Monadische Logik 2. Ordnung Ausarbeitung zum Seminarvortrag im Rahmen der Seminarveranstaltung Logische Aspekte von XML SS03 Gert Smolka PS-Lab Universität des Saarlandes

Ilhan Aslan

October 22, 2003

Monadische Logik(MSO) ist auf Büchi zurück zuführen, der in den 60er die erste Definition zu dieser Logik lieferte, MSO ist entscheidbar und sehr expressive. Büchi zeigte zu der Zeit die Korrespondenz zu Endlichen Automaten! Im Folgenden wird die Syntax von Monadischer Logik genauer betrachtet, dabei wird auch versucht die Einschränkungen die MSO mit sich bringt zu erklären. Im Rahmen der Seminarveranstaltung ist dieses Thema in sofern sehr wichtig, weil man mit Mondadischer Logik 2. Ordnung Eigenschaften von Endlichen wie auch von Unendlichen Bäumen beschreiben kann. D.h. man kann Bäume in MSO kodieren! Die Struktur eines XML-Dokuments ist nichts anderes als eine Baumstruktur. Das XML-Dokument selber ist ein gelabelter Baum! Die Anordnung der Label kann man als eine Eigenschaft bzw. Kodierung betrachten.

Inhaltsangabe

Logik 1.Ordnung vs. 2.Ordnung vs. 3.Ordnung

In der Logik 2. Ordnung gibt es 2 Varianten von Variablen. Zusätzlich zu den Individuen-Variablen aus der Logik 1. Ordnung, die auf einzelne Individuen des Universums abgebildet werden gibt es die so genannten Mengen-Variablen. Diese Mengen-Variablen werden nicht mehr auf einzelne Individuen, sondern auf Mengen von Individuen abgebildet! Wenn man diesen Vorgang weiter anhebt und Variablen auf Mengen von Mengen von Individuen-Variablen abbildet, bekommt man Variablen der 3.Ordnung. In der Logik 3. Ordnung hat man somit 3 Varianten von Variablen! Man kann zeigen das Formeln mit Variablen 3.Ordnung unentscheidbar sind. Jede Logik die höher als die Logik 3. Ordnung ist, ist damit auch unentscheidbar. Aus diesem Grund wird eine Logik die entscheidbar sein muss sich auf Variablen 1. und 2. Ordnung einschränken. Wenn man zusätzlich von der Logik verlangt dass sie möglichst expressive ist, endet die Suche bei der Logik 2. Ordnung!

Syntax von MSO

MSO steht für monadic second order logic. Ausserdem gibt es in der Literatur noch die Bezeichnungen S1S, WS1S, S2S, WS2S ... . Dabei steht das W für weak(schwach), bei schwachen Definitionen der Monadischen Logik 2. Ordnung werden Mengen-Variablen auf endliche Mengen eingeschränkt. Die Zahl zwischen den Beiden S steht für die Anzahl der Nachfolgerfunktion in der jeweiligen Definition. Gegeben ein Alphabet kann man mit einer Nachfolgerfunktion Eigenschaften von Wörtern beschreiben. Eine Eigenschaft für ein Wort aus einem Alphabeten mit dem Zeichen a wäre z.B. dass dieses Wort ein jeder zweiten Position ein a stehen hat. Man beachte das jede Position in einem Wort mit einer einzigen Nachfolgerfunktion beschrieben werden kann. Dabei gilt das die Anfangsposition im Wort die Position ist, die kein Nachfolger ist und die Position k kann man dann durch k-mal anwenden der Nachfolgerfunktion auf die Anfangsposition erreichen. Mit zwei Nachfolgerfunktionen kann Eigenschaften von binären Bäumen beschreiben usw. . Es ist nicht notwendig Logiken mit mehr als 2 Nachfolgerfunktioenen zu betrachten, da man Eigenschaften von k-nären äumen mit binären kodieren kann.

