

Einführung in die Logik - 10

Logikkalküle für die automatische (Sprach-)Verarbeitung

(2) Beschreibungslogiken (Description Logics)

Beschreibungslogiken

HappyCLErsti \equiv

Human \sqcap Adult \sqcap

(Female \sqcup \neg Female) \sqcap

\exists studiertFach.CL \sqcap

1hatFachsemester.Semester \sqcap

≤ 4 bekommtKlausuraufgabe.KNS \sqcap

\forall wirdGeprüftIn.Bestanden

Literatur

Korrektheit, Vollständigkeit, Entscheidbarkeit

Boolos, George S. & Richard C. Jeffrey (1980): *Computability and Logic*. Second Edition. Cambridge: Cambridge University Press.

Hermes, Hans (1961; 3. Auflage 1978): *Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit*. Berlin, Heidelberg, New York: Springer-Verlag (Heidelberger Taschenbücher Sammlung Informatik, Band 87)

Spies, Marcus (2004): *Einführung in die Logik. Werkzeuge für Wissensrepräsentation und Wissensmanagement*. Heidelberg/Berlin: Spektrum Akademischer Verlag.

Beschreibungslogiken

Baader, Franz & Werner Nutt (2003): Basic Description Logics. In F. Baader et al. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge u.a.: Cambridge University, pp. 47-100.

Baader, Franz; Ian Horrocks, Ulrike Sattler (2005): Description Logics as Ontology Languages for the Semantic Web. In: D. Hutter & W. Stephan (eds.): *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Artificial Intelligence*. Berlin/Heidelberg: Springer-Verlag, pp. 228–248.

Baader, Franz; Ian Horrocks, Ulrike Sattler (2008): Description Logics. In: F. van Harmelen et al. (eds.): *Handbook of Knowledge Representation*, Chapter 3. Elsevier, pp. 135-180.

Donini, Francesco M. (2003): Complexity of Reasoning. In F. Baader et al. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge u.a.: Cambridge University, pp. 101-141.

Nardi, Daniele & Ronald J. Brachman (2003): An Introduction to Description Logics. In F. Baader et al. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge u.a.: Cambridge University, pp. 5-44.

Owsnicki-Klewe, Bernd; Kai von Luck, Bernhard Nebel (2000): Wissensrepräsentation und Logik – Eine Einführung. In: G. Görz et al. (eds.): *Handbuch der Künstlichen Intelligenz*. 3. Auflage. München/Wien: Oldenbourg Verlag, pp. 153-197.

Was sind Beschreibungslogiken?

verschiedene Bezeichnungen:

- aktuell: Description Logics (Beschreibungslogiken)
 - Abkürzung im weiteren: **DL**
- früher auch:
 - Begriffslogiken
 - Terminologische Logiken, Terminologiesprachen
 - Termsubsumptionssprachen
 - KL-ONE-basierte Sprachen

Was sind Beschreibungslogiken?

Familie von Formalismen für die Wissensrepräsentation (KR)

❖ **Terminologie einer Domäne:** Repräsentation des Wissens über eine Domäne qua Definition ihrer relevanten Konzepte

➤ **T-Box**

- C_1 ist definiert als C_2 & C_3
- C_4 ist Hyponym zu C_5
- ...

❖ **Beschreibung einer Domäne:** Spezifikation der Eigenschaften von Objekten aus der Domäne mittels der definierten Domänenkonzepte

➤ **A-Box** ('Assertion')

- a ist Instanz von C_1
- b ist Instanz von C_2 & C_3
- ...

Was sind Beschreibungslogiken?

- wesentliche Eigenschaften von DL
 - präzise, logik-basierte **Semantik**
 - vgl. modelltheoretische Semantik der Prädikatenlogik
 - **Schlussverfahren** (Inferenzalgorithmen, 'Reasoning') als zentraler Service
 - Folgerung von implizitem Wissen aus explizit (in T- und A-Box) repräsentiertem Wissen
 - Schlussregeln (Inference Patterns) entsprechen wesentlichen Anforderungen intelligenter Systeme der Informationsverarbeitung:
 - Strukturierung von Begriffsmengen
 - Ist C1 Unterbegriff von C2? Sind C3 und C4 synonym? Wie ist C5 definiert?
 - Klassifikation von Individuen
 - Ist a eine Instanz von C1? Gibt es ein Objekt, das in der Relation R zu a steht?

Was sind Beschreibungslogiken?

