Konsistenz von Wissensbasen in Beschreibungslogiken mit Rollenoperatoren

Ralf Molitor
Matrikelnummer 191286

Diplomarbeit
im Fach Informatik

Lehr- und Forschungsgebiet Theoretische Informatik
Rheinisch-Westfälische Technische Hochschule Aachen
Prof. Dr.-Ing. Franz Baader

Betreuung:
Prof. Dr.-Ing. Franz Baader

Aachen, Juni 1997

Für meine Eltern, ohne deren Hilfe
mir ein erfolgreiches Studium
nicht möglich gewesen wäre.

Erklärung

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, im Juni 1997 ....

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

In dieser Arbeit werden einige Beschreibungslogiken untersucht. Beschreibungslogiken werden zur strukturierten Darstellung von Informationen aus einem Anwendungsbereich in Wissensrepräsentationssystemen eingesetzt. Sie stammen ab von dem von Brachman entwickelten System KL-ONE [BS85]. Die Idee ist hier, daß man aus atomaren *Konzepten* (einstelligen Prädikaten) und *Rollen* (zweistelligen Relationen) mit Hilfe von Konstruktoren komplexere Konzepte und Rollen aufbaut.

Der gemeinsame Kern der in dieser Arbeit untersuchten Beschreibungslogiken ist die Logik $\mathcal{ALC}$. In $\mathcal{ALC}$ definiert man, ausgehend von atomaren Konzepten und atomaren Rollen, komplexe Konzepte mit Hilfe der Konstruktoren *Konjunktion* ($C \sqcap D$), *Disjunktion* ($C \sqcup D$), *Negation* ($\neg C$), *Werterestriktion* ($\forall R.C$) und *Existenzrestriktion* ($\exists R.C$). Dabei erlaubt man keine komplexen Rollen.

Beispiel 1.1
In den Beispielen dieses Abschnitts werden atomare Konzepte groß und atomare Rollen klein geschrieben.

Der Konzeptterm

$$\mathsf{Person} \sqcap \mathsf{Weiblich} \sqcap \exists \mathsf{verheiratet\_mit}.\mathsf{Männlich} \sqcap \exists \mathsf{hat\_Kind}.\mathsf{Person}$$

beschreibt alle verheirateten Frauen, die Kinder haben. ◇

Bei den in der vorliegenden Arbeit untersuchten Erweiterungen von $\mathcal{ALC}$ verwendet man *Rollenterme*, um komplexe Beziehungen zwischen Elementen zu beschreiben. Interessiert man sich für die Frage, mit wie vielen Elementen ein Element in einer bestimmten Beziehung stehen darf bzw. muß, so verwendet man *Zahlenrestriktionen*, um diese Bedingungen zu formulieren.

Beispiel 1.2 (Beispiel 1.1 Fortsetzung)
Frauen, die mindestens vier Enkel und keine Urenkel haben, beschreibt man durch den Konzeptterm

$$\begin{array}{l} \mathsf{Person} \sqcap \mathsf{Weiblich} \sqcap \\ (\geq 4\ \mathsf{hat\_Kind} \circ \mathsf{hat\_Kind}) \sqcap \\ (\leq 0\ \mathsf{hat\_Kind} \circ \mathsf{hat\_Kind} \circ \mathsf{hat\_Kind}). \end{array}$$

Dabei wird die Anzahl ihrer Kinder nicht eingeschränkt. ◇

Demgegenüber verwendet man *Role-Value-Maps*, um für ein Element zu beschreiben, welche Arten von Beziehungen zwischen ihm und anderen Elementen bestehen müssen bzw. nicht bestehen dürfen.

Beispiel 1.3 (Beispiel 1.1 Fortsetzung)
Der Konzeptterm

$$\begin{array}{l} \mathsf{Person} \sqcap \mathsf{Weiblich} \sqcap \exists \mathsf{hat\_Kind.Person} \sqcap \\ (\mathsf{hat\_Kind} \circ \mathsf{hat\_Freund} \sqsubseteq \mathsf{kennt}) \end{array}$$

verwendet Role-Value-Maps, um Mütter zu beschreiben, die alle Freunde ihrer Kinder kennen und der Term

$$\exists \mathsf{hat\_Mutter.Person} \sqcap \neg(\mathsf{hat\_Lehrer} \sqsubseteq \mathsf{mag})$$

verwendet negative Role-Value-Maps, um Kinder zu beschreiben, die nicht jeden ihrer Lehrer mögen. ◇

Mit Hilfe der Beschreibungslogiken lassen sich also Zusammenhänge aus einem Anwendungsbereich durch geeignete Konzept- und Rollenterme repräsentieren. Man ist aber auch daran interessiert, konkrete Welten, d.h., konkrete Situationen im Anwendungsbereich, darzustellen. Dazu verwendet man sogenannte *ABoxen*. Eine ABox ist eine endliche Menge von *Assertionen*. Eine Assertion steht für eine Bedingung, die an ein *Individuum* gestellt wird.

Beispiel 1.4 (Beispiel 1.1 Fortsetzung)
Es sei folgende Menge {Emma, Fritz, Hans, Heinz, Anna, Susi, Rudi} von Individuen gegeben.

Die folgende ABox beschreibt einen Auszug eines Familienstammbaums:

$$\begin{array}{ll} \mathcal{A} = \{ & \mathsf{Emma\ verheiratet\_mit\ Fritz}, \\ & \mathsf{Emma\ hat\_Kind\ Heinz}, \\ & \mathsf{Emma\ hat\_Kind\ Hans}, \\ & \mathsf{Emma\ hat\_Kind\ Anna}, \\ & \mathsf{Hans\ hat\_Kind\ Susi}, \\ & \mathsf{Heinz\ hat\_Kind\ Rudi}, \\ & \mathsf{Emma} : (\leq 3\ \mathsf{hat\_Kind}) \sqcap (\geq 7\ \mathsf{hat\_Kind} \circ \mathsf{hat\_Kind}), \\ & \mathsf{Hans} : (\leq 2\ \mathsf{hat\_Kind}), \\ & \mathsf{Heinz} : (\geq 2\ \mathsf{hat\_Kind}), \\ & \mathsf{Anna} : (\leq 0\ \mathsf{hat\_kind}) \qquad \} \end{array}$$

Die letzten vier Assertionen beschreiben die Bedingungen an die Anzahl von Kindern und Enkeln der Familienmitglieder. Oma Emma hat drei Kinder und wird keine mehr bekommen, 'verlangt' aber sieben Enkel. Die ABox $\mathcal{A}$ ist offensichtlich nicht widersprüchlich, da Heinz die nötige Anzahl Kinder haben kann. Fügt man aber die Assertion $\mathsf{Heinz} : (\leq\ 4\ \mathsf{hat\_Kind})$ hinzu, so erhält man eine widersprüchliche ABox, da die drei Kinder von Emma zusammen höchstens sechs Enkel haben können. ◇

Neben der Darstellung expliziten Wissens interessiert man sich auch dafür, implizites Wissen aus der Repräsentation des expliziten Wissens zu gewinnen. Dazu verwendet man Schlußfolgerungsmechanismen und ist dabei natürlich insbesondere an effektiven Verfahren interessiert. Um nun festzustellen, ob es solche effektiven Verfahren überhaupt gibt, untersucht man, wie im Fall der klassischen Logik, spezielle *Entscheidbarkeitsprobleme* für die betrachteten Beschreibungslogiken. Dazu zählen das *Erfüllbarkeitsproblem*, das *Subsumtionsproblem* und das *Konsistenzproblem*.

Das Ziel dieser Arbeit besteht nun darin, zum einen das Konsistenzproblem für eine spezielle Beschreibungslogik mit Zahlenrestriktionen und zum anderen das Erfüllbarkeits- und Subsumtionsproblem für eine bestimmte Beschreibungslogik mit Role-Value-Maps zu untersuchen.

1.2 Grundlagen

In diesem Abschnitt werden die grundlegenden Definitionen und Sätze eingeführt.

Definition 1.5 (Konzept- und Rollenterme: Syntax)
Es seien $\mathcal{N_C}$ und $\mathcal{N_R}$ disjunkte Mengen von Konzept- und Rollennamen. Die Menge der Konzeptterme und die Menge der Rollenterme ist induktiv definiert:

1. Jeder Konzeptname ist ein *Konzeptterm* und jeder Rollenname ist ein *Rollenterm*.
2. Sind C_1, C_2 Konzeptterme, R_1, R_2 Rollenterme und $n \in \mathbb{N}$, so sind
   - $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\neg C_1$
   - $\forall R_1.C_1$, $\exists R_1.C_1$
   - $(\geq n\ R_1)$, $(\leq n\ R_1)$
   - $R_1 \sqsubseteq R_2$

   Konzeptterme und

   - $R_1 \circ R_2$

   ist ein *Rollenterm*. ◇

Die in den Kapiteln 2 und 3 untersuchten Sprachen werden die Verwendung einiger der hier angegebenen Konstruktoren noch einschränken bzw. ganz verbieten.

Wie im Bereich der klassischen Logik definiert man die Semantik der Beschreibungslogiken mit Hilfe von *Interpretationen*. Eine Interpretation $\mathcal{I}$ besteht aus einem Interpretationsbereich Δ_I und einer Interpretationsfunktion, die jedem Konzeptnamen A eine Menge $A^I \subseteq \Delta_I$ und jedem Rollennamen R eine binäre Relation $R^I \subseteq \Delta_I \times \Delta_I$ zuweist. Ähnlich wie im Fall der klassischen Logik wird die Interpretationsfunktion auf komplexe Konzepte und Rollen erweitert und eine Interpretation ist genau dann ein *Modell* eines Konzeptes C, wenn $C^I \neq \emptyset$ gilt.

Definition 1.6 (Konzept- und Rollenterme: Semantik)
Eine Interpretation ist eine Struktur $\mathcal{I}$, die aus einem nicht-leeren Interpretationsbereich Δ_I und einer Interpretationsfunktion besteht, die

- jedem $A \in \mathcal{N_C}$ eine Teilmenge A^I von Δ_I und
- jedem $R \in \mathcal{N_R}$ eine binäre Relation $R^I \subseteq \Delta_I \times \Delta_I$

zuweist. Die Interpretationsfunktion wird auf Konzept- und Rollenterme wie folgt erweitert:

$$\begin{array}{rcl}
(C_1 \sqcap C_2)^I & := & C_1^I \cap C_2^I \\
(C_1 \sqcup C_2)^I & := & C_1^I \cup C_2^I \\
(\neg C_1)^I & := & \Delta_I \setminus C_1^I \\
(\forall R_1.C_1)^I & := & \{d \in \Delta_I \mid \forall e \in \Delta_I : (d,e) \in R_1^I \Rightarrow e \in C_1^I\} \\
(\exists R_1.C_1)^I & := & \{d \in \Delta_I \mid \exists e \in \Delta_I : (d,e) \in R_1^I \wedge e \in C_1^I\} \\
(\geq n\ R_1)^I & := & \{d \in \Delta_I \mid |R_1^I(d)| \geq n\} \\
(\leq n\ R_1)^I & := & \{d \in \Delta_I \mid |R_1^I(d)| \leq n\} \\
(R_1 \sqsubseteq R_2)^I & := & \{d \in \Delta_I \mid R_1^I(d) \subseteq R_2^I(d)\} \\
(R_1 \circ R_2)^I & := & \{(d,e) \in \Delta_I \times \Delta_I \mid \exists d' \in \Delta_I : (d,d') \in R_1^I \wedge (d',e) \in R_2^I\}
\end{array}$$

wobei $R^I(d) := \{e \in \Delta_I \mid (d,e) \in R^I\}$. Eine Interpretation $\mathcal{I} = (\Delta_I, \cdot^I)$ ist *Modell* eines Konzeptterms C bzw. eines Rollenterms R, falls $C^I \neq \emptyset$ bzw. $R^I \neq \emptyset$. Ein Element $a \in \Delta_I$ mit $a \in C^I$ heißt *Instanz von C*. ◇

Bei der Untersuchung des Erfüllbarkeits- und Konsistenzproblems kann man sich auf Konzeptterme in Negationsnormalform (NNF) beschränken.

Ein Konzeptterm ist in NNF, wenn keine der folgenden Regeln auf ihn anwendbar ist:

$$\begin{array}{rcl}
\neg\neg C & \longrightarrow & C \\
\neg(C \sqcap D) & \longrightarrow & \neg C \sqcup \neg D \\
\neg(C \sqcup D) & \longrightarrow & \neg C \sqcap \neg D \\
\neg(\forall R.C) & \longrightarrow & \exists R.\neg C \\
\neg(\exists R.C) & \longrightarrow & \forall R.\neg C \\
\neg(\leq n\ \mathcal{R}) & \longrightarrow & (\geq (n+1)\ \mathcal{R}) \\
\neg(\geq n\ \mathcal{R}) & \longrightarrow & \begin{cases} (\leq (n-1)\ \mathcal{R}) & ; \quad n \geq 1 \\ A \sqcap \neg A & ; \quad n = 0, A \in \mathcal{N_C} \end{cases}
\end{array}$$

Offensichtlich liefern diese Regeln einen jeweils äquivalenten Konzeptterm. Dabei heißen zwei Konzeptterme C, D *äquivalent*, falls für jede Interpretation $\mathcal{I} = (\Delta_I, \cdot^I)$ $C^I = D^I$ gilt. Durch iterierte Anwendung der obigen Regeln erhält man also zu jedem Konzeptterm C einen äquivalenten Term C' in NNF.

Bei der Formulierung der Algorithmen sowie den Beweisen zur Korrektheit, Vollständigkeit und Terminierung benötigt man die folgenden Begriffe:

Definition 1.7
Sei C ein Konzeptterm in NNF.

$|C|_{\sqcap,\sqcup}$ bezeichnet die *und/oder-Größe von C* und steht für die Anzahl von Konjunktions- und Disjunktionskonstruktoren in C.

$\mathsf{Sub}(C)$ steht für die *Menge der Unterkonzepte von C* und ist induktiv definiert durch:

$$\begin{array}{rcl} C \in \mathsf{Sub}(C) & & \\ C_1 \sqcap C_2 \in \mathsf{Sub}(C) & \Rightarrow & C_1, C_2 \in \mathsf{Sub}(C) \\ C_1 \sqcup C_2 \in \mathsf{Sub}(C) & \Rightarrow & C_1, C_2 \in \mathsf{Sub}(C) \\ \neg C_1 \in \mathsf{Sub}(C) & \Rightarrow & C_1 \in \mathsf{Sub}(C) \\ \forall R.C_1 \in \mathsf{Sub}(C) & \Rightarrow & C_1 \in \mathsf{Sub}(C) \\ \exists R.C_1 \in \mathsf{Sub}(C) & \Rightarrow & C_1 \in \mathsf{Sub}(C) \end{array}$$

Mit $\mathsf{depth}(C)$ bezeichnet man die *maximale Rollentiefe von C*. Diese ist induktiv definiert durch:

$$\begin{array}{l} \mathsf{depth}(A) := \mathsf{depth}(\neg A) := 0, \\ \mathsf{depth}(C_1 \sqcap C_2) := \mathsf{depth}(C_1 \sqcup C_2) := \mathsf{max}\{\mathsf{depth}(C_1), \mathsf{depth}(C_2)\}, \\ \mathsf{depth}(\forall R_1.C_1) := \mathsf{depth}(\exists R_1.C_1) := 1 + \mathsf{depth}(C_1), \\ \mathsf{depth}(\geq n\ R_1 \circ \ldots \circ R_m) := \mathsf{depth}(\leq n\ R_1 \circ \ldots \circ R_m) := m, \\ \mathsf{depth}(R_1 \circ \cdots \circ R_m \sqsubseteq S_1 \circ \cdots \circ S_m) := m, \end{array}$$

wobei $A \in \mathcal{N}_\mathcal{C}$ und R_i, S_j Rollennamen sind. ◇

Beachte: Für Role-Value-Maps ($\mathcal{R} \sqsubseteq \mathcal{S}$) mit Rollentermen unterschiedlicher Länge ist die maximale Rollentiefe nicht definiert. Die Beschreibungslogik, für die in Kapitel 3 das Erfüllbarkeitsproblem untersucht wird, erlaubt aber nur Role-Value-Maps mit Rollentermen gleicher Länge, so daß obige Definition ausreichend ist.

Unter bestimmten Voraussetzungen läßt sich das *Erfüllbarkeitsproblem für Konzepte* auf das *Konsistenzproblem für ABoxen* reduzieren (siehe Definition 1.10 und Theorem 1.11). Mit den in den Kapiteln 2 und 3 vorgestellten Algorithmen soll die *Konsistenz von ABoxen* entschieden werden.

Definition 1.8 (ABox)
Es sei $\mathcal{N}_\mathcal{I}$ eine endliche Menge von Individuen und $\tau = \{x_0, x_1, x_2, \ldots\}$ eine abzählbar unendliche Menge von Variablen.

Eine *Assertion* ist von der Form

- $x : C$ (*Konzeptassertion*),
- $x \neq y$ (*Ungleichheitsassertion*),
- xRy (*Rollenassertion*),
- $\neg(x\mathcal{R}y)$ (*negative Rollenassertion*),

wobei $x, y \in \mathcal{N}_\mathcal{I} \cup \tau$, C ein Konzeptterm, R ein Rollenname und $\mathcal{R}$ ein Rollenterm ist.

Eine *ABox* $\mathcal{A}$ ist eine nicht-leere, endliche Menge von Assertionen.

Für eine ABox $\mathcal{A}$ bezeichnet $\tau_\mathcal{A}$ die Menge der in $\mathcal{A}$ auftretenden Individuen und Variablen.

Für eine ABox $\mathcal{A}$, Elemente $x, y \in \tau_\mathcal{A}$ und einen Rollenterm $\mathcal{R} = R_1 \circ \ldots \circ R_n$ mit Rollennamen $R_1, \ldots, R_n \in \mathcal{N}_\mathcal{R}$ ist y *$\mathcal{R}$-Nachfolger von* x bzw. x *$\mathcal{R}$-Vorfahre von* y in $\mathcal{A}$, wenn Elemente $x_0, \ldots, x_n \in \tau_\mathcal{A}$ existieren mit

- $x_0 = x, x_n = y$ und
- $\{x_i R_{i+1} x_{i+1} \mid 0 \leq i < n\} \subseteq \mathcal{A}$.

Man schreibt dann kurz $y \in \mathcal{R}(x)$ in $\mathcal{A}$.

Für eine ABox $\mathcal{A}$ beschreibt $\mathcal{A}[x/y]$ die ABox, die aus $\mathcal{A}$ entsteht, wenn alle Vorkommen von x in $\mathcal{A}$ durch y ersetzt werden.

Für die Menge der Individuen wird im allgemeinen die *Unique Name Assumption* (UNA) angenommen. Diese fordert, daß zwei verschiedene Individuen auch stets verschieden interpretiert werden. Mit Hilfe der Ungleichheitsassertionen wird die UNA in der ABox $\mathcal{A}$ explizit gemacht, d.h., sollen zwei Elemente $x, y \in \tau_\mathcal{A}$ verschieden interpretiert werden, so enthält $\mathcal{A}$ die Assertion $x \neq y$. Diese Assertionen sind symmetrisch, d.h., ist $x \neq y \in \mathcal{A}$, so wird auch $y \neq x \in \mathcal{A}$ angenommen. Zwei Elemente x, y werden in $\mathcal{A}$ *unterschieden*, falls $x \neq y \in \mathcal{A}$ gilt. Eine *verallgemeinerte ABox* $\mathcal{A}$ ist eine ABox, die für je zwei Individuen $a, b \in \mathcal{N}_\mathcal{I}$ eine Ungleichheitsassertion $a \neq b$ und nur Konzeptassertionen $x : C$ enthält, die nur Konzeptterme C in NNF enthalten.

Eine ABox $\mathcal{A}$ enthält einen *clash* genau dann, wenn einer der folgenden Fälle zutrifft:

- $\{x : A, x : \neg A\} \subseteq \mathcal{A}$ für ein $x \in \tau_\mathcal{A}$ und ein $A \in \mathcal{N}_\mathcal{C}$,
- $x : (\leq n\, \mathcal{R}) \in \mathcal{A}$ und x hat $l > n$ $\mathcal{R}$-Nachfolger $x_1, \ldots, x_l$ in $\mathcal{A}$ mit $\{x_i \neq x_j \mid 1 \leq i < j \leq l\} \subseteq \mathcal{A}$,
- $\neg(x\mathcal{R}y) \in \mathcal{A}$ und y ist $\mathcal{R}$-Nachfolger von x in $\mathcal{A}$,

- $x \neq x \in \mathcal{A}$.

Enthält $\mathcal{A}$ einen clash, so heißt $\mathcal{A}$ *geschlossen*, sonst *offen*.

Eine Interpretation $\mathcal{I} = (\Delta_I, \cdot^I)$ ist *Modell der ABox* $\mathcal{A}$, falls $\mathcal{I}$ die Elemente aus $\tau_{\mathcal{A}}$ so interpretiert, daß $\mathcal{I}$ jede Assertion aus $\mathcal{A}$ erfüllt, d.h.,

$$\begin{array}{rll} (x^I, y^I) \in R^I & \text{für alle} & xRy \in \mathcal{A}, \\ x^I \neq y^I & \text{für alle} & x \neq y \in \mathcal{A}, \\ x^I \in C^I & \text{für alle} & x : C \in \mathcal{A}, \\ (x^I, y^I) \notin \mathcal{R}^I & \text{für alle} & \neg(x\mathcal{R}y) \in \mathcal{A}. \end{array}$$

Eine ABox $\mathcal{A}$ ist *konsistent* genau dann, wenn ein Modell von $\mathcal{A}$ existiert. Andernfalls ist $\mathcal{A}$ *inkonsistent*. $\diamond$

Bemerkung 1.9

Zu jeder ABox $\mathcal{A}$ erhält man eine verallgemeinerte ABox $\mathcal{A}'$ wie folgt:
Man ersetzt jede Konzeptassertion $x : C \in \mathcal{A}$ durch $x : \hat{C}$, wobei $\hat{C}$ der zu C äquivalente Konzeptterm in NNF sei. $\hat{\mathcal{A}}$ bezeichne die so erhaltene ABox. Dann definiert man $\mathcal{A}'$ durch

$$\mathcal{A}' := \hat{\mathcal{A}} \cup \{a \neq b \mid a, b \in \mathcal{N}_{\mathcal{I}},\ a, b \text{ verschieden}\}.$$

Unter Berücksichtigung der UNA gilt dann offensichtlich für jede Interpretation $\mathcal{I}$: $\mathcal{I}$ ist genau dann Modell von $\mathcal{A}$, wenn $\mathcal{I}$ Modell von $\mathcal{A}'$ ist. $\diamond$

Es folgt eine Übersicht über die wichtigsten Entscheidbarkeitsprobleme für Beschreibungslogiken und ihre Zusammenhänge.

Definition 1.10 (Entscheidbarkeitsprobleme)

1. Das Erfüllbarkeitsproblem:
   Gegeben: ein Konzeptterm C
   Frage: Ist C erfüllbar, d.h., existiert eine Interpretation $\mathcal{I}$ mit $C^I \neq \emptyset$?

2. Das Subsumtionsproblem:
   Gegeben: zwei Konzeptterme C, D
   Frage: Wird C von D subsumiert, i.Z. $C \sqsubseteq D$, d.h., gilt für alle Interpretationen $\mathcal{I}$ die Inklusion $C^I \subseteq D^I$?

3. Das Konsistenzproblem:
   Gegeben: eine ABox $\mathcal{A}$
   Frage: Ist $\mathcal{A}$ konsistent, d.h., existiert eine Interpretation $\mathcal{I}$, die Modell von $\mathcal{A}$ ist? $\diamond$

Der folgende Satz formuliert die Zusammenhänge zwischen diesen Problemen.

Theorem 1.11
Seien C, D Konzeptterme. Dann gilt:

1. C ist erfüllbar genau dann, wenn $\{x_0 : C\}$ konsistent ist.
2. $C \sqsubseteq D$ genau dann, wenn $C \sqcap \neg D$ unerfüllbar ist.
3. C ist unerfüllbar genau dann, wenn $C \sqsubseteq A \sqcap \neg A$ für ein $A \in \mathcal{N}_\mathcal{C}$. ⌟

Bemerkung 1.12
Der Beweis von Theorem 1.11 ist trivial und es folgt offensichtlich:

1. Ist eine Sprache abgeschlossen unter Negation, so läßt sich das Subsumtionsproblem auf das Erfüllbarkeitsproblem reduzieren. Man ist deshalb daran interessiert, daß die untersuchten Sprachen unter Negation abgeschlossen sind.
2. Ist für eine Sprache das Konsistenzproblem entscheidbar, so ist auch das Erfüllbarkeitsproblem für diese Sprache entscheidbar. Insbesondere läßt sich ein Konsistenzalgorithmus leicht als Erfüllbarkeitsalgorithmus nutzen. Umgekehrt ist dies im allgemeinen nicht so leicht möglich, wie man in Kapitel 2 sehen wird. ◇

Es folgt die formale Definition der in den folgenden Kapiteln untersuchten Sprachen.

Definition 1.13 ($\mathcal{ALCN}$, $\mathcal{ALCN}(\circ)$, $\mathcal{ALC} + (\sqsubseteq)$, $\mathcal{ALC} + (\sqsubseteq_n)$)
Seien $\mathcal{N}_\mathcal{C}$ eine Menge von *Konzeptnamen* und $\mathcal{N}_\mathcal{R}$ eine Menge von *Rollennamen.*
Die Menge der $\mathcal{ALCN}(\circ)$*-Konzepte* enthält alle Konzepte, die aufgebaut sind über $\mathcal{N}_\mathcal{C}$ und $\mathcal{N}_\mathcal{R}$ unter Verwendung der Konstruktoren

- Konjunktion $(C \sqcap D)$,
- Disjunktion $(C \sqcup D)$,
- Negation $(\neg C)$,
- Werterestriktion $(\forall R.C)$,
- Existenzrestriktion $(\exists R.C)$ und
- Zahlenrestriktion $\big((\leq n\, \mathcal{R}),\ (\geq n\, \mathcal{R})\big)$,

wobei in Konzepten der Form $\forall R.C$ und $\exists R.C$ nur Rollennamen zugelassen sind.
Die Menge der $\mathcal{ALCN}$*-Konzepte* enthält genau die $\mathcal{ALCN}(\circ)$-Konzepte, die auch in den Zahlenrestriktionen nur Rollennamen enthalten, d.h., für Teilkonzepte der Form $(\leq n\, \mathcal{R})$ und $(\geq n\, \mathcal{R})$ gilt stets $\mathcal{R} \in \mathcal{N}_\mathcal{R}$.
Die Menge der $\mathcal{ALC} + (\sqsubseteq)$*-Konzepte* enthält alle Konzeptterme, die über $\mathcal{N}_\mathcal{C}$, $\mathcal{N}_\mathcal{R}$ und den Konstruktoren

- Konjunktion ($C \sqcap D$),
- Disjunktion ($C \sqcup D$),
- Negation ($\neg C$),
- Werterestriktion ($\forall R.C$),
- Existenzrestriktion ($\exists R.C$) und
- Role-Value-Maps ($\mathcal{R} \sqsubseteq \mathcal{S}$)

aufgebaut sind. Dabei dürfen Werte- und Existenzrestriktionen wie bei $\mathcal{ALCN}(\circ)$ nur Rollennamen enthalten, in den Role-Value-Maps dürfen komplexe Rollenterme auftreten.
Die Menge der $\mathcal{ALC} + (\sqsubseteq_n)$-*Konzepte* enthält genau die $\mathcal{ALC} + (\sqsubseteq)$-Konzepte, die nur Role-Value-Maps enthalten, die Rollenterme gleicher Länge enthalten, d.h., die die Form $(R_1 \circ \ldots \circ R_n \sqsubseteq S_1 \circ \ldots \circ S_n)$ für ein $n \in \mathbb{N}$ und Rollennamen R_i, $S_i \in \mathcal{N}_\mathcal{R}$ haben. $\diamond$

In den Beweisen der Entscheidbarkeitsresultate stützt man sich unter anderem auf folgende, mögliche Eigenschaften einer Beschreibungslogik ab.

Definition 1.14 (Eigenschaften)
Eine Beschreibungslogik hat die

Endliche-Modell-Eigenschaft, falls ein Konzept C genau dann erfüllbar ist, wenn ein *endliches Modell* von C existiert,

Baummodell-Eigenschaft, falls zu jedem erfüllbaren Konzept C auch ein *Baummodell* existiert,

Levelstruktur-Eigenschaft, falls zu jedem erfüllbaren Konzept C auch ein *Modell mit Levelstruktur* existiert.

Ein *Baummodell von C* ist eine Interpretation $\mathcal{I}$, für die gilt:

1. die Elemente aus Δ_I bilden einen Baum, d.h., es gibt eine Wurzel $a_0 \in \Delta_I$, die Instanz von C ist, so daß zu jedem $a \in \Delta_I$ genau ein Pfad von a_0 nach a in $\mathcal{I}$ existiert;
2. a_0 hat keinen Vorgänger, d.h., es ex. kein $b \in \Delta_I$ sodaß $(b, a_0) \in R^I$ für ein $R \in \mathcal{N}_\mathcal{R}$;
3. jedes andere Element $a \in \Delta_I$ hat genau einen Rollenvorgänger, d.h., es ex. genau ein $R \in \mathcal{N}_\mathcal{R}$ und genau ein $b \in \Delta_I$ mit $(b, a) \in R^I$.

Ein *Modell mit Levelstruktur von* C ist eine Interpretation $\mathcal{I}$, für die gilt:

1. es gibt ein $a_0 \in \Delta_I$, das Instanz von C ist, so daß zu jedem $a \in \Delta_I$ ein Pfad von a_0 nach a existiert;

2. a_0 hat keinen Vorgänger;

3. zu jedem anderen Element $a \in \Delta_I$ existiert ein $m \in \mathbb{N}$, so daß jeder Pfad von a_0 nach a in $\mathcal{I}$ die Länge m hat.

Es existiert ein *Pfad von* a_0 *nach* a_m *in* $\mathcal{I}$, falls es Elemente $a_1, \dots, a_{m-1} \in \Delta_I$ und Rollennamen $R_1, \dots, R_m$ gibt mit $(a_{i-1}, a_i) \in R_i^I$ für $1 \leq i \leq m$. ◇

Bemerkung 1.15
Aus der Definition der Baummodell- bzw. Levelstruktur-Eigenschaft in Definition 1.14 folgt offensichtlich, daß jede Beschreibungslogik, die die Baummodell-Eigenschaft hat, auch die Levelstruktur-Eigenschaft hat. Die Umkehrung gilt nicht. In 2.1 wird man sehen, daß $\mathcal{ALCN}(\circ)$ die Levelstruktur- aber nicht die Baummodell-Eigenschaft hat. ◇

In [BS96] wird bewiesen, daß das Erfüllbarkeitsproblem für $\mathcal{ALCN}(\circ)$ entscheidbar ist. Aus den Ergebnissen in [Sch89] folgt, daß Erfüllbarkeit für $\mathcal{ALC} + (\sqsubseteq)$ unentscheidbar ist. Im Beweis des Entscheidbarkeitsresultats nutzt man aus, daß $\mathcal{ALCN}(\circ)$ die *Levelstruktur-Eigenschaft* besitzt und im Beweis zur Unentscheidbarkeit des Erfüllbarkeitsproblems für $\mathcal{ALC} + (\sqsubseteq)$ nutzt man aus, daß $\mathcal{ALC} + (\sqsubseteq)$ *nicht* diese Eigenschaft hat. Das Fragment $\mathcal{ALC} + (\sqsubseteq_n)$ hat aber die Levelstruktur-Eigenschaft. Dies motiviert die Untersuchung des Erfüllbarkeitsproblems für $\mathcal{ALC} + (\sqsubseteq_n)$ in Kapitel 3. In Kapitel 2 wird das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ untersucht.

Kapitel 2

Die Sprache $\mathcal{ALCN}(\circ)$

Mit Hilfe von Zahlenrestriktionen kann die Zahl von $\mathcal{R}$-Nachfolgern für ein Individuum beschränkt werden. Dabei ist für die Beschreibungslogik $\mathcal{ALCN}(\circ)$ $\mathcal{R}$ von der Form $R_1 \circ \ldots \circ R_n$ für ein $n \geq 1$ und Rollennamen $R_i \in \mathcal{N}_\mathcal{R}$. Also können auch Bedingungen an die Anzahl der Nachfolger, die mit Rollenketten der Länge $n > 1$ mit einem Element verbunden sind, formuliert werden.

Im Abschnitt 2.1 wird ein regelbasiertes Verfahren aus [BS96] angegeben, das Erfüllbarkeit für $\mathcal{ALCN}(\circ)$ entscheidet. Die Idee besteht darin, zu einem erfüllbaren Konzept C ein *kanonisches Modell* durch iteriertes Anwenden sogenannter *Vervollständigungsregeln* zu erzeugen.

Leider ist mit diesem Verfahren das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ nicht entscheidbar, da eine iterierte Regelanwendung auf beliebige $\mathcal{ALCN}(\circ)$-ABoxen im allgemeinen nicht terminiert. Allerdings läßt sich für das Fragment $\mathcal{ALCN}$ durch Angabe einer einfachen Strategie zur Steuerung der Regelanwendung der Algorithmus so erweitern, daß er das Konsistenzproblem für beliebige $\mathcal{ALCN}$-ABoxen entscheidet. Auf Grund der Komposition von Rollentermen in Zahlenrestriktionen ist ein entsprechender Erweiterungsschritt für $\mathcal{ALCN}(\circ)$ aber wesentlich aufwendiger.

Ein Grund für die zusätzlichen Probleme, die sich bei der Untersuchung von $\mathcal{ALCN}(\circ)$-ABoxen ergeben, liegt darin, daß $\mathcal{ALCN}$ die Baummodell-Eigenschaft hat, $\mathcal{ALCN}(\circ)$ aber nicht mehr. Das $\mathcal{ALCN}(\circ)$-Konzept

$$C = (\geq 2\, R) \sqcap (\forall R.\exists S.A) \sqcap (\leq 1\, R \circ S)$$

ist offensichtlich erfüllbar, hat aber kein Baummodell, da in einem Modell von C mindestens zwei S-Vorfahren zu einem $(R \circ S)$-Nachfolger der Wurzel x_0 existieren.

Allerdings hat $\mathcal{ALCN}(\circ)$ noch die Levelstruktur-Eigenschaft, was sich zum einen im Terminierungsbeweis zum Erfüllbarkeitsalgorithmus und zum anderen auch für einige Aussagen zum Konsistenzproblem nutzen läßt.

Erfüllbarkeitsalgorithmus für $\mathcal{ALCN}(\circ)$:

Eingabe:	Ein $\mathcal{ALCN}(\circ)$-Konzept C
Datenstruktur:	Ein mit $\mathcal{ALCN}(\circ)$-ABoxen beschrifteter Baum.
1. Schritt:	Generiere einen Baum, der nur aus einer Wurzel besteht, die mit $\{x_0 : C_0\}$ beschriftet ist, wobei C_0 das zu C äquivalente $\mathcal{ALCN}(\circ)$-Konzept in NNF und x_0 eine Variable sei.
2. Schritt:	Wende solange wie möglich Regeln auf Blattbeschriftungen an, die offen sind. Sind alle Blattbeschriftungen vollständig, so terminiere.
Ausgabe:	„C ist erfüllbar" , falls eine vollständige, offene ABox im Baum existiert „C ist unerfüllbar", sonst

Abbildung 2.1: Erfüllbarkeitsalgorithmus für $\mathcal{ALCN}(\circ)$

2.1 Das Erfüllbarkeitsproblem

In diesem Abschnitt wird zunächst der Beweis aus [BS96], daß das Erfüllbarkeitsproblem für $\mathcal{ALCN}(\circ)$ entscheidbar ist, skizziert. Die Hilfsaussagen in Lemma 2.2 werden in Abschnitt 2.2 bei der Untersuchung des Konsistenzproblems für $\mathcal{ALCN}(\circ)$ bewiesen. Einige Aspekte werden bei der Betrachtung des Erfüllbarkeitsproblems für $\mathcal{ALC} + (\sqsubseteq_n)$ wiederverwendet.

