Description Logics und Reasoning über XML-Dokumente

Giuliana Sabbatini

June 6, 2000

Wissensbasierter Zugriff auf Web Informationen

- **Grundidee:** Wissensbasierte Techniken und Informationen über die Struktur von Web Dokumenten benutzen, um Suche und Browsing in Web zu erleichtern.

- **Beispiele:**
  - Optimierung der Suche in einer Menge von Dokumenten.
  - Suche auf Basis der Seitenstruktur *: Auf Basis von expliziten Informationen über Seitenstruktur kann das Tool Benutzeranfrage beantworten, in dem es auf die Seite, welche die gewünschte Information enthält, zeigt.

- Zu diesem Zweck kann man **Description Logic** benutzen:

  Calvanese, De Giacomo, Lenzerini "Representing and Reasoning on XML Documents: A Description Logic Approach", 1999

*Calvanese, De Giacomo, Iocchi, Lenzerini, Nardi "Knowledge-based access to the Web".

Grundlage des Ansatzes

- Wissensrepräsentation in hierarchischen Strukturen auf Basis von **Konzepten** ($\equiv$ Klassen) und **Rollen** (binäre Relationen)
- DTD $\equiv$ Beschreibung eines Konzepts
- Web Seite $\equiv$ Instanzierung eines Konzepts
- Eine Document Base ist in diesem Sinne eine Menge struktureller Beschreibungen von Web Seiten und deren Inhalte
- Description Logic Reasoning Prozeduren ermöglichen damit, Reasoning über Struktur und Eigenschaften der Seiten durchzuführen

[Analog zu Schema-Reasoning Techniken in DB Umgebung -z.B. Constraint Inference in relationalen DB, Inheritance Inference in OODB]

Alternativer Ansatz: Reasoning über XML Dokumenten mit OODB Schemas *

*Christophides, Abiteboul, Cluet, Scholl "From Structured Documents to Novel Query Facilities".

Relevante Reasoning Tasks

- Verifizierung der **Entsprechung** eines Dokuments einer DTD (in polynomieller Zeit)

- **Äquivalenz** zwischen DTDs, sowie Inklusion und Disjunktion (in exponentieller Zeit, wobei bekannte Prozeduren doppelexponentielle Zeit brauchen)

- Alternative Begriffe und Varianten der Entsprechung und Äquivalenz:

  – Strukturelle Entsprechung und Äquivalenz

  – $\mathcal{R}$-Äquivalenz und -Entsprechung, wobei $\mathcal{R}$ eine Äquivalenz-Relation über die Menge von Elemententypen einer DTD ist.

- Optimierung von Query-Answering über eine Document Base: Steigerung der Effektivität des Suchprozesses, in dem eine (kurze) Beschreibung eines (langen) Dokuments genügt, um seine Relevanz zur Suche beurteilen zu können, ohne das ganze Dokument durchsuchen zu müssen

- Weitere Reasoning Möglichkeiten über DTDs

Definitionen und Notationen

- XML-Dokument: (**D**,d), wobei **D** eine DTD und d eine Dokument Instanz, die aus von Tags abgegrenzten Elementen (= logische Komponenten des Dokuments) besteht, ist.

  Beachte: d muß **D** entsprechen!

- Zwei Alphabethe **T** (Terminals) und **E** (Elemententypen mit assoziierten Start- und End-Tags)

- DTD: (**P**,R), wobei

  - **P**: Menge von Elemententypsdefinitionen der Form $E \to \alpha$, wobei E $\in$ **E** und α= Inhaltsmodell in Form einer Expression über **T** $\cup$ **E**:

    $$\alpha ::= S \mid \epsilon \mid \alpha_1 | \alpha_2 \mid \alpha_1, \alpha_2 \mid \alpha? \mid \alpha^* \mid \alpha^+$$

  - R: Wurzelelemententyp, R $\in$ **E**

- Bemerkung: DTDs sind **ECFGs** [*]

- docs(**P**,S) induktiv definiert:

