Expresiones regulares y MSO

Teorema (Büchi): Un lenguaje de palabras L es regular si y sólo si L es definible por una fórmula en MSO.

MSO es una buena alternativa para el caso de lenguajes regulares sobre palabras.

- ¿Qué pasa en el caso de documentos XML?

Teorema (Thatcher & Wright): Un lenguaje de árboles L es regular si y sólo si L es definible por una fórmula en MSO.

¡MSO es el nuevo objetivo!

- Nótese que es costoso evaluar una consulta en MSO.

¿Es suficiente con Regular XPath?

Nótese que Regular XPath $\not\subseteq$ Conditional XPath.

- $(\texttt{child}/?\mathit{section}/\texttt{child}/?\mathit{section})^*$ es una consulta en Regular XPath que no puede ser expresada en Conditional XPath.

¿Basta con Regular XPath para capturar MSO?

Teorema [GM05, BSSS06]: Hay una consulta en MSO que **no** es expresable en Regular XPath.

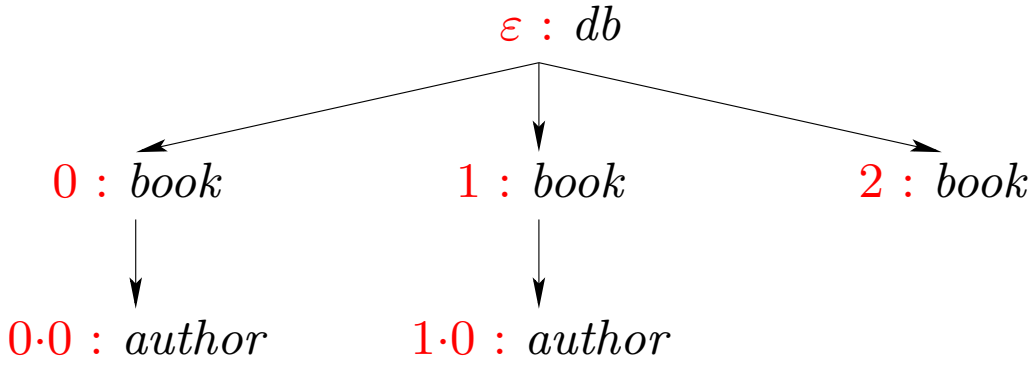
¿Existe un lenguaje eficiente que capture MSO?

Monadic Datalog

Este lenguaje utiliza reglas datalog para definir un conjunto de nodos.

- A diferencia de XPath, este lenguaje no construye caminos, sino que extrae nodos que satisfacen alguna propiedad.

Dado árbol T:



Representamos T usando los predicados extensionales: `root`, `leaf`, $\texttt{label}_{db}$, $\texttt{label}_{book}$, $\texttt{label}_{author}$, `firstchild`, `nextsibling`, `lastsibling`.

Monadic Datalog

Donde:

$$
\begin{aligned}
\texttt{root} &= \{\varepsilon\} \\
\texttt{leaf} &= \{2, 0{\cdot}0, 1{\cdot}0\} \\
\texttt{label}_{db} &= \{\varepsilon\} \\
\texttt{label}_{book} &= \{0, 1, 2\} \\
\texttt{label}_{author} &= \{0{\cdot}0, 1{\cdot}0\} \\
\texttt{firstchild} &= \{(\varepsilon, 0), (0, 0{\cdot}0), (1, 1{\cdot}0)\} \\
\texttt{nextsibling} &= \{(0, 1), (1, 2)\} \\
\texttt{lastsibling} &= \{2, 0{\cdot}0, 1{\cdot}0\}
\end{aligned}
$$

Monadic Datalog

Para definir una consulta usamos predicados intensionales.

- Estos predicados son unarios.

Ejemplo: La siguiente consulta extrae el conjunto de hijos de la raíz

$$Q(x) \leftarrow \texttt{root}(y), \texttt{firstchild}(y, x).$$
$$Q(x) \leftarrow Q(y), \texttt{nextsibling}(y, x).$$

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$Q_{\mathbf{i+1}}(x) \leftarrow \texttt{root}(y), \texttt{firstchild}(y, x).$$
$$Q_{\mathbf{i+1}}(x) \leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x).$$

Entonces:

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$\begin{aligned} Q_{\mathbf{i+1}}(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\ Q_{\mathbf{i+1}}(x) &\leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x). \end{aligned}$$

Entonces:

$$Q_0 = \emptyset$$

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$\begin{aligned} Q_{\mathbf{i+1}}(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\ Q_{\mathbf{i+1}}(x) &\leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x). \end{aligned}$$