Theorem 2.1
Das Erfüllbarkeitsproblem für $\mathcal{ALCN}(\circ)$ ist entscheidbar, d.h., für ein gegebenes $\mathcal{ALCN}(\circ)$-Konzept C ist es entscheidbar, ob eine Interpretation $\mathcal{I}$ existiert mit $C^I \neq \emptyset$. ⌟

Zum Beweis von Theorem 2.1:
Man formuliert ein regelbasiertes Verfahren, das für ein $\mathcal{ALCN}(\circ)$ -Konzept C entscheidet, ob es erfüllbar ist oder nicht. Der Algorithmus ist in Abbildung 2.1 dargestellt und Abbildung 2.2 führt die Vervollständigungsregeln ein, die im Algorithmus verwendet werden. Daß dieser Algorithmus zusammen mit diesen Vervollständigungsregeln das Gewünschte leistet, folgt aus den Aussagen in Lemma 2.2.

Zunächst noch einige Notationen:
Eine ABox $\mathcal{A}$ heißt *vollständig*, wenn $\mathcal{A}$ geschlossen ist oder keine der Vervollständigungsregeln mehr auf $\mathcal{A}$ anwendbar ist. Der durch den Algorithmus generierte Baum wird als *Vervollständigungsbaum* bezeichnet.
Man bezeichnet $\longrightarrow_\forall$, $\longrightarrow_\sqcap$, $\longrightarrow_\sqcup$ als *propagierende Regeln*, $\longrightarrow_\leq$ als *identifizierende Regel* und $\longrightarrow_\exists$, $\longrightarrow_\geq$ als *generierende Regeln.*

1. **Konjunktion:**
   Sei $x : (C_1 \sqcap C_2) \in \mathcal{A}$ und $x : C_1 \notin \mathcal{A}$ oder $x : C_2 \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\sqcap} \mathcal{A} \cup \{x : C_1, x : C_2\}$

2. **Disjunktion:**
   Sei $x : (C_1 \sqcup C_2) \in \mathcal{A}$ und $x : C_1 \notin \mathcal{A}$ und $x : C_2 \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\sqcup} \mathcal{A}_1 = \mathcal{A} \cup \{x : C_1\}$
   $\mathcal{A} \longrightarrow_{\sqcup} \mathcal{A}_2 = \mathcal{A} \cup \{x : C_2\}$

3. **Werterestriktion:**
   Sei $x : (\forall R.C) \in \mathcal{A}$ für ein $R \in \mathcal{N_R}$, $xRy \in \mathcal{A}$ und $y : C \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\forall} \mathcal{A} \cup \{y : C\}$

4. **Existenzrestriktion:**
   Sei $x : (\exists R.C) \in \mathcal{A}$ für ein $R \in \mathcal{N_R}$ und es gibt kein y mit $xRy \in \mathcal{A}$ und $y : C \in \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\exists} \mathcal{A} \cup \{xRz, z : C\}$
   für eine neue Variable $z \in \tau \setminus \tau_{\mathcal{A}}$

5. **Zahlenrestriktion:**
   Sei $x : (\geq n\ R_1 \circ \ldots \circ R_m) \in \mathcal{A}$ für Rollennamen $R_1, \ldots, R_m$ und x hat weniger als n $(R_1 \circ \ldots \circ R_m)$-Nachfolger in $\mathcal{A}$, dann
   $$\begin{aligned} \mathcal{A} \longrightarrow_{\geq} \mathcal{A} \quad & \cup \{xR_1y_1, y_1R_2y_2, \ldots, y_{m-1}R_my_m\} \\ & \cup \{y_m \neq z \mid z \in (R_1 \circ \ldots \circ R_m)(x) \text{ in } \mathcal{A}\} \end{aligned}$$
   für neue Variablen $y_1, \ldots, y_m$ aus $\tau \setminus \tau_{\mathcal{A}}$

6. **Zahlenrestriktion:**
   Sei $x : (\leq\ n\ R_1 \circ \ldots \circ R_m) \in \mathcal{A}$, x hat mehr als n $(R_1 \circ \ldots \circ R_m)$-Nachfolger in $\mathcal{A}$, und es gibt $(R_1 \circ \ldots \circ R_m)$-Nachfolger y_1, y_2 von x in $\mathcal{A}$ mit $(y_1 \neq y_2) \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\leq} \mathcal{A}_{y_1,y_2} = \mathcal{A}[y_2/y_1]$
   für alle Paare y_1, y_2 von $(R_1 \circ \ldots \circ R_m)$-Nachfolgern von x mit $(y_1 \neq y_2) \notin \mathcal{A}$

Abbildung 2.2: Vervollständigungsregeln für $\mathcal{ALCN}(\circ)$

Lemma 2.2
Sei C_0 ein $\mathcal{ALCN}(\circ)$-Konzept in NNF und $\mathcal{A}$ eine ABox, die mit den Vervollständigungsregeln aus $\{x_0 : C_0\}$ abgeleitet wurde. Dann gilt:

1. Für jede Regel $\longrightarrow$, die auf $\mathcal{A}$ anwendbar ist, gilt: $\mathcal{A}$ ist genau dann konsistent, wenn ein $\mathcal{A}'$ existiert mit $\mathcal{A} \longrightarrow \mathcal{A}'$ und $\mathcal{A}'$ ist konsistent.

2. Ist $\mathcal{A}$ eine vollständige, offene ABox, so ist $\mathcal{A}$ konsistent.

3. Ist $\mathcal{A}$ geschlossen, so ist $\mathcal{A}$ inkonsistent.

4. Der Vervollständigungsalgorithmus 2.1 terminiert stets bei Eingabe eines $\mathcal{ALCN}(\circ)$-Konzeptes C. ⌟

Mit den Aussagen des Lemmas läßt sich Theorem 2.1 leicht beweisen:

Aus Punkt (4) folgt, daß nach endlich vielen Schritten ein Baum generiert wird, dessen Blätter alle mit vollständigen ABoxen beschriftet sind. Falls C_0 erfüllbar ist, ist auch $\{x_0 : C_0\}$ konsistent und wegen Punkt (1) auch mindestens eine Blattbeschriftung. Wegen Punkt (3) ist diese ABox offen, der Algorithmus gibt „C ist erfüllbar" aus. Existiert umgekehrt eine vollständige, offene ABox $\mathcal{A}$ im Baum, so folgt mit (2), daß ein Modell $\mathcal{I}$ von $\mathcal{A}$ existiert. Wegen $x_0 : C_0 \in \mathcal{A}$ gilt $x_0^I \in C_0^I$. Also ist C_0 und damit auch C erfüllbar. □

Bemerkung 2.3 (zu Punkt (4) von Lemma 2.2)
Aus der Definition der Vervollständigungsregeln in Abbildung 2.2 ergeben sich leicht zwei Folgerungen für die ABoxen $\mathcal{A}$, die aus $\mathcal{A}_0 = \{x_0 : C_0\}$ mit diesen Regeln abgeleitet werden:

1. Jede Variable $x \in \tau_{\mathcal{A}}$, $x \neq x_0$, ist ein $(R_1 \circ \ldots \circ R_m)$-Nachfolger von x_0 für eine Rollenkette der Länge $m \geq 1$ und jede andere Rollenkette, die x_0 und x verbindet, hat ebenfalls Länge m.

2. Ist x von x_0 aus über eine Rollenkette der Länge m erreichbar, so gilt für jede Assertion $x : C$ in $\mathcal{A}$, daß C maximal Rollentiefe $m_0 - m$ hat mit $m_0 = \mathsf{depth}(C_0)$. Damit ist m beschränkt durch m_0.

Der erste Punkt besagt, daß jede Variable $x \in \tau_{\mathcal{A}}$ einen eindeutigen Level $\mathsf{level}(x)$ besitzt. Aus dem zweiten Punkt folgt, daß dieser Level stets zwischen 0 und m_0 liegt. Im Beweis von Punkt (4) aus Lemma 2.2 in [BS96] nutzt man diese Eigenschaft wie folgt aus: Man definiert eine Abbildung κ, die jede ABox $\mathcal{A}$ auf ein $5(m_0 + 1)$-Tupel über $\mathbb{N}$ abbildet und zeigt für jede Regel $\longrightarrow$ aus Abbildung 2.2, daß gilt:

$$\mathcal{A} \longrightarrow \mathcal{A}' \quad \Longrightarrow \quad \kappa(\mathcal{A}) \succ \kappa(\mathcal{A}'),$$

Konsistenzalgorithmus für $\mathcal{ALCN}(\circ)$:

Eingabe:	Eine $\mathcal{ALCN}(\circ)$-ABox $\hat{\mathcal{A}}$
Datenstruktur:	Ein mit $\mathcal{ALCN}(\circ)$-ABoxen beschrifteter Baum.
1. Schritt:	Generiere einen Baum, der nur aus einer Wurzel besteht, die mit $\mathcal{A}_0$ beschriftet ist, wobei $\mathcal{A}_0$ die verallgemeinerte ABox zu $\hat{\mathcal{A}}$ sei.
2. Schritt:	Wende solange wie möglich Regeln auf Blattbeschriftungen an, die mit offenen ABoxen beschriftet sind. Sind alle Blattbeschriftungen geschlossen oder vollständig, so terminiere.
Ausgabe:	„$\hat{\mathcal{A}}$ ist konsistent" , falls eine offene und vollständige ABox im Baum existiert, „$\hat{\mathcal{A}}$ ist inkonsistent", sonst

Abbildung 2.3: Konsistenzalgorithmus für $\mathcal{ALCN}(\circ)$

wobei $\succ$ die lexikographische Ordnung auf $\mathbb{N}^{5(m_0+1)}$ bezeichnet. Mit dieser Behauptung folgt dann aus der Annahme einer unendlichen Folge von Regelanwendungen offensichtlich ein Widerspruch zur Wohlfundiertheit der lexikographischen Ordnung $\succ$ auf $\mathbb{N}^{5(m_0+1)}$. $\diamond$

Dies schließt die Beweisskizze zu Theorem 2.1 ab. Der Beweis von Lemma 2.2 ergibt sich auch aus den Beweisen im folgenden Abschnitt 2.2.

2.2 Das Konsistenzproblem

Das Konsistenzproblem ist für $\mathcal{ALC}$ sowie verschiedene Erweiterungen von $\mathcal{ALC}$ entscheidbar [BBH94],[BHs91],[Hol94]. Zum Beweis wird jeweils ein regelbasiertes Verfahren angegeben, daß für eine gegebene ABox entscheidet, ob sie konsistent ist oder nicht. Die verschiedenen Arbeiten verwenden dabei unterschiedliche Techniken für diese Algorithmen und auch unterschiedliche Vorgehensweisen zum Beweis, daß sie das Gewünschte leisten. Eine Möglichkeit besteht z.B. darin, ausgehend von einem Algorithmus, der Erfüllbarkeit für ein Konzept entscheidet, ein Verfahren zu entwickeln, daß diesen Erfüllbarkeitsalgorithmus erweitert, um das Konsistenzproblem zu entscheiden. Im folgenden wird nun zunächst mit diesem Ansatz die Entscheidbarkeit des Konsistenzproblems für $\mathcal{ALCN}(\circ)$ untersucht.

Die prinzipielle Idee besteht darin, den Vervollständigungsalgorithmus aus Abbildung 2.3 sowie die Regeln aus Abbildung 2.2 wiederzuverwenden, wobei man statt eines $\mathcal{ALCN}(\circ)$-Konzeptes C eine $\mathcal{ALCN}(\circ)$-ABox als Eingabe erhält.

Um mit diesem Algorithmus das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ entscheiden zu können, müßte man wiederum die vier Punkte aus Lemma 2.2 zeigen. Die Punkte (1), (2) und (3) aus Lemma 2.2 lassen sich leicht übertragen.

2.2.1 Zur Korrektheit und Vollständigkeit des Algorithmus

Lemma 2.4
Sei $\mathcal{A}_0$ eine verallgemeinerte $\mathcal{ALCN}(\circ)$-ABox und $\mathcal{A}$ eine ABox, die mit den Vervollständigungsregeln aus $\mathcal{A}_0$ abgeleitet wurde. Dann gilt:

1. Für jede Regel $\longrightarrow$, die auf $\mathcal{A}$ anwendbar ist, gilt: $\mathcal{A}$ ist genau dann konsistent, wenn ein $\mathcal{A}'$ existiert mit $\mathcal{A} \longrightarrow \mathcal{A}'$ und $\mathcal{A}'$ ist konsistent.
2. Ist $\mathcal{A}$ eine vollständige, offene ABox, so ist $\mathcal{A}$ konsistent.
3. Ist $\mathcal{A}$ geschlossen, so ist $\mathcal{A}$ inkonsistent. $\lrcorner$

Beweis:
Man geht im folgenden davon aus, daß nur verallgemeinerte ABoxen betrachtet werden, die mit den Vervollständigungsregeln in Abbildung 2.2 aus $\mathcal{A}_0$ abgeleitet wurden.

Beweis zu Punkt (1) von Lemma 2.4:
Auf $\mathcal{A}$ sei eine Regel $\longrightarrow$ anwendbar. Man zeigt nun die Behauptung durch eine Fallunterscheidung über die Regeln.

- Konjunktion: $\mathcal{A} \longrightarrow_{\sqcap} \mathcal{A}'$:
  Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell zu $\mathcal{A}$.
  Umgekehrt: Sei $\mathcal{I}$ Modell von $\mathcal{A}$ und $\longrightarrow_{\sqcap}$ wurde auf $x : C_1 \sqcap C_2$ angewendet. Dann ist $\mathcal{I}$ auch Modell von $\mathcal{A}' = \mathcal{A} \cup \{x : C_1,\ x : C_2\}$, denn es gilt $x^I \in (C_1 \sqcap C_2)^I$, also $x^I \in C_1^I$ und $x^I \in C_2^I$. Damit erfüllt $\mathcal{I}$ jede Assertion aus $\mathcal{A}'$.

- Disjunktion: $\mathcal{A} \longrightarrow_{\sqcup} \mathcal{A}'$:
  Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell zu $\mathcal{A}$.
  Umgekehrt: Sei $\mathcal{I}$ Modell von $\mathcal{A}$ und $\longrightarrow_{\sqcup}$ wurde auf $x : C_1 \sqcup C_2$ angewendet. Dann ist $\mathcal{I}$ auch Modell von $\mathcal{A}_1 = \mathcal{A} \cup \{a : C_1\}$ oder $\mathcal{A}_2 = \mathcal{A} \cup \{a : C_2\}$, denn $\mathcal{I}$ erfüllt $a : C_1 \sqcup C_2$ und damit gilt $x^I \in C_1^I$ oder $x^I \in C_2^I$. Damit erfüllt $\mathcal{I}$ jede Assertion aus $\mathcal{A}_1$ oder $\mathcal{A}_2$.

- Werterestriktion: $\mathcal{A} \longrightarrow_{\forall} \mathcal{A}'$:
  Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell zu $\mathcal{A}$.
  Umgekehrt: Sei $\mathcal{I}$ Modell von $\mathcal{A}$ und $\longrightarrow_{\forall}$ wurde auf $xRy, y : \forall R.C$ angewendet. $\mathcal{I}$ ist auch Modell von $\mathcal{A}' = \mathcal{A} \cup \{y : C\}$, denn wegen $(x^I, y^I) \in R^I$ und $x^I \in (\forall R.C)^I$ gilt $y^I \in C^I$, also erfüllt $\mathcal{I}$ jede Assertion aus $\mathcal{A}'$.

- Existenzrestriktion: $\mathcal{A} \longrightarrow_{\exists} \mathcal{A}'$:
  Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell zu $\mathcal{A}$.
  Umgekehrt: Sei $\mathcal{I}$ Modell von $\mathcal{A}$ und $\longrightarrow_{\exists}$ wurde auf $x : \exists R.C$ angewendet.

Dann läßt sich $\mathcal{I}$ zu einem Modell $\mathcal{I}'$ von $\mathcal{A}' = \mathcal{A} \cup \{xRy, y : C\}$ für eine neue Variable $y \in \tau \setminus \tau_{\mathcal{A}}$ erweitern, denn:
Da $\mathcal{I}$ Modell zu $\mathcal{A}$ ist, existiert ein $d \in \Delta_I$ mit $(x^I, d) \in R^I$ und $d \in C^I$. Dann sei $\mathcal{I}'$ wie $\mathcal{I}$ definiert, wobei die Interpretationsfunktion um $y^{I'} := d$ erweitert wird. Damit ist $\mathcal{I}'$ Modell von $\mathcal{A}'$.

- Zahlenrestriktion: $\mathcal{A} \longrightarrow_{\geq} \mathcal{A}'$:

  Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell zu $\mathcal{A}$.
  Umgekehrt: Sei $\mathcal{I}$ Modell von $\mathcal{A}$ und $\longrightarrow_{\geq}$ wurde auf $x : (\geq n\, R_1 \circ \ldots \circ R_m) \in \mathcal{A}$ angewendet und es sei

  $$\begin{aligned} \mathcal{A}' = \mathcal{A} \; & \cup \; \{xR_1y_1, y_1R_2y_2, \cdot\mathcal{A}, y_{m-1}R_my_m\} \\ & \cup \; \{y_m \neq z \mid z \in (R_1 \circ \ldots \circ R_m)(x) \text{ in } \mathcal{A}\} \end{aligned}$$

  für neue Variablen $y_1, \ldots, y_m \in \tau \setminus \tau_{\mathcal{A}}$. Dann läßt sich $\mathcal{I}$ zu einem Modell $\mathcal{I}'$ von $\mathcal{A}'$ erweitern, denn:
  Da $\mathcal{I}$ Modell von $\mathcal{A}$ ist, hat x^I in Δ_I mindestens n verschiedene $(R_1 \circ \ldots \circ R_m)$-Nachfolger. Da aber $\longrightarrow_{\geq}$ auf $x : (\geq n\, R_1 \circ \ldots \circ R_m)$ anwendbar ist, existieren weniger als n paarweise unterschiedene $(R_1 \circ \ldots \circ R_m)$-Nachfolger von x in $\mathcal{A}$. Also existiert ein $d_m \in \Delta_I$ mit $y^I \neq d_m$ für alle $(R_1 \circ \ldots \circ R_m)$-Nachfolger y von x in $\mathcal{A}$. Außerdem existieren $d_1, \cdots, d_{m-1} \in \Delta_I$ mit $(x^I, d_1) \in R_1^I, (d_1, d_2) \in R_2^I, \ldots, (d_{m-1}, d_m) \in R_m$. Dann sei $\mathcal{I}'$ wie $\mathcal{I}$ definiert, wobei die Interpretationsfunktion um $y_i^{I'} := d_i$ für $1 \leq i \leq m$ erweitert wird.

- Zahlenrestriktion: $\mathcal{A} \longrightarrow_{\leq} \mathcal{A}'$:

  $\longrightarrow_{\leq}$ sei auf $x : (\leq n\, R_1 \circ \ldots \circ R_m) \in \mathcal{A}$ anwendbar, d.h., es existieren $l > n$ $(R_1 \circ \ldots \circ R_m)$-Nachfolger $y_1, \ldots, y_l$ von x in $\mathcal{A}$ und es sei $\mathcal{A}_{ij} = \mathcal{A}[y_i/y_j]$. Zu einem Modell $\mathcal{I}$ von $\mathcal{A}_{ij}$ definiert man ein Modell $\mathcal{I}'$ von $\mathcal{A}$, indem man y_i wie y_j interpretiert, d.h., $y_i^{I'} := y_j^I$. Dann ist $\mathcal{I}'$ offensichtlich ein Modell von $\mathcal{A}$.
  Sei umgekehrt $\mathcal{I}$ ein Modell von $\mathcal{A}$, so ist $\mathcal{I}$ auch Modell für ein $\mathcal{A}_{ij}$, denn: Da $\longrightarrow_{\leq}$ auf $x : (\leq n\, R_1 \circ \ldots \circ R_m)$ anwendbar ist, existieren mehr als n $(R_1 \circ \ldots \circ R_m)$-Nachfolger von x in $\mathcal{A}$. Darunter befinden sich mindestens zwei Elemente $y_1, y_2 \in \tau_{\mathcal{A}}$ mit $y_1 \neq y_2 \notin \mathcal{A}$ und $y_1^I = y_2^I$, sonst wäre $\mathcal{I}$ kein Modell von $\mathcal{A}$. Dann ist $\mathcal{A}_{12}$ aus $\mathcal{A}$ mit $\longrightarrow_{\leq}$ ableitbar und $\mathcal{I}$ ist Modell von $\mathcal{A}_{12}$. $\square$

Beweis zu Punkt (2) von Lemma 2.4:
Man definiert zu einer offenen und vollständigen ABox $\mathcal{A}$ die kanonische Interpretation $\mathcal{I}_{\mathcal{A}}$ wie folgt:

- $\Delta_{\mathcal{I}_{\mathcal{A}}}$ ist die Menge $\tau_{\mathcal{A}}$ der in $\mathcal{A}$ vorkommenden Individuen und Variablen, $x^{\mathcal{I}_{\mathcal{A}}} = x$ für alle $x \in \tau_{\mathcal{A}}$,
- $R^{\mathcal{I}_{\mathcal{A}}} := \{(x, y) \mid xRy \in \mathcal{A}\}$
- $A^{\mathcal{I}_{\mathcal{A}}} := \{x \mid x : A \in \mathcal{A}\}$

Behauptung: $\mathcal{I}_\mathcal{A}$ ist Modell von $\mathcal{A}$.
Zum Beweis der Behauptung ist zu zeigen, daß $\mathcal{I}_\mathcal{A}$ jede Assertion aus $\mathcal{A}$ erfüllt.

- Rollenassertionen: $xRy \in \mathcal{A}$
  Per def. gilt $(x, y) \in R^{\mathcal{I}_\mathcal{A}}$.
- Ungleichheitsassertionen: $x \neq y$
  Da $\mathcal{A}$ offen ist, sind x und y verschieden, also gilt $x \neq y$.
- Konzeptassertionen: $x : C$
  Hier zeigt man durch Induktion über den Aufbau von Konzepttermen, daß $x^{\mathcal{I}_\mathcal{A}} \in C^{\mathcal{I}_\mathcal{A}}$ gilt.
  - $x : A \in \mathcal{A}$ für einen Konzeptnamen $A \in \mathcal{N}_\mathcal{C}$
    Per def. gilt $x \in A^{\mathcal{I}_\mathcal{A}}$.
  - $x : \neg A \in \mathcal{A}$ für einen Konzeptnamen $A \in \mathcal{N}_\mathcal{C}$
    Da $\mathcal{A}$ offen ist, gilt $x : A \notin \mathcal{A}$. Per def. ist dann $x \notin A^{\mathcal{I}_\mathcal{A}}$, d.h., $x \in (\neg A)^{\mathcal{I}_\mathcal{A}}$.
    Beachte: Nach Voraussetzung ist $\mathcal{A}_0$ eine verallgemeinerte ABox und $\mathcal{A}$ aus $\mathcal{A}_0$ mit den Vervollständigungsregeln abgeleitet. Daher enthalten alle Konzeptassertionen $x : C$ aus $\mathcal{A}$ nur Konzeptterme C in NNF, d.h., Negation tritt nur unmittelbar vor Konzeptnamen auf.
  - $x : C_1 \sqcap C_2 \in \mathcal{A}$
    Da $\mathcal{A}$ vollständig ist, ist die $\sqcap$-Regel nicht auf $\mathcal{A}$ anwendbar, d.h., es gilt $\{x : C_1, x : C_2\} \subseteq \mathcal{A}$. Nach Induktionsvoraussetzung gilt dann $x \in C_1^{\mathcal{I}_\mathcal{A}}$ und $x \in C_2^{\mathcal{I}_\mathcal{A}}$, also $x \in (C_1 \sqcap C_2)^{\mathcal{I}_\mathcal{A}}$.
  - $x : C_1 \sqcup C_2 \in \mathcal{A}$
    Analog zur Konjunktion folgt $x \in (C_1 \sqcup C_2)^{\mathcal{I}_\mathcal{A}}$.
  - $x : \exists R.C \in \mathcal{A}$
    Da $\mathcal{A}$ vollständig ist, existiert ein $y \in \tau_\mathcal{A}$ mit $\{xRy, y : C\} \subseteq \mathcal{A}$. Aus $xRy \in \mathcal{A}$ folgt $(x, y) \in R^{\mathcal{I}_\mathcal{A}}$ und aus $y : C \in \mathcal{A}$ folgt mit der Induktionsvoraussetzung $y \in C^{\mathcal{I}_\mathcal{A}}$, also $x \in (\exists R.C)^{\mathcal{I}_\mathcal{A}}$.
  - $x : \forall R.C \in \mathcal{A}$
    Sei $(x, y) \in R^{\mathcal{I}_\mathcal{A}}$ beliebig. Nach Definition von $R^{\mathcal{I}_\mathcal{A}}$ folgt $xRy \in \mathcal{A}$. Da $\mathcal{A}$ vollständig ist, ist die $\forall$-Regel nicht auf $\{x : \forall R.C, xRy\} \subseteq \mathcal{A}$ anwendbar, d.h., es ist $y : C \in \mathcal{A}$. Aus $y : C \in \mathcal{A}$ folgt mit der Induktionsvoraussetzung $y \in C^{\mathcal{I}_\mathcal{A}}$. Da y mit $(x, y) \in R^{\mathcal{I}_\mathcal{A}}$ beliebig gewählt war, folgt also $x \in (\forall R.C)^{\mathcal{I}_\mathcal{A}}$.
  - $x : (\geq n\, R_1 \circ \ldots \circ R_m) \in \mathcal{A}$
    Da $\mathcal{A}$ vollständig ist, existieren mindestens n verschiedene $R_1 \circ \ldots \circ R_m$-Nachfolger von x in $\mathcal{A}$ mit $\{y_i \neq y_j \mid 1 \leq i < j \leq n\} \subseteq \mathcal{A}$. Damit sind $(x, y_i) \in (R_1 \circ \ldots \circ R_m)^{\mathcal{I}_\mathcal{A}}$ für $1 \leq i \leq n$, also
    $x \in (\geq n\, R_1 \circ \ldots \circ R_m)^{\mathcal{I}_\mathcal{A}}$.

- $x : (\geq n\, R_1 \circ \ldots \circ R_m) \in \mathcal{A}$
  Angenommen, es gibt $n+1$ verschiedene $(R_1 \circ \ldots \circ R_m)$-Nachfolger $y_1, \ldots,$ y_{n+1} von x in $\Delta_{\mathcal{I}_\mathcal{A}}$. Dann sind die y_i auch $(R_1 \circ \ldots \circ R_m)$-Nachfolger von x in $\mathcal{A}$. Da $\mathcal{A}$ vollständig ist, ist die $\leq$-Regel nicht anwendbar, d.h., $\mathcal{A}$ enthält die Assertionen $y_i \neq y_j$ für $1 \leq i < j \leq n+1$. Damit enthält $\mathcal{A}$ aber einen clash, im Widerspruch zur Voraussetzung, daß $\mathcal{A}$ offen ist. Also hat x höchstens n $(R_1 \circ \ldots \circ R_m)$-Nachfolger, und es folgt $x \in (\leq n\, R_1 \circ \ldots \circ R_m)^{\mathcal{I}_\mathcal{A}}$.
  □

Beweis zu Punkt (3) von Lemma 2.4:
Dieser Punkt ist trivial: enthält eine ABox $\mathcal{A}$ einen clash, so kann es offensichtlich keine Interpretation geben, die jede Assertion aus $\mathcal{A}$ erfüllt. □

Bemerkung 2.5
Im Falle des Erfüllbarkeitsalgorithmus aus Abbildung 2.1 kann man im Beweis von Lemma 2.2 offensichtlich von speziellen ABoxen $\mathcal{A}_0$ ausgehen. Diese sind stets von der Form $\{x_0 : C_0\}$ für ein $\mathcal{ALCN}(\circ)$-Konzept C_0 und eine Variable x_0. Also liefert der Beweis von Lemma 2.4 auch einen Beweis der Punkte (1), (2) und (3) von Lemma 2.2. ◇

Es fehlt noch der Nachweis der Terminierung für den Algorithmus aus Abbildung 2.3 bei Eingabe einer beliebigen $\mathcal{ALCN}(\circ)$-ABox $\hat{\mathcal{A}}$.

2.2.2 Zur Terminierung des Algorithmus

Mit Bemerkung 2.3 folgt, daß jede ABox $\mathcal{A}$, die mit den Regeln in Abbildung 2.2 aus einer ABox $\mathcal{A}_0 = \{x_0 : C_0\}$ abgeleitet wurde, eine Levelstruktur besitzt, wobei x_0 die Wurzel bildet und jedes $x \in \tau_\mathcal{A}$ einen eindeutigen Level $\mathsf{level}(x)$ besitzt.

Im Falle des Konsistenzalgorithmus besitzen die abgeleiteten ABoxen im allgemeinen keine Levelstruktur mehr, da in einer $\mathcal{ALCN}(\circ)$-ABox mehrere Individuen auftreten können, so daß keine eindeutige Wurzel existiert. Außerdem kann es zwischen zwei Individuen Pfade unterschiedlicher Länge geben. Folglich läßt sich der in Bemerkung 2.3 skizzierte Terminierungsbeweis aus [BS96] nicht auf den Konsistenzalgorithmus übertragen.

Das folgende Beispiel zeigt sogar, daß Terminierung bei einer ungesteuerten Anwendung der Vervollständigungsregeln auf $\mathcal{ALCN}$- und damit auch $\mathcal{ALCN}(\circ)$-ABoxen nicht gegeben ist.

Beispiel 2.6
Gegeben sei folgende verallgemeinerte $\mathcal{ALCN}$-ABox

$$\begin{aligned}\mathcal{A}_0 = \{ \quad & a_0 R a_0, \\ & a_0 : (\leq 1\, R), \\ & a_0 : \exists R.A, \\ & a_0 : \forall R.((\leq 1\, R) \sqcap \exists R.A) \quad \}.\end{aligned}$$

Ausgehend von $\mathcal{A}_0$ sind folgende Regelanwendungen möglich:

$$\begin{array}{lll}\mathcal{A}_0 & \longrightarrow_{\exists} & \mathcal{A}_1 := \mathcal{A}_0 \cup \{a_0Rx_1, x_1 : A\} \\ & \longrightarrow_{\forall} & \mathcal{A}_2 := \mathcal{A}_1 \cup \{x_1 : (\leq 1\, R) \sqcap \exists R.A)\} \\ & \longrightarrow_{\sqcap} & \mathcal{A}_3 := \mathcal{A}_2 \cup \{x_1 : (\leq 1\, R), x_1 : \exists R.A\} \\ & \longrightarrow_{\exists} & \mathcal{A}_4 := \mathcal{A}_3 \cup \{x_1Rx_2, x_2 : A\} \\ & \longrightarrow_{\leq} & \mathcal{A}_5 := \mathcal{A}_4[x_1/a_0]\end{array}$$

Die so erhaltene ABox $\mathcal{A}_5$ beschreibt die gleiche Situation wie die ABox $\mathcal{A}_1$, wobei nun die Variable x_2 statt x_1 verwendet wird. Offensichtlich können auf $\mathcal{A}_5$ die Regeln wie auf $\mathcal{A}_1$ angewendet werden, d.h., eine unendliche Folge von Regelanwendungen ist möglich. Damit ist Terminierung nicht gewährleistet. Abbildung 2.4 veranschaulicht den sogenannten *Jojo-Effekt*, der bei der skizzierten Regelanwendung auftritt. $\diamond$

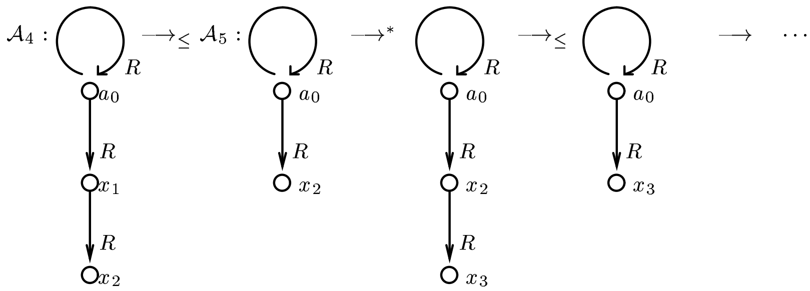


Abbildung 2.4: Der Jojo-Effekt.

Bemerkung 2.7
Bei der in Beispiel 2.6 angegebenen ABox $\mathcal{A}_0$ handelt es sich um eine verallgemeinerte $\mathcal{ALCN}$-ABox, d.h., weder für $\mathcal{ALCN}$ noch für $\mathcal{ALCN}(\circ)$ ist Terminierung bei ungesteuerter Regelanwendung gewährleistet. Die Ursachen für dieses Problem liegen zum einen in dem Rollenzykel a_0Ra_0, zum anderen in der vorschnellen Anwendung der $\exists$-Regel. Der Jojo-Effekt entsteht erst dadurch, daß durch die Anwendung der $\exists$-Regel ein Individuum zunächst 'angehangen' (also erzeugt) und dann durch anwenden der $\leq$-Regel über den Rollenzykel wieder 'hochgezogen' (also identifiziert) wird. $\diamond$

Um mit Hilfe von Algorithmus 2.3 dennoch das Konsistenzproblem entscheiden zu können, muß man also geeignete Erweiterungen oder Modifikationen vornehmen. Eine Möglichkeit besteht darin, eine *Strategie* zu entwickeln, die die Regelanwendung so beeinflußt, daß eine unendliche Folge von Regelanwendungen nicht mehr möglich ist.

Zur Formulierung und Analyse möglicher Strategien benötigt man noch die folgenden Notationen:

Strategie für $\mathcal{ALCN}$-ABoxen:

- Ersetze bei Anwendung der $\leq$-Regel stets Variablen durch Individuen. (Das Identifizieren zweier Variablen bleibt unbeeinflußt.)
- Unterteile die Regelanwendung in zwei Phasen:
  1. Phase: Wende Regeln nur auf Individuen an, d.h., auf Assertionen $a : C$, wo a Individuum ist.
  2. Phase: Wende Regeln beliebig an.

Abbildung 2.5: Eine Strategie für $\mathcal{ALCN}$

1. $\hat{\mathcal{A}}$ bezeichnet die $\mathcal{ALCN}(\circ)$-ABox, für die Konsistenz zu entscheiden ist. $\mathcal{A}_0$ sei die zu $\hat{\mathcal{A}}$ äquivalente, verallgemeinerte ABox, die im 1. Schritt des Konsistenzalgorithmus bestimmt wird.

2. Man unterscheidet im folgenden zwischen den in $\mathcal{A}_0$ enthaltenen *Individuen* und den durch die generierenden Regeln $\longrightarrow_{\geq}$ und $\longrightarrow_{\exists}$ erzeugten *Variablen.* Spricht man von einem *Element* aus $\mathcal{A}$, so meint man damit ein beliebiges Element aus $\tau_{\mathcal{A}}$.

3. $\mathsf{Sub}(\mathcal{A}_0)$ bezeichnet die Menge aller Unterkonzepte von Konzepttermen aus $\mathcal{A}_0$:
$$\mathsf{Sub}(\mathcal{A}_0) = \{C \mid \text{ex. } a_0 : C_0 \in \mathcal{A}_0 \text{ und } C \in \mathsf{Sub}(C_0)\}.$$

4. $\mathsf{maxdepth}$ bezeichnet die maximale Rollentiefe aller Konzepte aus $\mathcal{A}_0$,
$$\mathsf{maxdepth} = \mathsf{max}\{\mathsf{depth}(C) \mid \text{ex. } a \in \tau_{\mathcal{A}_0} \text{ mit } a : C \in \mathcal{A}_0\}$$

5. Bei den weiteren Überlegungen setzt man $\mathsf{maxdepth} > 0$ voraus, denn der Fall $\mathsf{maxdepth} = 0$ ist trivial. Eine ABox $\mathcal{A}_0$ mit $\mathsf{maxdepth} = 0$ enthält nur Konzeptterme $C \in \mathsf{Sub}(\mathcal{A}_0)$, die nur Boolesche Operatoren enthalten. Also ist $\mathcal{A}_0$ eine $\mathcal{ALC}$-ABox. Für $\mathcal{ALC}$ ist das Konsistenzproblem entscheidbar.