Eine Minimal Syntax für die Monadische Logik ist:

$$\psi = \forall X \psi \mid \neg\psi \mid \psi \vee \phi \mid S_i(X,Y) \mid X \subseteq Y$$

In dieser Definition werden nur Mengen-Variablen benutzt. Man hätte gerne das $S_i(X,Y)$ bedeutet das X und Y Mengen sind, die genau ein Element besitzen und dass das Element in Y der Nachfolger vom Element in X. Da man das Prädikat in der Definition nicht nur auf einelementige Mengen beschränken darf, definiert man das Prädikat folgendermassen: das groesste Element in X ist Nachfolger vom groessten Element in Y! Man kann zusätzlich ein Prädikat Einelementig definieren das bestimmt ob eine Menge X nur aus einem Element besteht.

$$Einelementig(X) :\Leftrightarrow \exists Y((Y \subseteq X) \wedge \neg(Y = X) \wedge \neg\forall Z(Z \subseteq X \leftarrow Z = X \vee Z = Y))$$

Mit diesen beiden Definitionen hat man das was man eigentliche wollte. Mit der Negation und dem logischen und, kann man das logische oder definieren, genauso kann man mit der Negation und dem Allquantor den Existenzquantor definieren.

Variationen in der Syntax

In der Literatur findet man oft unterschiedliche Definitionen von Monadischer Logik 2. Ordnung. Oft hat man 2 Varianten von Variablen, die Individuen-Variablen und die Mengen-Variablen. Wenn man Individuen Variablen zulässt, kommen zusätzlich zu der Minimal-Definition die Quantoren 1.Ordnung, so nennt die Quantoren über Individuen-Variablen. Wenn man einmal Individuen-Variablen hat, kann man die Nachfolgerfunktion mit Mengen-Variablen streichen und stattdessen die Nachfolgerfunktion auf Individuen-Variablen definieren. Ausserdem kann man die Teilmengen-Relation auf Mengen-Variablen durch die Element-von-Relation ersetzen. Damit hat man eine alternative Definition der Monadischen Logik 2. Ordnung. Eine dritte Möglichkeit ergibt sich wenn man anstatt der Nachfolgerfunktion in S1S eine kleiner-Relation einsetzt. Diese Option ist für SkS mit k grösser 1 nicht möglich.

$$S(x,y) :\Leftrightarrow (x < y \wedge \neg \exists z (x < z \wedge z < y))$$

Semantik von MSO

Die Interpretation von Variablen ist ein wichtiger Bestandteil der Semantik von MSO. Da man hier 2 unterschiedliche Varianten von Variablen hat. Abhängig von der Anzahl der Nachfolgerfunktionen in der Definition der Monadischen Logik werden die Variablen über einem Univerum interpretiert der aus einem, zwei oder mehreren Elementen aufgebaut ist. Bei einem Nachfolger und dem Element 1 besteht das Universum aus e, 1, 11, 111, 1111, ... Man betrachtet diese Elemente als Grundstrukturen(Shapes) von Wörtern. Jedes Wort hat eine Anfangspostion(1) eine Nachfolgerposition(11) usw. . Wenn man etwas genauer hinschaut erkennt man eine Isomorphie zwischen dem Universum und den Natürlichen Zahlen. Bei zwei Nachfolgerfunktionen wird das Universum aus zwei Elementen 1 und 2 aufgebaut. Das Universum besteht dann aus e, 1, 2, 11, 22, 12, 21, 111, ... Diese kann man als Grundstrukturen(Shapes) von Pfaden in binären Bäumen betrachten. Jedes Element aus dem Universum identifiziert dabei eine Pfadstruktur(eine Position im binären Baum). In einem binären Baum hat ein Knoten eine eindeutige Position die mit einer Folge von einsen und zweien idetifiziert wird. Es ist zum Beispiel ein Knoten auf Position 121 linkes Kind vom rechten Kind vom linken Kind der Wurzel. Bei k Nachfolgerfunktionen mit k grösser 2 läuft das Ganze analog. Die Individuen-Variablen werden dabei auf einzelne Shapes(oder Pfade) abgebildet, während Mengen-Variablen auf Mengen von Shapes(oder Pfaden) abgebildet werden. Bei (W)S2S kann eine Menge von Pfaden einen Baum definieren, wenn diese Menge Prefixabgeschlossen ist d.h. falls z.B. 121 in der Menge ist muss auch 12, 1 und e in der Menge sein.

Erfüllbarkeit in MSO

An dieser Stelle werden nur die Atomaren Formeln betrachtet die spezifisch für die Monadische Logik sind.

Sei β eine Belegungsfunktion die Variablen auf Elemente aus dem Universum bzw. Mengen von Elementen aus dem Universum abbildet.