F. Baader, W. Nutt

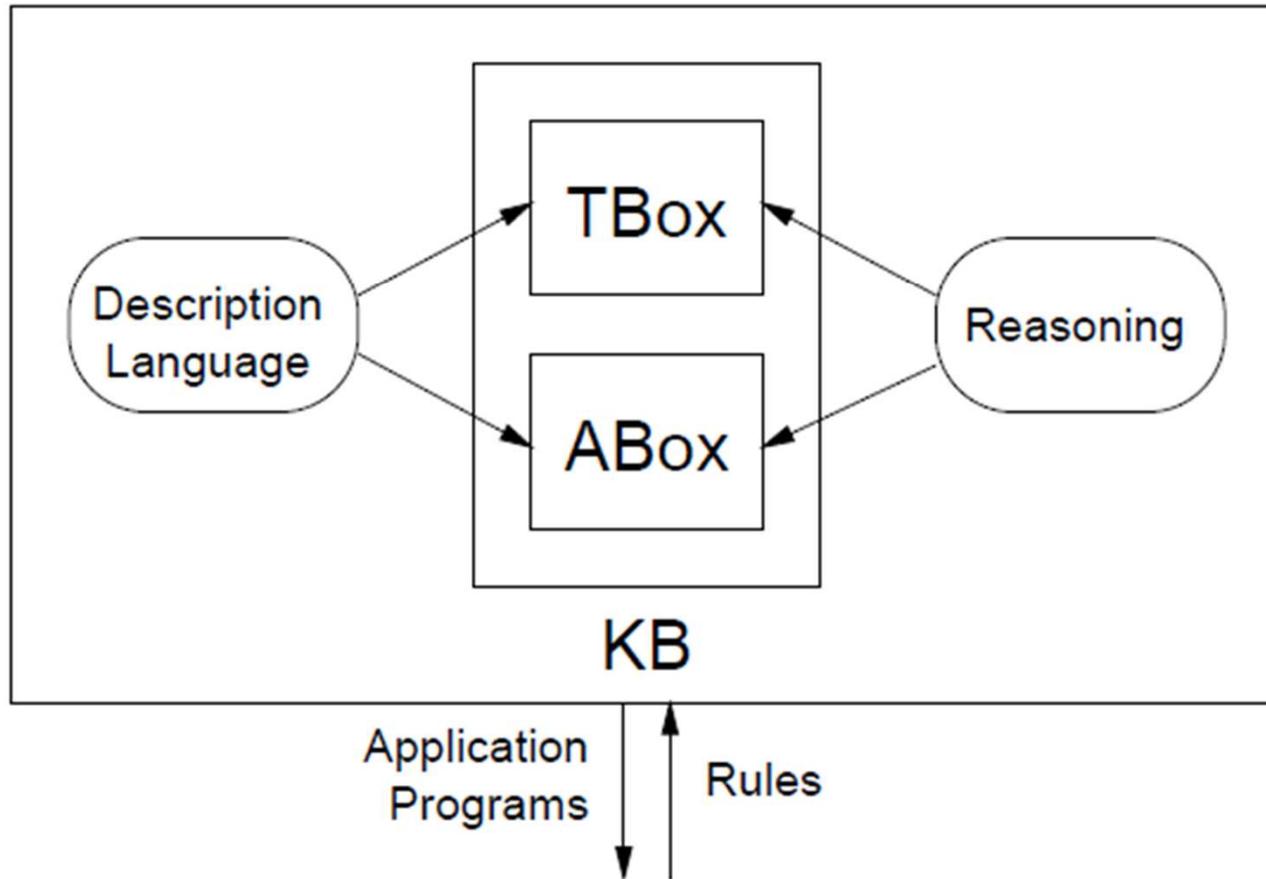


Fig. 2.1. Architecture of a knowledge representation system based on Description Logics.

Exkurs: Korrektheit, Vollständigkeit, Entscheidbarkeit - 1

vgl. Folien 05

Korrektheit (a.: Widerspruchsfreiheit):

Wenn $\alpha \vdash A$, dann $\alpha \models A$

- Wenn A aus einer Formelmenge α syntaktisch ableitbar ist, dann folgt A auch semantisch aus α
- Wenn A syntaktisch beweisbar ist, dann ist A auch semantisch allgemeingültig ► Jedes Theorem ist ein Tautologie.

Vollständigkeit:

Wenn $\alpha \models A$, dann $\alpha \vdash A$

- Wenn A aus einer Formelmenge α semantisch folgt, dann ist A auch aus α syntaktisch ableitbar
- Wenn A semantisch allgemeingültig ist, dann ist A auch syntaktisch beweisbar ► Jede Tautologie ist ein Theorem.

