
 - Cada relación puede verse como una interpretación de un predicado; interpretación en el sentido de que a cada instancia posible del predicado se le asocia un valor de verdad, si el registro/átomo está en la base de datos entonces es "verdadero" si no es "falso".
 - La semántica de CRT está restringida, no requiere teoría de modelos, sólo interesa saber la validez de una formula en un modelo fijo que es la instancia de bases de datos dada.

Expresividad de los Lenguajes de Consultas

- **Comparación CRT y LPO.** CRT es una especialización de la LPO, adaptada a bases de datos donde la única estructura de representación necesaria son las relaciones.
 - Cada relación puede verse como una interpretación de un predicado; interpretación en el sentido de que a cada instancia posible del predicado se le asocia un valor de verdad, si el registro/átomo está en la base de datos entonces es "verdadero" si no es "falso".
 - La semántica de CRT está restringida, no requiere teoría de modelos, sólo interesa saber la validez de una formula en un modelo fijo que es la instancia de bases de datos dada.
- **Comparación AR y CRT.** Bajo ciertas condiciones el CRT tiene el mismo poder expresivo que el Álgebra Relacional.

CRT - Expresiones Seguras

- **Expresión Segura.** En Cálculo Relacional es aquella que garantiza producir una cantidad finita de tuplas como resultado. Caso contrario se denomina **Expresión Insegura**
- Ejemplo: $\{t \mid \neg (t \in EMPLEADO)\}$ es una **Expresión Insegura** porque produce una cantidad infinita de tuplas: Todo el **universo** de posibles empleados que no forman parte de la relación *EMPLEADO*

CRT - Expresiones Seguras

- **Expresión Segura.** En Cálculo Relacional es aquella que garantiza producir una cantidad finita de tuplas como resultado. Caso contrario se denomina **Expresión Insegura**
- Ejemplo: $\{t \mid \neg (t \in EMPLEADO)\}$ es una **Expresión Insegura** porque produce una cantidad infinita de tuplas: Todo el **universo** de posibles empleados que no forman parte de la relación *EMPLEADO*
- **Dominio de una Expresión del CRT.** Sea E una expresión del CRT, $dom(E)$ es el conjunto de valores que aparecen tanto a) como valores constantes en E , como b) los valores pertenecientes a cualquier atributo de cualquier tupla de las relaciones mencionadas en E
- Ejemplo: $dom(\{t \mid (t \in EMPLEADO)\})$ es el conjunto de todos los valores que toman los atributos en todas las tuplas de la relación *EMPLEADO*

CRT - Expresiones Seguras

- **Expresión Segura.** En Cálculo Relacional es aquella que garantiza producir una cantidad finita de tuplas como resultado. Caso contrario se denomina **Expresión Insegura**
- Ejemplo: $\{t | \neg (t \in EMPLEADO)\}$ es una **Expresión Insegura** porque produce una cantidad infinita de tuplas: Todo el **universo** de posibles empleados que no forman parte de la relación *EMPLEADO*
- **Dominio de una Expresión del CRT.** Sea E una expresión del CRT, $dom(E)$ es el conjunto de valores que aparecen tanto a) como valores constantes en E , como b) los valores pertenecientes a cualquier atributo de cualquier tupla de las relaciones mencionadas en E
- Ejemplo: $dom(\{t | (t \in EMPLEADO)\})$ es el conjunto de todos los valores que toman los atributos en todas las tuplas de la relación *EMPLEADO*
- **Definición alternativa.** Una **Expresión** es **Segura** si todos los valores en el resultado son parte del dominio de la expresión
- **Observación.** Notar que el resultado de $\{t | \neg (t \in EMPLEADO)\}$ es una **Expresión Insegura**, ya que incluye valores por fuera de los incluidos en la relación *EMPLEADO*. Dichos valores no pertenecen al dominio de la expresión
- **Expresividad.** CRT restringido a expresiones seguras es equivalente en poder de expresividad al Álgebra Relacional básica.

DRC - Cálculo Relacional de Dominio

- **CRT**. Utiliza tuplas a modo de variables
- **CRD**. Utiliza atributos a modo de variables
- **Expresividad**. **CRD** tiene el mismo poder de expresividad que **CRT**

LPO (CRT) como Lenguaje de Consulta

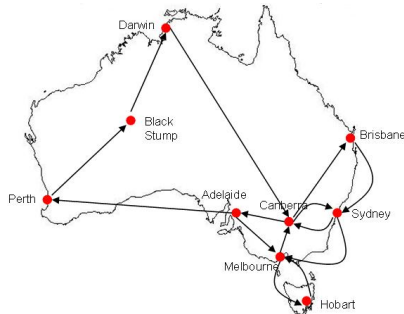
- ¿Podemos expresar cualquier consulta/propiedad que se nos ocurra en LPO?
- Existen ciertas consultas de BDs que requieren de un mayor poder de expresividad que la provista por AR y CRT (o incluso LPO completa).
- Miremos un ejemplo...