2.2.3 Eine Lösung für $\mathcal{ALCN}$

Abbildung 2.5 enthält eine Strategie, mit der die in Bemerkung 2.7 genannten Ursachen für die Nichtterminierung bei Eingabe einer $\mathcal{ALCN}$-ABox beseitigt werden. Dies ist ein erster Schritt in Richtung eines terminierenden $\mathcal{ALCN}(\circ)$-Algorithmus.

Satz 2.8
Der Vervollständigungsalgorithmus in Abbildung 2.3 terminiert bei Eingabe einer $\mathcal{ALCN}$-ABox $\mathcal{A}$, falls die Strategie in Abbildung 2.5 eingehalten wird. ⌟

Beweis[1]**:**
Die Behauptung des Satzes folgt leicht aus den Aussagen der folgenden Lemmata. Diese beziehen sich auf die Begriffe aus dem Algorithmus in Abbildung 2.3 und der Strategie in Abbildung 2.5. □

Lemma 2.9
Phase 1 terminiert. ⌟

Beweis:
Die Behauptung folgt leicht aus den folgenden Beobachtungen:

1. Die Anzahl der Individuen bleibt gleich oder nimmt ab, d.h., sie nimmt insbesondere nicht zu.

2. Auf jedes Individuum a können nur endlich oft Regeln angewendet werden, denn:

   - Für alle Assertionen $x : C$ in den abgeleiteten ABoxen gilt, daß $C \in \mathsf{Sub}(\mathcal{A}_0)$ ist. Je Individuum kann es also nur endlich viele solcher Assertionen geben.
   - Für ein Individuum a kann die $\sqcap-, \sqcup-, \exists-, \geq -$ Regel nur endlich oft angewendet werden. Damit ist die Zahl der mit $\longrightarrow_{\geq}$ und $\longrightarrow_{\exists}$ zu a erzeugten Nachfolger beschränkt.
   - Dadurch ist die Anzahl der $\longrightarrow_{\leq}$- und $\longrightarrow_{\forall}$-Anwendungen auf ein Individuum a ebenfalls beschränkt. □

Lemma 2.10
In Phase 2 werden keine Regeln mehr auf Individuen angewendet. ⌟

Beweis:
Auf Grund der Unterscheidung zwischen Individuen und Variablen und da gemäß Strategie stets Variablen durch Individuen ersetzt werden, sind Individuen nie Nachfolger von Variablen. Damit kann durch Regelanwendung auf Variablen keine neue Assertion für Individuen entstehen, sodaß nach Phase 1 keine Regelanwendung auf Individuen mehr nötig wird. □

Lemma 2.11
Phase 2 terminiert. ⌟

Beweis:
In einer aus $\mathcal{A}_0$ mit der Strategie abgeleiteten ABox $\mathcal{A}$ gilt für eine Variable x stets:

[1]Der Beweis stammt aus einer Mitschrift der Vorlesung 'Logische Methoden in der Wissensrepräsentation', die im SS 96 von Prof. Baader an der RWTH Aachen gelesen wurde.

1. x hat genau einen Rollenvorgänger, d.h., es gibt genau ein $R \in \mathcal{N}_\mathcal{R}$ und genau ein $x' \in \tau_\mathcal{A}$ mit $x'Rx \in \mathcal{A}$.

   Denn: x entsteht durch Anwenden der $\exists$- oder $\geq$-Regel auf ein Individuum oder eine Variable x'. Durch das Identifizieren von Elementen entstehen keine weiteren, direkten Vorfahren von x, da immer nur direkte Nachfolger eines Elements miteinander identifiziert werden.

2. Gilt $x'Rx \in \mathcal{A}$, so ist

$$\mathsf{max}\{|C| \mid x : C \in \mathcal{A}\} > \mathsf{max}\{|D| \mid x' : D \in \mathcal{A}\}$$

   Denn: Assertionen zu x ergeben sich stets aus größeren Assertionen zu direkten Vorgängern von x. Beim Identifizieren bleibt die Aussage offensichtlich erhalten.

Es folgt: Betrachtet man alle von einer Variablen ausgehenden Rollenpfade in einer abgeleiteten ABox $\mathcal{A}$, so erhält man wegen 1. einen Baum. Dieser ist endlich verzweigt, da die $\exists$- und $\geq$-Regel auf jedes Element nur endlich oft angewendet wird. Wegen 2. hat der Baum nur endliche Tiefe. Damit können in Phase 2 nur endlich viele Variablen entstehen und auf jede dieser Variablen kann nur endlich oft eine Regel angewendet werden. Also terminiert Phase 2. □

Dies schließt den Beweis zu Satz 2.8 ab. Zusammen mit Lemma 2.4 folgt aus diesem Satz, daß der Algorithmus 2.3 zusammen mit der Strategie in Abbildung 2.5 das Konsistenzproblem für $\mathcal{ALCN}$-ABoxen entscheidet.

Theorem 2.12
Das Konsistenzproblem für $\mathcal{ALCN}$ ist entscheidbar. ⌟

Bemerkung 2.13
In [Hol94] wird ein Verfahren angegeben, das das Konsistenzproblem für eine Erweiterung von $\mathcal{ALCN}$ um qualifizierende Zahlenrestriktionen entscheidet. Mit den Ergebnissen in [Hol94] erhält man also einen alternativen Beweis zu Theorem 2.12. Ein Vorteil des Beweises aus [Hol94] ist, daß er sogar ein PSPACE-Resultat liefert. ◇

Wesentlich für den soeben geführten Terminierungsbeweis ist die Gültigkeit der folgenden Punkte bei Einhaltung der Strategie aus Abbildung 2.5:

1. Ein Individuum ist nie Nachfolger einer Variablen.
2. Nach Phase 1 wird keine Regel mehr auf Individuen angewendet.
3. Die von einer Variablen ausgehenden Rollenpfade bilden einen Baum.

Bei der Untersuchung von $\mathcal{ALCN}(\circ)$-ABoxen zeigt sich leider sehr schnell, daß zum einen eine Einhaltung der Strategie aus Abbildung 2.5 bei der Regelanwendung keine vollständigen ABoxen liefert. Zum anderen gelten die Punkte 1. und 3. im allgemeinen nicht mehr.

Beispiel 2.14
Es seien $\mathcal{N}_{\mathcal{C}} = \{A, B\}$ und $\mathcal{N}_{\mathcal{R}} = \{R, S\}$ gegeben sowie

$$\begin{aligned} \mathcal{A}_0 = \{ \quad & a_0 R a_1, \\ & a_1 R a_2, \\ & a_2 R a_3, \\ & a_0 : (\leq 2\, R \circ R), \\ & a_0 : \exists R.\exists R.\forall R.(A \sqcap B), \\ & a_1 : \exists S.(((\exists S.\exists S.A) \sqcap (\exists S.\exists S.B)) \sqcap (\leq 1\, S \circ S)) \quad \}. \end{aligned}$$

Abbildung 2.6 zeigt eine ABox, die durch Regelanwendungen aus $\mathcal{A}$ abgeleitet werden kann. Dabei werden sowohl auf Individuen als auch Variablen Regeln angewendet.

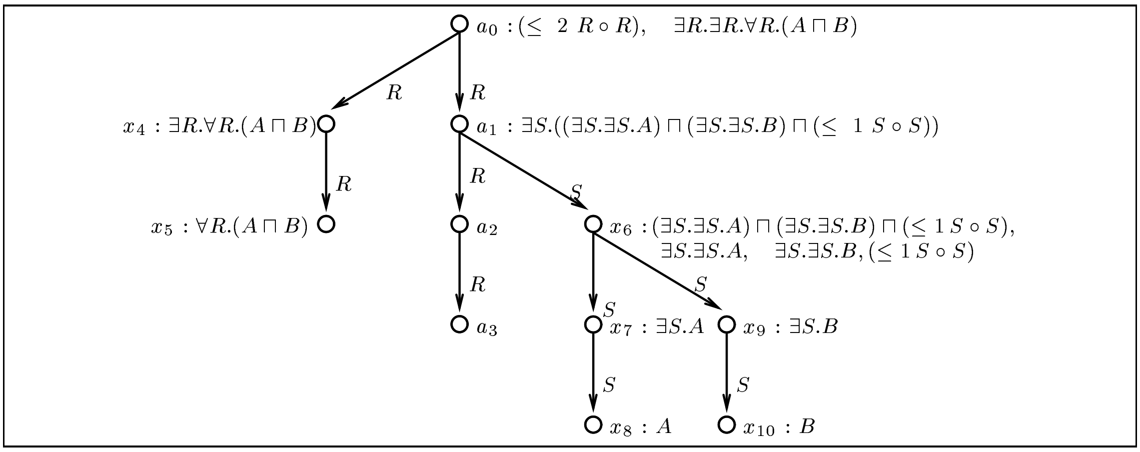


Abbildung 2.6: Die abgeleitete $\mathcal{ALCN}(\circ)$-ABox.

Hält man die Strategie aus Abbildung 2.5 ein, so kann die $\leq$-Regel nicht mehr auf $a_0 : (\leq 1\, R \circ R)$ und x_5, a_2 angewendet, nachdem die $\exists$-Regel auf x_4 angewendet wurde. Also ist die erhaltene ABox nicht vollständig. Vervollständigt man aber die ABox aus Abbildung 2.6 durch Anwendungen der $\leq$-, $\forall$- und $\sqcap$-Regel, so bilden die Rollenassertionen der erhaltenen ABox den in Abbildung 2.7 dargestellten gerichteten Graphen.

In der abgeleiteten ABox aus Abbildung 2.7 sind nun offensichtlich die Punkte 1. und 3. von Seite 31 verletzt:

1. Das Individuum a_2 hat als R-Vorfahren die Variable x_4.

2. Die von der Variablen x_6 ausgehenden Rollenpfade bilden keinen Baum mehr.

$\diamond$

Das Beispiel zeigt also, daß sich der Terminierungsbeweis für den Konsistenzalgorithmus für $\mathcal{ALCN}$-ABoxen nicht auf den Fall $\mathcal{ALCN}(\circ)$ übertragen läßt.

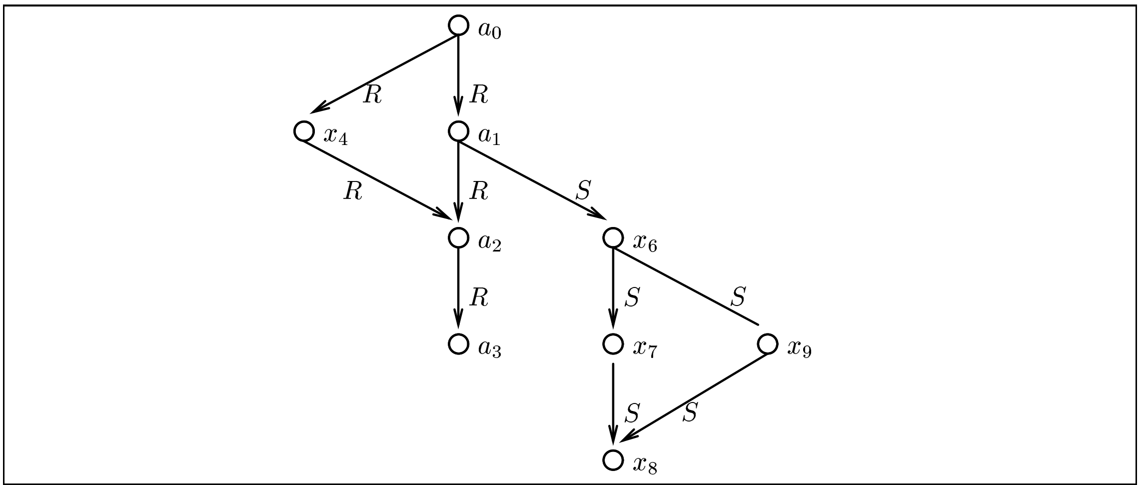


Abbildung 2.7: Das Ergebnis der Ableitung.

2.2.4 Probleme bei der Lösungssuche für $\mathcal{ALCN}(\circ)$

Um dennoch eine Strategie zu finden, die die Terminierung von Algorithmus 2.3 bei Eingabe beliebiger $\mathcal{ALCN}(\circ)$-ABoxen gewährleistet, wird zunächst untersucht, unter welchen Voraussetzungen der Algorithmus nicht terminiert. Im folgenden wird vorausgesetzt, daß bei der Anwendung der $\leq$-Regel die Ersetzungsstrategie aus Abbildung 2.5 eingehalten wird, d.h., sind ein Individuum und eine Variable zu identifizieren, so ersetzt man stets die Variable durch das Individuum.

Die folgende Definition liefert ein Analogon zu der in Bemerkung 2.3 eingeführten Levelfunktion für $\mathcal{ALCN}(\circ)$-ABoxen.

Definition 2.15
Sei $\mathcal{A}_0$ eine beliebige, verallgemeinerte $\mathcal{ALCN}(\circ)$-ABox, $\mathcal{A}$ sei eine mit den Regeln aus Abbildung 2.2 aus $\mathcal{A}_0$ abgeleitete ABox und es seien $x, y \in \tau_{\mathcal{A}}$.

Unter einer *Rollenkette der Länge n von x nach y in $\mathcal{A}$* versteht man eine Menge von Rollenassertionen der Form

$$\{x_0 R_1 x_1, \cdots, x_{n-1} R_n x_n\} \text{ mit } x_0 = x \text{ und } x_n = y.$$

Eine Rollenkette heißt *zyklenfrei*, wenn alle Elemente $x_0, \cdots, x_n$ paarweise verschieden sind.

Für eine Variable x definiert man den *minimalen Abstand von x zu Individuen in $\mathcal{A}$* als

$$\triangle(x) := \mathsf{min}\{|\mathcal{R}| \mid \text{ex. } a \in \tau_{\mathcal{A}_0} \text{ mit } x \in \mathcal{R}(a) \text{ in } \mathcal{A}\}.$$

Für ein Individuum $a \in \tau_{\mathcal{A}_0}$ definiert man $\triangle(a) := 0$. ◇

Bei den $\mathcal{ALCN}(\circ)$-ABoxen, die aus $\{x_0 : C_0\}$ abgeleitet werden, kann der maximal erreichbare Level beschränkt werden durch die Rollentiefe $\mathsf{depth}(C_0)$. Entsprechendes gilt für den minimalen Abstand in $\mathcal{ALCN}(\circ)$-ABoxen, die aus einer verallgemeinerten ABox $\mathcal{A}_0$ abgeleitet werden.

Lemma 2.16
Für eine aus $\mathcal{A}_0$ abgeleitete ABox $\mathcal{A}$ gilt: Zu jeder Variablen x aus $\mathcal{A}$ existiert ein Individuum a, so daß eine Rollenkette der Länge $n \leq \mathsf{maxdepth}$ von a nach x existiert, d.h., es existiert keine Variable x mit $\triangle(x) > \mathsf{maxdepth}$. ⌟

Beweis:
Für eine Variable $x \in \tau_{\mathcal{A}}$ existiert mindestens ein Individuum als Vorfahre (keine Variable ensteht aus dem Nichts) und damit auch ein Individuum a mit einer Rollenkette der Länge $\triangle(x)$ von a nach x. Aus der Definition der Vervollständigungsregeln folgt:

$$\mathsf{depth}(C) \leq \mathsf{maxdepth} - \triangle(x) \text{ für jede Assertion } x : C \in \mathcal{A}, \qquad (*)$$

denn: Für ein Individuum a und eine Assertion $a : C$ gilt $\mathsf{depth}(C) \leq \mathsf{maxdepth}$. Entsteht eine Assertion $x : C'$ durch m-maliges anwenden der $\exists$- und $\forall$-Regel auf Assertionen $x_i : C_i$ mit $C_i \in \mathsf{Sub}(C)$, so gilt offensichtlich $\mathsf{depth}(C') \leq \mathsf{maxdepth} - m$. Für ein Element x mit $\triangle(x) = m$ entsteht eine Assertion $x : C'$, wenn die $\exists$- oder $\forall$-Regel ausgehend von einem $a : C$, $a \in \tau_{\mathcal{A}_0}$, mindestens m-mal auf Assertionen $x_i : C_i$ mit $C_i \in \mathsf{Sub}(C)$ angewendet wird. Also gilt $(*)$.
Angenommen, es gibt ein x' mit $\triangle(x') > \mathsf{maxdepth}$. Man sieht leicht, daß dann auch ein Vorfahre x'' von x' existiert, so daß es einen Vorfahren x von x'' gibt mit

- $\triangle(x) \leq \mathsf{maxdepth}$ und
- x'' wird durch anwenden einer generierenden Regel auf eine Assertion $x : C$ erzeugt.

Daraus folgt aber mit $(*)$ ein Widerspruch der Form

$$\mathsf{maxdepth} < \triangle(x'') \leq \triangle(x) + \mathsf{depth}(C) \leq \triangle(x) + \mathsf{maxdepth} - \triangle(x) = \mathsf{maxdepth}.$$

Also kann keine Variable x' mit $\triangle(x') > \mathsf{maxdepth}$ erzeugt werden. □

Mit Hilfe der Aussagen im folgenden Lemma lassen sich nun die Situationen näher charakterisieren, in denen eine unendliche Folge von Regelanwendungen möglich ist.

Lemma 2.17
Sei $\mathcal{A}_0$ eine verallgemeinerte ABox und $\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \cdots$ eine Folge von Regelanwendungen. Dann gilt:

1. Enthält die Folge nur endlich oft eine Anwendung der $\leq$-Regel, so bricht sie nach endlich vielen Schritten ab.
2. Enthält die Folge nur endlich oft Anwendungen der generierenden Regeln, so bricht sie nach endlich vielen Schritten ab. ⌟

Beweis:
Zu Punkt (1) **von Lemma 2.17:** Da die $\leq$-Regel nur endlich oft angewendet wird, existiert ein minimaler Index $j_0 \geq 0$ so, daß in der Folge $\mathcal{A}_{j_0} \longrightarrow \mathcal{A}_{j_0+1} \longrightarrow \cdots$ die $\leq$-Regel nicht mehr auftritt. Also gilt

$$\mathcal{A}_j \subset \mathcal{A}_{j+1} \text{ und } \tau_{\mathcal{A}_j} \subseteq \tau_{\mathcal{A}_{j+1}} \text{ für alle } j \geq j_0$$

Sei $x \in \tau_{\mathcal{A}_{j_0}}$ ein Element, auf das noch Regeln angewendet werden. Durch Anwendung der $\sqcap$- bzw. $\sqcup$-Regel wird die Zahl der Assertionen zu x echt größer. Bei Anwendung der $\forall$-Regel auf $x : \forall R.C$ und xRy mit y aus $\tau_{\mathcal{A}_{j_0}}$ wird die Zahl der Assertionen zu y echt größer. Da die $\leq$-Regel ab Index j_0 nicht mehr angewendet wird, bilden die von x aus ab Index j_0 erzeugten Variablen einen Baum mit Wurzel x. Zu jeder ab Index j_0 erzeugten Variablen y existiert also genau ein Element $x \in \tau_{\mathcal{A}_{j_0}}$ mit einer Rollenkette minimaler Länge zu y und genau ein Rollenvorgänger y', d.h., genau eine Rollenassertion der Form $y'Ry$. Diese Bäume von Variablen sind offensichtlich endlich verzweigt und in der Tiefe durch $\mathsf{maxdepth}$ beschränkt. Also erzeugt man nur noch endlich viele Variablen und wendet auch die propagierenden Regeln nur noch endlich oft an.

Zu Punkt (2) **von Lemma 2.17:** Da nur endlich oft generierende Regeln angewendet werden, existiert ein minimaler Index $j_0 \geq 0$ so, daß in der Folge $\mathcal{A}_{j_0} \longrightarrow \mathcal{A}_{j_0+1} \longrightarrow \cdots$ keine generierende Regel mehr auftritt. $\tau_{\mathcal{A}_{j_0}}$ ist endlich und ab Index j_0 werden keine weiteren Variablen mehr erzeugt. Auf jedes der endlich vielen Elemente kann eine propagierende Regel nur endlich oft angewendet werden. Da bei jeder Anwendung von $\longrightarrow_{\leq}$ die Anzahl von Elementen aus $\tau_{\mathcal{A}_{j_0}}$ echt kleiner wird, kann die $\leq$-Regel nur endlich oft angewendet werden. Folglich bricht die Folge $\mathcal{A}_{j_0} \longrightarrow \mathcal{A}_{j_0+1} \longrightarrow \cdots$ nach endlich vielen Schritten mit einer vollständigen ABox ab. $\square$

Mit Lemma 2.17 folgt also, daß ausgehend von einer verallgemeinerten $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$ jede Folge von Regelanwendungen, in der die $\leq$-Regel nur endlich oft auftritt, nach endlich vielen Schritten mit einer vollständigen ABox abbricht. Ebenso bricht jede Folge von Regelanwendungen nach endlich vielen Schritten ab, die nur endlich oft generierende Regeln enthält. Folglich erzeugt man in jeder unendlichen Folge von Regelanwendungen unendlich oft Variablen, die anschließend durch die $\leq$-Regel mit bereits existierenden Elementen identifiziert werden.

Beispiel 2.6 zeigt, daß es tatsächlich eine solche unendliche Folge von Regelanwendungen geben kann.

Um nun eine solche unendliche Folge von Regelanwendungen auf $\mathcal{ALCN}(\circ)$-ABoxen zu vermeiden, könnte man ähnlich wie für $\mathcal{ALCN}$-ABoxen eine Strategie angeben, die gerade diese Ursache der Nichtterminierung beseitigt.

Die Strategie 2.5 von Seite 29 basierte auf der Idee, die Regelanwendung in zwei Phasen aufzuteilen und für jede Phase Terminierung nachzuweisen. In Beispiel 2.14 wurde aber gezeigt, daß eine Phase, in der Regeln nur auf Individuen anzuwenden sind, für $\mathcal{ALCN}(\circ)$-ABoxen ggf. mehrfach zu durchlaufen ist. So ist z.B. auf

a_2 zunächst keine Regel anwendbar. Erst nachdem auf die Variable x_4 die $\exists$-Regel angewendet wurde, ist die $\leq$-Regel auf das Individuum a_0 und dann die $\forall$-Regel auf $a_2 : \forall R.(A \sqcap B)$ anwendbar. Folglich reicht eine Einteilung der Regelanwendung in diese zwei Phasen nicht aus, um auch bei Eingabe von $\mathcal{ALCN}(\circ)$-ABoxen einen terminierenden und vollständigen Algorithmus zu erhalten.

Die Idee der Einteilung in Phasen

Eine der Ursachen für den Jojo-Effekt in Beispiel 2.6 ist die vorschnelle Anwendung einer generierenden Regel. Im konkreten Beispiel bedeutet das, daß die $\exists$-Regel angewendet wird, obwohl die $\leq$-Regel schon vorher angewendet werden könnte. Man kann eine Strategie zur Kontrolle der Regelanwendungen auf $\mathcal{ALCN}(\circ)$-ABoxen angeben, die dieses Problem beseitigt und dazu die Idee der Einteilung der Regelanwendung in Phasen aufgreift. Man definiert zwei Arten von Phasen, die abwechselnd solange durchlaufen werden, bis keine Regel mehr anwendbar ist. In der ersten Phase, der sogenannten *Generierungsphase*, verbietet man die Anwendung der $\leq$-Regel und wendet alle anderen Regeln solange wie möglich an. In der zweiten Phase, der sogenannten *Identifizierungsphase*, verbietet man die Anwendung der generierenden Regeln und wendet alle anderen Regeln solange wie möglich an. Der Beweis zu Lemma 2.17 läßt sich leicht so modifizieren, daß Terminierung jeder Generierungs- bzw. Identifizierungsphase folgt. Damit diese Strategie nun tatsächlich dafür sorgt, daß der Konsistenzalgorithmus bei Eingabe einer $\mathcal{ALCN}(\circ)$-ABox stets terminiert, darf es keine unendliche Folge von Phasendurchläufen geben.

Beachte: Unter Berücksichtigung der soeben skizzierten Strategie terminiert der Algorithmus bei Eingabe der $\mathcal{ALCN}$-ABox $\mathcal{A}_0$ aus Beispiel 2.6 nach dem Durchlaufen einer Generierungs- und einer Identifizierungsphase.

Das folgende Beispiel liefert aber eine $\mathcal{ALCN}(\circ)$-ABox, für die es eine unendliche Folge von Phasendurchläufen geben kann.

Beispiel 2.18
Es bezeichne $\mathcal{A} \longrightarrow_G \mathcal{A}'$ das Ergebnis einer Generierungsphase, d.h., $\mathcal{A}'$ wurde durch Anwenden der Regeln $\longrightarrow_\sqcap$, $\longrightarrow_\sqcup$, $\longrightarrow_\forall$, $\longrightarrow_\exists$ und $\longrightarrow_\geq$ aus $\mathcal{A}$ abgeleitet und man schreibt kurz $\mathcal{A} \longrightarrow_I \mathcal{A}'$, falls $\mathcal{A}'$ aus $\mathcal{A}$ durch Anwenden der Regeln $\longrightarrow_\sqcap$, $\longrightarrow_\sqcup$, $\longrightarrow_\forall$ und $\longrightarrow_\leq$ abgeleitet wurde. Es sei

$$\begin{aligned} \mathcal{A}_0 = \{ \quad & a_0 : \exists R.A,\ \forall R.\exists R.A, \\ & a_0 : (\leq 2\, R \circ R), \\ & a_0 R a_0 \}. \end{aligned}$$

Durchläuft man ausgehend von $\mathcal{A}_0$ abwechselnd Generierungs- und Identifizierungsphasen, so ist eine unendliche Folge von Regelanwendungen möglich:

$$\begin{array}{llll}
\mathcal{A}_0 & \longrightarrow_G & \mathcal{A}_1 = \mathcal{A}_0 & \cup \quad \{a_0Rx_1,\ x_1 : A,\ x_1 : \exists R.A, \\
& & & \qquad x_1Rx_2,\ x_2 : A\} \\
& \longrightarrow_I & \mathcal{A}_2 = \mathcal{A}_1[x_1/a_0] & \cup \quad \{x_2 : \exists R.A\} \\
& \longrightarrow_G & \mathcal{A}_3 = \mathcal{A}_2 & \cup \quad \{x_2Rx_3,\ x_3 : A\} \\
& \longrightarrow_I & \mathcal{A}_4 = \mathcal{A}_3[x_2/a_0] & \cup \quad \{x_3 : \exists R.A\}
\end{array}$$

Offensichtlich beschreibt die ABox $\mathcal{A}_3$ genau die gleiche Situation wie $\mathcal{A}_1$, wobei nun lediglich die Namen a_0, x_2, x_3 statt a_0, x_1, x_2 verwendet werden. Ebenso unterscheiden sich die ABoxen $\mathcal{A}_2$ und $\mathcal{A}_4$ nur in der Menge der verwendeten Namen, d.h., es ist eine unendliche Folge von Regelanwendungen ähnlich wie in Beispiel 2.6 möglich. *Beachte:* In der ersten Identifizierungsphasekann man x_1 und a_0 identifizieren, da $a_0Ra_0Ra_0$ und $a_0Ra_0Rx_1$ $(R \circ R)$-Rollenketten in $\mathcal{A}_1$ sind. Wegen $a_0Rx_1Rx_2$ in $\mathcal{A}_1$ kann man auch x_2 und a_0 identifizieren. Dann terminiert die Regelanwendung mit der vollständigen, offenen ABox $\mathcal{A}_2' = \mathcal{A}_1[x_2/a_0]$. Zum Nachweis der Vollständigkeit des Algorithmus müssen aber alle möglichen Folgen von Regelanwendungen und damit auch alle möglichen Identifizierungen berücksichtigt werden. ◇

Mit Beispiel 2.18 folgt also, daß eine Strategie, die die Regelanwendung in Generierungs- und Identifizierungsphasen einteilt, nicht das Gewünschte leistet.

Die Idee des direkten Identifizierens

Eine alternative Strategie beruht auf der Idee, jeweils unmittelbar nach dem Anwenden einer generierenden Regel ($\longrightarrow_\exists$ oder $\longrightarrow_\geq$) alle $\leq$-Assertionen zu erfüllen, d.h., solange wie möglich die $\leq$-Regel anzuwenden. Diese Idee stammt aus [BHs91], wo für eine Erweiterung von $\mathcal{ALC}$ die Entscheidbarkeit des Konsistenzproblems bewiesen wird. Leider läßt sich auch mit dieser Strategie eine unendliche Folge von Regelanwendungen nicht verhindern.

Beispiel 2.19 (Beispiel 2.18 Fortsetzung)
Gegeben sei die ABox $\mathcal{A}_0$ aus Beispiel 2.18. Man schreibt kurz $\mathcal{A} \longrightarrow_P \mathcal{A}'$, wenn $\mathcal{A}'$ aus $\mathcal{A}$ durch Anwenden der propagierenden Regeln ($\longrightarrow_\sqcap$, $\longrightarrow_\sqcup$ und $\longrightarrow_\forall$) abgeleitet wird. Man schreibt $\mathcal{A} \longrightarrow_{\exists,\leq} \mathcal{A}'$ bzw. $\mathcal{A} \longrightarrow_{\geq,\leq} \mathcal{A}'$, falls ABoxen $\mathcal{A}_1, \ldots, \mathcal{A}_n$, $n \geq 0$, existieren mit

$$\mathcal{A} \longrightarrow_\exists \mathcal{A}_1 \longrightarrow_\leq \mathcal{A}_2 \longrightarrow_\leq \cdots \longrightarrow_\leq \mathcal{A}_n \longrightarrow_\leq \mathcal{A}'$$

bzw.

$$\mathcal{A} \longrightarrow_\geq \mathcal{A}_1 \longrightarrow_\leq \mathcal{A}_2 \longrightarrow_\leq \cdots \longrightarrow_\leq \mathcal{A}_n \longrightarrow_\leq \mathcal{A}'.$$

Ausgehend von $\mathcal{A}_0$ sind nun folgende Regelanwendungen möglich:

$$\begin{array}{lll}
\mathcal{A}_0 & \longrightarrow_{\exists,\leq} & \mathcal{A}_1 = \mathcal{A}_0 \cup \{a_0Rx_1,\ x_1 : A\} \\
& \longrightarrow_P & \mathcal{A}_2 = \mathcal{A}_1 \cup \{x_1 : \exists R.A\} \\
& \longrightarrow_{\exists,\leq} & \mathcal{A}_3 = \big(\mathcal{A}_2 \cup \{x_1Rx_2,\ x_2 : A\}\big)[x_1/a_0] \\
& \longrightarrow_P & \mathcal{A}_4 = \mathcal{A}_3 \cup \{x_2 : \exists R.A\} \\
& \longrightarrow_{\exists,\leq} & \mathcal{A}_5 = \big(\mathcal{A}_4 \cup \{x_2Rx_3,\ x_3 : A\}\big)[x_2/a_0]
\end{array}$$

Wiederum beschreiben nun $\mathcal{A}_5$ und $\mathcal{A}_3$ bzw. $\mathcal{A}_4$ und $\mathcal{A}_2$ die gleichen Situationen, nur mit geänderten Namen, sodaß wiederum ähnlich wie in Beispiel 2.6 eine unendliche Folge von Regelanwendungen möglich ist. Also reicht auch diese Strategie nicht aus, um die Terminierung des Konsistenzalgorithmus zu gewährleisten. $\diamond$

Bemerkung 2.20
Bei den in den Beispielen 2.18 und 2.19 skizzierten Folgen von Regelanwendungen wird die Strategie in Abbildung 2.5 in dem Sinne eingehalten, daß zunächst solange wie möglich Regeln auf Individuen (hier jeweils nur a_0) angewendet werden und dann erst auf Variablen. Dann ist aber erneut eine Regel (hier die $\leq$-Regel) auf das Individuum a_0 anwendbar. Folglich würde die Einhaltung der Strategie in Abbildung 2.5 keine vollständige ABox liefern, mit der ein kanonisches Modell von $\mathcal{A}_0$ definiert werden könnte.

Die in den bisherigen Beispielen untersuchten Folgen von Regelanwendungen auf $\mathcal{ALCN}(\circ)$-ABoxen lieferten stets nur ABoxen, die jeweils nur endlich viele Elemente enthalten. Die Beispiele im folgenden Abschnitt werden aber zeigen, daß es $\mathcal{ALCN}(\circ)$-ABoxen gibt, so daß sogar für die soeben skizzierten Strategien unendliche Folgen von Regelanwendungen möglich sind, durch die man im Limes eine ABox erhält, die unendlich viele Elemente enthält. $\diamond$

Die Beispiele 2.18 und 2.19 zeigen, daß es sehr schwierig ist, eine Strategie zu entwickeln, die sicherstellt, daß Algorithmus 2.3 stets terminiert und damit ein Entscheidungsverfahren für das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ liefert.

2.2.5 Zur Endlichen-Modell-Eigenschaft

Eine weitere Möglichkeit, das gewünschte Entscheidbarkeitsresultat zu erhalten, besteht darin, direkt die Endliche-Modell-Eigenschaft für $\mathcal{ALCN}(\circ)$-ABoxen nachzuweisen. Dazu muß man zeigen, daß es zu jeder konsistenten ABox ein endliches Modell gibt. Da außerdem ein Verfahren existiert, das bei Eingabe einer inkonsistenten ABox $\mathcal{A}$ nach endlicher Zeit mit Ausgabe „$\mathcal{A}$ ist inkonsistent." terminiert, ließe sich ein effektives Verfahren angeben, das das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ entscheidet.

Zur Existenz eines widerlegungsvollständigen Verfahrens: Mit den Ergebnissen aus [Bor96] läßt sich zu einer verallgemeinerten $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$ eine Formel $\Phi(\mathcal{A}_0)$ der Prädikatenlogik 1. Stufe definieren für die gilt:

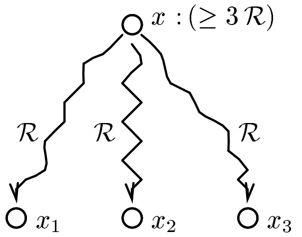
$$\mathcal{A}_0 \text{ ist konsistent} \iff \Phi(\mathcal{A}_0) \text{ ist erfüllbar.}$$

Die Menge der unerfüllbaren Formeln der Prädikatenlogik 1. Stufe ist rekursiv aufzählbar. Also existiert ein effektives Verfahren, daß bei Eingabe einer unerfüllbaren Formel ϕ nach endlicher Zeit mit Ausgabe „ϕ ist unerfüllbar." terminiert. Damit leistet ein Verfahren, das zu einer $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$ die zugehörige Formel $\Phi(\mathcal{A}_0)$

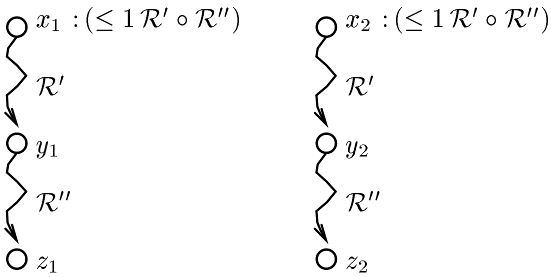
bestimmt und diese auf Unerfüllbarkeit testet, das Gewünschte.