Entscheidbarkeit:

Ein Logikkalkül ist entscheidbar, wenn man für jeden seiner wfA durch ein effektives formales Verfahren (d.h. einen Algorithmus; eine Prozedur, die nach endlichen Schritten terminiert) angeben kann, ob dieser ein Theorem ist oder nicht.

Exkurs: Korrektheit, Vollständigkeit, Entscheidbarkeit - 2

- **1928, David Hilbert: Entscheidungsproblem** - Der Nachweis der (Un-) Entscheidbarkeit der PL ist eines der wichtigsten mathematischen Probleme.
- **1929, Kurt Gödel: Vollständigkeitstheorem** - PL1 ist korrekt und vollständig (ebenso AL), PL2 (und höhere PL) ist unvollständig.
- **1936/37, Alonzo Church, Alan Turing – Unentscheidbarkeit** von PL1; Allgemeingültigkeit, Erfüllbarkeit, Folgerung sind in PL1 (und höheren PL) im Gegensatz zur AL unentscheidbar.
 - PL1 ist aber semi-entscheidbar: Wenn B aus A folgt, kann dies erkannt werden. Wenn B jedoch nicht aus A folgt, heißt das nicht, dass $\neg B$ aus A folgt. Daher kann nicht immer erkannt werden, dass ein wfA kein Theorem ist.

Unentscheidbarkeit einer vollständigen Theorie bedeutet, dass wir zwar wissen, dass wir alle logischen Folgerungen aus den Axiomen ableiten können, aber nicht generell wissen, welche Sätze dies im Einzelnen sind. Ingenium und Intuition sind zumindest bisweilen erforderlich, um in einer solchen Theorie Folgerungen aufzustellen oder (...) behauptete Folgerungen zu überprüfen. [Spies 2004: 226]

Exkurs: Korrektheit, Vollständigkeit, Entscheidbarkeit - 3

[Owsnicki-Klewe et al. 2000]

- Die **Unentscheidbarkeit** der Folgerung in PL1 bedeutet, dass wir kein Deduktionssystem für PL1 implementieren können, das (a) korrekt ist, (b) vollständig ist und (c) immer terminiert.
- Die Unterscheidung von entscheidbaren und unentscheidbaren Problemen ist aber noch sehr grob: Im Prinzip entscheidbare Probleme können so kompliziert sein, dass impraktikable Berechnungszeiten entstehen.
 - **Komplexitätstheorie**
- Die volle PL1 ist von ihrer Ausdruckskraft zu mächtig, als dass man effiziente Verfahren für ihre Verarbeitung angeben könnte.
- In der Wissensrepräsentation werden deshalb Repräsentationsformalismen entwickelt, die effizientere Verfahren zulassen als die allgemeinen Beweisverfahren für PL1.
 - **Balance zwischen Ausdrucksfähigkeit und Berechenbarkeits-/Verarbeitungseigenschaften** (Komplexität des Schließens – s. Brachmann & Levesque 1984)

Exkurs: Korrektheit, Vollständigkeit, Entscheidbarkeit - 4

[Lutz, VL U Bremen WS 2010]

- Durch verschiedene syntaktische Einschränkungen können entscheidbare Fragmente von PL1 erzeugt werden:
 - monadische PL (Klassenlogik) – nur 1-stellige Relationen, keine Funktionssymbole
 - wfA mit max. 2 Variablen
 - wfA in PNF (Pränexer Normalform) – Quantorenpräfix $\exists^* \forall^*$ gefolgt von quantorenfreier Matrix
- Hier setzen **Description Logics / Beschreibungslogiken** an
 - Familie von logikbasierten Formalismen für die Wissensrepräsentation mit unterschiedlichen Ausdrucksstärken, von denen viele mit entscheidbaren Fragmenten der PL1 äquivalent sind.
 - Ontologien / Terminologien für div. Anwendungsbereiche:
 - Allgemeine WR
 - Semantic Web
 - spezielle Domänen
 - » Biomedizinische Informatik u.a.