Grafos y Modelo Relacional

- Sea una BD donde se almacenan *i*) ciudades y *ii*) los pares (a,b) , tq una aerolínea ofrece vuelo sin escalas intermedias entre ellas

Grafos y Modelo Relacional

- Sea una BD donde se almacenan *i)* ciudades y *ii)* los pares (a,b) , tq una aerolínea ofrece vuelo sin escalas intermedias entre ellas
- Podemos pensar la BD como grafo $G=(V,E)$, donde V es el conjunto de ciudades y $E(a,b)$ representa que existe un vuelo non-stop de a hacia b
- **Ejemplo.**



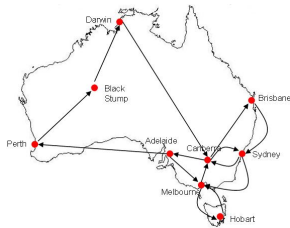
- $G = (V, E)$
- $V = \{ Adelaide; Perth; Black Stump; Darwin; \dots \}$
- $E = \{ E(Adelaide, Perth); E(Perth, Black Stump); E(Black Stump, Darwin); \dots \}$

Grafos y Modelo Relacional (Cont.)

- Una consulta en LPO sobre un grafo puede ser:
$$\varphi(x, y) = E(x, y) \vee (\exists z)(E(x, z) \wedge E(z, y))$$

Grafos y Modelo Relacional (Cont.)

- Una consulta en LPO sobre un grafo puede ser:
$$\varphi(x, y) = E(x, y) \vee (\exists z)(E(x, z) \wedge E(z, y))$$
- Ejemplo.**



$\{(Adelaide, Perth); (Perth, Black Stump); \dots (Adelaide, Black Stump); (Perth, Darwin); \dots \}$

Propiedades que no podemos expresar

- "¿Es posible viajar desde x hacia y ? (sin importar la cantidad de escalas)?"
- "¿Dado un grafo, ¿Tiene cantidad par de nodos?"
- "¿Existe un camino de longitud impar desde x hacia y ?"

Propiedades que no podemos expresar

- "¿Es posible viajar desde x hacia y ? (sin importar la cantidad de escalas)?"
- "¿Dado un grafo, ¿Tiene cantidad par de nodos?"
- "¿Existe un camino de longitud impar desde x hacia y ?"
- Podemos preguntar "¿Existen caminos de longitud 24 entre x e y ?" pero no podemos preguntar si "¿Existe un camino de longitud k entre x e Y ?", donde k es un número arbitrario **no fijo**.

Propiedades que no podemos expresar

- "¿Es posible viajar desde x hacia y ? (sin importar la cantidad de escalas)?"
- "¿Dado un grafo, ¿Tiene cantidad par de nodos?"
- "¿Existe un camino de longitud impar desde x hacia y ?"
- Podemos preguntar "¿Existen caminos de longitud 24 entre x e y ?" pero no podemos preguntar si "¿Existe un camino de longitud k entre x e Y ?", donde k es un número arbitrario **no fijo**.
- ¿Cómo sabemos qué consultas podemos expresar y cuáles no?

Propiedades que no podemos expresar

- "¿Es posible viajar desde x hacia y ? (sin importar la cantidad de escalas)?"
- "¿Dado un grafo, ¿Tiene cantidad par de nodos?"
- "¿Existe un camino de longitud impar desde x hacia y ?"
- Podemos preguntar "¿Existen caminos de longitud 24 entre x e y ?" pero no podemos preguntar si "¿Existe un camino de longitud k entre x e Y ?", donde k es un número arbitrario **no fijo**.
- ¿Cómo sabemos qué consultas podemos expresar y cuáles no?
- Existen herramientas matemáticas que permiten demostrar que ciertas propiedades (consultas) no son expresables en LPO: **Teorema Ehrenfeucht-Fraisse**.

Juegos de Ehrenfeucht-Fraisse

- Demostración basada en Teoría de Juegos
- Es una técnica para determinar si dos estructuras son elementalmente equivalentes (isomorfas).