Zum Nachweis der Endlichen-Modell-Eigenschaft: Es ist zu zeigen, daß zu jeder konsistenten $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$ eine *endliche Interpretation* $\mathcal{I}$ existiert, die jede Assertion aus $\mathcal{A}_0$ erfüllt. Dazu versucht man, ausgehend von einem unendlichen Modell $\mathcal{I}'$ von $\mathcal{A}_0$, eine endliche Interpretation $\mathcal{I}$ zu definieren, die Modell von $\mathcal{A}_0$ ist.

Um $\mathcal{I}$ zu definieren bestimmt man zunächst eine endliche Klassifizierung der Elemente der Trägermenge $\triangle_{\mathcal{I}'}$ des unendlichen Modells $\mathcal{I}'$ und definiert dann mit Hilfe dieser Charakterisierung eine Äquivalenzrelation $\sim$ mit endlichem Index auf $\triangle_{\mathcal{I}'}$. Faßt man anschließend die unendlich vielen Elemente aus $\triangle_{\mathcal{I}'}$ zu endlich vielen Äquivalenzklassen zusammen, so erhält man eine endliche Interpretation $\mathcal{I} = (\Delta_I, \cdot^I)$ mit Trägermenge $\Delta_I = \triangle_{\mathcal{I}'}/\sim$. Allerdings ändert sich im allgemeinen durch dieses Zusammenfassen die Anzahl der Vorgänger und Nachfolger eines Elementes bzw. einer Klasse von Elementen, so daß in der erhaltenen Interpretation $\mathcal{I}$ ggf. einige Zahlenrestriktionen nicht mehr erfüllt sind. Abbildung 2.8 verdeutlicht die Situationen, in denen durch einfaches Zusammenfassen zueinander äquivalenter Elemente Zahlenrestriktionen für Vorgänger in Δ_I verletzt werden.



Gilt $x_1 \sim x_2 \sim x_3$, so wird die $\geq$-Assertion verletzt, wenn keine weiteren $\mathcal{R}$-Nachfolger von $[x]_\sim$ existieren.



Gilt $x_1 \not\sim x_2$, $y_1 \sim y_2$ und $z_1 \not\sim z_2$, so werden die $\leq$-Assertionen zu $[x_1]_\sim$ und $[x_2]_\sim$ verletzt.

Abbildung 2.8: Äquivalenzklassen und verletzte Zahlenrestriktionen.

Man kann versuchen, diese Probleme zu umgehen bzw. zu lösen, indem man ein spezielles, unendliches Modell untersucht. Dazu zeigt man, daß ausgehend von einer konsistenten ABox $\mathcal{A}_0$ und einer unendlichen Folge $\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \cdots$ von Regelanwendungen durch den 'Limes' eine kanonische Interpretation $\mathcal{I}'$ definiert werden kann, die Modell von $\mathcal{A}_0$ ist. Ist $\mathcal{I}$ endlich, so ist nichts mehr zu zeigen. Sonst ist ausgehend von dem speziellen, unendlichen Modell $\mathcal{I}'$ eine endliche Interpretation $\mathcal{I}$ zu definieren, die Modell von $\mathcal{A}_0$ ist.

Bei der Untersuchung solch spezieller, kanonischer Modelle wurde zunächst festgestellt, daß es unendliche Folgen von Regelanwendungen geben kann, durch die im Limes eine ABox mit unendlich vielen Variablen erzeugt wird. Die mit dieser unendlichen ABox definierte kanonische Interpretation enthält dann ebenfalls unendlich viele Elemente. Daher ergeben sich auch bei einer endlichen Klassifizierung der Elemente solch spezieller, unendlicher Modelle die in Abbildung 2.8 skizzierten Probleme.

Beispiel 2.21
Gegeben sei folgende Ausgangs-ABox $\mathcal{A}_0$:

$$\begin{aligned}\mathcal{A}_0 = \{ \quad & a_0 : \exists R.(\geq 1\, S \circ S \circ S),\ \forall R.\exists R.(\geq 1\, S \circ S \circ S),\\ & a_0 : (\leq 2\, R \circ R),\ (\leq 1\, S \circ S),\\ & a_0 R a_0 \}.\end{aligned}$$

Beispiel 2.18 hat gezeigt, daß mit dieser Menge von Assertionen eine unendliche Folge von Regelanwendungen möglich ist. Dabei werden unendlich viele R- und $(R \circ R)$-Nachfolger von a_0 erzeugt. Im vorliegenden Fall kann zu jedem dieser R-Nachfolger ein $(S \circ S \circ S)$-Nachfolger erzeugt werden, bevor der R-Nachfolger mit a_0 identifiziert wird. Dadurch erhält a_0 unendlich viele $(S \circ S \circ S)$-Nachfolger. Die unendlich vielen $(S \circ S)$-Nachfolger werden alle mit einer Variablen x_0 identifiziert, wodurch sowohl für das Individuum a_0 als auch die Variable x_0 im Limes unendlich viele S-Nachfolger existieren. Abbildung 2.9 stellt das Ergebnis der soeben skizzierten unendlichen Folge von Regelanwendungen dar. Dabei ist die kanonische Interpretation, die mit Hilfe der unendlichen ABox definiert werden kann, ein unendliches Modell von $\mathcal{A}_0$. $\diamond$

Bemerkung 2.22
Auch für das Beispiel 2.21 gilt die Aussage aus Bemerkung 2.20, daß zunächst nur auf das Individuum a_0 Regeln angewendet werden. Desweiteren ist aber eine unendliche Folge von Regelanwendungen, die zu der in Abbildung 2.9 skizzierten ABox führt, auch unter Berücksichtigung der in den Beispielen 2.18 und 2.19 untersuchten Strategien möglich. $\diamond$

Beispiel 2.23
Dieses Beispiel zeigt, daß man ausgehend von der konsistenten ABox $\mathcal{A}_0$ eine unendliche ABox ableiten kann, die eine unendlich lange, zyklenfreie Rollenkette enthält.

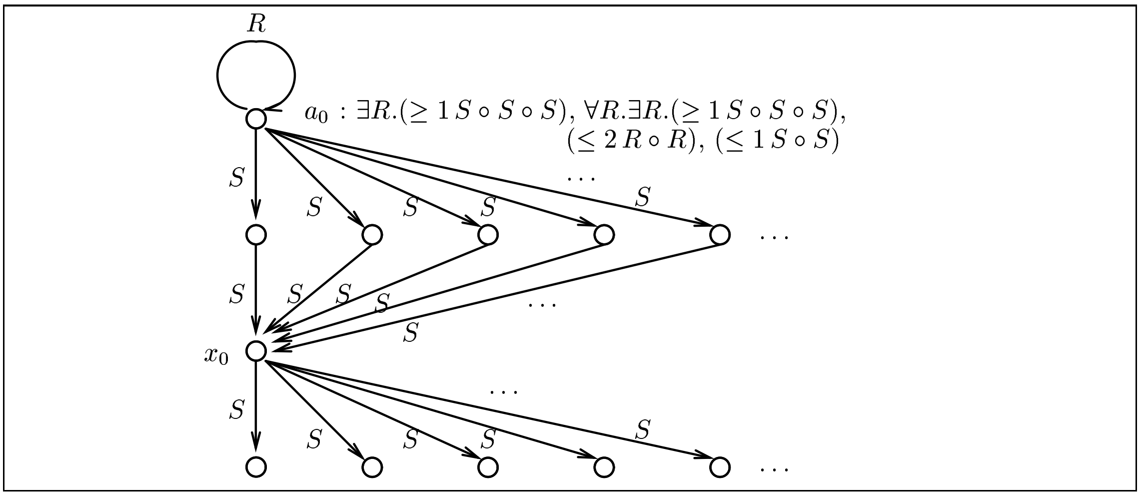


Abbildung 2.9: Ein unendliches kanonisches Modell.

$$\begin{aligned}\mathcal{A}_0 = \{ \quad & a_0 : \exists R.\exists S.A,\ \forall R.\exists R.\exists S.A,\\ & a_0 : \forall S.(\geq 1\ R \circ S),\\ & a_0 : (\leq 1\, R),\ (\leq 1\, S \circ R \circ S),\\ & a_0 R a_0,\ a_0 S a_0\}.\end{aligned}$$

Man erzeugt wiederum unendlich viele R- und $(R \circ R)$-Nachfolger von a_0 und zu jedem $(R \circ R)$-Nachfolger einen S-Nachfolger. Diese Variablen werden durch das Identifizieren der $(R \circ R)$-Nachfolger mit a_0 zu S-Nachfolger von a_0, so daß auf Grund der $\forall S$-Assertion jeweils ein $(R \circ S)$-Nachfolger erzeugt werden kann. Dadurch erhält a_0 beliebig viele $(S \circ R \circ S)$-Nachfolger. Um ein Modell von $\mathcal{A}_0$ zu erhalten, müßten alle Variablen mit a_0 identifiziert werden. Man kann aber auch unendlich viele Variablen so ersetzen, daß man eine unendlich lange, zyklenfreie Rollenkette erhält (Abbildung 2.10). Durch die Zyklen $a_0 R a_0$ und $a_0 S a_0$ ist nämlich jeder S-Nachfolger von a_0 auch $(S \circ R \circ S)$-Nachfolger von a_0, sodaß ein $(S \circ R \circ S)$-Nachfolger mit einem S-Nachfolger von a_0 auf Grund der Assertion $a_0 : (\leq 1\, S \circ R \circ S)$ identifiziert werden kann. Dabei ist die kanonische Interpretation, die mit Hilfe der erhaltenen unendlichen ABox definiert werden kann, aber kein Modell von $\mathcal{A}_0$. $\diamond$

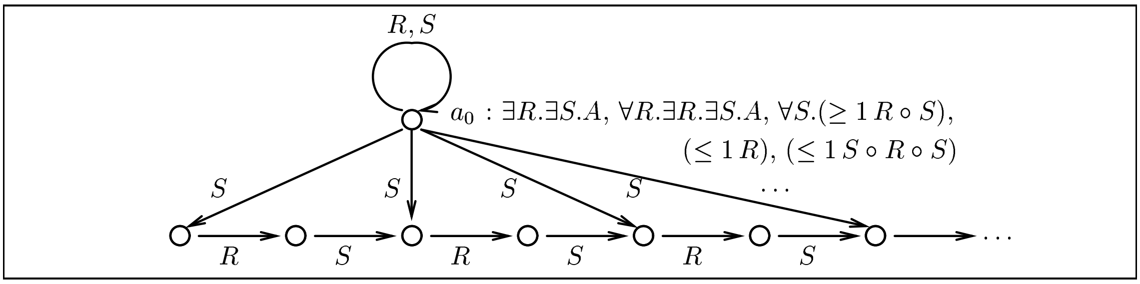


Abbildung 2.10: Erzeugen unendlich langer, zyklenfreier Rollenketten.

Mit Hilfe von Lemma 2.16 sieht man leicht, daß stets unendliche Verzweigungen erzeugt werden, wenn eine mit den Vervollständigungsregeln abgeleitete ABox unendlich viele Elemente enthält. Beispiel 2.23 zeigt, daß man sogar unendlich lange, zyklenfreie Rollenketten generieren kann, wobei man im konkreten Fall allerdings kein unendliches Modell der Ausgangs-ABox erhält. Die Folge der Regelanwendungen, die in Beispiel 2.23 zu der unendlich langen Rollenkette führt ist aber in folgendem Sinn *unfair*: Es sei $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \cdots$ die unendliche Folge der abgeleiteten ABoxen. Dann ist offensichtlich die $\leq$-Regel ab einem Index j_0 in jeder ABox $\mathcal{A}_j$, $j \geq j_0$, auf die Assertion $a_0 : (\leq 1\, S \circ R \circ S)$ anwendbar, ohne daß ein Index $k > j_0$ existiert, so daß $\longrightarrow_{\leq}$ in $\mathcal{A}_k$ nicht mehr auf $a_0 : (\leq 1\, S \circ R \circ S)$ angewendet werden kann.

Demgegenüber gilt eine Folge von Regelanwendungen als *fair*, falls zu jeder Assertion $x : C \in \mathcal{A}_i$, auf die eine Regel $\longrightarrow$ anwendbar ist, ein $j > i$ existiert, so daß $\longrightarrow$ in $\mathcal{A}_j$ nicht mehr auf $x : C$ anwendbar ist.

Unter der Voraussetzung, daß diese Fairneßannahme bei der Regelanwendung erfüllt wird, liefert eine unendliche Folge von Regelanwendungen ausgehend von einer konsistenten ABox $\mathcal{A}_0$ eine ggf. unendliche kanonische Interpretation, die Modell von $\mathcal{A}_0$ ist. An dieser Stelle erhält man also spezielle Modelle von $\mathcal{A}_0$. Beispiel 2.21 hat gezeigt, daß solche kanonischen Modelle durchaus noch unendlich viele Elemente enthalten können. Aus der Untersuchung dieser speziellen, unendlichen Modelle ergaben sich unter anderem folgende Feststellungen.

1. Bei dem Versuch, eine Äquivalenzrelation mit endlichem Index auf der Trägermenge zu definieren, die anschließend ein endliches Modell von $\mathcal{A}_0$ liefern soll, stößt man wiederum auf die in Abbildung 2.8 skizzierten Probleme.

2. Unter der Annahme, daß ein solches unendliches, kanonisches Modell keine unendlich langen, zyklenfreien Rollenketten enthält, kann man einen Algorithmus definieren, der in endlich vielen Schritten eine endliche Menge $\triangle$ von Elementen aus der Trägermenge markiert, so daß die unendliche Interpretation eingeschränkt auf diese Menge $\triangle$ ein endliches Modell von $\mathcal{A}_0$ ist.

Leider bleibt das Problem offen, ob es unter Berücksichtigung der Fairneßannahme in einem solchen unendlichen, kanonischen Modell ausgehend von einem Individuum $a_0 \in \tau_{\mathcal{A}_0}$ eine unendlich lange, zyklenfreie Rollenkette geben kann.

Damit wird auch mit diesem Ansatz nicht das angestrebte Ziel, der Beweis der Entscheidbarkeit des Konsistenzproblems für $\mathcal{ALCN}(\circ)$, erreicht.

2.2.6 Eine entscheidbare Klasse von $\mathcal{ALCN}(\circ)$-ABoxen

Um dieses Kapitel dennoch mit einem Entscheidbarkeitsresultat abzuschließen, überlegt man sich, unter welchen Voraussetzungen der Algorithmus aus Abbildung 2.3 terminiert und somit Konsistenz für die Eingabe-ABox entscheidet. Insbesondere sucht man Bedingungen, die die Eingabe-ABox $\hat{\mathcal{A}}$ erfüllen muß, damit sich der Terminierungsbeweis aus [BS96] auf den Konsistenzalgorithmus übertragen läßt.

Die in den 'bösen' Beispielen 2.6 und 2.18 betrachteten ABoxen enthalten jeweils eine zyklische Rollenkette. Deshalb untersucht man das Terminierungsverhalten des Algorithmus zunächst für azyklische ABoxen.

Beispiel 2.24
Gegeben sei folgende azyklische Ausgangs-ABox $\mathcal{A}_0$:

$$\begin{aligned}\mathcal{A}_0 = \{ \quad & a_0 : (\leq 1\, R \circ R), \\ & a_2 : \exists R.A,\ \forall R.\exists R.A, \\ & a_0 R a_1,\ a_1 R a_2,\ a_0 R a_2 \}.\end{aligned}$$

Ausgehend von $\mathcal{A}_0$ sind folgende Regelanwendungen möglich:

$$\begin{array}{lll}\mathcal{A}_0 & \longrightarrow_\exists & \mathcal{A}_1 := \mathcal{A}_0 \cup \{a_2 R x_1,\ x_1 : A\} \\ & \longrightarrow_\forall & \mathcal{A}_2 := \mathcal{A}_1 \cup \{x_1 : \exists R.A\} \\ & \longrightarrow_\exists & \mathcal{A}_3 := \mathcal{A}_2 \cup \{x_1 R x_2,\ x_2 : A\} \\ & \longrightarrow_\leq & \mathcal{A}_4 := \mathcal{A}_3[x_1/a_2] \\ & \longrightarrow_\forall & \mathcal{A}_5 := \mathcal{A}_4 \cup \{x_2 : \exists R.A\} \\ & \longrightarrow_\exists & \mathcal{A}_6 := \mathcal{A}_5 \cup \{x_2 R x_3,\ x_3 : A\} \\ & \longrightarrow_\leq & \mathcal{A}_7 := \mathcal{A}_6[x_2/a_2]\end{array}$$

Abbildung 2.11 veranschaulicht diese Folge von Regelanwendungen. Die ABoxen $\mathcal{A}_4$ und $\mathcal{A}_7$ beschreiben die gleiche Situation, wobei lediglich verschiedene Variablen verwendet werden. Damit ist auch ausgehend von einer azyklischen ABox eine unendliche Folge von Regelanwendungen möglich. Insbesondere läßt sich die ABox $\mathcal{A}_0$ so modifizieren, daß man azyklische ABoxen erhält, so daß ausgehend von diesen ABoxen, ähnlich wie in den Beispielen 2.18 und 2.19 skizziert, unendliche Folgen von Regelanwendungen möglich sind, durch die ABoxen erzeugt werden, die unendlich viele Elemente enthalten. ◇

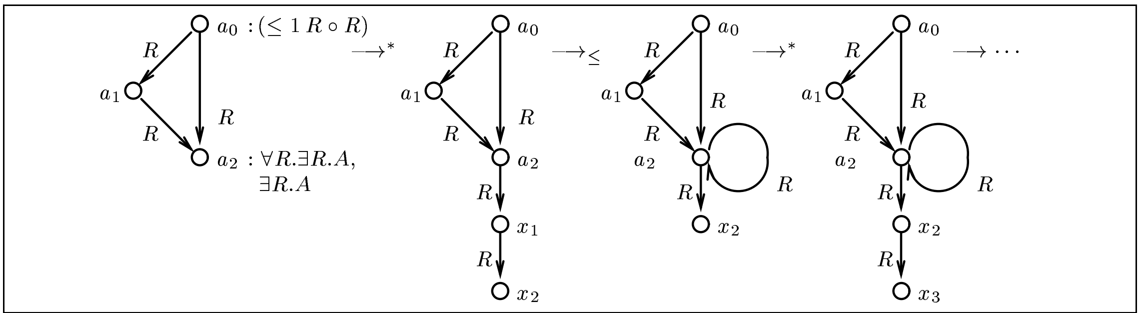


Abbildung 2.11: Noch ein Jojo-Effekt.

Beispiel 2.24 zeigt also, daß die Forderung, daß die Ausgangs-ABox $\mathcal{A}_0$ zyklenfrei sein muß, nicht ausreicht, um sicherzustellen, daß jede Folge von Regelanwendungen ausgehend von $\mathcal{A}_0$ terminiert. Außerdem erkennt man, daß sich der Jojo-Effekt nicht unmittelbar aus einer zyklischen Rollenkette sondern vielmehr aus unterschiedlich langen Rollenketten zwischen zwei Individuen ergibt. Dabei entspricht die *Länge einer Rollenkette* der Anzahl verschiedener Elemente aus $\mathcal{A}$, die diese Rollenkette bilden.

Diese Beobachtung motiviert nun die Untersuchung von Ausgangs-ABoxen $\mathcal{A}_0$, die folgende, schärfere Bedingung erfüllen:

($*$) Zwischen je zwei verschiedenen Individuen existieren nur Rollenketten gleicher Länge in $\mathcal{A}_0$.

Durch die Bedingung ($*$) schließt man die ABox $\mathcal{A}_0$ aus Beispiel 2.24 aus. Erfüllt eine Ausgangs-ABox $\mathcal{A}_0$ die Bedingung ($*$), so überträgt sich die Bedingung ($*$) allerdings nicht auf ABoxen $\mathcal{A}$, die aus $\mathcal{A}_0$, abgeleitet werden können. Das folgende Beispiel zeigt sogar, daß auch für Ausgangs-ABoxen, die die Bedingung ($*$) erfüllen, unendliche Folgen von Regelanwendungen möglich sind.

Beispiel 2.25

Folgende Ausgangs-ABox $\mathcal{A}_0$ genügt offensichtlich der Bedingung ($*$).

$$
\begin{aligned}
\mathcal{A}_0 = \{ \quad & a_1 : (\leq 1\, R \circ R \circ R), \\
& a_4 : \exists R.A,\ \forall R.\exists R.A,\ \exists R.B, \\
& a_5 : (\leq 1\, R \circ R), \\
& a_7 : \exists R.B, \\
& a_1 R a_2,\ a_2 R a_3,\ a_3 R a_4, \\
& a_5 R a_6,\ a_6 R a_4,\ a_5 R a_7,\ a_1 R a_7 \}.
\end{aligned}
$$

Durch folgende Regelanwendungen erhält man dann eine ABox, die eine zyklische Rollenkette enthält.

$$
\begin{aligned}
\mathcal{A}_0 \quad & \longrightarrow_{\exists} \quad \mathcal{A}_1 := \mathcal{A}_0 \cup \{a_7 R x_1,\ x_1 : B\} \\
& \longrightarrow_{\leq} \quad \mathcal{A}_2 := \mathcal{A}_1[x_1/a_4] \\
& \longrightarrow_{\exists} \quad \mathcal{A}_3 := \mathcal{A}_2 \cup \{a_4 R x_2,\ x_2 : B\} \\
& \longrightarrow_{\leq} \quad \mathcal{A}_4 := \mathcal{A}_3[x_2/a_4]
\end{aligned}
$$

$\mathcal{A}_4$ enthält nun den Rollenzykel $a_4 R a_4$. Insbesondere gilt

$$\{a_4 R a_4,\ a_4 : \exists R.A,\ a_4 : \forall R.\exists R.A\} \subseteq \mathcal{A}_4.$$

Mit Beispiel 2.6 folgt also, daß eine unendliche Folge von Regelanwendungen ausgehend von einer ABox $\mathcal{A}_0$, die die Bedingung ($*$) erfüllt, möglich ist. In Abbildung 2.12 sind die ABoxen $\mathcal{A}_0$ und $\mathcal{A}_4$ dargestellt. $\diamond$

Für die Ausgangs-ABoxen $\mathcal{A}_0$ aus den Beispielen 2.24 und 2.25 ist die Terminierung deshalb nicht gesichert, weil jeweils unterschiedlich lange Rollenketten zwischen zwei

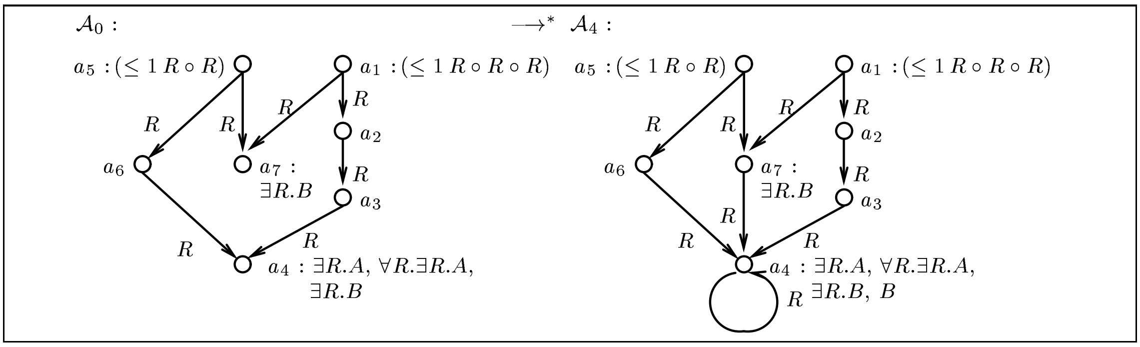


Abbildung 2.12: Zur Entstehung des Jojo-Effekts.

Elementen existieren bzw. erzeugt werden können, wodurch der Jojo-Effekt und damit eine unendliche Folge von Regelanwendungen ermöglicht wird. Bei der Untersuchung der Ausgangs-ABoxen erkennt man aber leicht, daß sie keine Levelstruktur besitzen, die folgender Intuition genügt: Sind zwei Elemente x und y durch eine Rollenkette der Länge m verbunden, so gilt $\mathsf{level}(y) = \mathsf{level}(x) + m$.

Die folgende Definition setzt nun genau diese Intuition in eine hinreichende Bedingung an die Ausgangs-ABoxen um, so daß sich die Terminierung leicht mit der in [BS96] angegebenen Terminierungsordnung folgern läßt.

Definition 2.26
Sei $\hat{\mathcal{A}}$ eine beliebige $\mathcal{ALCN}(\circ)$-ABox und $\mathcal{A}_0$ die zugehörige verallgemeinerte ABox. Dann heißt $\mathcal{A}_0$ *zulässig* genau dann, wenn eine Partition $I_1 \dot\cup \ldots \dot\cup I_\nu$ von $\tau_{\mathcal{A}_0}$ und eine Abbildung $\mathsf{level} : \tau_{\mathcal{A}_0} \longrightarrow \mathbb{N}$ existiert mit

1. für je zwei Individuen $a \in I_i$ und $a' \in I_j$ mit $i \neq j$ existiert keine Rollenassertion aRa' in $\mathcal{A}_0$,
2. für alle $1 \leq j \leq \nu$ existiert ein $a_j \in I_j$ mit $\mathsf{level}(a_j) = 0$ und
3. für jede Rollenassertion aRa' in $\mathcal{A}_0$ gilt $\mathsf{level}(a') = \mathsf{level}(a) + 1$.

◇

Satz 2.27
Für eine gegebene, endliche $\mathcal{ALCN}(\circ)$-ABox ist es effektiv entscheidbar, ob sie zulässig ist oder nicht. ⌟

Beweis:
Sei $\mathcal{A}_0$ eine endliche $\mathcal{ALCN}(\circ)$-ABox. Dann ist $\tau_{\mathcal{A}_0}$ endlich und es gibt nur endlich viele Partitionen $I_1 \dot\cup \ldots \dot\cup I_\nu$ von $\tau_{\mathcal{A}_0}$. Mit der 3. Bedingung aus Definition 2.26 folgt, daß eine zulässige ABox keine zyklischen Rollenketten enthält. Die Länge einer maximalen, zyklenfreien Rollenkette in $\mathcal{A}_0$ kann durch $|\tau_{\mathcal{A}_0}|$ beschränkt werden. Also folgt mit der 2. Bedingung

$$\mathsf{level}(a) \in \{0, \ldots, |\tau_{\mathcal{A}}|\} \text{ für alle } a \in \tau_{\mathcal{A}}.$$

Ist $\tau_{\mathcal{A}_0}$ endlich, so gibt es offensichtlich nur endlich viele mögliche Abbildungen $\mathsf{level} : \tau_{\mathcal{A}_0} \longrightarrow \{0, \ldots, |\tau_{\mathcal{A}_0}|\}$. Für eine endliche $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$, eine Partition $I_1 \dot\cup \ldots \dot\cup I_\nu$ von $\tau_{\mathcal{A}_0}$ und eine Funktion $\mathsf{level} : \tau_{\mathcal{A}_0} \longrightarrow \{0, \ldots, |\tau_{\mathcal{A}_0}|\}$ ist es offensichtlich effektiv entscheidbar, ob sie die Bedingungen 1., 2. und 3. aus Definition 2.26 erfüllen. □

Bemerkung 2.28
Die Definition einer zulässigen Ausgangs-ABox schließt die Ausgangs-ABoxen aus den Beispielen 2.24 und 2.25 aus. Abbildung 2.13 zeigt die Rollenassertionen einer zulässigen ABox $\mathcal{A}$. Die zugehörige Partition von $\tau_{\mathcal{A}}$ lautet $I_1 = \{a_1, \ldots, a_5\}$, $I_2 = \{a_6, \ldots, a_{10}\}$ und die zugehörige Levelfunktion liefert

- $\mathsf{level}(a) = 0$ für $a \in \{a_1, a_2, a_6\}$,
- $\mathsf{level}(a) = 1$ für $a \in \{a_3, a_4, a_7, a_8, a_9\}$ und
- $\mathsf{level}(a) = 2$ für $a \in \{a_5, a_{10}\}$.

Der Beweis von Satz 2.27 liefert ein effektives, aber sehr ineffizientes Verfahren, um für eine $\mathcal{ALCN}(\circ)$-ABox zu entscheiden, ob sie zulässig ist oder nicht. Für eine zulässige $\mathcal{ALCN}(\circ)$-ABox $\mathcal{A}_0$ ist aber sowohl die Partition als auch die zugehörige Levelfunktion durch die drei Punkte der Definition 2.26 eindeutig bestimmt. Dies läßt sich ggf. nutzen, um ein effizienteres Verfahren zu formulieren, um für eine zulässige ABox die Partition und die zugehörige Levelfunktion zu bestimmen. Auf die Angabe und Untersuchung eines geeigneten Verfahrens wird hier verzichtet, da die Entscheidbarkeit der Klasse der zulässigen $\mathcal{ALCN}(\circ)$-ABoxen für das folgende Entscheidbarkeitsresultat ausreichend ist.
Außerdem gilt für eine zulässige ABox $\mathcal{A}_0$ und zwei Individuen $a, a' \in \tau_{\mathcal{A}_0}$, die durch eine $\mathcal{R}$-Rollenkette der Länge m in $\mathcal{A}_0$ verbunden sind: $\mathsf{level}(a') = \mathsf{level}(a) + m$. ◇

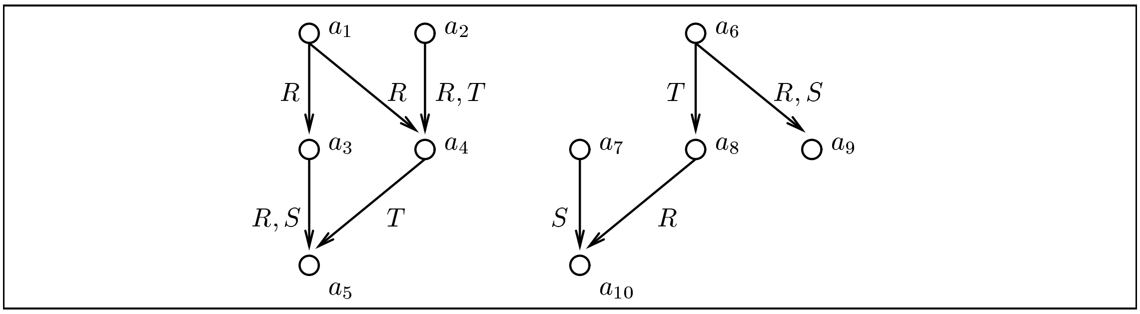


Abbildung 2.13: Die Rollenassertionen einer zulässigen ABox.

Ausgehend von einer zulässigen ABox $\mathcal{A}_0$ und einer Folge von Regelanwendungen $\mathcal{A}_0 \longrightarrow^* \mathcal{A}$ erweitert man die Levelfunktion zu $\mathcal{A}_0$ zu einer Levelfunktion zu $\mathcal{A}$ durch:

- Wird y durch anwenden der $\exists$-Regel auf $x : \exists R.C$ erzeugt, so sei $\mathsf{level}(y) := \mathsf{level}(x) + 1$.
- Werden $y_1, \ldots, y_m$ durch anwenden der $\geq$-Regel auf $x : (\geq\ n\, R_1 \circ \ldots \circ R_m)$ erzeugt, so sei $\mathsf{level}(y_i) := \mathsf{level}(x) + i$ für $1 \leq i \leq m$.

Bemerkung 2.29
In einer zulässigen ABox $\mathcal{A}_0$ werden nur Individuen a, a' identifiziert, die denselben Level besitzen und nicht unterschieden sind, denn: Ist $\longrightarrow_{\leq}$ auf $a'' : (\leq\ n\,\mathcal{R})$ und $a, a' \in \mathcal{R}(a'')$ in $\mathcal{A}_0$ anwendbar, so ist $\mathsf{level}(a) = \mathsf{level}(a') = \mathsf{level}(a'') + |\mathcal{R}|$ und $a \neq a' \notin \mathcal{A}_0$. In einer aus $\mathcal{A}_0$ abgeleiteten ABox $\mathcal{A}$ werden dann ebenfalls nur Elemente mit gleichem Level identifiziert. Also besitzt jedes Element $x \in \tau_{\mathcal{A}}$ einen eindeutigen Level $\mathsf{level}(x)$. Dieser Level ergibt sich für Individuen aus der Levelfunktion der zulässigen Ausgangs-ABox $\mathcal{A}_0$ und für Variablen aus $\mathcal{A}$ aus der Erweiterung dieser Levelfunktion auf $\mathcal{A}$. ◇

Lemma 2.30
Sei $\mathcal{A}_0$ eine zulässige Ausgangs-ABox. Dann gilt für jede ABox $\mathcal{A}$ mit $\mathcal{A}_0 \longrightarrow^* \mathcal{A}$ und jeden Rollenterm $\mathcal{R}$

$$y \in \mathcal{R}(x) \text{ in } \mathcal{A} \implies \mathsf{level}(y) = \mathsf{level}(x) + |\mathcal{R}|.$$

⌟

Beweis:
Man zeigt die Behauptung durch vollständige Induktion über der Anzahl n der Regelanwendungen, mit denen $\mathcal{A}$ aus $\mathcal{A}_0$ abgeleitet wird.

$n = 0$: Da $\mathcal{A}_0$ zulässig ist, folgt die Behauptung mit Bemerkung 2.28.

$n \longrightarrow n+1$: Hier zeigt man die Behauptung durch eine Fallunterscheidung über die Regeln aus Abbildung 2.2.

- $\mathcal{A}_n \longrightarrow \mathcal{A}_{n+1}$ mit $\longrightarrow \in \{\longrightarrow_{\sqcap}, \longrightarrow_{\sqcup}, \longrightarrow_{\forall}\}$:
  Die Menge der Rollenassertionen in $\mathcal{A}_{n+1}$ ist gleich der Menge der Rollenassertionen in $\mathcal{A}_n$, also folgt die Behauptung aus der Induktionsvoraussetzung.
- $\mathcal{A}_n \longrightarrow_{\exists} \mathcal{A}_{n+1} = \mathcal{A}_n \cup \{xRy,\ y : C\}$:
  Aus der Definition der $\exists$-Regel folgt, daß y nicht in Rollenketten von x' nach y' in $\mathcal{A}_{n+1}$ mit $x', y' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y\}$ auftritt. Also folgt für Rollenketten von x' nach y' in $\mathcal{A}_{n+1}$ mit $x', y' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y\}$ die Behauptung per Induktion. Für $x' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y\}$ und y ist jede Rollenkette in $\mathcal{A}_{n+1}$ von x' nach y von der Form $x'\mathcal{R}xRy$. Mit der Induktionsvoraussetzung und der Definition der Levelfunktion zu $\mathcal{A}_{n+1}$ folgt

  $$\mathsf{level}(y) = \mathsf{level}(x') + |\mathcal{R}| + 1 = \mathsf{level}(x') + |\mathcal{R} \circ R|.$$

Es gibt offensichtlich keine Rollenkette in $\mathcal{A}_{n+1}$, die mit y beginnt. Daher sind dies alle Fälle.

- $\mathcal{A}_n \longrightarrow_{\geq} \mathcal{A}_{n+1} = \mathcal{A}_n \;\cup\; \{xR_1y_1, y_1R_2y_2, \ldots, y_{m-1}R_my_m\} \;\cup\; \{y_m \neq z \mid z \in (R_1\circ\ldots\circ R_m)(x) \text{ in } \mathcal{A}_n\}$:
  Aus der Definition der $\geq$-Regel folgt, daß kein y_i in einer Rollenkette von x' nach y' in $\mathcal{A}_{n+1}$ mit $x', y' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y_1, \ldots, y_m\}$ auftritt. Also folgt für Rollenketten von x' nach y' in $\mathcal{A}_{n+1}$ mit $x', y' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y_1, \ldots, y_m\}$ die Behauptung per Induktion. Für y_i und $x' \in \tau_{\mathcal{A}_{n+1}} \setminus \{y_1, \ldots, y_m\}$ ist jede Rollenkette in $\mathcal{A}_{n+1}$ von x' nach y_i von der Form $x'\mathcal{R}xR_1y_1\ldots y_{i-1}R_iy_i$. Mit der Induktionsvoraussetzung und der Definition der Levelfunktion zu $\mathcal{A}_{n+1}$ folgt

  $$\mathsf{level}(y_i) = \mathsf{level}(x') + |\mathcal{R}| + i = \mathsf{level}(x') + |\mathcal{R} \circ R_1\circ\ldots\circ R_i|.$$

  Für $y_i, y_j \in \{y_1, \ldots, y_m\}$ mit $i < j$ existiert nur die $(R_{i+1}\circ\ldots\circ R_j)$-Rollenkette von y_i nach y_j in $\mathcal{A}_{n+1}$. Mit der Definition der Levelfunktion folgt

  $$\mathsf{level}(y_j) = \mathsf{level}(y_i) + |R_{i+1}\circ\ldots\circ R_j|.$$

  Wiederum gibt es keine Rollenketten, die mit einem y_i beginnen, also sind dies alle Fälle.