Eine grundlegende Beschreibungslogik: *ALC* (Attributive Language with full Complements)

C	\rightarrow	B	Begriffssymbole
		$C \sqcap C'$	Begriffskonjunktion
		$C \sqcup C'$	Begriffsdisjunktion
		$\neg C$	Begriffsnegation
		$\forall R: C$	Warterestriktion
		$\exists R: C$	Existentielle Restriktion

- Atomare Begriffe: C, C', \dots (1-stellige Prädikate)
- Atomare Rollen: R, R', \dots (2-stellige Relationen)
- Begriffskonstruktoren: \sqcap, \sqcup, \neg

Beziehungen zwischen Begriffen ► Terminologische Axiome

- \sqsubseteq - (unechte) Unterbegriffs-Beziehung; partielle Definition
- \equiv oder \doteq - Äquivalenz-Beziehung; vollständige Definition

Korrespondenz zwischen DL (\mathcal{ALC}) und Prädikatenlogik (PL1)

DL-Repräsentationssprache	Prädikatenlogik 1. Stufe
$C \sqcap D$	$C(x) \wedge D(x)$
$C \sqcup D$	$C(x) \vee D(x)$
$\neg C$	$\neg C(x)$
$\forall R:C$	$\forall y(R(x,y) \rightarrow C(y))$
$\exists R:C$	$\exists y(R(x,y) \wedge C(y))$
$C \sqsubseteq D$	$\forall x(C(x) \rightarrow D(x))$
$C \equiv D$ bzw. $C \doteq D$	$\forall x(C(x) \leftrightarrow D(x))$
$a \in C$	$C(a)$
$\langle a,b \rangle \in R$	$R(a,b)$

DL-Konzepte/Rollen vs. PL1-Prädikate /Relationen

- Im Unterschied zur Prädikatenlogik erlauben Beschreibungslogiken komplexe (zusammengesetzte) einstellige Prädikate/Konzepte ($C \sqcap D$, $C \sqcup D$, $\neg C$).
- Dafür ist die Struktur der Aussagen/Formeln eingeschränkt:
 - **Rollen:** Rollenspezifikationen der Form $R1 \sqsubseteq R2$, $R2 \equiv R3$ sowie komplexe Rollen ($R3 \sqcap R4$, $R4 \sqcup R5$, $\neg R1$) werden nur von wenigen DL-Sprachen unterstützt, da dies zu einer erheblichen Erhöhung der Ausdrucksmächtigkeit und damit zu Komplexitäts- und Entscheidbarkeitsproblemen führt.
 - **Negation:** Negation ist in DL nur auf Konzeptebene anwendbar, nicht auf der Ebene von Aussagen/Formeln.
 - **Quantifikation:** Werterestriktionen $\forall R:C$ und $\exists R:C$ führen nur eine zusätzliche Variable ein; die Interpretation eines solchen DL-Ausdrucks beinhaltet damit maximal 2 Variablen.
 - ❖ Beschränkungen der DL-Sprachen bewirken Vorteile für Entscheidbarkeit und Verarbeitung.

Terminologien in Beschreibungslogiken

semantische Relation	DL-Repräsentation
Hypo-/Hyperonymie	Violine \sqsubseteq Streichinstrument
	Viola \sqsubseteq Streichinstrument
	Gitarre \sqsubseteq Zupfinstrument
	Streichinstrument \sqsubseteq Saiteninstrument
	Zupfinstrument \sqsubseteq Saiteninstrument
Synonymie	Violine \equiv Geige
	Viola \equiv Bratsche
Antonymie	laut \sqsubseteq \neg leise, tief \sqsubseteq \neg hoch
Komplementarität	lebendig \equiv \neg tot
(K. in begrenzter Domäne)	? gestrichenerTon \equiv \neg gezupfterTon ?

Definitionstyp	DL-Repräsentation
partiell	Streichinstrument \sqsubseteq Violine \sqcup Viola \sqcup Violoncello \sqcup Kontrabass \sqcup Kokyu \sqcup Erhu \sqcup Rebab \sqcup Gudok
total	Std_Streichinstr_Im_Europ_Orchester \equiv Violine \sqcup Viola \sqcup Violoncello \sqcup Kontrabass