Entonces:

$$\begin{aligned} Q_0 &= \emptyset \\ Q_1 &= \{0\} \end{aligned}$$

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$\begin{aligned} Q_{\mathbf{i+1}}(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\ Q_{\mathbf{i+1}}(x) &\leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x). \end{aligned}$$

Entonces:

$$\begin{aligned} Q_0 &= \emptyset \\ Q_1 &= \{0\} \\ Q_2 &= \{0, 1\} \end{aligned}$$

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$\begin{aligned} Q_{\mathbf{i+1}}(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\ Q_{\mathbf{i+1}}(x) &\leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x). \end{aligned}$$

Entonces:

$$\begin{aligned} Q_0 &= \emptyset \\ Q_1 &= \{0\} \\ Q_2 &= \{0, 1\} \\ Q_3 &= \{0, 1, 2\} \end{aligned}$$

Monadic Datalog: Semántica

Semántica de Monadic Datalog: Menor punto fijo.

Para todo $\mathbf{i}$:

$$\begin{aligned} Q_{\mathbf{i+1}}(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\ Q_{\mathbf{i+1}}(x) &\leftarrow Q_{\mathbf{i}}(y), \texttt{nextsibling}(y, x). \end{aligned}$$

Entonces:

$$\begin{aligned} Q_0 &= \emptyset \\ Q_1 &= \{0\} \\ Q_2 &= \{0, 1\} \\ Q_3 &= \{0, 1, 2\} \\ Q_4 &= \{0, 1, 2\} \end{aligned}$$

Monadic Datalog: Algunos ejemplo

Conjunto de libros que aparecen como hijos de la raíz:

$$
\begin{aligned}
Q(x) &\leftarrow P(x), \texttt{label}_{book}(x). \\
P(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) &\leftarrow P(y), \texttt{nextsibling}(y, x).
\end{aligned}
$$

Conjunto de libros que tienen al menos un autor:

$$
\begin{aligned}
Q(x) &\leftarrow P_1(x), \texttt{label}_{book}(x). \\
P_1(x) &\leftarrow \texttt{firstchild}(x, y), P_2(y). \\
P_2(x) &\leftarrow \texttt{label}_{author}(x). \\
P_2(x) &\leftarrow P_2(y), \texttt{nextsibling}(x, y).
\end{aligned}
$$

Monadic Datalog: Complejidad

En general es costoso evaluar una consulta en Datalog .

- ¿Por qué queremos usar Monadic Datalog?

Teorema [GK04]: Es posible evaluar una consulta Q en Monadic Datalog sobre un árbol T en tiempo $O(|T| \cdot |Q|)$.

Veamos el algoritmo ...

Primer paso: Crear instancia ground

$$
\begin{aligned}
Q(x) &\leftarrow P(x), \texttt{label}_{book}(x). \\
P(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) &\leftarrow P(y), \texttt{nextsibling}(y, x).
\end{aligned}
$$