  - Falls S ein Terminal F ist, denn docs(**P**,F)=F

  - Falls S ein Elementtyp E ist, und $E \to \alpha \in$ **P**, denn

    $$docs(\mathbf{P}, E) = \{< E > d_1 \ldots d_k < /E > |$$

    $$\exists S_1 \ldots S_k \ generated \ by \ \alpha : d_i \in docs(\mathbf{P}, S_i), \forall i \in \{1, \ldots, k\}\}$$

- d entspricht einer DTD **D**, falls d $\in$ docs(**D**)

[*]Siehe auch: Wood "Standard Generalized Markup Language: Mathematical and Philosophical Issues", 1995

Einführung in die Descritpion Logic $\mathcal{DL}$ *

- **Individuen** (Objekte), **Konzepte** (einstellige Prädikaten) und **Rollen** (binäre Prädikaten) beschreiben das Wissen

- Auf Basis von atomaren Konzepten und Regeln können in $\mathcal{DL}$ komplexe Konzepte und Regeln aufgebaut werden, z.B.:

Konzepte	**Syntax**	**Semantics**
Atom.Konz.	A	$A^I \subseteq \Delta^I$
Top	$\top$	Δ^I
Negation	$\neg C$	$\Delta^I \setminus C^I$
Konjunktion	$C_1 \sqcap C_2$	$C_1^I \cap C_2^I$
Univ.Quant.	$\forall R.C$	$\{o \mid \forall o' : (o, o') \in R^I \rightarrow o' \in C^I\}$
Rollen		
Atom.Rolle	P	$P^I \subseteq \Delta^I \times \Delta^I$
Vereinigung	$R_1 \cup R_2$	$R_1^I \cup R_2^I$
Verkettung	$R_1 \circ R_2$	$R_1^I \circ R_2^I$
Refl.Tr.Clos.	R^*	$(R^I)^*$
Trans.Clos.	R^+	$(R^I)^+$
Identität	$id(C)$	$\{(o, o) \mid o \in C^I\}$

  - **Abkürzungen:** $\bot$ ($\equiv \neg\top$), $C_1 \sqcup C_2$ ($\equiv \neg(\neg C_1 \sqcap \neg C_2)$), $\exists R.C$ ($\equiv \neg\forall R.\neg C$)

  - **Beispiel:** $Person \sqcap \forall child.Male \sqcap \exists child.Doctor$

  - **Semantics:** Eine **Interpretation** I ist ein Paar $(\Delta^I, \cdot^I)$, wobei Δ^I die Interpretationsdomäne und $\cdot^I$ eine Interpretationsfunktion ist, die jedem Konzept C eine Menge $C^I \subseteq \Delta^I$ und jeder Rolle R eine binäre Relation $R^I \subseteq \Delta^I \times \Delta^I$ assoziiert.

  - C^I und R^I werden **Extensionen** von C bzw. R genannt.

*Andere Logiken derselben Familie: siehe auch Donini, Lenzerini, Nardi, Schaerf "Reasoning in Description Logics", 1996.

$\mathcal{DL}$ Wissensbasis und Reasoning

- Eine $\mathcal{DL}$ Wissensbasis K ist eine Menge von Sätzen der Form $C_1 \sqsubseteq C_2$ (C_1, C_2 beliebige $\mathcal{DL}$ Konzepte)
  - Eine Interpretation I erfüllt $C_1 \sqsubseteq C_2$, falls $C_1^I \subseteq C_2^I$.
  - Eine Interpretation ist ein Modell für K, falls sie alle Sätze in K erfüllt.
- Reasoning in $\mathcal{DL}$:
  - **Erfüllbarkeit eines Konzepts** C in K: K besitzt ein Modell, in dem die Extension von C nicht leer ist ($K \models C \not\equiv \bot$).
  - **Erfüllbarkeit einer Wissensbasis** K: K besitzt ein Modell.
  - **Subsumption zwischen Konzepten** C_1 und C_2 in K: $C_1^I \subseteq C_2^I$ für jedes Modell I von K ($K \models C_1 \sqsubseteq C_2$).
- Erfüllbarkeit von K und Subsumption können zu Konzeptserfüllbarkeit reduziert werden:
  - K ist erfüllbar $\Leftrightarrow$ $\top$ ist erfüllbar in K
  - Subsumption zwischen C_1 und C_2 $\Leftrightarrow$ $C_1 \sqcap \neg C_2$ ist nicht erfüllbar in K
- Konzeptserfüllbarkeit in $\mathcal{DL}$ ist EXPTIME-vollständig.