$1.\beta \models X \in X \Leftrightarrow \beta(x) \in \beta(X)$
$2.\beta \models S_1(x, y) \Leftrightarrow \beta(x)1 = \beta(y)$
$3.\beta \models S_2(x, y) \Leftrightarrow \beta(x)2 = \beta(y)$
$4.\beta \models \exists X\psi \Leftrightarrow \beta[B/X] \models \psi$
für eine Menge von Pfaden B.

Die 1. Formel, x ist-Element-von X ist erfüllbar, wenn das Element auf das x abgebildet wird in der Menge von Elementen ist, auf die X abgebildet wird. Das Element im Universum auf das x abgebildet wird ist 1. Nachfolger von dem Element das auf y abgebildet wird, wenn das Element das auf x abgebildet wird konkatiniert mit 1 das Element ergibt das auf y abgebildet wird. In diesem Fall ist die Formel y ist-erster-Nachfolger-von x erfüllbar. Analog läuft das für die k-te Nachfolgerfunktion für k grösser 2. Die fünfte Formel ist erfüllbar wenn eine Substitution exitiert unter dessen Einsatz die Formel erfüllbar wird. Die Erfüllbarkeit von allen anderen atomaren Formeln ist analog zu der Erfüllbarkeit von Atomaren Formeln in der allgemeinen Prädikatenlogik (First Order Logic). Eine Formel ist in Monadischer Logik 2. Ordnung ist gültig, wenn sie unter allen Belegungen erfüllbar ist.

Prädikate in Monadischer Logik 2.Ordnung

Es existiert eine Beziehung zwischen der Definition eines Prädikaten und einer Teilmenge aus dem Universum. Dabei gilt, zu jeder Teilmenge des Universums kan man ein charakteristisches Prädikat definieren, indem man das Prädikat auf 1 abbildet für Elemente aus der Menge und sonst wird auf 0 abgebildet. Umgekehrt gibt es zu jedem Prädikaten eine charakteristische Menge, in der Menge sind die Elemente aus dem Universum für die das Prädikat auf 1 auswertet, alle anderen Elemente werden nicht in die charakteristische Menge aufgenommen. Mit dieser Idee kann man nun leicht zeigen wie man Wörter bzw. Bäume in Monadischer Logik 2. Ordnung kodieren kann. Dabei beachte man das Mengenvariablen für Teilmengen aus dem Universum stehen, wie oben erklärt gibt es fü jede dieser Mengen eine charakteristisches Prädikat. D.h. In der Monadischen Logik 2. Ordunung quantifizieren wir indirekt Prädikate.

Kodiere Strings in S1S

In S1S kann man unendliche Wörter kodieren. Ein Beispiel-Wort ist das Wort aus dem Alphabeten das nur aus den Zeichen a und b besteht, dass an jeder geraden Position ein a und an jeder ungeraden Position ein b hat. Jede Position auf der ein a steht ist in der Menge La und jede Position auf der ein b steht ist in der Menge Lb. Allgemein kann man so für jedes Zeichen aus dem Alphabeten eine Menge definieren. Diese Mengen bestehen aus Elementen des Universums und haben deshalb auch charakteristische Prädikate.

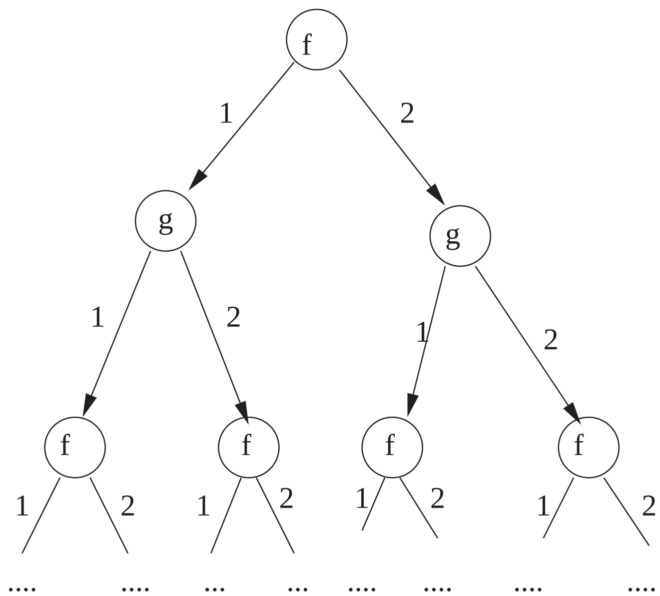
	e	1	11	111	...
	a	b	a	b	...
La	1	0	1	0	...
Lb	0	1	0	1	...