Primer paso: Crear instancia ground

$$
\begin{aligned}
Q(x) &\leftarrow P(x), \texttt{label}_{book}(x). \\
P(x) &\leftarrow \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) &\leftarrow P(y), \texttt{nextsibling}(y, x).
\end{aligned}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(\varepsilon) & \leftarrow & P(\varepsilon), \texttt{label}_{book}(\varepsilon). \\
Q(0) & \leftarrow & P(0), \texttt{label}_{book}(0). \\
Q(1) & \leftarrow & P(1), \texttt{label}_{book}(1). \\
Q(2) & \leftarrow & P(2), \texttt{label}_{book}(2). \\
Q(0{\cdot}0) & \leftarrow & P(0{\cdot}0), \texttt{label}_{book}(0{\cdot}0). \\
Q(1{\cdot}0) & \leftarrow & P(1{\cdot}0), \texttt{label}_{book}(1{\cdot}0). \\
P(x) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0), \texttt{label}_{book}(0). \\
Q(1) & \leftarrow & P(1), \texttt{label}_{book}(1). \\
Q(2) & \leftarrow & P(2), \texttt{label}_{book}(2). \\
P(x) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(x) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(x) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, x). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(\varepsilon) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, \varepsilon). \\
P(0) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, 0). \\
P(1) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, 1). \\
P(2) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, 2). \\
P(0{\cdot}0) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, 0{\cdot}0). \\
P(1{\cdot}0) & \leftarrow & \texttt{root}(y), \texttt{firstchild}(y, 1{\cdot}0). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \texttt{root}(\varepsilon), \texttt{firstchild}(\varepsilon, 0). \\
P(0{\cdot}0) & \leftarrow & \texttt{root}(0), \texttt{firstchild}(0, 0{\cdot}0). \\
P(1{\cdot}0) & \leftarrow & \texttt{root}(1), \texttt{firstchild}(1, 1{\cdot}0). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \texttt{root}(\varepsilon), \texttt{firstchild}(\varepsilon, 0). \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \\
P(x) & \leftarrow & P(y), \texttt{nextsibling}(y, x).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \\
P(\varepsilon) & \leftarrow & P(y), \texttt{nextsibling}(y, \varepsilon). \\
P(0) & \leftarrow & P(y), \texttt{nextsibling}(y, 0). \\
P(1) & \leftarrow & P(y), \texttt{nextsibling}(y, 1). \\
P(2) & \leftarrow & P(y), \texttt{nextsibling}(y, 2). \\
P(0{\cdot}0) & \leftarrow & P(y), \texttt{nextsibling}(y, 0{\cdot}0). \\
P(1{\cdot}0) & \leftarrow & P(y), \texttt{nextsibling}(y, 1{\cdot}0).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{aligned}
Q(0) &\leftarrow P(0).\\
Q(1) &\leftarrow P(1).\\
Q(2) &\leftarrow P(2).\\
P(0) &\leftarrow \\
P(1) &\leftarrow P(0), \texttt{nextsibling}(0,1).\\
P(2) &\leftarrow P(1), \texttt{nextsibling}(1,2).
\end{aligned}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \\
P(1) & \leftarrow & P(0). \\
P(2) & \leftarrow & P(1).
\end{array}
$$

Primer paso: Crear instancia ground

$$
\begin{array}{lcl}
Q(0) & \leftarrow & P(0). \\
Q(1) & \leftarrow & P(1). \\
Q(2) & \leftarrow & P(2). \\
P(0) & \leftarrow & \\
P(1) & \leftarrow & P(0). \\
P(2) & \leftarrow & P(1).
\end{array}
$$

Segundo Paso: Calcular menor punto fijo

Ahora calculamos el menor punto fijo de la instancia ground.

Instancia ground:

- Tamaño: $O(|T| \cdot |D|)$.
- Tiempo para calcular menor punto fijo: $O(|T| \cdot |D|)$.

Monadic Datalog: Expresividad

Teorema [GK04]: Sobre árboles se tiene que MSO = Monadic Datalog.

¡Monadic Datalog es un buen lenguaje!

- Ha sido usado exitosamente en la practica [GKB$^+$04].

Monadic Datalog: Expresividad

Ejercicio: Demuestre que toda consulta en Monadic Datalog puede ser expresada en MSO.

Ejercicio: Sea $\Sigma = \{a, b\}$. Construya una consulta $Q(x)$ en Monadic Datalog tal que para todo Σ-árbol T se tiene que ε está en la respuesta a Q sobre T si y sólo si T contiene un número par de nodos con etiqueta a.

Referencias

[BSSS06] Mikolaj Bojanczyk, Mathias Samuelides, Thomas Schwentick, and Luc Segoufin. On the expressive power of pebble automata. In *ICALP*, 2006.

[Cod72] E. F. Codd. Relational completeness of data base sublanguages. *In: R. Rustin (ed.): Database Systems: 65-98, Prentice Hall and IBM Research Report RJ 987, San Jose, California*, 1972.

[GK04] Georg Gottlob and Christoph Koch. Monadic datalog and the expressive power of languages for web information extraction. *JACM*, 51(1):74–113, 2004.

[GKB+04] Georg Gottlob, Christoph Koch, Robert Baumgartner, Marcus Herzog, and Sergio Flesca. The lixto data extraction project - back and forth between theory and practice. In *PODS*, pages 1–12, 2004.

[GKP05] Georg Gottlob, Christoph Koch, and Reinhard Pichler. Efficient algorithms for processing xpath queries. *TODS*, 30(2):444–491, 2005.

[GM05] Evan Goris and Maarten Marx. Looping caterpillars. In *LICS*,

pages 51–60, 2005.

[Mar04a] Maarten Marx. Conditional xpath, the first order complete xpath dialect. In *PODS*, pages 13–22, 2004.

[Mar04b] Maarten Marx. Xpath with conditional axis relations. In *EDBT*, pages 477–494, 2004.

[Mar05] Maarten Marx. First order paths in ordered trees. In *ICDT*, pages 114–128, 2005.