Von DTDs zu $\mathcal{DL}$ (1)

Wie kann die Wissensbasis K, die einer DTD **D** entspricht, in $\mathcal{DL}$ definiert werden?

- Der Alphabet von K inkludiert die atomaren Konzepte `Tag` und `Terminal`; ein atomares Konzept F für jedes $F \in$ **T**; die atomaren Konzepte `StartE` und `EndE` für jedes $E \in$ **E**; ein atomares Konzept E_D für jedes $(E \to \alpha) \in$ **P**; die atomaren Rollen `f` und `r`.

- Die Menge von Sätzen in K besteht aus $K_{T,E} \cup K_R \cup K_D$

- $\mathbf{K_{T,E}}$ kodiert die allgemeinen strukturellen Eigenschaften eines Dokuments in eine Baumstruktur, z.B.:
  - $\mathtt{Tag} \sqsubseteq \forall(\mathtt{f} \cup \mathtt{r}).\bot$
  - $\mathtt{Terminal} \sqsubseteq \forall(\mathtt{f} \cup \mathtt{r}).\bot \sqcap \neg\mathtt{Tag}$
  - $F \sqsubseteq \mathtt{Terminal} \quad \forall F \in \mathbf{T}$
  - $\mathtt{StartE} \sqsubseteq \mathtt{Tag} \quad \forall E \in \mathbf{E}$
  - $\mathtt{EndE} \sqsubseteq \mathtt{Tag} \quad \forall E \in \mathbf{E}$

- $\mathbf{K_R}$ kodiert die Äquivalenzrelation
$$\mathcal{R} = \{\{E_1^1, \ldots, E_{n_1}^1\}, \ldots, \{E_1^m, \ldots, E_{n_m}^m\}\}$$
und besteht aus folgenden Sätzen:
  - $\mathtt{StartE}_\mathtt{1}^\mathtt{i} \sqsubseteq \neg\mathtt{StartE}_\mathtt{1}^\mathtt{j} \; \forall i, j \in \{1, \ldots, m\}, i \neq j$
  - $\mathtt{EndE}_\mathtt{1}^\mathtt{i} \sqsubseteq \neg\mathtt{EndE}_\mathtt{1}^\mathtt{j} \; \forall i, j \in \{1, \ldots, m\}, i \neq j$
  - $\mathtt{StartE}_\mathtt{i}^\mathtt{j} \equiv \mathtt{StartE}_\mathtt{i+1}^\mathtt{j} \; \forall i \in \{1, \ldots, n_j - 1\}, j \in \{1, \ldots, m\}$
  - $\mathtt{EndE}_\mathtt{i}^\mathtt{j} \equiv \mathtt{EndE}_\mathtt{i+1}^\mathtt{j} \; \forall i \in \{1, \ldots, n_j - 1\}, j \in \{1, \ldots, m\}$

Von DTDs zu $\mathcal{DL}$ (2)

- $\mathbf{K_D}$ kodiert die Elementtypendefinitionen in **D**, welche die Struktur der entsprechenden Dokumenten bestimmen:
  - Für jedes $(E \to \alpha) \in$ **P** enthält K_D den Satz

$$E_D \equiv \exists \mathtt{f}.\mathtt{StartE} \sqcap \exists(\mathtt{r} \circ \tau(\alpha)).\mathtt{EndE}$$