Es ist erkennbar das diese Mengen disjunkt sein müssen weil auf einer Position nicht zwei Zeichen stehen können. Es muss auch gelten dass die

Vereinigung der Mengen gleich dem Universum sind. D.h. jede Position im Wort ist mit genau einem Zeichen aus dem Alphabet belegt.
In WS1S kann man nur endliche Wörter kodieren. Wie bei S1S gilt das der Schnitt der Labelmengen(im Beispiel sind as La und Lb) leer sein muss, d.h. an jeder Position im Wort steht nur ein Zeichen, aber die Vereinigung ergibt nicht mehr das ganze Universum. Um auszusagen das an jeder Position im Wort ein Zeichen steht muss man Verlangen das die Vereinigung der Labelmengen prefix-abgeschlossen ist.

Kodiere Bäume in S2S

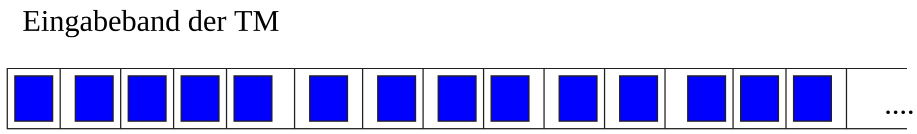
In der Monadischen Logik mit 2 Nachfolgerfunktionen kann man Binäre unendliche-Bäume kodieren. Ein Beispiel-Baum, über dem Alphabeten mit den Zeichen f und g mit den Aritäten 2, ist der Baum der an den Knoten deren Tiefe eine gerade Zahl ist ein f stehen hat und an den Knoten deren Tiefe eine ungerade Zahl ist ein g stehen hat. In der Labelmenge Lf stehen damit alle Elemente (Pfade) aus dem Universum, die eine gerade Länge haben und alle Elemente die eine ungerade Länge haben stehen in der Labelmenge Lg.

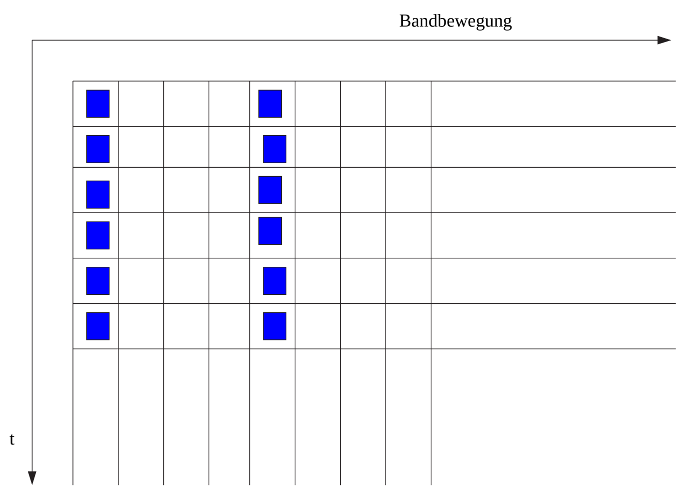


Auch hier gilt, dass der Schnitt der Labelmengen leer sein muss. Die Vereinigung der Labelmengen ist wiederum das ganze Universum. Zusätzlich muss noch gelten das die Vereinigung der Labelmengen nicht leer sein darf, weil es per Definition keinen leeren Baum gibt. Wenn man WS2S betrachtet hat man wie bei S1S und WS1S die Bedingung der prefix-abgeschlossenheit der Vereinigung der Labelmengen und es muss die Aritätenkonsistenz verlangt. D.h. Ein Zeichen mit Arität 0 darf keine Kinder-Knoten besitzen genauso wie eine Zeichen mit Arität 2 nicht mehr oder weniger Kinder-Knoten haben darf.