- $\mathcal{A}_n \longrightarrow_{\leq} \mathcal{A}_{n+1} = \mathcal{A}_n[x/y]$:
  Die $\leq$-Regel werde auf die Assertion $z : (\leq k\,\mathcal{R}) \in \mathcal{A}_n$ und $x, y \in \mathcal{R}(z)$ angewendet. Mit der Induktionsvoraussetzung folgt

  $$\mathsf{level}(x) = \mathsf{level}(y) = \mathsf{level}(z) + |\mathcal{R}|.$$

  Für Rollenketten aus $\mathcal{A}_{n+1}$, die auch in $\mathcal{A}_n$ sind, folgt die Behauptung mit der Induktionsvoraussetzung.
  Rollenketten $x_0R_1x_1\ldots x_{m-1}R_mx_m$ aus $\mathcal{A}_{n+1}$, die nicht in $\mathcal{A}_n$ sind, sind von der Form $x_0\mathcal{R}'y\mathcal{R}''x_m$ mit

  - $x_0\mathcal{R}'x$ und $y\mathcal{R}''x_m$ in $\mathcal{A}_n$ oder
  - $x_0\mathcal{R}'y$ und $x\mathcal{R}''x_m$ in $\mathcal{A}_n$.

  Mit der Induktionsvoraussetzung folgt

  $$\mathsf{level}(x_m) = \mathsf{level}(y) + |\mathcal{R}''| = \mathsf{level}(x_0) + |\mathcal{R}'| + |\mathcal{R}''|.$$

$\square$

Lemma 2.31
Sei $\mathcal{A}_0$ eine zulässige Ausgangs-ABox und $\mathcal{A}$ aus $\mathcal{A}_0$ abgeleitet. Dann gilt für jedes Element $x \in \tau_{\mathcal{A}}$

$$0 \leq \mathsf{level}(x) \leq \mathsf{maxdepth} + |\tau_{\mathcal{A}_0}| \;=:\; m_0.$$

$\lrcorner$

Beweis:
Für ein Individuum $a \in \tau_{\mathcal{A}_0}$ folgt mit Bemerkung 2.28
$$0 \leq \mathsf{level}(a) \leq |\tau_{\mathcal{A}_0}|.$$
Mit Bemerkung 2.29 und Lemma 2.30 folgt, daß jedes Element einen eindeutigen Level besitzt und daß sich der Level eines Elementes aus $\mathcal{A}$ nicht durch weitere Regelanwendungen auf $\mathcal{A}$ ändert. Mit Lemma 2.16 folgt, daß für jede Variable x aus $\mathcal{A}$ ein Individuum $a \in \tau_{\mathcal{A}_0}$ und ein Rollenterm $\mathcal{R}$ existiert mit

- $1 \leq |\mathcal{R}| \leq \mathsf{maxdepth}$,
- $x \in \mathcal{R}(a)$ und
- $\mathsf{level}(x) = \mathsf{level}(a) + |\mathcal{R}|$.

Damit folgt
$$0 \leq \mathsf{level}(x) \leq |\tau_{\mathcal{A}_0}| + \mathsf{maxdepth}.$$
□

Mit Hilfe des Begriffs der zulässigen Ausgangs-ABox und den Aussagen in den Lemmata 2.30 und 2.31 läßt sich der Terminierungsbeweis aus [BS96] auf den Konsistenzalgorithmus übertragen.

Der Terminierungsbeweis

Lemma 2.32
Der Algorithmus in Abbildung 2.3 terminiert, falls die Ausgangs-ABox $\mathcal{A}_0$ zulässig ist. ⌟

Beweis:
Man definiert eine Abbildung κ, die einer ABox $\mathcal{A}$ ein $5(m_0 + 1)$-Tupel über $\mathbb{N}$ zuweist und zeigt folgende Behauptung:
$$\mathcal{A} \longrightarrow \mathcal{A}' \quad \Longrightarrow \quad \kappa(\mathcal{A}) \succ \kappa(\mathcal{A}') \tag{$*$}$$
Dabei bezeichnet $\succ$ die lexikographische Ordnung auf $\mathbb{N}^{5(m_0+1)}$. Aus $(*)$ folgt dann die Behauptung des Lemmas:
Angenommen, der Algorithmus terminiert nicht. Dann existiert eine unendliche Folge von ABoxen mit
$$\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \cdots \text{ und } \kappa(\mathcal{A}_i) \succ \kappa(\mathcal{A}_{i+1}) \text{ für alle } i \geq 0.$$
Damit bilden die $\kappa(\mathcal{A}_i)$ bzgl. $\succ$ eine unendliche absteigende Folge im Widerspruch zur Wohlfundiertheit von $\succ$.

Zur Definition der Abbildung κ: Sei $\mathcal{A}$ eine aus $\mathcal{A}_0$ abgeleitete ABox. Dann sei
$$\kappa(\mathcal{A}) := (\kappa_0, \cdots, \kappa_{m_0}),$$
mit $\kappa_l = (k_{l,1}, k_{l,2}, k_{l,3}, k_{l,4}, k_{l,5}) \in \mathbb{N}^5$ und

- $k_{l,1}$ = Anzahl der Elemente x in $\mathcal{A}$ mit $\mathsf{level}(x) = l$.
- $k_{l,2}$ = Summe der und/oder-Größen $|C|_{\sqcap,\sqcup}$ aller Assertionen $x : C$, auf die $\longrightarrow_\sqcap$ oder $\longrightarrow_\sqcup$ in $\mathcal{A}$ anwendbar ist, mit $\mathsf{level}(x) = l$.
- $k_{l,3}$ steht für die Summe der fehlenden Anwendungen der $\geq$-Regel: zu einer Assertion $x : (\geq n\, R_1 \circ \ldots \circ R_m)$ bezeichne k_x die maximale Größe aller Mengen M von $(R_1 \circ \ldots \circ R_m)$-Nachfolgern von x, für die $x_i \neq x_j \in \mathcal{A}$ für alle Paare verschiedener Elemente x_i, x_j aus M gilt. Dann setze $r_x := n - k_x$, falls $n \geq k_x$ und $r_x := 0$ sonst. $k_{l,3}$ ist gleich der Summe aller r_x zu Assertionen $x : (\geq n\, R_1 \circ \ldots \circ R_m)$ mit $\mathsf{level}(x) = l$.
- $k_{l,4}$ = Anzahl aller Assertionen $x : \exists R.C \in \mathcal{A}$ mit $\mathsf{level}(x) = l$ und die $\exists$-Regel ist auf $x : \exists R.C$ in $\mathcal{A}$ anwendbar.
- $k_{l,5}$ = Anzahl aller Paare $x : (\forall R.C),\ xRy$ aus $\mathcal{A}$ mit $\mathsf{level}(x) = l$ auf die die $\forall$-Regel anwendbar ist.

Zum Beweis der Behauptung $(*)$ zeigt man für jede Vervollständigungsregel aus Abbildung 2.2, daß aus $\mathcal{A} \longrightarrow \mathcal{A}'$ stets $\kappa(\mathcal{A}) \succ \kappa(\mathcal{A}')$ folgt.

1. Konjunktionsregel: $\mathcal{A} \longrightarrow_\sqcap \mathcal{A}'$

Die Regel werde auf $x : C_1 \sqcap C_2$ angewendet und es seien $l = \mathsf{level}(x)$ und κ_l bzw. κ'_l die mit Level l in $\mathcal{A}$ bzw. $\mathcal{A}'$ assoziierten 5-Tupel.
Da die $\sqcap$-Regel weder die Anzahl noch den Level von Elementen ändert, bleiben die ersten Komponenten gleich. Die zweite Komponente von κ'_l ist kleiner als die von κ_l, da $\longrightarrow_\sqcap$ in $\mathcal{A}'$ nicht mehr auf $x : C_1 \sqcap C_2$ anwendbar ist und $|C_1 \sqcap C_2|_{\sqcap,\sqcup} > |C_1|_{\sqcap,\sqcup} + |C_2|_{\sqcap,\sqcup}$ gilt. Da ein lexikographischer Vergleich stattfindet, ist eine Erhöhung der Werte in den letzten drei Komponenten von κ'_l und in Tupeln κ'_m, mit $m > l$ vernachlässigbar. Betrachte also noch Tupel κ_m mit $m < l$. Die ersten drei Komponenten dieser Tupel bleiben stets unverändert, die letzten beiden Werte bleiben gleich oder werden kleiner. Dies ist genau dann möglich, wenn Assertionen $yRx,\ y : (\forall R.C')$ (oder $y : \exists R.C'$) in $\mathcal{A}$ existieren mit $\mathsf{level}(y) = l - 1$ und $C' \in \{C_1, C_2\}$.

2. Disjunktionsregel: $\mathcal{A} \longrightarrow_\sqcup \mathcal{A}'$

Für diese Regel läßt sich die Behauptung wie für die Konjunktionsregel zeigen.

3. Werterestriktion: $\mathcal{A} \longrightarrow_\forall \mathcal{A}'$

Die Regel werde auf $x : \forall R.C,\ xRy$ angewendet und es seien $l = \mathsf{level}(x)$ und κ_l bzw. κ'_l die mit Level l in $\mathcal{A}$ bzw. $\mathcal{A}'$ assoziierten 5-Tupel.
Die ersten drei Komponenten von κ'_l bleiben gegenüber κ_l unverändert, die vierte Komponente bleibt gleich oder wird kleiner (falls $\mathcal{A}$ Assertionen $zR'y$ und $z : \exists R'.C$ enthält) und die fünfte Komponente wird kleiner, da $\longrightarrow_\forall$ nicht mehr auf das Paar $x : \forall R.C,\ xRy$ anwendbar ist.
Mit Lemma 2.30 folgt $\mathsf{level}(y) = \mathsf{level}(x) + 1$. Also vergrößert sich durch die

neue Assertion $y : C$ in $\mathcal{A}'$ höchstens das Tupel κ'_{l+1}. Durch den lexikographischen Vergleich brauchen Änderungen in Tupeln κ'_m mit $m > l$ aber nicht berücksichtigt zu werden.
Alle Tupel κ'_m mit $m < l$ bleiben gegenüber κ_m unverändert, da y nur direkte Vorfahren mit Level l hat.

4. Existenzrestriktion: $\mathcal{A} \longrightarrow_{\exists} \mathcal{A}'$
Die Regel werde auf $x : \exists R.C$ angewendet und es seien xRy und $y : C$ die neuen Assertionen in $\mathcal{A}'$, $l = \mathsf{level}(x)$ und κ_l bzw. κ'_l die mit Level l in $\mathcal{A}$ bzw. $\mathcal{A}'$ assoziierten 5-Tupel.
Offensichtlich bleiben die ersten beiden Komponenten von κ'_l gegenüber κ_l gleich, ebenso die dritte Komponente, da die neue Variable y in keiner Ungleichheitsassertion auftritt. Die vierte Komponente wird kleiner, also ist eine mögliche Erhöhung der fünften Komponente irrelevant und ebenso mögliche Änderungen in Tupeln κ'_m mit $m > l$. Insbesondere folgt wiederum mit Lemma 2.30, daß $\mathsf{level}(y) = \mathsf{level}(x) + 1$ ist, so daß sich höchstens das Tupel κ'_{l+1} gegenüber κ_{l+1} vergrößert.
Tupel κ'_m mit $m < l$ werden durch die Anwendung von $\longrightarrow_{\exists}$ auf Level l nicht geändert. Insbesondere bleibt auch hier jeweils die dritte Komponente gleich, da die neue Variable y in keiner Ungleichheitsassertion auftritt.

5. Zahlenrestriktion: $\mathcal{A} \longrightarrow_{\geq} \mathcal{A}'$
Die Regel werde auf $x : (\geq n\ R_1 \circ \ldots \circ R_m)$ angewendet und es seien $l = \mathsf{level}(x)$, $xR_1y_1,\ y_1R_2y_2,\ \cdots\ , y_{m-1}R_my_m$ die neuen Rollenassertionen in $\mathcal{A}'$ und κ_l bzw. κ'_l die mit Level l in $\mathcal{A}$ bzw. $\mathcal{A}'$ assoziierten 5-Tupel.
Wie bei der $\exists$-Regel bleiben die ersten beiden Komponenten von κ'_l gegenüber κ_l gleich. Die dritte Komponente wird kleiner, da die neue Variable y_m den maximalen Mengen paarweise unterschiedener $(R_1 \circ \ldots \circ R_m)$-Nachfolger von x hinzugefügt wird. Die vierte Komponente bleibt gleich und die mögliche Erhöhung der fünften Komponente sowie Änderungen in Tupeln κ'_m mit $m > l$ sind wiederum irrelevant. Auch hier gilt $\mathsf{level}(y_i) = \mathsf{level}(x) + i$.
Tupel κ'_m mit $m < l$ bleiben entweder unverändert oder die dritte Komponente wird auf Grund der neuen Ungleichheitsassertionen kleiner.

6. Zahlenrestriktion: $\mathcal{A} \longrightarrow_{\leq} \mathcal{A}'$
Die Regel werde auf $x : (\leq\ n\ R_1 \circ \ldots \circ R_m)$ angewendet und es seien $\mathcal{A}' = \mathcal{A}[y1/y2]$, $l = \mathsf{level}(x)$ und κ_l bzw. κ'_l die mit Level l in $\mathcal{A}$ bzw. $\mathcal{A}'$ assoziierten 5-Tupel.
Mit Lemma 2.30 folgt $\mathsf{level}(y_1) = \mathsf{level}(y_2) = \mathsf{level}(x) + m$. Also wird auf Level $l + m$ die erste Komponente des Tupels κ'_{l+m} gegenüber κ_{l+m} kleiner, sodaß Änderungen in den anderen Komponenten oder folgenden Tupeln irrelevant sind.
Die Tupel κ'_r mit $r < l + m$ bleiben gleich oder werden kleiner. Komponenten in κ_{l+m} und κ_{l+m-1} werden eventuell kleiner, weil y_2 in $\mathcal{A}'$ neben seinen Assertionen aus $\mathcal{A}$ auch noch die von y_1 erhält. Da durch das Identifizieren keine

Assertionen entfernt werden, bleiben alle zuvor erfüllten Werte- und Existenzrestriktionen erfüllt. Die dritte Komponente der Tupel κ_r erhöht sich nicht, weil $y_1 \neq y_2$ nicht in $\mathcal{A}$ ist (sonst wären y_1, y_2 nicht identifiziert worden), so daß keine der Mengen M paarweise unterschiedener Nachfolger kleiner wird.

□

Bemerkung 2.33
Der Beweis von Lemma 2.32 liefert offensichtlich auch einen Beweis zu Punkt (4) von Lemma 2.2. Im Falle des Erfüllbarkeitsalgorithmus ist die Ausgangs-ABox $\mathcal{A}_0 = \{x_0 : C_0\}$ offensichtlich zulässig. Die Levelfunktion zu $\mathcal{A}_0$ weist x_0 den Wert 0 zu und der Level einer Variablen $x \in \tau_{\mathcal{A}}$ mit $\mathcal{A}_0 \longrightarrow^* \mathcal{A}$ ist gleich der Länge einer Rollenkette von x_0 nach x in $\mathcal{A}$. ◇

Das Entscheidbarkeitsresultat

Mit den Aussagen in Lemma 2.4 und 2.32 läßt sich nun folgendes Entscheidbarkeitsresultat zeigen:

Theorem 2.34
Für die Klasse der gemäß Definition 2.26 zulässigen $\mathcal{ALCN}(\circ)$-ABoxen ist das Konsistenzproblem entscheidbar. ⌟

Beweis:
Der Beweis von Theorem 2.34 verläuft analog zum Beweis von Theorem 2.1 auf Seite 22. □

Kapitel 3

Die Sprache $\mathcal{ALC} + (\sqsubseteq_n)$

Mit Hilfe von Role-Value-Maps formuliert man für ein Objekt a Bedingungen an die Art der Beziehungen zwischen a und anderen Objekten. Eine Beziehung wird durch einen Rollenterm $\mathcal{R} = R_1 \circ \ldots \circ R_m$ mit $m \geq 1$ und Rollennamen R_i beschrieben.

Role-Value-Maps sind sehr ausdrucksstark. Unterabschnitt 3.1 zeigt, daß für die Sprache $\mathcal{ALC} + (\sqsubseteq)$ das Subsumtionsproblem und damit auch das Erfüllbarkeitsproblem unentscheidbar ist. Für den in Abschnitt 3.1 skizzierten Beweis ist es aber entscheidend, Konzeptterme der Form $(R_1 \circ \ldots \circ R_n \sqsubseteq S_1 \circ \ldots \circ S_m)$ für beliebige Werte $m, n \geq 1$ definieren zu können. Für $\mathcal{ALC} + (\sqsubseteq_n)$ geht dieser Beweis nicht durch, da $\mathcal{ALC} + (\sqsubseteq_n)$ die Levelstruktur-Eigenschaft hat.

In Abschnitt 3.2 wird untersucht, ob das Erfüllbarkeitsproblem und (wegen Abschluß unter Negation) auch das Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq_n)$ entscheidbar ist. Dabei stützt man sich wieder auf ein regelbasiertes Verfahren ab, dessen Vervollständigungsregeln zu einem erfüllbaren $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C eine ABox erzeugen, durch die sich ein kanonisches Modell definieren läßt. Dabei ergeben sich aber wie im Falle des Konsistenzproblems für $\mathcal{ALCN}(\circ)$ Probleme mit der Terminierung, d.h., man erhält im allgemeinen unendliche kanonische Modelle. Ausgehend von solchen speziellen Modellen folgen dann der Nachweis der Endlichen-Modell-Eigenschaft und damit die gewünschten Entscheidbarkeitsresultate für Fragmente von $\mathcal{ALC} + (\sqsubseteq_n)$.

3.1 Das Unentscheidbarkeitsresultat

Der folgende Satz formuliert das Unentscheidbarkeitsresultat für $\mathcal{ALC} + (\sqsubseteq)$.

Theorem 3.1
Für die Sprache $\mathcal{ALC} + (\sqsubseteq)$ ist das Subsumtionsproblem und das Erfüllbarkeitsproblem unentscheidbar. ⌟

Beweisskizze:
Die Beweisidee stammt aus [Sch89] und aus den Ergebnissen dieses Artikels ergibt

sich leicht ein vollständiger Beweis zu Theorem 3.1. Deshalb beschränkt man sich im folgenden auf eine Beweisskizze und zeigt zusätzlich, warum der Beweis für die Teilsprache $\mathcal{ALC} + (\sqsubseteq_n)$ von $\mathcal{ALC} + (\sqsubseteq)$ nicht durchgeht.

In [Sch89] wird bewiesen, daß das Subsumtionsproblem für Konzeptterme unentscheidbar ist, die aufgebaut sind unter Verwendung der folgenden Konstruktoren:

- Konjunktion $C_1 \sqcap C_2$
- Werterestriktion $\forall R.C$
- Role-Value-Maps für Rollen mit Komposition $(R_1 \circ \ldots \circ R_n = S_1 \circ \ldots \circ S_m)$

Dabei steht $(R_1 \circ \ldots \circ R_n = S_1 \circ \ldots \circ S_m)$ als Abkürzung für den Term $(R_1 \circ \ldots \circ R_n \sqsubseteq S_1 \circ \ldots \circ S_m) \sqcap (S_1 \circ \ldots \circ S_m \sqsubseteq R_1 \circ \ldots \circ R_n)$. Bei Konzepttermen, die über diese Konstruktoren aufgebaut sind, handelt es sich also um $\mathcal{ALC} + (\sqsubseteq)$-Konzeptterme.

Man zeigt nun zuerst, daß sich das Wortproblem für Gruppen auf das Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq)$-Konzeptterme reduzieren läßt.

Definition 3.2 (Wortproblem für Gruppen)
Es sei Σ ein endliches Alphabet und $S = \{(u_1, v_1), \ldots, (u_n, v_n)\} \subset \Sigma^+ \times \Sigma^+$ ein Thue-System.
Man definiert die Relation $\underset{S}{\longrightarrow} \subseteq \Sigma^+ \times \Sigma^+$ durch

$$u \underset{S}{\longrightarrow} v \quad \text{gdw} \quad \exists (u_i, v_i) \in S \; \exists x, y \in \Sigma^* : \\ u = x u_i y \text{ und } v = x v_i y.$$

Die Kongruenzrelation $\overset{*}{\underset{S}{\longleftrightarrow}}$ auf Σ^+ ist definiert als der reflexive, transitive, symmetrische Abschluß von $\underset{S}{\longrightarrow}$, d.h.,

$$u \overset{*}{\underset{S}{\longleftrightarrow}} v \quad \text{gdw} \quad u = v \text{ oder es ex. } u_0, \ldots, u_n,\ n \geq 1, \text{ mit} \\ u_0 = u,\ u_n = v \text{ und } u_i \underset{S}{\longrightarrow} u_{i+1} \text{ oder } u_{i+1} \underset{S}{\longrightarrow} u_i$$

Die durch S definierte Halbgruppe sei $H_S := \Sigma^+ / \overset{*}{\underset{S}{\longleftrightarrow}}$ mit

- Trägermenge = Menge der $\overset{*}{\underset{S}{\longleftrightarrow}}$-Äquivalenzklassen
  $[u] = \{v \in \Sigma^+ \mid u \overset{*}{\underset{S}{\longleftrightarrow}} v\}$ und
- assoziativer Operation $\circ : [u] \circ [v] := [uv]$.

Das System S ist eine *Gruppenpräsentation*, wenn H_S eine Gruppe ist, d.h., ein neutrales Element und zu jedem Element ein Inverses existiert. Unter dem *Wortproblem* für S versteht man das folgende Problem:

gegeben: $u, v \in \Sigma^+$

Frage: Gilt $u \overset{*}{\underset{S}{\longleftrightarrow}} v$?

◇

In [Boo59] wird gezeigt, daß es eine endliche Gruppenpräsentation S gibt, für die das Wortproblem unentscheidbar ist. Es sei nun S so ein System mit

- $\Sigma = \{R_1, \dots, R_n\}$
- $S = \{(u_1, v_1), \dots, (u_m, v_m)\}$

sowie $u_0, v_0 \in \Sigma^+$.

Man konstruiert nun Konzeptterme C_S und $C_{u_0v_0}$ so, daß gilt:

$$C_S \sqsubseteq C_{u_0v_0} \quad \Longleftrightarrow \quad u_0 \overset{*}{\underset{S}{\longleftrightarrow}} v_0 \tag{$*$}$$

Da die Konstruktion der Konzeptterme C_S und $C_{u_0v_0}$ effektiv ist, folgt mit der Behauptung $(*)$ das Gewünschte:
Angenommen, das Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq)$ wäre entscheidbar. Dann wäre auch das Wortproblem für die betrachtete endliche Gruppenpräsentation entscheidbar. Dies steht aber im Widerspruch zur Wahl von S als eine endliche Gruppenpräsentation mit unentscheidbarem Wortproblem. Also ist auch das Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq)$ unentscheidbar.

Es bleibt der Beweis der Behauptung $(*)$ und die Definition der Konzeptterme C_S und $C_{u_0v_0}$. Der Beweis von $(*)$ ergibt sich leicht aus den Aussagen und der Vorgehensweise in [Sch89] und liefert für das Folgende keine wesentlichen Erkenntnisse. Deshalb wird hier auf diesen Beweis verzichtet. Da sich aus der Konstruktion der Konzepte C_S und $C_{u_0v_0}$ leicht folgern läßt, warum Theorem 3.1 nicht auch für $\mathcal{ALC} + (\sqsubseteq_n)$ gelten muß, wird im folgenden noch auf diesen Punkt eingegangen.

Zur Definition von C_S und $C_{u_0v_0}$ verwendet man die Elemente $R_1, \dots, R_n$ des Alphabets und eine zusätzliche, „universelle" Rolle R als Rollennamen.
Ein Konzept C definiert eine *universelle Rolle* R, wenn für jedes Modell $\mathcal{I}$ für C und jede Instanz a von C in $\mathcal{I}$ jeder $\mathcal{S}$-Nachfolger von a auch R-Nachfolger von a in $\mathcal{I}$ ist.[1]

[1]In C_S stimmt das nicht ganz, da man nicht $R \circ R = R$ als Subkonzept von C_S hat. Also ist nicht notwendig jeder $R \circ R$-Nachfolger auch R-Nachfolger ist in einem Modell von C_S.

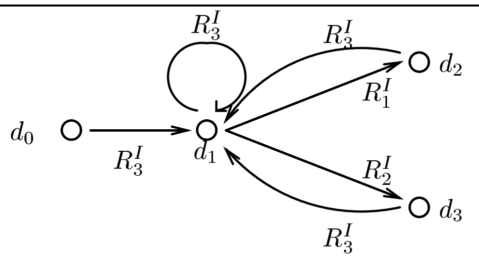
1. $$C_S := \begin{array}{l} (R \circ R_1 = R) \sqcap \ldots \sqcap (R \circ R_n = R) \sqcap \\ \forall R.((u_{11} \circ \ldots \circ u_{1k_1} \;=\; v_{11} \circ \ldots \circ v_{1l_1}) \sqcap \\ \qquad\qquad \vdots \\ (u_{m1} \circ \ldots \circ u_{mk_m} \;=\; v_{m1} \circ \ldots \circ v_{ml_m})) \end{array}$$

2. $C_{u_0 v_0} := \quad \forall R.(u_{01} \circ \ldots \circ u_{0k_0} \;=\; v_{01} \circ \ldots \circ v_{0l_0})$

Die Idee hinter diesen Definitionen ist die folgende:
$u_{j1} \circ \ldots \circ u_{jk_j} = v_{1j} \circ \ldots \circ v_{jl_j}$ sorgt dafür, daß eine $\xrightarrow[S]{}$-Ersetzung simuliert wird. Damit diese einfache Idee durchgeht, benötigt man aber noch die Teilterme $\forall R. \cdots$ und $R \circ R_i = R$, da man ansonsten nur Ersetzungen am Wortanfang simulieren würde.

Beispiel 3.3
Es seien $\Sigma = \{R_1, R_2, R_3\}$ und $S = \{(R_1, R_2)\}$. Dann gilt

- $R_1 R_3 \overset{*}{\underset{S}{\longleftrightarrow}} R_2 R_3$
- $R_3 R_1 \overset{*}{\underset{S}{\longleftrightarrow}} R_3 R_2$
- $(R_1 = R_2) \sqsubseteq (R_1 \circ R_3 = R_2 \circ R_3)$, denn jedes Modell von $R_1 = R_2$ ist auch Modell von $R_1 \circ R_3 = R_2 \circ R_3$:
  Sei $\mathcal{I} = (\Delta_I, \cdot^I)$ Modell von $R_1 = R_2$. Dann gilt für alle $d \in (R_1 = R_2)^I$: $R_1^I(d) = R_2^I(d)$ und es folgt $(R_1 \circ R_3)^I(d) = (R_2 \circ R_3)^I(d)$ und $d \in (R_1 \circ R_3 = R_2 \circ R_3)^I$. Also ist $\mathcal{I}$ auch Modell von $(R_1 \circ R_3 = R_2 \circ R_3)$.
- $(R_1 = R_2) \not\sqsubseteq (R_3 \circ R_1 = R_3 \circ R_2)$, denn für ein $d \in (R_1 = R_2)^I$ gilt $R_1^I(d) = R_2^I(d)$, aber für ein $(d, d') \in R_3^I$ muß $R_1^I(d') = R_2^I(d')$ nicht gelten. Abbildung 3.2 zeigt eine Interpretation $\mathcal{I}$, die Modell von $(R_1 = R_2)$ aber nicht von $(R_3 \circ R_1 = R_3 \circ R_2)$ ist. $\diamond$



Es ist $(R_1 = R_2)^I = \{d_0, d_2, d_3\}$, aber
$d_i \notin (R_3 \circ R_1 = R_3 \circ R_2)^I$, denn
$(R_3 \circ R_1)^I(d_i) = \{d_2\}$ und
$(R_3 \circ R_2)^I(d_i) = \{d_3\}$
für $i = 0, \ldots, 3$

Abbildung 3.1: Gegenbeispiel ohne universelle Rolle

Die Teilterme $(R \circ R_i \;=\; R)$ sorgen nun dafür, daß man ausgehend von einem $d_0 \in C_S^I$ nicht nur den Anfang sondern jede Stelle in der Rollenkette mit einem R^I-Schritt erreichen kann. Durch den $\forall R.$-Teilterm erfüllt dann jeder $(R \circ R_1 \circ \ldots \circ R_k)^I$-Nachfolger d von d_0 die Role-Value-Maps, die genau die Gleichungen aus S beschreiben. Also kann an jeder Stelle in der Rollenkette ein Ersetzungsschritt simuliert werden.

Beispiel 3.4 (Beispiel 3.3 Fortsetzung)
Mit den Voraussetzungen aus Beispiel 3.3 erhält man

$$\begin{array}{lll} C_S & := & (R \circ R_1 = R) \sqcap (R \circ R_2) \sqcap (R \circ R_3) \sqcap \\ & & \forall R.(R_1 = R_2)\ , \\ C_{u_0 v_0} & := & \forall R.(R_3 \circ R_1 = R_3 \circ R_2) \end{array}$$

und in jedem Modell $\mathcal{I}$ von C_S gilt für $d_0 \in C_S^I$: ist $(d_0, d') \in (R \circ R_3)^I$, so ist $(d_0, d') \in R^I$ und damit gilt $R_1^I(d') = R_2^I(d')$. Abbildung 3.2 verdeutlicht diese Situation. ◇

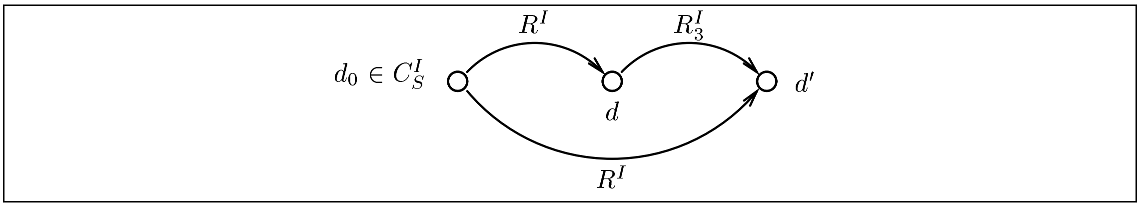


Abbildung 3.2: Modelle mit universeller Rolle

Die Beispiele 3.3 und 3.4 zeigen, daß man bei der oben skizzierten Vorgehensweise zur Reduktion des Wortproblems auf das Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq)$ in der Lage sein muß, unterschiedlich lange Rollenketten von einem Individuum zu einem anderen zu erzwingen. In $\mathcal{ALC} + (\sqsubseteq)$ ist dies offensichtlich möglich, in $\mathcal{ALC} + (\sqsubseteq_n)$ aber nicht. Die Konzepte C_S und $C_{u_0 v_0}$ sind keine $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepte, also läßt sich der soeben skizzierte Beweis von Theorem 3.1 nicht auf $\mathcal{ALC} + (\sqsubseteq_n)$ übertragen. Es wird sich sogar zeigen, daß $\mathcal{ALC} + (\sqsubseteq_n)$ wie $\mathcal{ALCN}(\circ)$ die *Levelstruktur-Eigenschaft* besitzt. Also hat man gar nicht die Möglichkeit, unter Verwendung der in $\mathcal{ALC} + (\sqsubseteq_n)$ erlaubten Konstruktoren Konzepte zu definieren, die unterschiedlich lange Pfade zwischen einer Instanz und einem Nachfolger erzwingen.

3.2 Das Erfüllbarkeitsproblem für $\mathcal{ALC} + (\sqsubseteq_n)$

In diesem Abschnitt wird das Erfüllbarkeitsproblem für $\mathcal{ALC} + (\sqsubseteq_n)$ untersucht. Man formuliert ein Vervollständigungsverfahren, das für ein $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 in NNF ausgehend von $\{x_0 : C_0\}$ versucht, eine vollständige und offene ABox $\mathcal{A}$ zu erzeugen, aus der sich dann ein kanonisches Modell zu C_0 konstruieren läßt.

3.2.1 Der Vervollständigungsalgorithmus für $\mathcal{ALC} + (\sqsubseteq_n)$

Man greift auf den Erfüllbarkeitsalgorithmus in Abbildung 2.1 zurück, wobei man nun statt $\mathcal{ALCN}(\circ)$-Konzepte $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepte als Eingabe erwartet. Abbildung 3.3 faßt die Vervollständigungsregeln für $\mathcal{ALC} + (\sqsubseteq_n)$-ABoxen zusammen.

Gegenüber den Vervollständigungsregeln aus Abbildung 2.2 ergeben sich natürlich einige Änderungen. Die Vervollständigungsregeln für die Konstruktoren $\geq$ und $\leq$

1. **Konjunktion:**
   falls $x : (C_1 \sqcap C_2) \in \mathcal{A}$ und $x : C_1 \notin \mathcal{A}$ oder $x : C_2 \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\sqcap} \mathcal{A} \cup \{x : C_1, x : C_2\}$

2. **Disjunktion:**
   falls $x : (C_1 \sqcup C_2) \in \mathcal{A}$ und $x : C_1 \notin \mathcal{A}$ und $x : C_2 \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\sqcup} \mathcal{A}_1 = \mathcal{A} \cup \{x : C_1\}$
   $\mathcal{A} \longrightarrow_{\sqcup} \mathcal{A}_2 = \mathcal{A} \cup \{x : C_2\}$

3. **Werterestriktion:**
   falls $x : (\forall R.C) \in \mathcal{A}$ für ein $R \in \mathcal{N_R}$, $xRy \in \mathcal{A}$ und $y : C \notin \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\forall} \mathcal{A} \cup \{y : C\}$

4. **Existenzrestriktion:**
   falls $x : (\exists R.C) \in \mathcal{A}$ für ein $R \in \mathcal{N_R}$ und es gibt kein y mit $xRy \in \mathcal{A}$ und $y : C \in \mathcal{A}$, dann
   $\mathcal{A} \longrightarrow_{\exists} \mathcal{A} \cup \{xRz, z : C\}$
   für eine neue Variable $z \in \tau \setminus \tau_{\mathcal{A}}$

5. **Positive Role-Value-Maps:**
   falls $x : (R_1 \circ \ldots \circ R_m \sqsubseteq S_1 \circ \ldots \circ S_m) \in \mathcal{A}$ und es ex. ein y in $\mathcal{A}$ mit $y \in (R_1 \circ \ldots \circ R_m)(x)$ und $y \notin (S_1 \circ \ldots \circ S_m)(x)$, dann
   $\mathcal{A} \longrightarrow_{\sqsubseteq} \mathcal{A} \cup \{xS_1y_2, y_{m-1}S_my\} \cup \{y_iS_iy_{i+1} \mid 2 \leq i < m-1\}$
   für neue Variablen $y_2, \ldots, y_{m-1}$ aus $\tau \setminus \tau_{\mathcal{A}}$

6. **Negative Role-Value-Maps:**
   falls $x : \neg(R_1 \circ \ldots \circ R_m \sqsubseteq S_1 \circ \ldots \circ S_m) \in \mathcal{A}$ und es ex. kein y in $\mathcal{A}$ mit $y \in (R_1 \circ \ldots \circ R_m)(x)$ und $\neg(x\,(S_1 \circ \ldots \circ S_m)\,y) \in \mathcal{A}$, dann
   $$\begin{aligned} \mathcal{A} \longrightarrow_{\not\sqsubseteq} \mathcal{A} \quad & \cup \{xR_1y_2\} \cup \{y_iR_iy_{i+1} \mid 2 \leq i \leq m-1\} \\ & \cup \{\neg(x\,(S_1 \circ \ldots \circ S_m)\,y_m)\} \end{aligned}$$
   für neue Variablen $y_2, \ldots, y_m$ aus $\tau \setminus \tau_{\mathcal{A}}$

Abbildung 3.3: Vervollständigungsregeln für $\mathcal{ALC} + (\sqsubseteq_n)$

werden ersetzt durch Regeln zur Behandlung von Konzepten der Form $(\mathcal{R} \sqsubseteq \mathcal{S})$ und $\neg(\mathcal{R} \sqsubseteq \mathcal{S})$.