Eine Beispiel-T-Box in \mathcal{ALC}

<i>Komponente</i>	\sqsubseteq	<i>Ding</i>
<i>Motor</i>	\sqsubseteq	<i>Komponente</i>
<i>Lampe</i>	\sqsubseteq	<i>Komponente</i> \sqcap \neg <i>Motor</i>
<i>Stecker</i>	\sqsubseteq	<i>Komponente</i> \sqcap \neg <i>Motor</i> \sqcap \neg <i>Lampe</i>
<i>Saugrohr</i>	\sqsubseteq	<i>Komponente</i> \sqcap \neg <i>Motor</i> \sqcap \neg <i>Lampe</i> \sqcap \neg <i>Stecker</i>
<i>Mixstab</i>	\sqsubseteq	<i>Komponente</i> \sqcap \neg <i>Motor</i> \sqcap \neg <i>Lampe</i> \sqcap \neg <i>Stecker</i> \sqcap \neg <i>Saugrohr</i>
<i>Geraet</i>	\sqsubseteq	\forall <i>Teil</i> : <i>Komponente</i> \sqcap <i>Ding</i> \sqcap \neg <i>Komponente</i>
<i>E-Geraet</i>	\doteq	<i>Geraet</i> \sqcap \exists <i>Teil</i> : <i>Stecker</i>
<i>Staubsauger</i>	\doteq	<i>E-Geraet</i> \sqcap \forall <i>Teil</i> : <i>Komponente</i> \sqcap \exists <i>Teil</i> : <i>Motor</i> \sqcap \exists <i>Teil</i> : <i>Lampe</i> \sqcap \exists <i>Teil</i> : <i>Saugrohr</i>
<i>Mixer</i>	\doteq	<i>E-Geraet</i> \sqcap \forall <i>Teil</i> : <i>Komponente</i> \sqcap \exists <i>Teil</i> : <i>Motor</i> \sqcap \exists <i>Teil</i> : <i>Lampe</i> \sqcap \exists <i>Teil</i> : <i>Mixstab</i>

[Baader & Nutt 2003,
Baader et al. 2008]

Ausdrucksstärkere Beschreibungslogiken: *ALCN* – Attributive Language with full Complements and Number Restrictions

F. Baader, W. Nutt

Woman	≡	Person \sqcap Female
Man	≡	Person \sqcap \neg Woman
Mother	≡	Woman \sqcap \exists hasChild.Person
Father	≡	Man \sqcap \exists hasChild.Person
Parent	≡	Father \sqcup Mother
Grandmother	≡	Mother \sqcap \exists hasChild.Parent
MotherWithManyChildren	≡	Mother \sqcap ≥ 3 hasChild
MotherWithoutDaughter	≡	Mother \sqcap \forall hasChild. \neg Woman
Wife	≡	Woman \sqcap \exists hasHusband.Man

Fig. 2.2. A terminology (TBox) with concepts about family relationships.

[Baader & Nutt 2003,
Baader et al. 2008]

Ausdrucksstärkere Beschreibungslogiken: *ALCN* – Attributive Language with full Complements and Number Restrictions

Person $\sqcap (\leq 1 \text{ hasChild} \sqcup (\geq 3 \text{ hasChild} \sqcap \exists \text{hasChild.Female}))$

➤ Was heißt das?

BinaryTree \equiv Tree $\sqcap \leq 2 \text{ has-branch} \sqcap \forall \text{has-branch.BinaryTree}$.

➤ Was für ein Definitonstyp ist das?

✓ zyklische / rekursive Definition

Ausdrucksstärkere Beschreibungslogiken: \mathcal{ALCN} – Attributive Language with full Complements and Number Restrictions

$$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b. (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$$

$$(\exists R.\top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}}\}.$$

$$(\geq n R)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \geq n \right\}$$

$$(\leq n R)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \leq n \right\}$$

Schlussverfahren in Beschreibungslogiken

Schmidt-Schauß & Smolka (1991): Inferenzalgorithmen für Tests auf Subsumption und Erfüllbarkeit

➤ Begründung als Hausaufgabe (s.a. F2 & F5)

(1) Constraint-Lösungsmethode:

- Subsumption zwischen 2 Begriffen ($C \sqsubseteq C'$ bzw. $C \preceq C'$ als S. in der leeren Terminologie) wird auf Inkonsistenz des Begriffsausdrucks $C \sqcap \neg C'$ reduziert.
- Inkonsistenz-Entscheidungsalgorithmus: Jede mögliche Vervollständigung des Wurzel-Constrainsystems $\{(x: C \sqcap \neg C')\}$ führt zu einem Widerspruch.

(2) Tableauverfahren

- Nachweis der Konsistenz einer Wissensbasis $\mathcal{K} = \{\mathcal{T}, \mathcal{A}\}$ durch Konstruktion eines Modells für \mathcal{K}
- Umformung der DL-Ausdrücke in Normalform (h.: NNF – Negation Normal Form)
 - NNF: Negation nur vor Begriffsnamen
 - $\neg(\exists x.A \sqcap \forall y.B)$ wird umgeformt zu äquivalentem $(\forall x. \neg A \sqcap \exists y. \neg B)$
- Ausgehend von der in \mathcal{A} beschriebenen Situationen werden zusätzliche Constraints für das Modell expliziert, die durch die Konzepte in \mathcal{A} und die Axiome/Konzeptbeziehungen in \mathcal{T} impliziert werden.
 - Tableau-typische Transformations- bzw. Propagationsregeln (s. nächste Seite)