  - $\tau(\alpha)$ ist z.B:
    - $\tau(\alpha_1|\alpha_2) = \tau(\alpha_1) \cup \tau(\alpha_2)$
    - $\tau(\alpha_1, \alpha_2) = \tau(\alpha_1) \circ \tau(\alpha_2)$
    - $\tau(\alpha^*) = \tau(\alpha)^*$
    - $\tau(\alpha^+) = \tau(\alpha)^+$

Entsprechung eines Dokuments einer DTD

- Jedes Dokument d entspricht einem Modell von $K_{T,E}$ und K_R, und zwar existiert ein eindeutiges Mapping β zwischen Dokumenten und Modelle von $K_{T,E}$ und K_R. $\beta(d) = (\Delta^{\beta(d)}, \cdot^{\beta(d)})$ kann induktiv definiert werden:
  - Falls $d = F \in \mathbf{T}$, dann $\Delta^{\beta(d)} = F^{\beta(d)} = \{o\}$. Die Extensionen anderer Konzepte und Rollen sind leer.
  - Falls $d =< E > d_1 \ldots d_k < /E >$, $E \in \mathbf{E}$, $< E >$ und $< /E >$ Start- und Endtag von E, $d_1, \ldots, d_k$ Dokumente über $\mathbf{T} \cup \mathbf{E}$, dann:
    * $\Delta^{\beta(d)} = \{o, o_b, o_1, \ldots, o_k, o_e\} \uplus \biguplus_{1 \leq i \leq k} \Delta^{\beta(d_i)}$
    * $\mathtt{StartE'}^{\beta(d)} = \{o_b\} \uplus \biguplus_{1 \leq i \leq k} \mathtt{StartE'}^{\beta(d_i)}, \ \forall E' \in [E]_{\mathcal{R}}$
    * $\mathtt{EndE'}^{\beta(d)} = \{o_e\} \uplus \biguplus_{1 \leq i \leq k} \mathtt{EndE'}^{\beta(d_i)}, \ \forall E' \in [E]_{\mathcal{R}}$
    * $\mathtt{StartE'}^{\beta(d)} = \biguplus_{1 \leq i \leq k} \mathtt{StartE'}^{\beta(d_i)}, \ \forall E' \in \mathbf{E} \setminus [E]_{\mathcal{R}}$
    * $\mathtt{EndE'}^{\beta(d)} = \biguplus_{1 \leq i \leq k} \mathtt{EndE'}^{\beta(d_i)}, \ \forall E' \in \mathbf{E} \setminus [E]_{\mathcal{R}}$
    * $\mathtt{Tag}^{\beta(d)} = \{o_b, o_e\} \uplus \biguplus_{1 \leq i \leq k} \mathtt{Tag}^{\beta(d_i)}$
    * ... usw.

- Das Prüfen der Entsprechung eines Dokuments einer DTD kann zu Model-Checking in $\mathcal{DL}$ polynomiell reduziert und damit in polynomieller Zeit (in der Dimension von d und **D**) durchgeführt werden.

Inklusion und Äquivalenz zwischen DTDs

- Die Definition der Wissensbasis, die einer einzelnen DTD entspricht, kann zu einer Menge von DTDs $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_n\}$ über dieselben Alphabeten **T**,**E** und dieselbe Relation $\mathcal{R}$ erweitert werden: $K = K_{T,E} \cup K_R \cup K_{D_1} \cup \ldots \cup K_{D_n}$. Ein Modell von K ist damit eine Menge von Dokumenten.

- Seien **D** und **D'** zwei DTDs und K die charakteristische Wissensbasis:

  - $D \sqsubseteq_R D' \;\; \Leftrightarrow$ Subsumption zwischen R_D und $R'_{D'}$ in K ($K \models R_D \sqsubseteq R'_{D'}$)

    wobei R und R' die Wurzelelemententypen von **D** bzw. **D'** sind (mit den Definitionen $R \to \alpha_1$ bzw. $R' \to \alpha_2$), und $R_D \equiv \exists \mathtt{f}.\mathtt{StartR} \sqcap \exists(\mathtt{r} \circ \tau(\alpha_1)).\mathtt{EndR}$ (Analog für R'_D)

- Da logische Implikation in $\mathcal{DL}$ in exponentieller Zeit entschieden werden kann, ist EXPTIME (in der Dimension der DTDs) ein upper bound für $\mathcal{R}$-Inklusion und -Äquivalenz zwischen DTDs.