Bemerkung 3.5 (zur Idee hinter den Regeln $\longrightarrow_{\sqsubseteq}$ und $\longrightarrow_{\not\sqsubseteq}$)
Um eine Assertion der Form $x : (\mathcal{R} \sqsubseteq \mathcal{S}) \in \mathcal{A}$ zu erfüllen, muß offensichtlich jeder $\mathcal{R}$-Nachfolger von x auch ein $\mathcal{S}$-Nachfolger von x sein, d.h., hat x einen $\mathcal{R}$-Nachfolger y in $\mathcal{A}$, der noch kein $\mathcal{S}$-Nachfolger von x ist, so erzeugt man eine $\mathcal{S}$-Rollenkette von x nach y, so daß y zum $\mathcal{S}$-Nachfolger von x wird.

Um eine Assertion der Form $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S}) \in \mathcal{A}$ zu erfüllen, benötigt man einen expliziten $\mathcal{R}$-Nachfolger y von x, der kein $\mathcal{S}$-Nachfolger von x werden darf. Dazu führt man einen neuen $\mathcal{R}$-Nachfolger y von x zusammen mit einer Assertion $\neg(x\,\mathcal{S}\,y)$ ein. Wird durch weitere Regelanwendungen y zum $\mathcal{S}$-Nachfolger von x, so ist $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ nicht mehr erfüllt. Da man aber einen neuen $\mathcal{R}$-Nachfolger eingeführt hat, würde nun auch jeder weitere, neu erzeugte $\mathcal{R}$-Nachfolger von x zum $\mathcal{S}$-Nachfolger, so daß die Assertion $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ nie erfüllt werden kann. Wird also ein einmal eingeführter Zeuge y für eine Assertion $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ zum $\mathcal{S}$-Nachfolger von x, so liefert die $\mathcal{S}$-Rollenkette von x nach y zusammen mit der Assertion $\neg(x\,\mathcal{S}\,y)$ einen clash und die ABox ist inkonsistent. $\diamond$

Beispiel 3.6
Das Beispiel in Abbildung 3.4 zeigt, daß eine Variable x, zu der eine Assertion $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ existiert, auch $\mathcal{R}$-Nachfolger haben kann, die zugleich $\mathcal{S}$-Nachfolger von x sind. Das Beispiel zeigt außerdem, daß man nicht bereits existierende $\mathcal{R}$-Nachfolger von x als Zeugen für $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ wiederverwenden kann. Nachdem x_2 $(R \circ R)$- und $(R \circ S)$-Nachfolger von x_0 ist, muß man einen weiteren $(R \circ R)$-Nachfolger von x_0 erzeugen, damit die kanonische Interpretation die Assertion $x_0 : \neg(R \circ R \sqsubseteq R \circ S)$ erfüllt. $\diamond$

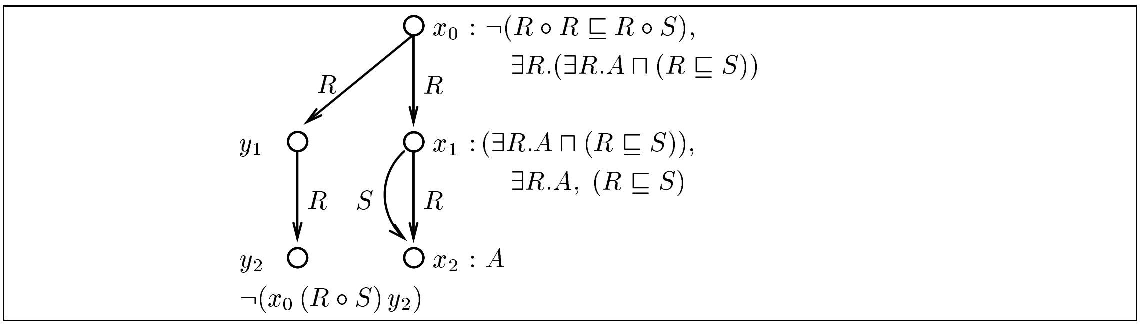


Abbildung 3.4: Zur Idee hinter der $\not\sqsubseteq$-Regel

Wie für $\mathcal{ALCN}(\circ)$ ergibt sich aus der Definition der Vervollständigungsregeln, daß jede ABox $\mathcal{A}$, die mit diesen Regeln aus $\mathcal{A}_0 = \{x_0 : C_0\}$ abgeleitet wurde, eine *Levelstruktur* besitzt, denn:

1. Jede Variable $x \in \tau_{\mathcal{A}}$, $x \neq x_0$, ist $(R_1 \circ \cdots \circ R_m)$-Nachfolger von x_0 für einen Rollenterm der Länge $m \geq 1$ und jede Rollenkette, die x_0 und x verbindet,

hat Länge m.
Denn: Eine Variable $x \neq x_0$ wird durch anwenden einer generierenden Regel erzeugt und ist offensichtlich ein Nachfolger von x_0. Für Assertionen $x : (\mathcal{R} \sqsubseteq \mathcal{S})$ in $\mathcal{A}$ gilt $|\mathcal{R}| = |\mathcal{S}|$. Durch anwenden der $\sqsubseteq$-Regel auf $x : (\mathcal{R} \sqsubseteq \mathcal{S})$ und $y \in \mathcal{R}(x)$ erzeugt man also eine weitere Rollenkette von x nach y, die die gleiche Länge hat wie die bereits existierenden Rollenketten von x nach y. Für einen Nachfahren y' von y entstehen durch die $\longrightarrow_{\sqsubseteq}$-Anwendung ebenfalls neue Rollenketten von x_0 nach y'. Diese haben dann aber ebenfalls die gleiche Länge wie bereits existierende Rollenketten von x_0 nach y'.

2. Ist x von x_0 aus über Rollenketten der Länge m erreichbar, so gilt für jede Assertion $x : C$ in $\mathcal{A}$, daß C maximal Rollentiefe $m_0 - m$ hat mit $m_0 = \mathsf{depth}(C_0)$. Damit ist m beschränkt durch m_0.

Beachte: Ausgehend von einem $\mathcal{ALCN}(\circ)$-Konzept C_0 in NNF gilt für eine ABox $\mathcal{A}$, die aus $\{x_0 : C_0\}$ mit den Vervollständigungsregeln aus Abbildung 2.2 abgeleitet wurde, daß alle Rollenketten von x_0 zu einer Variablen x in $\mathcal{A}$ von der Form $x_0 R_1 x_1 R_1 x_2 \cdots x_{m-1} R_m x$ für festes $m \in \{0, \ldots, m_0\}$ und feste Rollennamen $R_i \in \mathcal{N}_{\mathcal{R}}$ sind. Es kann allerdings verschiedene mögliche Zwischenindividuen $x_1, \ldots, x_{m-1}$ geben. In den $\mathcal{ALC} + (\sqsubseteq_n)$-ABoxen, die mit den Regeln aus Abbildung 3.3 aus $\{x_0 : C_0\}$ abgeleitet wurden, gilt dies nicht mehr. Zwischen x_0 und x können verschiedene Rollenketten existieren, d.h., ein x kann $\mathcal{R}$- und $\mathcal{S}$-Nachfolger von x_0 sein mit $\mathcal{R} \neq \mathcal{S}$.

Dennoch folgt aus dem ersten Punkt, daß jede Variable $x \in \tau_{\mathcal{A}}$ einen eindeutigen Level $\mathsf{level}(x)$ besitzt. Aus dem zweiten Punkt folgt, daß dieser Level stets zwischen 0 und m_0 liegt.

Zur Terminierung der iterierten Regelanwendung

Da man an einem Entscheidungsverfahren interessiert ist, untersucht man für den Erfüllbarkeitsalgorithmus wiederum Korrektheit, Vollständigkeit und Terminierung. Es gibt $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepte C_0 in NNF, so daß ausgehend von $\mathcal{A}_0 := \{x_0 : C_0\}$ unendliche Folgen von Regelanwendungen möglich sind. Beispiel 3.7 enthält zwei solcher Konzepte.

Beispiel 3.7
Es sei

$$
\begin{aligned}
C_1 &:= \exists R.A \sqcap \forall R.\exists R.A \sqcap (R \circ R \sqsubseteq R \circ S) \text{ und} \\
C_2 &:= \neg(R \circ R \circ R \circ R \sqsubseteq S \circ S \circ S \circ S) \sqcap \\
&\quad\ \forall R.(R \circ S \circ R \sqsubseteq R \circ R \circ R) \sqcap \\
&\quad\ (R \circ R \circ R \sqsubseteq R \circ R \circ S)
\end{aligned}
$$

Ausgehend von $\{x_0 : C_1\}$ erzeugt der Algorithmus nun durch unendliche Regelanwendung eine ABox mit einer unendlichen Verzweigung. Zunächst wird C_1 durch $\longrightarrow_{\sqcap}$-Anwendungen in Teilkonzepte aufgebrochen, dann wird ein $(R \circ R)$-Nachfolger

x_2 von x_0 erzeugt. Wegen $x_0 : (R \circ R \sqsubseteq R \circ S)$ wird eine $(R \circ S)$-Rollenkette von x_0 nach x_2 erzeugt und damit entsteht insbesondere ein neuer R-Nachfolger x_3 für x_0. Zu diesem ist wegen $x_0 : \forall R.\exists R.A$ ein R-Nachfolger x_4 zu generieren. x_4 ist außerdem $(R \circ R)$-Nachfolger von x_0, also muß wiederum eine $(R \circ S)$-Rollenkette von x_0 nach x_4 generiert werden und so weiter. Abbildung 3.5 verdeutlicht das Ergebnis dieser unendlichen Folge von Regelanwendungen. Man beachte, daß, wenn man den Teilterm $\forall R.\exists R.A$ durch den Term $\forall R.\neg(R \sqsubseteq S)$ ersetzt, ebenfalls eine unendliche Folge von Regelanwendungen entsteht, bei der man durch unendlich-maliges Anwenden der $\not\sqsubseteq$-Regel unendlich viele $(R \circ R)$-Nachfolger von x_0 erzeugt.

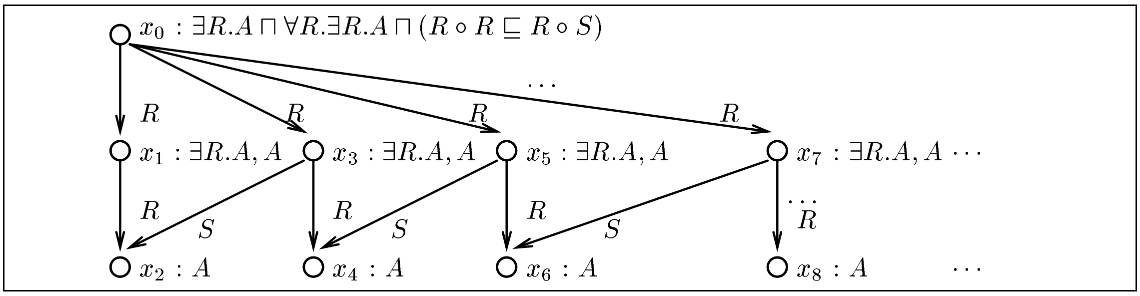


Abbildung 3.5: Erstes Beispiel für die Nichtterminierung

Bei der Untersuchung des Konzeptterms C_2 terminiert der Algorithmus ebenfalls nicht und erzeugt eine unendliche ABox. Das Ergebnis ist in Abbildung 3.6 dargestellt. Interessant an diesem Beispiel ist, daß unendlich viele $(R \circ R \circ R)$-Nachfolger von x_0 ohne Anwendung der $\exists$- oder $\not\sqsubseteq$-Regel auf Nachfolger von x_0 entstehen. Damit kann also jede der generierenden Regeln 4., 5. und 6. dafür verantwortlich sein, daß immer wieder ein neuer $\mathcal{R}$-Nachfolger einer Variable x erzeugt wird, für den wiederum eine $\mathcal{S}$-Rollenkette wegen $x : \mathcal{R} \sqsubseteq \mathcal{S}$ zu generieren ist. $\diamond$

3.2.2 Der Beweis über die Endliche-Modell-Eigenschaft

Man hat also ähnlich wie in Abschnitt 2.2 das Problem, daß der Algorithmus nicht bei jeder iterierten Regelanwendung terminiert. Folglich stellt der Algorithmus mit den Vervollständigungsregeln aus Abbildung 3.3 kein Verfahren dar, daß das Erfüllbarkeitsproblem für $\mathcal{ALC} + (\sqsubseteq_n)$ entscheidet.

Es läßt sich aber zeigen, daß der Erfüllbarkeitsalgorithmus unter Berücksichtigung einer *Fairneßannahme* widerlegungsvollständig ist, d.h., bei Eingabe eines Konzeptes C ein ggf. unendliches, kanonisches Modell erzeugt, falls C erfüllbar ist und sonst mit Ausgabe „C ist unerfüllbar." terminiert.

Aus der Existenz eines widerlegungsvollständigen Verfahrens und dem Nachweis der Endlichen-Modell-Eigenschaft für $\mathcal{ALC} + (\sqsubseteq_n)$ würde die Entscheidbarkeit des Erfüllbarkeitsproblems folgen.

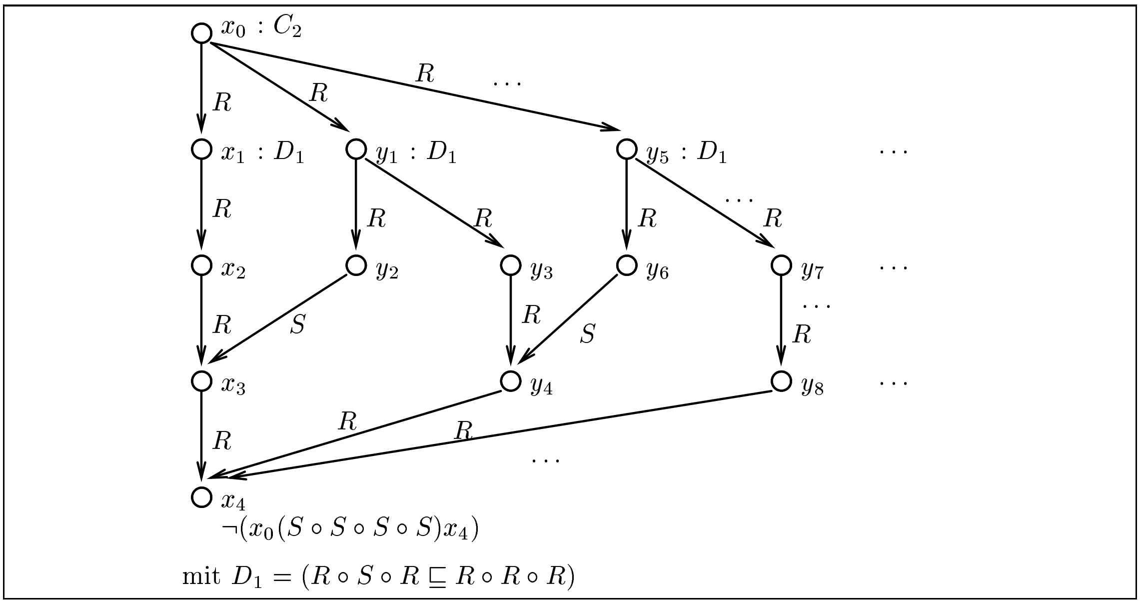


Abbildung 3.6: Zweites Beispiel für die Nichtterminierung

Leider konnte die Endliche-Modell-Eigenschaft für $\mathcal{ALC} + (\sqsubseteq_n)$ nicht nachgewiesen werden. Aus dem Beweisversuch ergibt sich aber für bestimmte Fragmente von $\mathcal{ALC} + (\sqsubseteq_n)$ der Nachweis der Endlichen-Modell-Eigenschaft, so daß man für diese Fragmente das Erfüllbarkeitsproblem entscheiden kann.

Zum Beweis der Widerlegungsvollständigkeit benötigt man noch den Begriff der *fairen Regelanwendung*:

Definition 3.8
Sei C ein $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept und C_0 das zu C äquivalente Konzept in NNF. Dann heißt die iterierte Regelanwendung durch den Vervollständigungsalgorithmus auf $\mathcal{A}_0 = \{x_0 : C_0\}$ *fair*, wenn für jeden Pfad $\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \cdots$ im Vervollständigungsbaum gilt:

- Ist $\longrightarrow \in \{\longrightarrow_\sqcup, \longrightarrow_\sqcap, \longrightarrow_\exists, \longrightarrow_{\not\sqsubseteq}\}$ auf $x : C$ in $\mathcal{A}_i$ anwendbar, so existiert ein $j > i$, so daß $\longrightarrow$ in $\mathcal{A}_j$ nicht mehr auf $x : C$ anwendbar ist.
- Ist $\longrightarrow_\forall$ auf $x : \forall R.C$, xRy in $\mathcal{A}_i$ anwendbar, so existiert ein $j > i$, so daß $\longrightarrow_\forall$ in $\mathcal{A}_j$ nicht mehr auf $x : \forall R.C$, xRy anwendbar ist.
- Ist $\longrightarrow_\sqsubseteq$ auf $x : (\mathcal{R} \sqsubseteq \mathcal{S})$, $x\,\mathcal{R}\,y$ in $\mathcal{A}_i$ anwendbar, so existiert ein $j > i$, so daß $\longrightarrow_\sqsubseteq$ in $\mathcal{A}_j$ nicht mehr auf $x : (\mathcal{R} \sqsubseteq \mathcal{S})$, $x\,\mathcal{R}\,y$ anwendbar ist. $\diamond$

Beispiel 3.9
Die Fairneßannahme ist notwendig, um ein widerlegungsvollständiges Verfahren zu erhalten, denn:
Ohne Berücksichtigung der Fairneßannahme ist für das Konzept

$$\begin{aligned} C_0 \;=\; & \big(A \sqcap \neg A\big) \sqcap \\ & \big(\exists R.A \sqcap \forall R.\exists R.A \sqcap (R \circ R \sqsubseteq R \circ S)\big) \end{aligned}$$

ausgehend von $\mathcal{A}_0 = \{x_0 : C_0\}$, wie in Beispiel 3.7 gezeigt, eine unendliche Folge von Regelanwendungen für das zweite Konjunkt möglich, ohne daß der offensichtliche Widerspruch im ersten Konjunkt erkannt wird. Bei einer fairen Regelanwendung ist aber sichergestellt, daß in jedem Pfad des Vervollständigungsbaumes irgendwann die $\sqcap$-Regel auf die Assertion $x_0 : A \sqcap \neg A$ angewendet wird, so daß jeder Pfad nach endlich vielen Schritten mit einer geschlossenen Blattbeschriftung endet. $\diamond$

Bemerkung 3.10
Man beachte, daß eine faire Regelanwendung stets möglich ist.
Auf $\mathcal{A}_0 = \{x_0 : C_0\}$ ist höchstens eine Regel anwendbar und bei jeder Regelanwendung entstehen nur endlich viele Assertionen bzw. Paare von Assertionen, auf die Regeln angewendet werden können. Dann läßt sich die Regelanwendung offensichtlich so steuern (z.B. durch eine FiFo-Abarbeitung der Situationen), daß auf jede dieser Situationen die entsprechende Regel nach endlich vielen Schritten angewendet wird. $\diamond$

Zur Korrektheit und Widerlegungsvollständigkeit des Verfahrens

Lemma 3.11
Seien C_0 ein $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept in NNF und $\mathcal{A}$ eine ABox, die mit den Vervollständigungsregeln aus $\{x_0 : C_0\}$ abgeleitet wurde. Dann gilt:

1. Ist $\longrightarrow$ auf $\mathcal{A}$ anwendbar, so gilt: $\mathcal{A}$ ist konsistent genau dann, wenn ein $\mathcal{A}'$ existiert mit $\mathcal{A} \longrightarrow \mathcal{A}'$ und $\mathcal{A}'$ ist konsistent.

2. Ist $\mathcal{A}$ geschlossen, so ist $\mathcal{A}$ inkonsistent.

3. Werden die Regeln fair angewendet, so ist der Algorithmus widerlegungsvollständig, d.h., ist C_0 unerfüllbar, so terminiert der Algorithmus nach endlich vielen Regelanwendungen mit Ausgabe „C_0 ist unerfüllbar.", sonst erzeugt er ein ggf. unendliches Modell zu C_0. $\lrcorner$

Beweis:
zu Punkt (1)**:** Auf $\mathcal{A}$ sei eine Vervollständigungsregel aus Abbildung 3.3 anwendbar. Für eine Regel $\longrightarrow$ aus $\{\longrightarrow_\sqcap, \longrightarrow_\sqcup, \longrightarrow_\forall, \longrightarrow_\exists\}$ folgt die Behauptung wie im Beweis von Lemma 2.4. Es bleibt der Beweis der Behauptung für die Regeln $\longrightarrow_\sqsubseteq$ und $\longrightarrow_{\not\sqsubseteq}$.

Positive Role-Value-Maps: $\mathcal{A} \longrightarrow_\sqsubseteq \mathcal{A}'$

$\longrightarrow_\sqsubseteq$ wurde auf

$x : (R_1 \circ \ldots \circ R_m \sqsubseteq S_1 \circ \ldots \circ S_m)$ und

$xR_1x_1,\ x_1R_2x_2, \ldots, x_{m-1}R_mx_m$

in $\mathcal{A}$ angewendet und es sei

$\mathcal{A}' = \mathcal{A} \cup \{xS_1y_1, y_1S_2y_2, \dots, y_{m-1}S_mx_m\}$.

Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell von $\mathcal{A}$.
Sei also umgekehrt $\mathcal{I}$ ein Modell zu $\mathcal{A}$. Dann läßt sich $\mathcal{I}$ wie folgt zu einem Modell von $\mathcal{A}'$ erweitern: $\mathcal{I}$ erfüllt jede Assertion aus $\mathcal{A}$, also gilt

$(x^I, x_1^I) \in R_1^I$ und $(x_i^I, x_{i+1}^I) \in R_{i+1}^I$ für $1 \leq i \leq m-1$ und
$x^I \in (R_1 \circ \dots \circ R_m \sqsubseteq S_1 \circ \dots \circ S_m)^I$.

Also existieren Elemente $b_1, \dots, b_{m-1} \in \Delta_I$ mit

$(x^I, b_1) \in S_1^I$, $(b_i, b_{i+1}) \in S_{i+1}^I$ für $1 \leq i \leq m-2$ und $(b_{m-1}, x_m^I) \in S_m^I$.

Man definiert nun $\mathcal{I}'$ wie $\mathcal{I}$ bis auf

$y_i^{I'} := b_i$, $1 \leq i \leq m-1$.

Damit erfüllt $\mathcal{I}'$ jede Assertion aus $\mathcal{A}'$, ist also ein Modell von $\mathcal{A}'$.

Negative Role-Value-Maps: $\mathcal{A} \longrightarrow_{\not\sqsubseteq} \mathcal{A}'$

$\longrightarrow_{\not\sqsubseteq}$ wurde auf

$x : \neg(R_1 \circ \dots \circ R_m \sqsubseteq S_1 \circ \dots \circ S_m)$

in $\mathcal{A}$ angewendet und es sei

$\mathcal{A}' = \mathcal{A} \cup \{xR_1y_1, y_1R_2y_2, \dots, y_{m-1}R_my_m, \neg(x\,(S_1 \circ \dots \circ S_m)\,y_m)\}$.

Wegen $\mathcal{A} \subset \mathcal{A}'$ ist jedes Modell $\mathcal{I}$ von $\mathcal{A}'$ auch Modell von $\mathcal{A}$.
Sei also umgekehrt $\mathcal{I}$ ein Modell zu $\mathcal{A}$. Dann läßt sich $\mathcal{I}$ wie folgt zu einem Modell von $\mathcal{A}'$ erweitern:
$\mathcal{I}$ erfüllt jede Assertion aus $\mathcal{A}$, also gilt

$$x^I \in (\neg(R_1 \circ \dots \circ R_m \sqsubseteq S_1 \circ \dots \circ S_m))^I,$$

d.h., es existieren Elemente $b_1, \dots, b_m \in \Delta_I$ mit

$(x^I, b_1) \in R_1^I$ und $(b_i, b_{i+1}) \in R_{i+1}^I$ für $1 \leq i \leq m-1$ und
$(x^I, b_m) \notin (S_1 \circ \dots \circ S_m)^I$.

Man definiert nun $\mathcal{I}'$ wie $\mathcal{I}$ bis auf

$y_i^{I'} := b_i$, $1 \leq i \leq m-1$.

Damit erfüllt $\mathcal{I}'$ jede Assertion aus $\mathcal{A}'$, ist also Modell von $\mathcal{A}'$.

zu Punkt (2)**:** Ist $\mathcal{A}$ geschlossen, so enthält $\mathcal{A}$ einen clash der Form

- $\{x : A, x : \neg A\} \subseteq \mathcal{A}$ oder

- $\neg(x\mathcal{R}y) \in \mathcal{A}$ und y ist $\mathcal{R}$-Nachfolger von x in $\mathcal{A}$.

Damit existiert offensichtlich keine Interpretation $\mathcal{I}$, die Modell von $\mathcal{A}$ ist, d.h., die jede Assertion aus $\mathcal{A}$ erfüllt.

zu Punkt (3)**:** Zum Beweis der Behauptung zeigt man: Terminiert der Algorithmus bei fairer Regelanwendung nicht nach endlich vielen Schritten mit Ausgabe „C_0 ist unerfüllbar", so ist C_0 erfüllbar.

Aus der Definition der Vervollständigungsregeln folgt, daß der Vervollständigungsbaum Verzweigungsgrad ≤ 2 hat, da jede Regel außer der Disjunktionsregel einen Nachfolgerknoten im Baum liefert und die Disjunktionsregel zwei.

Angenommen, der Algorithmus terminiert nicht nach endlich vielen Schritten mit Ausgabe „C_0 ist unerfüllbar". Dann existiert ein Pfad

$$\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \cdots$$

im Vervollständigungsbaum mit:

$$\mathcal{A}_i \text{ ist offen und } \mathcal{A}_i \subset \mathcal{A}_{i+1} \text{ für alle } i \geq 0.$$

Terminiert der Algorithmus nach endlich vielen Schritten mit Ausgabe „C_0 ist erfüllbar.", so ist offensichtlich jeder solche Pfad endlich. Terminiert der Algorithmus nicht nach endlich vielen Schritten, so folgt mit dem Lemma von König, daß ein solcher Pfad existiert, der unendlich ist (da der Vervollständigungsbaum Verzweigungszahl 2 hat.)

Sei $\mathcal{A}_\infty := \bigcup_{i \geq 0} \mathcal{A}_i$ die auf diesem Pfad erzeugte, offene ABox. Man zeigt nun, daß die durch $\mathcal{A}_\infty$ induzierte kanonische Interpretation Modell von $\mathcal{A}_\infty$ ist.
Man definiert die kanonische Interpretation $\mathcal{I}_\infty = (\triangle_{\mathcal{I}_\infty}, \cdot^{\mathcal{I}_\infty})$ von $\mathcal{A}_\infty$ durch

- $\triangle_{\mathcal{I}_\infty} := \tau_{\mathcal{A}_\infty}$,
- $A^{\mathcal{I}_\infty} := \{x \in \triangle_{\mathcal{I}_\infty} \mid x : A \in \mathcal{A}_\infty\}$ für $A \in \mathcal{N}_\mathcal{C}$ und
- $R^{\mathcal{I}_\infty} := \{(x, y) \mid xRy \in \mathcal{A}_\infty\}$ für $R \in \mathcal{N}_\mathcal{R}$.

Unter der zusätzlichen Voraussetzung, daß die Regeln fair angewendet werden, gilt nun folgende Behauptung:

Behauptung: Die kanonische Interpretation $\mathcal{I}_\infty$ von $\mathcal{A}_\infty$ ist ein Modell von C_0, d.h., es gilt $C_0^{\mathcal{I}_\infty} \neq \emptyset$.

Zum Beweis dieser Behauptung zeigt man durch Induktion über den Aufbau von $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepttermen:

$$x : C \in \mathcal{A}_\infty \implies x \in C^{\mathcal{I}_\infty}.$$

Wegen $x_0 : C_0 \in \mathcal{A}_\infty$ folgt dann die Behauptung. Sei also $x : C \in \mathcal{A}_\infty$.

- $C = A \in \mathcal{N_C}$:
  Aus der Definition von $\mathcal{I}_\infty$ folgt $x \in A^{\mathcal{I}_\infty}$.

- $C = \neg A$ für $A \in \mathcal{N_C}$:
  Da $\mathcal{A}_\infty$ offen ist, ist $x : A$ nicht in $\mathcal{A}_\infty$. Aus der Definition von $\mathcal{I}_\infty$ folgt $x \notin A^{\mathcal{I}_\infty}$, also $x \in (\neg A)^{\mathcal{I}_\infty}$.

- $C = C_1 \sqcap C_2$:
  Wegen $x : C \in \mathcal{A}_\infty = \bigcup_{i \geq 0} \mathcal{A}_i$ existiert ein i mit $x : C_1 \sqcap C_2 \in \mathcal{A}_i$. Aus der Fairneßannahme folgt, daß dann auch ein $j \geq i$ existiert, so daß $\longrightarrow_\sqcap$ nicht mehr auf $x : C_1 \sqcap C_2$ in $\mathcal{A}_j \supseteq \mathcal{A}_i$ angewendet werden kann, d.h., es gilt $\{x : C_1, x : C_2\} \in \mathcal{A}_j$. Also folgt $\{x : C_1, x : C_2\} \in \mathcal{A}_\infty$. Per Induktion gilt $x \in C_1^{\mathcal{I}_\infty}$ und $x \in C_2^{\mathcal{I}_\infty}$, also $x \in (C_1 \sqcap C_2)^{\mathcal{I}_\infty}$.

- $C = C_1 \sqcup C_2$:
  Analog zum Fall $C = C_1 \sqcap C_2$ folgt $x \in (C_1 \sqcup C_2)^{\mathcal{I}_\infty}$.

- $C = \exists R.C'$:
  Wie oben folgt, daß ein Index j existiert, so daß $\longrightarrow_\exists$ in $\mathcal{A}_j$ nicht mehr auf $x : \exists R.C'$ anwendbar ist. Also existiert ein y mit $\{xRy, y : C'\} \subseteq \mathcal{A}_j \subseteq \mathcal{A}_\infty$. Per Induktion folgt $y \in C'^{\mathcal{I}_\infty}$ und damit $x \in (\exists R.C')^{\mathcal{I}_\infty}$.

- $C = \forall R.C'$:
  Sei $(x, y) \in R^{\mathcal{I}_\infty}$ beliebig. Dann ist $xRy \in \mathcal{A}_\infty$. Es folgt wiederum wie oben, daß ein Index j existiert, so daß $\longrightarrow_\forall$ in $\mathcal{A}_j$ nicht mehr auf das Paar $x : \forall R.C'$, xRy anwendbar ist. Also ist $y : C' \in \mathcal{A}_j \subseteq \mathcal{A}_\infty$. Per Induktion folgt $y \in C'^{\mathcal{I}_\infty}$. Da y als beliebiger R-Nachfolger von x in $\mathcal{I}_\infty$ gewählt war, folgt $x \in (\forall R.C')^{\mathcal{I}_\infty}$.

- $C = (\mathcal{R} \sqsubseteq \mathcal{S})$:
  Sei y ein beliebiger $\mathcal{R}$-Nachfolger von x in $\mathcal{I}_\infty$. Dann ist y auch ein $\mathcal{R}$-Nachfolger von x in $\mathcal{A}_\infty$. Wie oben folgt: Es existiert ein Index i, so daß y $\mathcal{R}$-Nachfolger von x in $\mathcal{A}_i$ ist und es existiert ein Index $j > i$, so daß $\longrightarrow_\sqsubseteq$ in $\mathcal{A}_j$ nicht mehr auf $x : (\mathcal{R} \sqsubseteq \mathcal{S})$ und y anwendbar ist. Also ist y auch ein $\mathcal{S}$-Nachfolger von x in $\mathcal{A}_j \subseteq \mathcal{A}_\infty$. Da y als beliebiger $\mathcal{R}$-Nachfolger von x in $\mathcal{I}_\infty$ gewählt war, folgt $x \in (\mathcal{R} \sqsubseteq \mathcal{S})^{\mathcal{I}_\infty}$.

- $C = \neg(\mathcal{R} \sqsubseteq \mathcal{S})$:
  Es existiert ein Index i mit $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S}) \in \mathcal{A}_i$ und ein Index j, so daß $\longrightarrow_{\not\sqsubseteq}$ in $\mathcal{A}_j$ nicht mehr auf $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ anwendbar ist, d.h., es existiert ein $\mathcal{R}$-Nachfolger y von x in $\mathcal{A}_j$ mit $\neg(x\mathcal{S}y) \in \mathcal{A}_j$. Dann ist y auch $\mathcal{R}$-Nachfolger von x in $\mathcal{I}_\infty$. Angenommen, y wäre auch $\mathcal{S}$-Nachfolger von x in $\mathcal{I}_\infty$. Dann wäre y $\mathcal{S}$-Nachfolger von x in $\mathcal{A}_\infty$ und $\neg(x\mathcal{S}y) \in \mathcal{A}_\infty$ im Widerspruch dazu, daß $\mathcal{A}_\infty$ offen ist. Also ist $x \in (\neg(\mathcal{R} \sqsubseteq \mathcal{S}))^{\mathcal{I}_\infty}$.

$\square$

Zum Nachweis der Endlichen-Modell-Eigenschaft

Die folgende Definition führt die Fragmente von $\mathcal{ALC} + (\sqsubseteq_n)$ ein, für die die Endliche-Modell-Eigenschaft im folgenden bewiesen wird.

Definition 3.12 (Die Fragmente $\mathcal{ALUE} + (\sqsubseteq_n)$ und $\mathcal{F}$)
Seien $\mathcal{N_C}$ eine Menge von Konzeptnamen und $\mathcal{N_R}$ eine Menge von Rollennamen.

Die Menge der $\mathcal{ALUE} + (\sqsubseteq_n)$-*Konzepte* enthält alle Konzepte, die aufgebaut sind über $\mathcal{N_C}$ und $\mathcal{N_R}$ unter Verwendung der Konstruktoren

- primitive Negation ($\neg A$ für $A \in \mathcal{N_C}$),
- Konjunktion ($C \sqcap D$),
- Disjunktion ($C \sqcup D$),
- Werterestriktion ($\forall R.C$),
- Existenzrestriktion ($\exists R.C$) und
- Role-Value-Maps auf Rollenketten gleicher Länge ($R_1 \circ \ldots \circ R_m \sqsubseteq S_1 \circ \ldots \circ S_m$),

wobei in Konzepten der Form $\forall R.C$ und $\exists R.C$ nur Rollennamen zugelassen sind.