vgl. F9

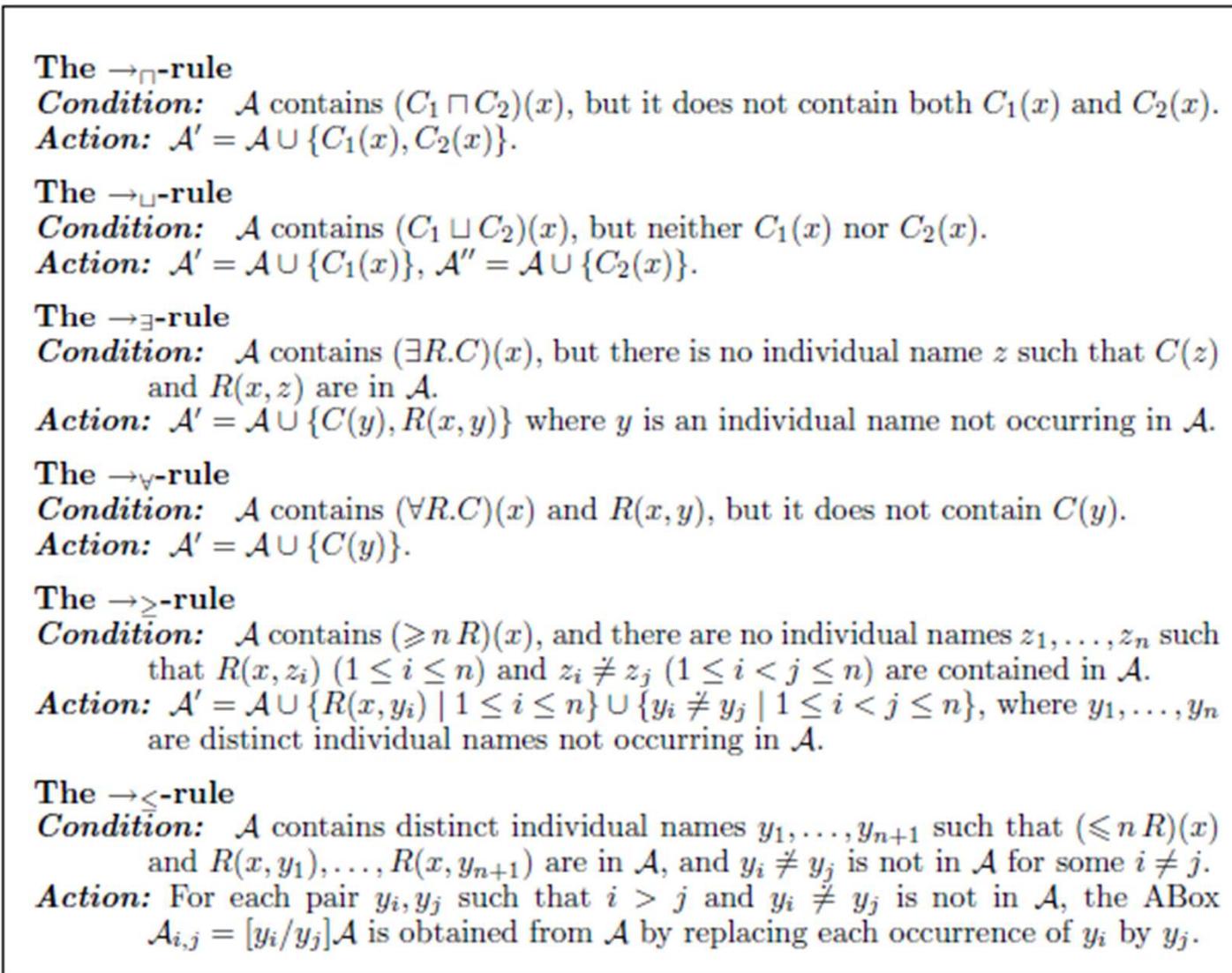


Fig. 2.6. Transformation rules of the satisfiability algorithm.