Anfrage Q an eine Dokumentenbasis B

- $B =< \mathcal{D}, I >$: $\mathcal{D}$ Menge von DTDs, für die es bekannt ist, ob $\mathbf{D_1} \sqsubseteq_R \mathbf{D_2}$ und ob $\mathbf{D_1} \otimes_R \mathbf{D_2}$; I Menge von Dokumenten, s.d. $\forall d \in I,\ \exists \mathbf{D} \in \mathcal{D} : d$ entspricht $\mathbf{D}$, und $\forall d \in I,\ \forall \mathbf{D} \in \mathcal{D}$ es ist bekannt, ob d entspricht $\mathcal{D}$.

- Query $Q \equiv$ DTD; Antwort $Q(B) = \{d \in B \mid d\ entspricht Q\}$

- Algorithmus für $Q(B)$: $S = \mathcal{D},\ \ J = I$

  1. `while` $S \neq \emptyset$ `do` wähle **D** aus S aus, s.d. $\not\exists \mathbf{D'} \in S : \mathbf{D} \sqsubseteq_R \mathbf{D'}$:
     (a) `if` $\mathbf{D} \equiv_R Q$ `then` $Q(B) = \{d \in I \mid d\ entspricht\ \mathbf{D}\}$; `stop`
     (b) `if` $\mathbf{D} \sqsubseteq_R Q$ `then`
        i. Verschiebe aus J in $Q(B)$ alle Dokumenten, die **D** entsprechen
        ii. Entferne aus S alle DTDs **D'** mit $\mathbf{D'} \sqsubseteq_R \mathbf{D}$
        iii. Geh weiter mit der nächsten `while`-Iteration
     (c) `if` $Q \sqsubseteq_R \mathbf{D}$ `then`
        i. Entferne **D** aus S
        ii. $\forall \mathbf{D'} \in S : \mathbf{D'} \otimes_R \mathbf{D}$ entferne **D'** aus S, und aus J alle Dokumenten, die **D'** entsprechen
        iii. Geh weiter mit der nächsten `while`-Iteration
     (d) `if` $\mathbf{D} \otimes_R Q$ `then`
        i. Entferne aus S alle DTDs **D'** mit $\mathbf{D'} \sqsubseteq_R \mathbf{D}$
        ii. Entferne aus J alle Dokumenten, die **D'** entsprechen
        iii. Geh weiter mit der nächsten `while`-Iteration
     (e) `else` entferne **D** aus S und geh weiter
  2. Füge zu $Q(B)$ alle Dokumenten $d \in J$ hinzu, die Q entsprechen

Zum Schluß

- DTD-Reasoning ist teuer: Die vorgeschlagenen Methoden zahlen sich aus, wenn die Dimension der DTDs im Vergleich zur Dimension der Dokumenten klein ist (z.B. logarithmisch).

- Wie kann man semantische Aspekte in die Struktur der Dokumenten kodieren?
  - DTDs können Attribute berücksichtigen: Attribute können in $\mathcal{DL}$ repräsentiert werden, in dem eine neue Rolle für jedes Attribut und die notwendigen neuen Sätze in der Wissensbasis eingefügt werden.
  - Dokumente können Links zu anderen Dokumenten enthalten: Links können mittels spezielle Konzepte repräsentiert werden, zusammen mit Rollen für den entsprechenden Namen und den assoziierten Anker.

- Offener Punkt: Verwendung der $\mathcal{DL}$ Logik zur Dokumentenklassifizierung (Welcher DTD besser entspricht ein gegebenes Dokument?)