Juegos de Ehrenfeucht-Fraisse

- Demostración basada en Teoría de Juegos
- Es una técnica para determinar si dos estructuras son elementalmente equivalentes (isomorfas).
- 2 jugadores: Spoiler y Duplicator
- 1 Tablero: Grafos (estructuras) A y B
- Los jugadores juegan una determinada cantidad de rondas, donde cada ronda consta de los siguientes pasos

Juegos de Ehrenfeucht-Fraisse

- Demostración basada en Teoría de Juegos
- Es una técnica para determinar si dos estructuras son elementalmente equivalentes (isomorfas).
- 2 jugadores: Spoiler y Duplicator
- 1 Tablero: Grafos (estructuras) A y B
- Los jugadores juegan una determinada cantidad de rondas, donde cada ronda consta de los siguientes pasos
 - Spoiler elige uno de los dos grafos (A ó B), para realizar su jugada
 - Spoiler elige un nodo perteneciente al grafo que seleccionó
 - Duplicator responde eligiendo un nodo del otro grafo
- Objetivos
 - Spoiler: mostrar que ambos grafos son distinguibles mediante una consulta en LPO
 - Duplicator: mostrar que ambos grafos son indistinguibles

EF - Ejemplo Nro. 1

Tablero



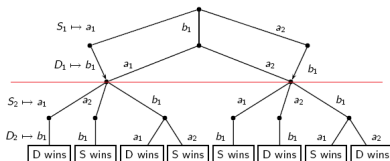
- **Análisis de las 2 primeras rondas**
 - **Primer Ronda:** Comienza Spoiler, por lo tanto puede seleccionar alguno de los 3 nodos del tablero. Si Spoiler elige a_1 o a_2 , Duplicator debe elegir un nodo del grafo B (su única opción es el nodo b_1). En cambio, si Spoiler elige b_1 , entonces Duplicator debe elegir algún nodo del grafo A (tiene dos opciones: a_1 o a_2).

EF - Ejemplo Nro. 1

Tablero



- **Análisis de las 2 primeras rondas**
 - **Primer Ronda:** Comienza Spoiler, por lo tanto puede seleccionar alguno de los 3 nodos del tablero. Si Spoiler elige a_1 o a_2 , Duplicator debe elegir un nodo del grafo B (su única opción es el nodo b_1). En cambio, si Spoiler elige b_1 , entonces Duplicator debe elegir algún nodo del grafo A (tiene dos opciones: a_1 o a_2).
 - **Árbol de movidas (primer ronda arriba de línea roja)**

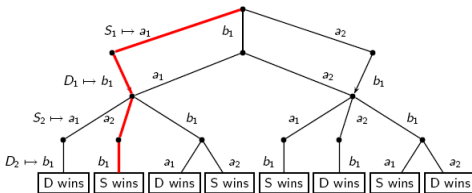


EF - Ejemplo Nro. 1 (Cont.)

Tablero

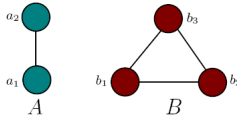


- Spoiler posee más de una estrategia ganadora utilizando 2 rondas.
- Una estrategia ganadora para Spoiler
 - Ronda 1: Spoiler elige a_1 . Esto fuerza a Duplicator a elegir b_1
 - Ronda 2: Spoiler elige a_2 . Esto fuerza a Duplicator a elegir b_1
- Spoiler gana porque el mapeo f , de $\{a_1, a_2\}$ a $\{b_1\}$, $f(a_1) = b_1$ y $f(a_2) = b_1$ no es un isomorfismo parcial.



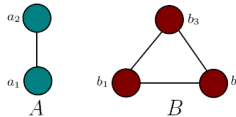
EF - Ejemplo Nro. 2

Tablero



EF - Ejemplo Nro. 2

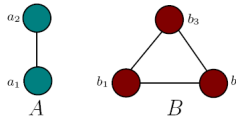
Tablero



- A 2 rondas, ¿qué jugador tiene estrategia ganadora?