- In vielen DLs sind Erfüllbarkeit und Subsumption deshalb entscheidbar (wenn auch u.U. hoch komplex), weil die Begriffsconstructoren (concept constructors) dieser DLs nur lokale Eigenschaften eines Objekts ausdrücken.
 - Beispiel: In \mathcal{ALC} drückt eine Aussage $C(x)$ Eigenschaften von x aus sowie von Objekten, die mit x über eine Kette von höchstens $|C|$ Rollenzuschreibungen verbunden sind. Dies bedeutet, dass ein Constraint bzgl. x nichts über Objekte aussagt, die beliebig weit (bzgl. Rollenverbindungen) von x entfernt sind.
 - \mathcal{ALC} entspricht einem PL1-Fragment aus wfA mit maximal 2 Variablen
- Nicht-lokale DLs können unentscheidbar sein.
 - Beispiel: \mathcal{ALR} (Schmidt-Schauß 1989) erlaubt 'equality role-value-maps' wie z.B. $\text{co-worker} = \text{relative}$ (für *persons whose co-workers coincide with their relatives*, z.B. in einem Begriffssystem für kleine Familienunternehmen) und ist nachweislich unentscheidbar.

Die Entwicklung von Beschreibungslogiken - 1

[Nardi & Brachmann 2003]

- Wissensbasierte Systeme können implizite Folgerungen aus ihrem explizit repräsentierten Wissen ermitteln.
 - logikbasierte Systeme
 - logische Folgerung
 - präzise denotationelle Semantik
 - netzwerkbasierte Systeme (Semantic Networks, Frames)
 - kognitiv motivierte Schlüsse
 - keine präzise Semantik (für frühe Systeme ca.1970-1985)
 - dadurch kein einheitlicher Folgerungsbegriff
- Beschreibungslogiken (DL - Description Logics) stellen eine Synthese aus logikbasierten und netzwerkbasierten Systemen dar
 - präziser Formalismus mit klar definierter Semantik inkl. Folgerungsbegriff
 - empirisch-experimenteller Ansatz:
 - verschiedene Formalismen mit unterschiedlicher Ausdrucksstärke (gemäß Modellierungsanforderungen der Domäne) und Verarbeitungseigenschaften (Komplexität)
 - parallele Entwicklung von Theorie (Syntax, Semantik, Entscheidbarkeit, Komplexität) und implementierten Systemen (DL-KRS, DL-based Knowledge Representation Systems)
 - Optimierung der Verarbeitung auch im Fall von höherer worst case complexity

Die Entwicklung von Beschreibungslogiken - 2

[Baader et al. 2008]

4 Phasen in der Forschung zur Balance (trade off) zwischen Ausdrucksstärken von DLn und der Komplexität des Schließens in diesen

1. [1980-1990] Schwerpunkt auf Implementierung von Systemen (KL-ONE, K-REP, BACK, LOOM, CLASSIC)
 - a. (+) effiziente Subsumptionsalgorithmen (structural subsumption algorithms) – Polynomialzeit
 - b. (-) vollständig nur für sehr ausdruckschwache DLs ► bei einigermaßen ausdrucksstarken DLs werden nicht alle Subsumptions- und Instanz-Beziehungen entdeckt.
2. [1990-1995] Entwicklung von Tableau-basierten Algorithmen als neuem Standard (KRIS, CRACK)
 - a. (+) vollständig auch für ausdrucksstarke DLs
 - b. (-) hohe Worst-Case-Komplexität (außerhalb Polynomialzeit)
 - i. (+) aber: optimierte Implementierungen der Algorithmen führen zu akzeptablem Verarbeitungsverhalten in der Praxis
3. [1995-2005] Inferenzalgorithmen für sehr ausdrucksstarke DLs mit gutem Verarbeitungsverhalten auch in großen Wissensbasen (Tableau-basierte Algorithmen oder mittels Übersetzung in Modallogiken)
4. [2005-dato] Produktive Anwendungen (z.B. Bio-Informatik) mit sehr ausdrucksstarken DLs und effizienten Tableau-basierten Algorithmen

Beschreibungslogiken als Ontologie-Sprachen für das Semantic Web

- Das Semantic Web (Berners-Lee et al. 2001) zielt auf maschinen-verstehbare Web-Ressourcen ab, deren Information sowohl von automatischen Tools (z.B. Suchmaschinen) als auch von menschlichen Nutzern verarbeitet werden kann.
- Der aktuelle Standard für Konzept-Annotationen im Semantic Web ist die Markup-Sprache *Web Ontology Language OWL*. Vorgänger von OWL war *DAML+OIL [DARPA Agent Markup Language plus Ontology Inference Layer]*.
 - aktuelle OWL-Varianten: OWL Lite, OWL DL, OWL Full
- Beschreibungslogiken (DL) sind nützlich für die Definition, Integration und Wartung der OWL bzw. DAML+OIL zugrunde liegenden Ontologien, da sie dem Semantic Web ein klar definiertes gemeinsames Verständnis der semantischen Konzepte liefern, die für die Annotation von Webseiten verwendet werden.