Bei der Definition des Fragmentes $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ schließt man alle $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepte aus, deren Negationsnormalform negative Role-Value-Maps als Subkonzept einer Werterestriktion enthalten. Formal definiert man $\mathcal{F}$ durch:

> Ein $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C ist in $\mathcal{F}$ genau dann, wenn für das zugehörige $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 in NNF gilt: Für alle $\forall R.C' \in \mathsf{Sub}(C_0)$ gilt: $\mathsf{Sub}(C')$ enthält keinen Konzeptterm der Form $\neg(\mathcal{R} \sqsubseteq \mathcal{S})$.

$\diamond$

Bemerkung 3.13
Das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ ist entscheidbar, d.h., für ein gegebenes Konzept C aus $\mathcal{ALC} + (\sqsubseteq_n)$ kann man effektiv entscheiden, ob C in $\mathcal{F}$ liegt oder nicht. Das $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 in NNF zu C läßt sich mit den Regeln auf Seite 12 effektiv bestimmen. Für ein $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 in NNF ist die Menge $\mathsf{Sub}(C_0)$ endlich, so daß man effektiv entscheiden kann, ob $\mathsf{Sub}(C_0)$ keine Werterestriktionen enthält, die negative Role-Value-Maps als Subkonzept enthalten.

Offensichtlich ist die Menge der $\mathcal{ALUE} + (\sqsubseteq_n)$-Konzepte echt in $\mathcal{F}$ und $\mathcal{F}$ wiederum echt in der Menge der $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepte enthalten.

Da weder $\mathcal{ALUE} + (\sqsubseteq_n)$ noch $\mathcal{F}$ unter Negation abgeschlossen sind, läßt sich für diese Fragmente von $\mathcal{ALC} + (\sqsubseteq_n)$ das Subsumtionsproblem nicht auf das Erfüllbarkeitsproblem reduzieren. Am Ende dieses Kapitels wird man aber ein weiteres Fragment von $\mathcal{ALC} + (\sqsubseteq_n)$ definieren, für das die Entscheidbarkeit des Subsumtionsproblems aus der Entscheidbarkeit des Erfüllbarkeitsproblems für das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ folgt. Dazu prüft man für zwei Konzeptterme C, D aus diesem Fragment das Konzept $C \sqcap \neg D$ auf Unerfüllbarkeit. Da diese Konzepte $C \sqcap \neg D$ im allgemeinen in $\mathcal{F}$ und nicht in $\mathcal{ALUE} + (\sqsubseteq_n)$ liegen, ist man an der Entscheidbarkeit des Erfüllbarkeitsproblems für das Fragment $\mathcal{F}$ interessiert. $\diamond$

Beachte: Bei den in Beispiel 3.7 angegebenen Konzepten handelt es sich bei C_1 um ein $\mathcal{ALUE} + (\sqsubseteq_n)$-Konzept und das Konzept C_2 liegt in $\mathcal{F}$. Also ist für beide Fragmente die Terminierung einer iterierten Regelanwendung nicht gegeben, so daß der Vervollständigungsalgorithmus mit den Regeln aus Abbildung 3.3 weder für $\mathcal{ALUE} + (\sqsubseteq_n)$ noch das Fragment $\mathcal{F}$ ein Entscheidungsverfahren für das Erfüllbarkeitsproblem liefert.

Man weist nun zunächst für das kleinere Fragment $\mathcal{ALUE} + (\sqsubseteq_n)$ die Endliche-Modell-Eigenschaft nach. Für das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ ist dann die im Beweis des folgenden Satzes definierte Äquivalenzrelation noch geeignet zu verfeinern, um auch für das größere Fragment die Endliche-Modell-Eigenschaft nachzuweisen. Dabei lassen sich aber wesentliche Aspekte des folgenden Beweises wiederverwenden.

Satz 3.14
$\mathcal{ALUE} + (\sqsubseteq_n)$ hat die Endliche-Modell-Eigenschaft, d.h., für jedes $\mathcal{ALUE} + (\sqsubseteq_n)$-Konzept C_0 sind die folgenden Aussagen äquivalent:

1. C_0 ist erfüllbar, d.h., es existiert eine Interpretation $\mathcal{I}$ mit $C_0^{\mathcal{I}} \neq \emptyset$.
2. C_0 ist endlich erfüllbar, d.h., es existiert eine endliche Interpretation $\mathcal{I}$ mit $C_0^{\mathcal{I}} \neq \emptyset$. ⌟

Beweis:
2. $\Longrightarrow$ 1.: Diese Richtung ist trivial.

1. $\Longrightarrow$ 2.: Sei C_0 erfüllbar und es sei $m_0 := \mathsf{depth}(C_0)$. Gemäß den Beweisen zu den Punkten (1) und (3) aus Lemma 3.11 existiert eine endliche oder unendliche Folge $\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \cdots$ von Regelanwendungen mit

- $\mathcal{A}_i$ ist offen und $\mathcal{A}_i \subset \mathcal{A}_{i+1}$ für alle $i \geq 0$,
- $\mathcal{A}_\infty = \bigcup_{i \geq 0} \mathcal{A}_i$ ist offen und
- die durch $\mathcal{A}_\infty$ induzierte kanonische Interpretation $\mathcal{I}_\infty$ ist Modell zu C_0.

Ist $\mathcal{A}_\infty$ und damit auch $\mathcal{I}_\infty$ endlich, so ist nichts zu zeigen. Sei also $\mathcal{I}_\infty = (\triangle_{\mathcal{I}_\infty}, \cdot^{\mathcal{I}_\infty})$ unendlich. Man definiert nun eine Äquivalenzrelation $\sim$ auf $\triangle_{\mathcal{I}_\infty}$ mit endlichem Index und weiter eine endliche Interpretation $\widetilde{\mathcal{I}}$ mit Trägermenge $\triangle_{\widetilde{\mathcal{I}}} = \triangle_{\mathcal{I}_\infty}/_\sim$. Dann zeigt man, daß $\widetilde{\mathcal{I}}$ Modell von C_0 ist.

Zur Definition der Äquivalenzrelation auf $\tau_{\mathcal{A}_\infty}$

Bei der Definition einer geeigneten Äquivalenzrelation $\sim$ auf $\tau_{\mathcal{A}_\infty}$ ist die Menge der Konzepte, die in den Assertionen eines Elementes auftreten, zu berücksichtigen, um bei der Äquivalenzklassenbildung die Entstehung von Widersprüchen der Form $[x]_\sim \in C^{\widetilde{\mathcal{I}}}$ und $[x]_\sim \in (\neg C)^{\widetilde{\mathcal{I}}}$ zu verhindern. Die Menge aller Konzepte, die in Assertionen zu x in $\mathcal{A}_\infty$ auftreten ist definiert durch $\mathcal{C}(x) = \{C \mid x : C \in \mathcal{A}_\infty\}$.

Das Beispiel in Abbildung 3.7 zeigt, daß auch der Level zweier Variablen x, x' in $\mathcal{A}_\infty$ bei der Definition zu berücksichtigen ist.

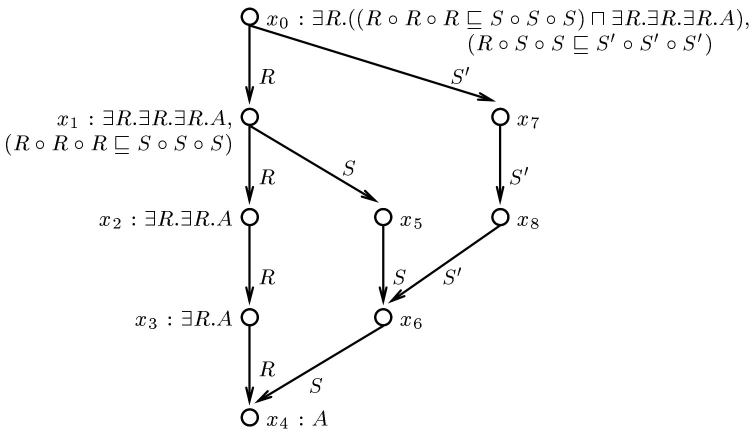


Abbildung 3.7: Zur Bedingung an den Level.

Sei $\mathcal{I}'$ die endliche Interpretation, die man erhält, wenn zur Äquivalenzklassenbildung nur die Bedingung $\mathcal{C}(x) = \mathcal{C}(x')$ berücksichtigt wird. Dann gilt im Beispiel aus Abbildung 3.7:

- $x_5^{\mathcal{I}'} = x_6^{\mathcal{I}'}$,
- $(x_5^{\mathcal{I}'}, x_5^{\mathcal{I}'}) \in R^{\mathcal{I}'}$,
- $(x_0^{\mathcal{I}'}, x_4^{\mathcal{I}'}) \in (R \circ S \circ S)^{\mathcal{I}'}$ und
- $(x_0^{\mathcal{I}'}, x_4^{\mathcal{I}'}) \notin (S' \circ S' \circ S')^{\mathcal{I}'}$.

Also folgt $x_0^{\mathcal{I}'} \notin C_0^{\mathcal{I}'}$. Insbesondere ist $\mathcal{I}'$ kein Modell von $\mathcal{A}_\infty$, was darauf zurückzuführen ist, daß $\mathcal{I}'$ keine Levelstruktur hat.

Eine Bedingung der Form $\mathsf{level}(x) = \mathsf{level}(x')$ in der Definition von $x \sim x'$ ist aber nicht ausreichend. Das Beispiel in Abbildung 3.8 zeigt, daß eine durch die Bedingung

$$\mathcal{C}(x) = \mathcal{C}(x') \text{ und } \mathsf{level}(x) = \mathsf{level}(x')$$

definierte, endliche Interpretation $\mathcal{I}''$ im allgemeinen kein Modell von $\mathcal{A}_\infty$ ist.

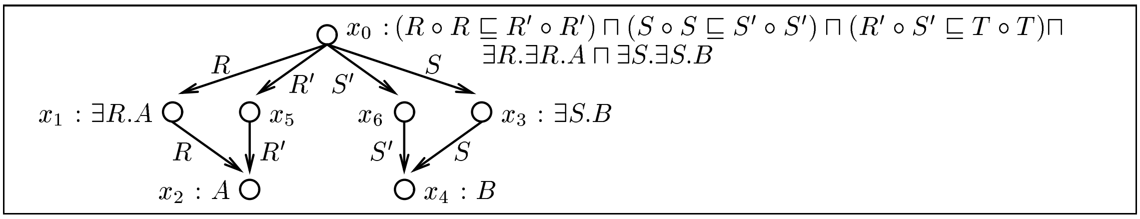


Abbildung 3.8: Zur Notwendigkeit einer stärkeren Bedingung.

Im Beispiel in Abbildung 3.8 gilt für x_5 und x_6 $\mathsf{level}(x_5) = \mathsf{level}(x_6) = 1$ und $\mathcal{C}(x_5) = \mathcal{C}(x_6) = \emptyset$, also gilt $x_5^{\mathcal{I}''} = x_6^{\mathcal{I}''}$ und damit $(x_0^{\mathcal{I}''}, x_2^{\mathcal{I}''}) \in (R' \circ S')^{\mathcal{I}''}$, aber $(x_0^{\mathcal{I}''}, x_2^{\mathcal{I}''}) \notin (T \circ T)^{\mathcal{I}''}$. Damit ist $x_0^{\mathcal{I}''} \notin (R' \circ S' \sqsubseteq T \circ T)^{\mathcal{I}''}$ und $\mathcal{I}''$ kein Modell von C_0.

Die folgende induktive Definition der Äquivalenzrelation $\sim$ auf $\tau_{\mathcal{A}_\infty}$ berücksichtigt die Erkenntnisse aus den Beispielen und ermöglicht stets die Definition eines endlichen Modells von $\mathcal{A}_\infty$ und damit von C_0.

Definition 3.15
Seien $x, x' \in \tau_{\mathcal{A}_\infty}$. Dann gelte

1. $x \sim_{m_0} x'$ genau dann, wenn
   - $\mathsf{level}(x) = \mathsf{level}(x') = m_0$ und
   - $\mathcal{C}(x) = \mathcal{C}(x')$ und
2. $x \sim_l x'$ genau dann, wenn $0 \leq l < m_0$ und
   - $\mathsf{level}(x) = \mathsf{level}(x') = l$,
   - $\mathcal{C}(x) = \mathcal{C}(x')$ sowie
   - für alle $xRy \in \mathcal{A}_\infty$ existiert ein $x'Ry' \in \mathcal{A}_\infty$ mit $y \sim_{l+1} y'$ und
   - für alle $x'Ry' \in \mathcal{A}_\infty$ existiert ein $xRy \in \mathcal{A}_\infty$ mit $y' \sim_{l+1} y$.

Es gelte $x \sim x'$ genau dann, wenn

- $\mathsf{level}(x) = \mathsf{level}(x') = l$ und
- $x \sim_l x'$. ◇

Da $\mathsf{level}(x)$ für alle $x \in \tau_{\mathcal{A}_\infty}$ eindeutig definiert ist, ist $\sim$ wohldefiniert. Zunächst zeigt man folgende Behauptung:

Behauptung: $\sim$ ist eine Äquivalenzrelation mit endlichem Index.

Beweis: Jede Variable x aus $\tau_{\mathcal{A}_\infty}$ hat einen eindeutigen Level $\mathsf{level}(x) \in \{0, \ldots, m_0\}$. Also zerfällt die Menge $\tau_{\mathcal{A}_\infty}$ in paarweise disjunkte Mengen $\tau_0, \ldots, \tau_{m_0}$ mit $\tau_i = \{x \in \tau_{\mathcal{A}_\infty} \mid \mathsf{level}(x) = i\}$ für $0 \leq i \leq m_0$. Man zeigt nun durch Induktion über den Level l, daß

1. $\sim_l$ eine reflexive, symmetrische und transitive Relation auf $\tau_{\mathcal{A}_\infty}$ ist und
2. endlichen Index auf τ_l hat, d.h., die Elemente aus τ_l bilden nur endlich viele $\sim_l$-Äquivalenzklassen.

Daraus folgt dann die Behauptung, denn: Jedes $\sim_l$, $0 \leq l \leq m_0$ ist reflexiv, symmetrisch und transitiv, also auch $\sim$. Da $\mathsf{level}(x)$ für alle $x \in \tau_{\mathcal{A}_\infty}$ eindeutig bestimmt ist und zwischen 0 und m_0 liegt, ist $\tau_{\mathcal{A}_\infty} = \tau_0 \dot\cup \ldots \dot\cup \tau_{m_0}$ und $\sim = \bigcup\limits_{l=0}^{m_0} \sim_l$ sowie $\tau_{\mathcal{A}_\infty} / \sim = \bigcup\limits_{l=0}^{m_0} \tau_l / \sim_l$. Also folgt aus 1. und 2., daß es nur endlich viele $\sim$-Klassen gibt. Es bleibt der Beweis der Punkte 1. und 2. für $\sim_0, \ldots, \sim_{m_0}$.

- Offensichtlich ist $\sim_{m_0}$ reflexiv, symmetrisch und transitiv. Eine $\sim_{m_0}$-Klasse $[x]_{\sim_{m_0}} = \{x' \in \tau_{m_0} \mid x \sim_{m_0} x'\}$ ist eindeutig charakterisiert durch $\mathcal{C}(x) \subseteq \mathsf{Sub}(C_0)$. Da $\mathsf{Sub}(C_0)$ endlich ist, bilden die Variablen aus τ_{m_0} nur endlich viele $\sim_{m_0}$-Klassen.
- Aus Definition 3.15 folgt, daß für alle $x, x' \in \tau_l$, $0 \leq l < m_0$, gilt: $x \sim_l x$ und $x \sim_l x' \iff x' \sim_l x$. Desweiteren gilt $x \sim_l x'$, $x' \sim_l x'' \implies x \sim_l x''$, denn: offensichtlich ist $\mathsf{level}(x) = \mathsf{level}(x') = \mathsf{level}(x'')$ und $\mathcal{C}(x) = \mathcal{C}(x') = \mathcal{C}(x'')$. Sei $xRy \in \mathcal{A}_\infty$ Wegen $x \sim_l x'$ existiert ein $x'Ry' \in \mathcal{A}_\infty$ mit $y \sim_{l+1} y'$ und wegen $x' \sim_l x''$ existiert ein $x''Ry'' \in \mathcal{A}_\infty$ mit $y' \sim_{l+1} y''$. Per Induktion folgt dann $y \sim_{l+1} y''$. Analog folgt, daß zu $x''Ry'' \in \mathcal{A}_\infty$ ein $xRy \in \mathcal{A}_\infty$ existiert mit $y'' \sim_{l+1} y$. Also gilt $x \sim_l x''$. Damit folgt, daß $\sim_l$ reflexiv, symmetrisch und transitiv ist.
  Eine $\sim_l$-Klasse $[x]_{\sim_l}$ auf τ_l ist eindeutig charakterisiert durch
  - die Menge der Konzeptassertionen von x in $\mathcal{A}_\infty$, $\mathcal{C}(x) \subseteq \mathsf{Sub}(C_0)$ und
  - die Menge $\rho(x)$ aller durch eine Assertion $xRy \in \mathcal{A}_\infty$ von x aus erreichbaren $\sim_{l+1}$-Klassen,
    $\rho(x) = \{\big(R, \sigma(R,x)\big) \mid R \in \mathcal{N_R} \text{ und } \sigma(R,x) = \{[y]_{\sim_{l+1}} \mid xRy \in \mathcal{A}_\infty\}\}$,
    mit $\rho(x) \subseteq \mathcal{N_R} \times 2^{\tau_{l+1}/\sim_{l+1}}$.

  $\mathsf{Sub}(C_0)$ sowie $\mathcal{N_R}$ sind endlich und per Induktion ist $\tau_{l+1} / \sim_{l+1}$ und damit auch $2^{\tau_{l+1}/\sim_{l+1}}$ endlich, also hat $\sim_l$ endlichen Index auf τ_l.

Zur Definition der endlichen Interpretation $\widetilde{\mathcal{I}}$

Man definiert die Interpretation $\widetilde{\mathcal{I}} = (\triangle_{\widetilde{\mathcal{I}}}, \cdot^{\widetilde{\mathcal{I}}})$ durch

- $\triangle_{\widetilde{\mathcal{I}}} := \{[x]_\sim \mid x \in \triangle_{\mathcal{I}_\infty}\}$

- $x^{\widetilde{\mathcal{I}}} := [x]_\sim$
- $A^{\widetilde{\mathcal{I}}} := \{[x]_\sim \mid x : A \in \mathcal{A}_\infty\}$
- $R^{\widetilde{\mathcal{I}}} := \{([x]_\sim, [y]_\sim) \mid \text{ ex. } x' \in [x]_\sim \text{ und } y' \in [y]_\sim \text{ mit } x'Ry' \in \mathcal{A}_\infty\}$

Da $\mathcal{C}(x) = \mathcal{C}(x')$ für $[x]_\sim = [x']_\sim$ gilt, ist $\widetilde{\mathcal{I}}$ eine wohldefinierte Interpretation. Es bleibt zu zeigen:

$$\widetilde{\mathcal{I}} \text{ ist Modell von } C_0. \qquad (*)$$

Dazu benötigt man folgende Hilfsaussage:

Lemma 3.16
Es seien y_0, y_m Variablen aus $\tau_{\mathcal{A}_\infty}$ und $([y_0]_\sim, [y_m]_\sim) \in \mathcal{R}^{\widetilde{\mathcal{I}}}$.
Dann existiert ein $y'_m \in \tau_{\mathcal{A}_\infty}$ mit $y'_m \in \mathcal{R}(y_0)$ in $\mathcal{A}_\infty$ und $y'_m \sim y_m$. ⌟

Beweis:
Man zeigt die Behauptung durch Induktion über $m = |\mathcal{R}|$.

- $m = 1$: Es sei $([y_0]_\sim, [y_1]_\sim) \in R_1^{\widetilde{\mathcal{I}}}$. Aus der Definition von $\widetilde{\mathcal{I}}$ folgt, daß ein $y'_0 \in [y_0]_\sim$ und ein $y'_1 \in [y_1]_\sim$ existiert mit $y'_0 R_1 y'_1 \in \mathcal{A}_\infty$. Sei $l = \mathsf{level}(y_0)$. Aus der Definition von $\sim$ folgt $y_0 \sim_l y'_0$. Also existiert ein $y''_1 \in \tau_{\mathcal{A}_\infty}$ mit $y_0 R_1 y''_1 \in \mathcal{A}_\infty$ und $y''_1 \sim_{l+1} y'_1$. Mit $y'_1 \sim y_1$ und der Definition von $\sim$ folgt $y''_1 \sim y_1$.

- $m \longrightarrow m{+}1$: Es sei $([y_0]_\sim, [y_{m+1}]_\sim) \in (R_1 \circ \mathcal{R})^{\widetilde{\mathcal{I}}}$. Aus der Definition von $\widetilde{\mathcal{I}}$ folgt, daß ein $[y_1]_\sim$ existiert mit $([y_0]_\sim, [y_1]_\sim) \in R_1^{\widetilde{\mathcal{I}}}$ und $([y_1]_\sim, [y_{m+1}]_\sim) \in \mathcal{R}^{\widetilde{\mathcal{I}}}$. Wie oben folgt, daß ein $y'_0 \in [y_0]_\sim$ und ein $y'_1 \in [y_1]_\sim$ existiert mit $y'_0 R_1 y'_1 \in \mathcal{A}_\infty$. Wiederum wie oben folgt, daß wegen $y_0 \sim y'_0$ ein y''_1 existiert mit $y_0 R_1 y''_1 \in \mathcal{A}_\infty$ und $y''_1 \sim y'_1$. Also gilt $y''_1 \sim y_1$ und damit $([y''_1]_\sim, [y_{m+1}]_\sim) \in \mathcal{R}^{\widetilde{\mathcal{I}}}$. Per Induktion folgt, daß ein $\mathcal{R}$-Nachfolger y'_{m+1} von y''_1 in $\mathcal{A}_\infty$ existiert mit $y'_{m+1} \sim y_{m+1}$. Wegen $y_0 R_1 y''_1 \in \mathcal{A}_\infty$ ist y'_{m+1} ein $(R_1 \circ \mathcal{R})$-Nachfolger von y_0 in $\mathcal{A}_\infty$ mit $y'_{m+1} \sim y_{m+1}$. □

Zum Beweis der Behauptung $(*)$ zeigt man durch Induktion über den Aufbau von $\mathcal{ALUE} + (\sqsubseteq_n)$-Konzepttermen

$$[x]_\sim \in C^{\widetilde{\mathcal{I}}} \text{ für alle } x : C \in \mathcal{A}_\infty.$$

Dann folgt wegen $x_0 : C_0 \in \mathcal{A}_\infty$ auch $[x_0]_\sim \in C_0^{\widetilde{\mathcal{I}}}$, also ist $\widetilde{\mathcal{I}}$ ein endliches Modell von C_0.
Sei also $x : C \in \mathcal{A}_\infty$.

- $C = A \in \mathcal{N}_\mathcal{C}$:
  Aus der Definition von $\widetilde{\mathcal{I}}$ folgt $[x]_\sim \in A^{\widetilde{\mathcal{I}}}$.

- $C = \neg A$ für $A \in \mathcal{N}_\mathcal{C}$:
  Da $\mathcal{A}_\infty$ offen ist, ist $x : A$ nicht in $\mathcal{A}_\infty$. Aus der Definition von $\widetilde{\mathcal{I}}$ folgt $[x]_\sim \notin A^{\widetilde{\mathcal{I}}}$, also $[x]_\sim \in (\neg A)^{\widetilde{\mathcal{I}}}$.

- $C = C_1 \sqcap C_2$:
  Aus der Fairneßannahme folgt $x : C_1 \in \mathcal{A}_\infty$ und $x : C_2 \in \mathcal{A}_\infty$. Per Induktion folgt $[x]_\sim \in C_1^{\widetilde{\mathcal{I}}}$ und $[x]_\sim \in C_2^{\widetilde{\mathcal{I}}}$, also $[x]_\sim \in (C_1 \sqcap C_2)^{\widetilde{\mathcal{I}}}$.

- $C = C_1 \sqcup C_2$:
  Analog zum Fall $C = C_1 \sqcap C_2$ folgt $[x]_\sim \in (C_1 \sqcup C_2)^{\widetilde{\mathcal{I}}}$.

- $C = \exists R.C'$:
  Wie oben folgt aus der Fairneßannahme, daß ein $y \in \tau_{\mathcal{A}_\infty}$ existiert mit xRy, $y : C' \in \mathcal{A}_\infty$. Per Induktion folgt $[y]_\sim \in C'^{\widetilde{\mathcal{I}}}$ und aus der Definition von $\widetilde{\mathcal{I}}$ folgt $([x]_\sim, [y]_\sim) \in R^{\widetilde{\mathcal{I}}}$, also $[x]_\sim \in (\exists R.C')^{\widetilde{\mathcal{I}}}$.

- $C = \forall R.C'$:
  Sei $[y]_\sim \in \triangle_{\widetilde{\mathcal{I}}}$ ein beliebiger R-Nachfolger von $[x]_\sim$ in $\widetilde{\mathcal{I}}$. Dann existiert ein $x' \in [x]_\sim$ und ein $y' \in [y]_\sim$ mit $x'Ry' \in \mathcal{A}_\infty$. Wegen $x \sim x'$ ist $x' : \forall R.C'$ in $\mathcal{A}_\infty$ und auf Grund der Fairneßannahme auch $y' : C' \in \mathcal{A}_\infty$. Wegen $y \sim y'$ ist dann auch $y : C' \in \mathcal{A}_\infty$ und per Induktion folgt $[y]_\sim \in C'^{\widetilde{\mathcal{I}}}$. Da $[y]_\sim$ als beliebiger R-Nachfolger gewählt war, ist also $[x]_\sim \in (\forall R.C')^{\widetilde{\mathcal{I}}}$.

- $C = (\mathcal{R} \sqsubseteq \mathcal{S})$:
  Sei $[y]_\sim \in \triangle_{\widetilde{\mathcal{I}}}$ ein beliebiger $\mathcal{R}$-Nachfolger von $[x]_\sim$ in $\widetilde{\mathcal{I}}$, d.h., $([x]_\sim, [y]_\sim) \in \mathcal{R}^{\widetilde{\mathcal{I}}}$. Dann folgt mit Lemma 3.16, daß ein $\mathcal{R}$-Nachfolger y' von x in $\mathcal{A}_\infty$ existiert mit $y \sim y'$. Mit der Fairneßannahme folgt, daß y' auch $\mathcal{S}$-Nachfolger von x in $\mathcal{A}_\infty$ ist. Also ist $([x]_\sim, [y']_\sim) \in \mathcal{S}^{\widetilde{\mathcal{I}}}$ und mit $[y']_\sim = [y]_\sim$ folgt $([x]_\sim, [y]_\sim) \in \mathcal{S}^{\widetilde{\mathcal{I}}}$. Da $[y]_\sim$ als beliebiger $\mathcal{R}$-Nachfolger gewählt war, ist $[x]_\sim \in (\mathcal{R} \sqsubseteq \mathcal{S})^{\widetilde{\mathcal{I}}}$. □

Um auch für das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ die Endliche-Modell-Eigenschaft wie im Beweis von Satz 3.14 zu zeigen, reicht die Äquivalenzrelation $\sim$ nicht aus, man benötigt eine geeignete Verfeinerung.

Beispiel 3.17
Für die Variablen der ABox aus Abbildung 3.9 folgt mit Definition 3.15 $x_2 \sim x_4$. Ohne eine weitere Bedingung würde man also die Variablen x_2 und x_4 zu einer $\sim$-Klasse zusammenfassen. Die so erhaltene Interpretation $\mathcal{I}$ erfüllt aber die Assertion $x_0 : \neg(R \circ R \sqsubseteq R \circ S)$ nicht mehr, da der Zeuge x_2 für diese Assertion mit x_4 zusammengefaßt wurde. Also ist $\mathcal{I}$ weder Modell der ABox noch des Ausgangskonzeptes C_0. ◇

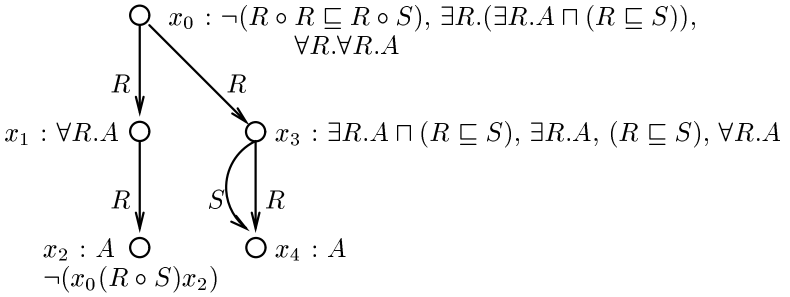


Abbildung 3.9: Zur Notwendigkeit einer weiteren Bedingung.

Um eine geeignete Verfeinerung der Äquivalenzrelation $\sim$ definieren zu können, benötigt man noch den Begriff der *Entstehungsgeschichte* $\mathsf{h}(x)$ einer Variablen x. Offensichtlich wird jede Variable $x \in \tau_{\mathcal{A}_\infty}$ durch eine Folge von Anwendungen generierender Regeln erzeugt. Genauer: eine Variable x mit $\mathsf{level}(x) = i$ wird durch anwenden einer generierenden Regel auf einen Vorfahren x_1 mit $\mathsf{level}(x_1) = i_1 < i$ erzeugt, x_1 wird durch einen Vorfahren x_2 mit $\mathsf{level}(x_2) = i_2 < i_1$ erzeugt und so weiter. Unter der Entstehungsgeschichte von x versteht man nun ein m_0-Tupel $\mathsf{h}(x)$, das genau die erzeugenden Konzepte enthält, durch die die x_i und x selber erzeugt wurden.

Definition 3.18 (Entstehungsgeschichte)
Die Abbildung $\mathsf{h} : \tau_{\mathcal{A}_\infty} \longrightarrow (\mathsf{Sub}(C_0) \cup \{\emptyset\})^{m_0}$ weist jedem $x \in \tau_{\mathcal{A}_\infty}$ seine Entstehungsgeschichte $\mathsf{h}(x)$ zu und ist induktiv definiert durch:

- $\mathsf{h}(x_0) := (\emptyset, \ldots, \emptyset)$ und
- $\mathsf{h}(y) := (D_0, \ldots, D_{j-1}, D, \emptyset, \ldots, \emptyset)$, falls
  - y durch anwenden einer generierenden Regel auf $x : D$ erzeugt wurde,
  - $j = \mathsf{level}(x)$ und
  - $\mathsf{h}(x) = (D_0, \ldots, D_{j-1}, \emptyset, \ldots, \emptyset)$.

  (Im folgenden schreibt man kurz $\mathsf{h}(x_i) = \mathsf{h}(x)[j \leftarrow D]$.)

Beachte: Es ist $\mathsf{level}(x) < m_0$, also ist $\mathsf{h}(x)$ stets ein m_0-Tupel über $\mathsf{Sub}(C_0) \cup \{\emptyset\}$.$\diamond$

Beispiel 3.19 (Beispiel 3.17 Fortsetzung)
Im Beispiel aus Abbildung 3.9 ist offensichtlich

- $\mathsf{h}(x_2) = \big(\neg(R \circ R \sqsubseteq R \circ S), \emptyset\big)$ sowie
- $\mathsf{h}(x_4) = \big(\exists R.(\exists R.A \sqcap (R \sqsubseteq S)), \exists R.A\big)$ und damit
- $\mathsf{h}(x_2) \neq \mathsf{h}(x_4)$.

Berücksichtigt man also die Entstehungsgeschichte bei der Äquivalenzklassenbildung, so faßt man x_2 und x_4 nicht zusammen, so daß die Assertion $x_0 : \neg(R \circ R \sqsubseteq R \circ S)$ aus Abbildung 3.9 erfüllt bleibt. $\diamond$

Satz 3.20
Das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ hat die Endliche-Modell-Eigenschaft, d.h., für jedes Konzept C_0 aus $\mathcal{F}$ sind die folgenden Aussagen äquivalent:

1. C_0 ist erfüllbar.

2. C_0 ist endlich erfüllbar. $\lrcorner$

Beweis:
2. $\Longrightarrow$ 1.: Diese Richtung ist wiederum trivial.
1. $\Longrightarrow$ 2.: Sei C_0 ein erfüllbares $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept aus $\mathcal{F}$ und es sei $m_0 := \mathsf{depth}(C_0)$. Wie im Beweis von Satz 3.14 sei $\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \cdots$ eine unendliche Folge von Regelanwendungen und $\mathcal{I}_\infty$ das unendliche, kanonische Modell von C_0. (Ist die Folge und damit $\mathcal{I}_\infty$ endlich, so ist nichts zu zeigen.)
Wie im Beweis von Satz 3.14 definiert man induktiv eine Äquivalenzrelation $\approx$ auf $\tau_{\mathcal{A}_\infty}$ mit endlichem Index und zeigt, daß die mit Hilfe von $\approx$ definierte endliche Interpretation $\overline{\mathcal{I}}$ ein Modell von $\mathcal{A}_\infty$ und damit von C_0 ist. Im Unterschied zur Relation $\sim$ wird bei der Definition von $\approx$ die Entstehungsgeschichte berücksichtigt.

Definition 3.21
Seien $x, x' \in \tau_{\mathcal{A}_\infty}$. Dann gelte

1. $x \approx_{m_0} x'$ genau dann, wenn
   - $\mathsf{level}(x) = \mathsf{level}(x') = m_0$,
   - $\mathcal{C}(x) = \mathcal{C}(x')$ und
   - $\mathsf{h}(x) = \mathsf{h}(x')$ und
2. $x \approx_l x'$ genau dann, wenn $0 \leq l < m_0$ und
   - $\mathsf{level}(x) = \mathsf{level}(x') = l$,
   - $\mathcal{C}(x) = \mathcal{C}(x')$,
   - $\mathsf{h}(x) = \mathsf{h}(x')$ sowie
   - für alle $xRy \in \mathcal{A}_\infty$ existiert ein $x'Ry' \in \mathcal{A}_\infty$ mit $y \approx_{l+1} y'$ und
   - für alle $x'Ry' \in \mathcal{A}_\infty$ existiert ein $xRy \in \mathcal{A}_\infty$ mit $y' \approx_{l+1} y$.

Es gelte $x \approx x'$ genau dann, wenn

- $\mathsf{level}(x) = \mathsf{level}(x') = l$ und
- $x \approx_l x'$. $\diamond$

Bemerkung 3.22
Die Definition der Relation $\approx$ erfolgt wie die Definition der Äquivalenzrelation $\sim$ induktiv, um die Aussagen über $\sim$ und $\widetilde{\mathcal{I}}$ leicht auf $\approx$ und die mit Hilfe von $\approx$ definierte Interpretation übertragen zu können.

Das Problem aus Beispiel 3.17 wird auch durch folgende Verfeinerung der Relation $\sim$ vermieden:

$$x \approx x' \text{ gdw } x \sim x' \text{ und } \mathsf{h}(x) = \mathsf{h}(x').$$

Um aber zu zeigen, daß die mit Hilfe der Relation $\approx$ definierte, endliche Interpretation $\mathcal{I}''$ ein Modell von $\mathcal{A}_\infty$ ist, benötigt man eine Hilfsaussage für $\approx$ und $\mathcal{I}''$, die der Aussage in Lemma 3.16 für $\sim$ und $\widetilde{\mathcal{I}}$ entspricht. Für die Relation $\approx$ läßt sich aber der Beweis von Lemma 3.16 nicht ohne weiteres auf den vorliegenden Fall übertragen, da zwar aus $x \approx x'$ auch $x \sim x'$ folgt, aber nicht umgekehrt. Für die durch Definition 3.21 bestimmte Relation $\approx$ und die mit $\approx$ definierte Interpretation $\overline{\mathcal{I}}$ läßt sich der Induktionsbeweis von Lemma 3.16 auf Seite 72 aber offensichtlich übertragen. $\diamond$

Behauptung: $\approx$ ist eine Äquivalenzrelation mit endlichem Index.
Beweis: Für alle $x \in \tau_{\mathcal{A}_\infty}$ gilt $\mathsf{h}(x) \in \big(\mathsf{Sub}(C_0) \cup \{\emptyset\}\big)^{m_0}$. Da m_0 und $\mathsf{Sub}(C_0)$ endlich sind, gibt es also nur endlich viele verschiedene Entstehungsgeschichten zu den Variablen aus $\tau_{\mathcal{A}_\infty}$. Mit dieser zusätzlichen Voraussetzung folgt der Beweis der Behauptung wie für die Relation $\sim$ aus Definition 3.15 auf Seite 70.