Die Unentscheidbarkeit der Logik mit binären Prädikaten

Eine Einschränkung in der Monadischen Logik ist das man nur einstellige (monadische) Prädikate zulässt. Diese Einschränkung ist deshalb notwendig, weil eine Logik mit binären Prädikaten unentscheidbar ist! Wenn man die Formeln über den Natürlichen Zahlen interpretiert kann man ein binäres Prädikat betrachten als ein 2 Dimensionales Gitter mit Einsen an den Stellen an denen das Prädikat zu 1 auswertet und Nullen an den Stellen an den es zu falsch auswertet. Solch ein Gitter kann man als einen unendlich nach oben und unendlich nach rechts laufenden Speicher betrachten! Mit diesem Speicher kann man den Lauf einer Turing Maschine simulieren. Dies geschieht indem man die horizontalen Bewegung auf dem Eingabeband der Turing Maschine mit horizontalen Bewegungen auf dem Gitter darstellt, während vertikale Bewegungen die zeitliche Folge darstellen.

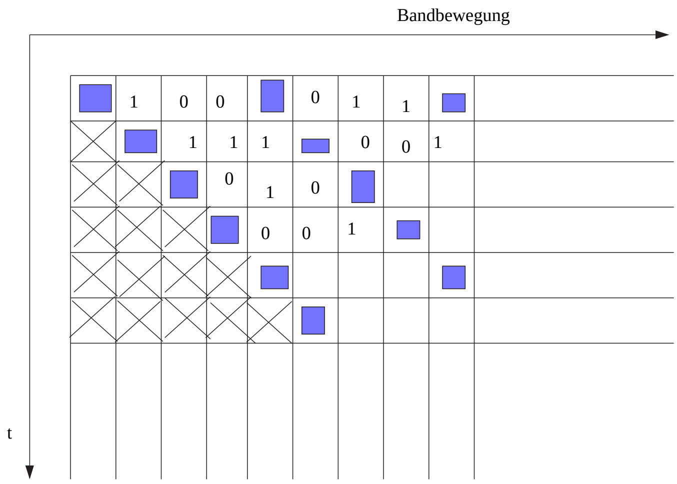




Die Bewegungen auf dem Band werden durch Bitmusterübergänge dargestellt. Diese Bitmuster haben eine feste endliche Länge, man kann sie mit einem binären Prädikat kodieren. Für jedes zu Kodierende Bit besteht die Formel aus einem auftreten des Prädikats. Die Prädikate sind mit logischen Und-Operatoren verknüpft. Wir können also den Lauf einer Turing Maschine simulieren, insbesondere können wir das Halte-Problem simulieren und deshalb ist die Logik mit binären Prädikaten unentscheidbar!

Die Unentscheidbarkeit der Logik 3. Ordnung

Die Monadische Logik 2.Ordnung ist eine Logik von 2. Ordnung weil eine Logik 3. Ordnung unentscheidbar ist! In der Logik dritter Ordnung haben wir Prädikate über Mengen-Variablen. Mann kann ein solches Prädikat auf zwei-elementige Mengen einschränken, indem man analog zum Prädikat Einelementig ein Prädikat Zweielementig definiert womit man dann wiederum ein halbes Gitter simulieren kann. Man kann nur ein halbes Gitter definieren weil man nun zweielementige Mengen betrachtet (es gilt 1,2 gleich 2,1) im vergleich zu Tupeln wo (1,2) ungleich (2,1) ist. Das ergebnis ist ein linear nach rechts verschobenes Gitter, es ist aber dennoch nach unten und rechts unendlich. Man muss die Bitmusterregeln aus dem Beispiel mit binären Prädikaten leicht verändern um das gleiche zu bekommen.



Referenzen

[1] Wolfgang Thomas,Languages, Automata, and Logic, May 1996,Bericht 9607 Institut für Informatik und Praktische Mathematik Der Christian-Albrechts-Universität zu Kiel D-24098 Kiel

[2] Khoussainov, Bakhodyr and Nerode,Anil: Automata Theory and Its Applications, Progress in Computer Science, Birkhäuser,Boston;Berlin(2001)

[3] David Basin, Felix Klaedke, Monadic Second-Order Logics in Theory and Practice, Albert-Universität, Freiburg

[5] Hupert Comon, Max Dauchet, Rmi Gilleron, Denis Lugiez, Sophie Tison, Marc Tommasi, Tree Automata Techniques and Applications

[6] Erich Grädel, Wolfgang Thomas, Thomas Wilke (Eds.), Automata, Logics, and Infinite Games, A Guide to Current Research (Part VI Monadic Second-Order Logic), Springer (2001)

[7] H.Hermes, Einführung in die Mathematische Logik, B.G. Teubner, Stuttgart (1972)