EF - Ejemplo Nro. 2

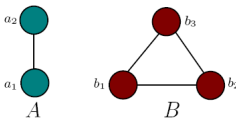
Tablero



- A 2 rondas, ¿qué jugador tiene estrategia ganadora? *Duplicator*

EF - Ejemplo Nro. 2

Tablero



- A 2 rondas, ¿qué jugador tiene estrategia ganadora? *Duplicator*
- Ronda 1
 - *Spoiler* puede elegir un nodo de A ó B
 - Si *Spoiler* elige uno de A, entonces *Duplicator* elegirá b_1 , si no elegirá a_1
 - Asumamos que *Spoiler* eligió a_2 . Entonces $x_1 := a_2, y_1 := b_1$
- Ronda 2
 - No importa qué elija *Spoiler*, *Duplicator* lo puede espejar. Por ejemplo, si *Spoiler* elige un nodo de B
 - Si *Spoiler* elige b_1 (mismo $y_1 = b_1$), entonces *Duplicator* debe repetir la elección de la ronda 1, y elige a_2
 - Si *Spoiler* elige b_2 o b_3 (nodos adyacentes a $y_1 = b_1$), entonces *Duplicator* debe elegir un nodo adyacente a $x_1 = a_2$.
 - Así $x_2 := a_2$ y $y_2 := b_1$, ó $x_2 := a_1$ e $y_2 := b_2/b_3$.
- Es fácil chequear que en ambos casos, f de X a Y es un isomorfismo parcial

EF - Intuición

- Duplicator gana el juego a n pasos si puede duplicar las n movidas que hace Spoiler. Por lo tanto, en las n rondas del juego EF no es posible distinguir entre A y B . En ese caso, lo escribimos como $A \sim_n B$. Caso contrario Spoiler gana las n rondas del juego EF y es posible distinguir entre A y B .
- Las estrategias pueden expresarse como fórmulas de LPO. Las rondas de juego equivalen a la cantidad de cuantificadores que se anidan en la fórmula, son números fijos.
- Si el duplicador gana, entonces quiere decir que en una cantidad fija de interacciones no se puede determinar que las estructuras son distintas por lo tanto tienen las mismas propiedades.

EF - Juego de EF y LPO - Teorema EF

Teorema de Ehrenfeucht-Fraisse

Sean A, B dos grafos. Las siguientes afirmaciones son equivalentes:

- A y B satisfacen las mismas sentencias con cantidad n de cuantificadores anidados
- $A \sim_n B$ (Duplicator gana la n -ésima movida del juego EF con tablero A y B)

EF - Aplicaciones

Una aplicación muy potente del Teorema de EF es la de mostrar los límites de expresividad de la lógica de primer orden

Demostración Inexpresibilidad de Propiedad usando Lógica de Primer Orden

Dada una propiedad P , no existe una sentencia en lógica de primer orden que exprese P , sí y sólo sí, para cada número entero positivo n , es posible hallar 2 grafos A y B tal que:

- La propiedad P es verdadera en A
- La propiedad P es falsa en B
- $A \sim_n B$ (Duplicator puede ganar el juego de EF de n movidas sobre A y B)

EF - ¿Es Par?

Demostración Inexpresibilidad de Propiedad usando Lógica de Primer Orden

Dada una propiedad P , no existe una sentencia en lógica de primer orden que exprese P , sí y sólo sí, para cada número entero positivo n , es posible hallar 2 grafos A y B tal que:

- La propiedad P es verdadera en A
 - La propiedad P es falsa en B
 - $A \sim_n B$ (Duplicador puede ganar el juego de EF de n movidas sobre A y B)
-
- Intuitivamente, no se puede expresar la propiedad de paridad con una sentencia en LPO con n cuantificadores anidados para cualquier n entero positivo.
 - Dados dos grafos A y B el duplicador siempre puede encontrar un elemento similar en la otra estructura para n pasos.

Expresividad de SQL

- En principio, la semántica de SQL está basada en el AR, por lo que la expresividad del lenguaje es equivalente a AR y CRT.
- Sin embargo, distintas implementaciones de SQL proveen predicados "extra-logicos" que le permiten expandir su poder expresivo:
 - Recursión
 - Funciones de agregación y agrupamiento ("group by")
 - Operaciones aritméticas sobre atributos numéricos
 - Store procedures... etc...

Expresividad de SQL

- En principio, la semántica de SQL está basada en el AR, por lo que la expresividad del lenguaje es equivalente a AR y CRT.
- Sin embargo, distintas implementaciones de SQL proveen predicados "extra-logicos" que le permiten expandir su poder expresivo:
 - Recursión
 - Funciones de agregación y agrupamiento ("group by")
 - Operaciones aritméticas sobre atributos numéricos
 - Store procedures... etc...
- **Comparación AR y CRT.** Bajo ciertas condiciones el CRT tiene el mismo poder expresivo que el Álgebra Relacional.

Bibliografía

- Ehrenfeucht-Fraïssé games, Math Explorers Club, Cornell Department of Mathematics. <http://www.math.cornell.edu/~mec/Summer2009/Raluca>
- Libkin. Elements of Finite Model Theory. Springer. 2012.
- Expressive Power of SQL. Leonid Libkin:
<https://homepages.inf.ed.ac.uk/libkin/papers/icdt01.pdf>