Man definiert nun die Interpretation $\overline{\mathcal{I}} = (\triangle_{\overline{\mathcal{I}}}, \cdot^{\overline{\mathcal{I}}})$ durch

- $\triangle_{\overline{\mathcal{I}}} := \{[x]_\approx \mid x \in \triangle_{\mathcal{I}_\infty}\}$,
- $x^{\overline{\mathcal{I}}} := [x]_\approx$,
- $A^{\overline{\mathcal{I}}} := \{[x]_\approx \mid x : A \in \mathcal{A}_\infty\}$ sowie
- $R^{\overline{\mathcal{I}}} := \{([x]_\approx, [y]_\approx) \mid \text{ ex. } x' \in [x]_\approx \text{ und } y' \in [y]_\approx \text{ mit } x'Ry' \in \mathcal{A}_\infty\}$

und zeigt: $\overline{\mathcal{I}}$ ist Modell von C_0.

Dazu zeigt man durch Induktion über die Länge von Konzepttermen[2] aus $\mathcal{F}$

$$[x]_\approx \in C^{\overline{\mathcal{I}}} \text{ für alle } x : C \in \mathcal{A}_\infty. \qquad (*)$$

Dann folgt wegen $x_0 : C_0 \in \mathcal{A}_\infty$ auch $[x_0]_\approx \in C_0^{\overline{\mathcal{I}}}$, also ist $\overline{\mathcal{I}}$ ein endliches Modell von C_0.

Zum Beweis der Behauptung $(*)$ benötigt man folgende Hilfsaussagen.

[2]Die Menge $\mathcal{F}$ ist nicht induktiv definiert, also ist eine Induktion über den Aufbau von Konzepttermen aus $\mathcal{F}$ nicht erklärt.

Lemma 3.23
Es seien y_0, y_m Variablen aus $\tau_{\mathcal{A}_\infty}$ und $([y_0]_\approx, [y_m]_\approx) \in \mathcal{R}^{\overline{\mathcal{I}}}$.
Dann existiert ein $y'_m \in \tau_{\mathcal{A}_\infty}$ mit $y'_m \in \mathcal{R}(y_0)$ in $\mathcal{A}_\infty$ und $y'_m \approx y_m$. ⌟

Beweis:
Analog zum Beweis von Lemma 3.16 auf Seite 72 zeigt man die Behauptung durch Induktion über $m = |\mathcal{R}|$. Dabei gelten die Aussagen und Folgerungen für $\widetilde{\mathcal{I}}$ und die $\sim$-Klassen auch für $\overline{\mathcal{I}}$ und die $\approx$-Klassen. □

Lemma 3.24
Es sei C_0 ein erfüllbares Konzept aus $\mathcal{F}$ und $\mathcal{A}_\infty$ eine aus $\{x_0 : C_0\}$ abgeleitete, unendliche ABox und $x \in \tau_{\mathcal{A}_\infty}$. Dann gilt:

1. Ist $\mathsf{h}(x) = (D_0, \ldots, D_{j-1}, \emptyset, \ldots, \emptyset)$ mit
   - $j = \mathsf{level}(x)$ und
   - $D_i = \exists R_i.D'_i$ für $0 \leq i < j$,

   so ist $[x]_\approx = \{x\}$.
2. Ist $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S}) \in \mathcal{A}_\infty$, so gilt:
   - $\mathsf{h}(x) = (D_0, \ldots, D_{j-1}, \emptyset, \ldots, \emptyset)$ mit
     - $j = \mathsf{level}(x)$ und
     - $D_i = \exists R_i.D'_i$ für $0 \leq i < j$ und
   - es existiert genau ein $y \in \tau_{\mathcal{A}_\infty}$ mit
     - $\neg(x\mathcal{S}y) \in \mathcal{A}_\infty$,
     - $\mathsf{h}(y) = \mathsf{h}(x)[j \leftarrow \neg(\mathcal{R} \sqsubseteq \mathcal{S})]$ und
     - $[y]_\approx = \{y\}$. ⌟

Beweis:
Zum Punkt (1)**:** Man zeigt durch vollständige Induktion über $j = \mathsf{level}(x)$, daß keine zwei verschiedenen Variablen aus $\tau_{\mathcal{A}_\infty}$ existieren, die die gleiche Geschichte h haben mit $\mathsf{h} = (\exists R_0.D'_0, \ldots, \exists R_{j-1}.D'_{j-1}, \emptyset, \ldots, \emptyset)$. Aus dieser Behauptung und der Definition 3.21 folgt dann das Gewünschte:

$$\mathsf{h}(x) = (\exists R_0.D'_0, \ldots, \exists R_{j-1}.D'_{j-1}, \emptyset, \ldots, \emptyset) \Longrightarrow [x]_\approx = \{x\}.$$

$j = 0$:
x_0 ist die einzige Variable aus $\tau_{\mathcal{A}_\infty}$ mit Level 0 und es ist $\mathsf{h}(x_0) = (\emptyset, \ldots, \emptyset)$. Für alle Variablen $x \in \tau_{\mathcal{A}_\infty} \setminus \{x_0\}$ ist $\mathsf{h}(x) \neq (\emptyset, \ldots, \emptyset)$. Also ist $[x_0]_\approx = \{x_0\}$.

$j - 1 \longrightarrow j$:
Angenommen, es existieren zwei verschiedene Variablen x, x' mit

- $\mathsf{h}(x) = (\exists R_0.D'_0, \ldots, \exists R_{j-1}.D'_{j-1}, \emptyset, \ldots, \emptyset)$ und
  $= \mathsf{h}(x_{j-1})[j \leftarrow \exists R_{j-1}.D'_{j-1}]$
- $\mathsf{h}(x') = (\exists R_0.D'_0, \ldots, \exists R_{j-1}.D'_{j-1}, \emptyset, \ldots, \emptyset)$
  $= \mathsf{h}(x'_{j-1})[j \leftarrow \exists R_{j-1}.D'_{j-1}]$.

Dann ist $\mathsf{h}(x_{j-1}) = \mathsf{h}(x'_{j-1})$ und per Induktion folgt $x_{j-1} = x'_{j-1}$. Aus der Definition der Vervollständigungsregeln folgt, daß die $\exists$-Regel höchstens einmal auf die Assertion $x_{j-1} : \exists R_{j-1}.D'_{j-1}$ angewendet wird, also ist $x = x'$ im Widerspruch zur Annahme, daß x und x' verschieden sind.

Zum Punkt (2)**:** Aus der Bedingung an die Konzeptterme aus $\mathcal{F}$ folgt, daß Assertionen der Form $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ nur durch Anwendungen der Regeln $\longrightarrow_\exists, \longrightarrow_\sqcap, \longrightarrow_\sqcup$ erzeugt werden. Also ist $\mathsf{h}(x) = (D_0, \ldots, D_{j-1}, \emptyset, \ldots, \emptyset)$ mit

- $j = \mathsf{level}(x)$ und
- $D_i = \exists R_i.D'_i$ für $0 \leq i < j$.

Aus der Definition der Vervollständigungsregeln folgt, daß die $\not\sqsubseteq$-Regel genau einmal auf eine Assertion $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ in $\mathcal{A}_\infty$ angewendet wird, d.h., es existiert genau ein $\mathcal{R}$-Nachfolger y von x in $\mathcal{A}_\infty$ mit

- $\neg(x\mathcal{S}y) \in \mathcal{A}_\infty$ und
- $\mathsf{h}(y) = \mathsf{h}(x)[j \leftarrow \neg(\mathcal{R} \sqsubseteq \mathcal{S})]$.

Zusammen mit Punkt (1) folgt, daß keine andere Variable $y' \in \tau_{\mathcal{A}_\infty}$ existiert mit $\mathsf{h}(y') = \mathsf{h}(y)$. Also ist $[y]_\approx = \{y\}$. □

Zum Beweis der Behauptung $(*)$**:**
Sei $x : C \in \mathcal{A}_\infty$.

- Ist C von der Form A, $\neg A$, $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\exists R.C_1$, $\forall R.C_1$ oder $(\mathcal{R} \sqsubseteq \mathcal{S})$, so folgt die Behauptung wie im Beweis von Satz 3.14.
  Beachte: Dabei ist die Induktionsvoraussetzung auf die Subkonzepte C_1 und C_2 anwendbar, da ihre Länge gegenüber C offensichtlich kleiner ist. Für Konzeptterme der Form $(\mathcal{R} \sqsubseteq \mathcal{S})$ folgt die Behauptung nun mit Lemma 3.23.
- $C = \neg(\mathcal{R} \sqsubseteq \mathcal{S})$:
  Mit der Fairneßannahme folgt, daß es $x_1, \ldots, x_m \in \tau_{\mathcal{A}_\infty}$ gibt mit
  - $\{xR_1x_1, x_1R_2x_2, \ldots, x_{m-1}R_mx_m, \neg(x\mathcal{S}x_m)\} \subseteq \mathcal{A}_\infty$ und
  - $\mathsf{h}(x_m) = \mathsf{h}(x)[\mathsf{level}(x) \leftarrow \neg(\mathcal{R} \sqsubseteq \mathcal{S})]$.

Aus der Definition von $\overline{\mathcal{I}}$ folgt $([x]_\approx, [x_m]_\approx) \in \mathcal{R}^{\overline{\mathcal{I}}}$ und es gilt $([x]_\approx, [x_m]_\approx) \notin \mathcal{S}^{\overline{\mathcal{I}}}$, denn:
Angenommen, es gilt $([x]_\approx, [x_m]_\approx) \in \mathcal{S}^{\overline{\mathcal{I}}}$. Dann folgt mit Lemma 3.23, daß ein x'_m in $\tau_{\mathcal{A}_\infty}$ existiert mit: x'_m ist $\mathcal{S}$-Nachfolger von x in $\mathcal{A}_\infty$ und $x'_m \approx x_m$. Mit Lemma 3.24 folgt $x'_m = x_m$, also $x_m \in \mathcal{S}(x)$ in $\mathcal{A}_\infty$ und $\neg(x\mathcal{S}x_m) \in \mathcal{A}_\infty$. Damit enthält $\mathcal{A}_\infty$ einen clash im Widerspruch zur Voraussetzung, daß $\mathcal{A}_\infty$ offen ist.
Es folgt $([x]_\approx, [x_m]_\approx) \notin \mathcal{S}^{\overline{\mathcal{I}}}$ und damit $[x]_\approx \in (\neg(\mathcal{R} \sqsubseteq \mathcal{S}))^{\overline{\mathcal{I}}}$. □

Beispiel 3.25 (Beispiel 3.7 Fortsetzung)
Abbildung 3.10 stellt die endlichen Interpretationen $\mathcal{I}'$ und $\mathcal{I}''$ dar, die man für das Konzept C_1 bzw. C_2 aus Beispiel 3.7 durch die Äquivalenzrelation $\sim$ bzw. $\approx$ erhält. Offensichtlich gilt: $C_1^{I'} \neq \emptyset$ und $C_2^{I''} \neq \emptyset$.
Beachte: Unter Verwendung der Relation $\approx$ aus Bemerkung 3.22 würde in $\mathcal{I}''$ gelten: $y_1 \approx y_5$ und $y_2 \approx y_6$. Die so erhaltene, kleinere Interpretation wäre ebenfalls ein Modell von C_2. Für beliebige Konzepte aus $\mathcal{F}$ bleibt aber eine entsprechende Behauptung noch zu zeigen. ◇

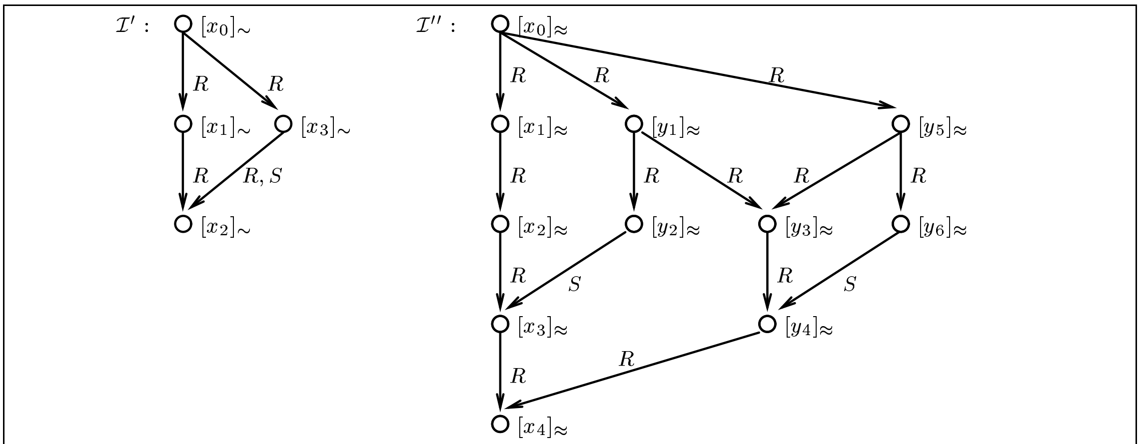


Abbildung 3.10: Zwei endliche Modelle.

Beispiel 3.26
Man betrachte folgendes $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept, das offensichtlich nicht in $\mathcal{F}$ ist:

$$C_0 := \exists R.\neg(R \sqsubseteq S) \sqcap \forall R.\neg(R \sqsubseteq S) \sqcap (R \circ R \sqsubseteq R \circ S).$$

Wie in Beispiel 3.7 skizziert, liefert eine unendliche Folge von Regelanwendungen das in Abbildung 3.11 dargestellte unendliche, kanonische Modell $\mathcal{I}_\infty$.

Die mit Hilfe der Äquivalenzrelation $\approx$ definierte endliche Interpretation $\overline{\mathcal{I}}$ ist in Abbildung 3.12 dargestellt. Offensichtlich ist $\overline{\mathcal{I}}$ aber kein Modell von C_0, da der R-Nachfolger $[x_3]_\approx$ von $[x_0]_\approx$ nicht in $\neg(R \sqsubseteq S)^{\overline{\mathcal{I}}}$ ist. Die ebenfalls in Abbildung 3.12 dargestellte, endliche Interpretation $\mathcal{I}'$ ist ein Modell von C_0.

Das Modell $\mathcal{I}'$ erhält man durch eine geeignete, disjunkte Zerlegung der Äquivalenzklasse $[x_4]_\approx$ in $\overline{x_4} = \{x_4, x_8, x_{12}, \dots\}$ und $\overline{x_6} = \{x_6, x_{10}, x_{14}, \dots\}$. Diese Zerlegung

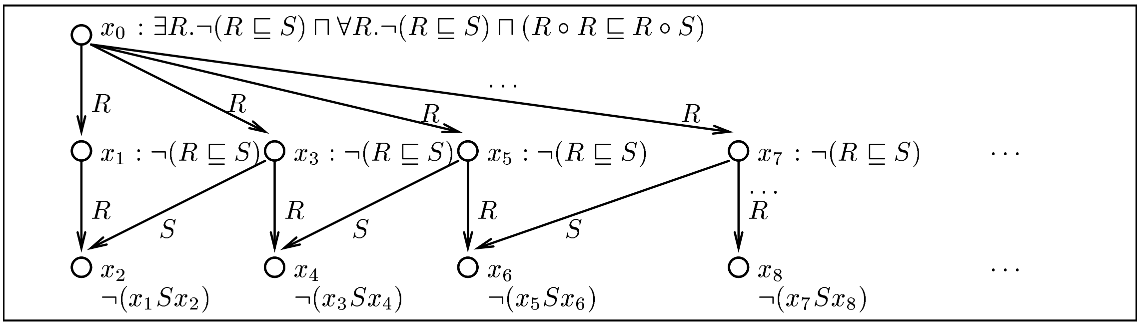


Abbildung 3.11: $\approx$ ist unzureichend für $\mathcal{ALC} + (\sqsubseteq_n)$.

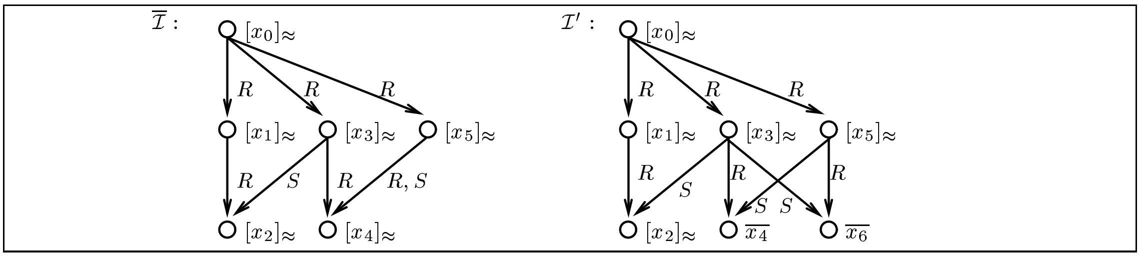


Abbildung 3.12: Die endliche Interpretation und ein endliches Modell von C_0.

beruht auf der folgenden Bedingung:

($*$) Fasse zwei Variablen y und y' nicht zusammen, wenn ein $x \in \tau_{\mathcal{A}_\infty}$ existiert mit $\neg(x\mathcal{R}y) \in \mathcal{A}_\infty$ und $y' \in \mathcal{R}(x)$ in $\mathcal{A}_\infty$.

Leider zeigt das obige Beispiel, daß ($*$) keine Verfeinerung der Äquivalenzrelation $\approx$ zu einer weiteren Äquivalenzrelation $\cong$ liefert. Für die Variablen aus Abbildung 3.11 gilt: Man kann x_4 mit x_8 und x_{10} zusammenfassen, aber x_8 und x_{10} dürfen nicht zusammengefaßt werden. Damit definiert die Bedingung ($*$) keine transitive Relation auf der Trägermenge eines unendlichen kanonischen Modells. Es ist also unklar, ob und wie man mit Hilfe der Bedingung ($*$) eine Äquivalenzrelation auf $\tau_{\mathcal{A}_\infty}$ mit endlichem Index definieren kann.

Beachte: Für eine unendliche ABox $\mathcal{A}_\infty$, die aus $\{x_0 : C_0\}$ mit einem beliebigen $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 in NNF abgeleitet wurde, und für die mit Hilfe der Äquivalenzrelation $\approx$ aus Definition 3.21 bestimmte endliche Interpretation $\overline{\mathcal{I}}$ gilt im allgemeinen nicht, daß $\overline{\mathcal{I}}$ Modell von C_0 ist. Die Ursache hierfür ist, daß es nun Assertionen $x : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ und $x' : \neg(\mathcal{R} \sqsubseteq \mathcal{S})$ mit $x \neq x'$ und $x \approx x'$ geben kann (z.B. gilt für die Variablen in Abbildung 3.11 $x_3 \approx x_5 \approx x_7 \approx \cdots$). Mit Lemma 3.24 folgt, daß dies ausgehend von Konzepten in $\mathcal{F}$ nicht möglich ist. Desweiteren gibt es y, y' mit $y \in \mathcal{R}(x)$, $\neg(x\mathcal{S}y) \in \mathcal{A}_\infty$ sowie $y' \in \mathcal{R}(x')$, $\neg(x'\mathcal{S}y') \in \mathcal{A}_\infty$. Gilt dann $y \approx y'$, so folgt aus $([x]_\approx, [y]_\approx) \in \mathcal{S}^{\overline{\mathcal{I}}}$ nicht mehr die Existenz einer $\mathcal{S}$-Rollenkette von x nach y in $\mathcal{A}_\infty$ (betrachte dazu die Variablen $x_4, x_6, \ldots$ aus Abbildung 3.11). Daher läßt sich der Widerspruchsbeweis der Behauptung ($*$) von Seite 78 nicht auf ABoxen $\mathcal{A}_\infty$ übertragen, die zu beliebigen $\mathcal{ALC} + (\sqsubseteq_n)$-Konzepten erzeugt wurden.$\diamond$

Entscheidbarkeitsresultate

Mit den Aussagen aus Lemma 3.11 und Satz 3.20 folgt nun die Entscheidbarkeit des Erfüllbarkeitsproblems für $\mathcal{ALUE} + (\sqsubseteq_n)$ und das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$.

Theorem 3.27
Für das Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$ ist das Erfüllbarkeitsproblem entscheidbar, d.h., für ein gegebenes $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C_0 aus $\mathcal{F}$ ist effektiv entscheidbar, ob eine Interpretation $\mathcal{I}$ existiert mit $C_0^I \neq \emptyset$. ⌟

Beweis:
Ist C_0 unerfüllbar, so folgt mit Lemma 3.11, daß der Vervollständigungsalgorithmus nach endlich vielen Regelanwendungen mit Ausgabe „C_0 ist unerfüllbar." terminiert. Ist C_0 erfüllbar, so existiert mit Satz 3.20 ein endliches Modell zu C_0. Zu jeder Kardinalität n existieren bis auf Isomorphie nur endlich viele verschiedene Interpretationen der Konzept- und Rollennamen aus $\mathsf{Sub}(C_0)$ und zu jeder Interpretation kann man offensichtlich effektiv entscheiden, ob sie ein Modell von C_0 ist oder nicht. Zählt man also rekursiv zu jeder Kardinalität $n \geq 1$ die endlich vielen 'verschiedenen' Interpretationen auf und prüft jeweils, ob es sich um ein Modell für C_0 handelt, so findet man nach endlich vielen Schritten ein endliches Modell zu C_0, falls C_0 erfüllbar ist. □

Die folgende Definition beschreibt das Fragment von $\mathcal{ALC} + (\sqsubseteq_n)$, für das das Subsumtionsproblem durch Reduktion auf das Erfüllbarkeitsproblem für $\mathcal{F}$ entschieden werden kann.

Definition 3.28 (Das Fragment $\mathcal{ALU} + (\sqsubseteq_n)$)
Seien $\mathcal{N_C}$ eine Menge von Konzeptnamen und $\mathcal{N_R}$ eine Menge von Rollennamen.

Die Menge der *$\mathcal{ALU} + (\sqsubseteq_n)$-Konzepte* enthält alle $\mathcal{ALUE} + (\sqsubseteq_n)$-Konzepte, die über $\mathcal{N_C}$ und $\mathcal{N_R}$ aufgebaut sind und nur *nicht-qualifizierende Existenzrestriktionen* der Form $(\exists R.\mathsf{Top})$ enthalten. Dabei steht Top als Abkürzung für ein Konzept $A \sqcup (\neg A)$, $A \in \mathcal{N_C}$, wird also stets als die gesamte Trägermenge interpretiert. ◇

Theorem 3.29
Das Subsumtionsproblem für $\mathcal{ALU} + (\sqsubseteq_n)$ ist entscheidbar, d.h., für zwei Konzepte C, D aus $\mathcal{ALU} + (\sqsubseteq_n)$ kann man effektiv entscheiden, ob $C \sqsubseteq D$ gilt oder nicht. ⌟

Beweis:
Für ein $\mathcal{ALU} + (\sqsubseteq_n)$-Konzept D ist das zu $\neg D$ effektiv bestimmbare $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept D' in NNF in $\mathcal{F}$, denn:
Aus der Definition der Menge der $\mathcal{ALU} + (\sqsubseteq_n)$-Konzepte folgt:

- $\mathsf{Sub}(D)$ enthält keine negativen Role-Value-Maps und
- für jedes Konzept der Form $\exists R.C'$ aus $\mathsf{Sub}(D)$ gilt: $C' = \mathsf{Top}$, also

- enthält keine Existenzrestriktion aus $\mathsf{Sub}(D)$ ein Subkonzept der Form $(\mathcal{R} \sqsubseteq \mathcal{S})$.

Damit enthält $\mathsf{Sub}(D')$ offensichtlich keine Konzepte der Form $\forall R.C'$, wo $\mathsf{Sub}(C')$ ein Konzept $\neg(\mathcal{R} \sqsubseteq \mathcal{S})$ enthält. Also liegt das zu $C \sqcap \neg D$ gehörige $\mathcal{ALC} + (\sqsubseteq_n)$-Konzept C' in NNF in $\mathcal{F}$ und ist offensichtlich effektiv bestimmbar. Gemäß Theorem 1.11 gilt $C \sqsubseteq D$ genau dann, wenn $C \sqcap (\neg D)$ unerfüllbar ist. Mit Theorem 3.27 folgt also, daß das Subsumtionsproblem für $\mathcal{ALU} + (\sqsubseteq_n)$ entscheidbar ist. $\square$

Kapitel 4

Zusammenfassung

Die Ergebnisse dieser Arbeit

In dieser Arbeit wurden Entscheidbarkeitsprobleme für die Beschreibungslogiken $\mathcal{ALCN}(\circ)$ und $\mathcal{ALC} + (\sqsubseteq_n)$ untersucht.

Ausgangspunkt war der Beweis der Entscheidbarkeit des Erfüllbarkeitsproblems für $\mathcal{ALCN}(\circ)$ in [BS96]. Zum Beweis wurde ein regelbasiertes Verfahren angegeben, das ausgehend von einem erfüllbaren Konzeptterm C_0 ein kanonisches Modell $\mathcal{I}$ durch iteriertes Anwenden sogenannter Vervollständigungsregeln erzeugt, so daß Δ_I eine Instanz von C_0 enthält.

In Abschnitt 2.2 wurde dann versucht, auch das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ mit diesem regelbasierten Verfahren zu entscheiden. Dabei stellte man aber fest, daß es Fälle gibt, für die es unendliche Folgen von Regelanwendungen geben kann. Man benötigte also geeignete Erweiterungen für den Konsistenzalgorithmus, um Terminierung zu gewährleisten.

Für das Fragment $\mathcal{ALCN}$ wurde eine Strategie angegeben, die die Regelanwendung so steuert, daß keine unendliche Folge von Regelanwendungen ausgehend von einer beliebigen, verallgemeinerten $\mathcal{ALCN}$-ABox möglich ist. Damit wurde die Entscheidbarkeit des Konsistenzproblems für $\mathcal{ALCN}$ bewiesen.

Daher wurde versucht, durch eine ähnliche Strategie auch die Terminierung iterierter Regelanwendungen auf $\mathcal{ALCN}(\circ)$-ABoxen zu sichern. Leider wurden für zwei vielversprechende Strategien Beispiele gefunden, die zeigen, daß auch bei Einhaltung dieser Strategien unendliche Folgen von Regelanwendungen möglich sind.

Ein weiterer Ansatz, der in dieser Arbeit nicht dokumentiert ist, basierte auf der Idee, die Vervollständigungsregeln so zu modifizieren, daß Terminierung leicht nachgewiesen werden kann. Für eine entsprechende Menge von Regeln wurde dann aber der Nachweis der Vollständigkeit des Algorithmus ebenso schwierig wie der Nachweis der Terminierung für das ursprüngliche Verfahren.

Bei dem anschließenden Versuch, die Endliche-Modell-Eigenschaft für $\mathcal{ALCN}(\circ)$-ABoxen nachzuweisen, wurde festgestellt, daß es faire, unendliche Folgen von Regelanwendungen geben kann, durch die unendliche, kanonische Modelle erzeugt werden. Leider konnte auch ausgehend von einem solchen speziellen unendlichen Modell kein endliches Modell für eine konsistente ABox definiert werden.

Damit wurde der Versuch, das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ im Rahmen dieser Arbeit zu entscheiden, aufgegeben. Um dennoch ein Entscheidbarkeitsresultat zu erhalten, wurden zum Abschluß des 2. Kapitels noch Bedingungen gesucht, die eine Klasse von $\mathcal{ALCN}(\circ)$-ABoxen charakterisieren, für die das Konsistenzproblem durch den Algorithmus entschieden wird. Dazu mußte die Terminierung jeder Folge von Regelanwendungen ausgehend von einer ABox aus dieser Klasse gesichert sein. Der Terminierungsbeweis zum Erfüllbarkeitsalgorithmus für $\mathcal{ALCN}(\circ)$ in [BS96] basiert auf der Beobachtung, daß jede aus $\{x_0 : C_0\}$ abgeleitete ABox eine Levelstruktur besitzt. Deshalb definierte man alle verallgemeinerten $\mathcal{ALCN}(\circ)$-ABoxen als zulässig für eine iterierte Regelanwendung, die eine Levelstruktur besitzen. Dabei ist für eine gegebene $\mathcal{ALCN}(\circ)$-ABox effektiv entscheidbar, ob sie zulässig ist. Für zulässige ABoxen ließ sich der Terminierungsbeweis aus [BS96] übertragen. Man erhielt also für eine spezielle, entscheidbare Klasse von $\mathcal{ALCN}(\circ)$-ABoxen das gewünschte Entscheidbarkeitsresultat.

Im 3. Kapitel wurde das Erfüllbarkeits- und Subsumtionsproblem für Erweiterungen von $\mathcal{ALC}$ um Role-Value-Maps mit Rollenkomposition untersucht. Mit den Ergebnissen aus [Sch89] wurde zunächst gezeigt, daß Erfüllbarkeit für $\mathcal{ALC} + (\sqsubseteq)$ unentscheidbar ist. Motiviert durch die Erkenntnis, daß das Fragment $\mathcal{ALC} + (\sqsubseteq_n)$ die Levelstruktur-Eigenschaft besitzt, wurde ein regelbasiertes Verfahren angegeben, das Erfüllbarkeit für $\mathcal{ALC} + (\sqsubseteq_n)$ entscheiden sollte.

Der erhaltene Erfüllbarkeitsalgorithmus wurde als widerlegungsvollständig nachgewiesen, aber es wurden wiederum Beispiele angegeben, die zeigen, daß die Terminierung einer Folge von Regelanwendungen im allgemeinen nicht gewährleistet ist. Um dennoch ein Entscheidbarkeitsresultat zu erhalten, griff man erneut die Idee auf, direkt die Endliche-Modell-Eigenschaft zu zeigen. Dies gelang für ein geeignet definiertes Fragment $\mathcal{F}$ von $\mathcal{ALC} + (\sqsubseteq_n)$. Es folgt die Entscheidbarkeit des Erfüllbarkeitsproblems für das Fragment $\mathcal{F}$ und daraus wiederum die Entscheidbarkeit des Subsumtionsproblems für $\mathcal{ALU} + (\sqsubseteq_n)$.

Ausblick

In [BS96] wird bewiesen, daß das Erfüllbarkeitsproblem für fast alle Erweiterungen von $\mathcal{ALCN}(\circ)$ um die Rollenoperatoren Disjunktion, Konjunktion und inverse Rollen unentscheidbar ist. Lediglich für $\mathcal{ALCN}(\circ, \sqcup)$ und $\mathcal{ALCN}(\circ, \cdot^{-1})$ ist das Erfüllbarkeitsproblem noch offen.

Bei dem Versuch, das Erfüllbarkeitsproblem für $\mathcal{ALCN}(\circ, \sqcup)$ mit Hilfe eines regelbasierten Verfahrens zu entscheiden, trifft man auf ähnliche Probleme wie bei der Untersuchung des Konsistenzproblems für $\mathcal{ALCN}(\circ)$. In beiden Fällen ist Terminierung im allgemeinen nicht gewährleistet und in beiden Fällen scheint es nicht möglich, ein unendliches bzw. beliebig großes Modell zu erzwingen, so daß ein bekanntes unentscheidbares Problem auf das Erfüllbarkeitsproblem für $\mathcal{ALCN}(\circ, \sqcup)$ oder das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ reduziert werden könnte.

Diese Arbeit beinhaltet einige Aspekte, die in weiterführenden Arbeiten zu untersuchen bleiben. Zunächst bleibt natürlich das Konsistenzproblem für $\mathcal{ALCN}(\circ)$ sowie das Erfüllbarkeits- und Subsumtionsproblem für $\mathcal{ALC} + (\sqsubseteq_n)$ offen. Der vorgestellte Erfüllbarkeitsalgorithmus für $\mathcal{ALCN}(\circ)$ ist EXPTIME und die Frage nach einem PSPACE-Verfahren ist ebenfalls offen ([BS96]).

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meines Studiums sowie bei der Erstellung meiner Diplomarbeit unterstützt haben.

Insbesondere danke ich meinem Vater für die Finanzierung meines Studiums und meiner Mutter für die 'allwochenendliche' Fürsorge, durch die ich für jede Woche meines Studiums neu gerüstet war.
Sowohl meinen Eltern als auch meinen Geschwistern Axel und Anne danke ich dafür, daß sie stets auf den nervenden und gestreßten Studenten Rücksicht genommen haben und mich immer wieder, bewußt oder unbewußt, motiviert und aufgebaut haben.

Mein Dank gilt außerdem allen Mitarbeitern des LuFG Theoretische Informatik. Ich danke Herrn Prof. Baader und Ulrike Sattler für die fachlichen Tips und Hilfen, ohne die mir die Erarbeitung dieser Diplomarbeit nicht möglich gewesen wäre. Ganz besonders möchte ich mich bei Ulrike Sattler für die vielen, stundenlangen Diskussionen über Regeln, Strategien und 'böse' Beispiele bedanken. Jörn Richts und Can Adam Albayrak danke ich für die schnelle und ausführliche Hilfe bei allen TeXnischen Problemen. Christopher Tresp und allen anderen Mitspielern danke ich außerdem für die (ent-)spannenden Runden Darts. Madjid Nassiri danke ich für die anregenden Diskussionen, die wir während der gemeinsamen Diplomandenzeit führen konnten.

Zum Schluß danke ich allen Freunden, die auch in den anstrengendsten Phasen meines Studiums zu mir gehalten haben. Namentlich möchte ich hier nur Ralf Küsters und Jörg Köller sowie Bernd Backhaus, Christian de Renet, Jörn Tschentscher und Marco Wischmeier erwähnen und mich für viele unterhaltsame, feucht-fröhliche und anregende Stunden bedanken, die wir zusammen verbracht haben.

Danke.

Literaturverzeichnis

[Baa96] F. Baader: Logik-basierte Wissensrepräsentation. *KI*, 3/96:8–16, 1996.

[BBH94] F. Baader, M. Buchheit und B. Hollunder: Cardinality Restrictions on Concepts. In: *Proceedings of the German AI Conference, KI'94*, Band 861 der Reihe *Lecture Notes in Computer Science*, Saarbrücken (Germany), 1994. Springer–Verlag.

[BHs91] Franz Baader und Philipp Hanschke: A Schema for Integrating Concrete Domains into Concept Languages. In: *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, Seiten 452–457, Sydney, 1991.

[Boo59] W.W. Boone: The Word Problem. *Ann. Math.*, 2(70):207–265, 1959.

[Bor96] A. Borgida: On the Relative Expressiveness of Description Logics and Predicate Logics. *Artificial Intelligence Journal*, 82:353–367, 1996.

[BS85] R. J. Brachman und J. Schmolze: An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2):171–216, 1985.

[BS96] F. Baader und U. Sattler: Number Restrictions on Complex Roles in Description Logics. In: *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96)*, Seiten 446–451. Morgan Kaufmann, Los Altos, 1996.

[HB91] B. Hollunder und F. Baader: Qualifying Number Restrictions in Concept Languages. In: *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, KR-91*, Seiten 335–346, Boston (USA), 1991.

[Hol94] B. Hollunder: *Algorithmic Foundations of Terminological Knowledge Representation Systems.* Doktorarbeit, Universität des Saarlandes, 1994.

[Sch89] M. Schmidt-Schauss: Subsumption in KL-ONE is Undecidable. In: *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning (KR-89)*, Seiten 421–431, Boston (USA), 1989.