Beschreibungslogiken als Ontologie-Sprachen für das Semantic Web

- OWL Lite ist äquivalent zur DL $\mathcal{SHDN}(\mathbf{D})$, OWL DL zu $\mathcal{SHOQN}(\mathbf{D})$:
 - Anreicherung von \mathcal{ALC} :
 - \mathcal{S} : \mathcal{ALC} plus transitive Rollen
 - \mathcal{H} : Sub-Rollen (Rollen-Hierarchien)
 - \mathcal{O} : Nominals
 - \mathcal{D} : Inverse Rollen
 - \mathcal{Q} : Qualified Number Restrictions (*)
 - \mathbf{D} : eine konkrete Domäne (Datentyp) ist integriert
 - $\mathcal{SHDN}/\mathcal{SHOQN}$ entsprechen entscheidbaren Fragmenten der PL1; Schließen/Folgern ist damit in diesen DLs entscheidbar.
 - OWL-Full erlaubt höherstufige Konstrukte und ist daher unentscheidbar.
 - Diese DLs haben zwar eine hohe Worst-Case-Komplexität (ExpTime), die sich aber wegen hoch optimierter Inferenzalgorithmen (Reasoners) in der Praxis nicht negativ auswirkt.

(*) nb: \mathcal{Q} ist auch in $\mathcal{SHDN}/\mathcal{SHOQN}$ enthalten; für den OWL-Vorgänger DAML+OIL hatten Baader et al. (2005) noch die DL \mathcal{SHDQ} vorgeschlagen, die aber von $\mathcal{SHDN}/\mathcal{SHOQN}$ eingeschlossen wird.

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor
someValuesFrom	$\exists r.C$	\exists hasChild.Lawyer
hasValue	$\exists r.\{x\}$	\exists citizenOf.{USA}
minCardinality	$(\geq n r)$	$(\geq 2$ hasChild)
maxCardinality	$(\leq n r)$	$(\leq 1$ hasChild)
inverseOf	r^-	hasChild $^-$

Figure 3.6: OWL constructors

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameAs	$\{x_1\} \equiv \{x_2\}$	{Pres_Bush} \equiv {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
TransitiveProperty	P transitive role	hasAncestor is a transitive role
FunctionalProperty	$\top \sqsubseteq (\leq 1 P)$	$\top \sqsubseteq (\leq 1$ hasMother)
InverseFunctionalProperty	$\top \sqsubseteq (\leq 1 P^-)$	$\top \sqsubseteq (\leq 1$ isMotherOf $^-)$
SymmetricProperty	$P \equiv P^-$	isSiblingOf \equiv isSiblingOf $^-$

Figure 3.7: OWL axioms

Anwendungsbereiche von Beschreibungslogiken & OWL

- Konfiguration (z.B. Telekommunikation, Automobil, Gebäudebau, ...)
- Software Engineering: SW Informations- und Dokumentationssysteme
- Datenbanken
 - Schema Design
 - Schema and Data Integration
 - Query Answering
- Ontologien für spezifische Domänen der Wissensrepräsentation
 - Biologie
 - Medizin
 - Bio-Medizinische Informatik
 - Medical Decision Support Systems
 - Ontologien für medizinisches Wissen
 - Geographie
 - Geologie
 - Astronomie
 - Landwirtschaft
 - Verteidigung
 - ...

Aufgaben zu Beschreibungslogiken finden Sie
in der Übungsklausur 2

Das war's (für's Erste ...) !

- Wozu Logik?
- Aussagenlogik: Syntax, Semantik
 - Schlussregel-Kalkül (Kalkül des Natürlichen Schließens), Tableau-Verfahren, Resolution
- Prädikatenlogik 1. Stufe: Syntax, Semantik
 - modelltheoretische / denotationelle Semantik der PL1 als Paradigma für formale Semantiken
- Höherstufige Logiksysteme: Typenlogik und λ -Kalkül
- Intensionale Logiksysteme: Temporallogik
- Logiksysteme für praktische CL-Anwendungen: Description Logics
- wer mehr wissen will: optionale Folien zu
 - weitere intensionale Logiksysteme (Modallogik, Epistemische Logik, Konditionale)
 - mereologische Domänen: Semantik von Pluralen und Massentermen
- und nun: *viel Erfolg bei der Abschlussklausur!*
- nutzen Sie für die Vorbereitung die *Übungsklausur!*