

REVISED

12:36 pm, Mar 18, 2010

Evaluation eines Ansatzes zur Muster-basierten Spezifikation von Konsistenzkriterien für Web-Dokumente

Mirjana Jakšić

Fakultät für Informatik und Mathematik, Universität Passau

Mirjana.Jaksic@uni-passau.de



Technischer Bericht, Nummer MIP-0906

Fakultät für Informatik und Mathematik

Universität Passau

März 2009

Zusammenfassung

In diesem Bericht werden erste Evaluationsergebnisse für eine neu entwickelte Methode zur Muster-basierten Spezifikation von Konsistenzkriterien für Web-Dokumente präsentiert. Es wird gezeigt, dass temporale Kriterien von unerfahrenen Nutzern mit wesentlich größerem Erfolg spezifiziert werden können als bei einer direkten Formulierung in einer temporalen Logik.

1 Einführung

Im Rahmen des Verdikt-Projekts [4] wird eine Methode zur automatischen Verifikation der Konsistenz von Web-Dokumenten entwickelt. Die Verifikation wurde durch Model-Checking implementiert, und die Kriterien werden mit der temporalen Beschreibungslogik \mathcal{ALCCTL} ausgedrückt [3]. Dabei wurde unter anderem untersucht, wie die Endnutzer bei der Formulierung der Konsistenzkriterien unterstützt werden können, weil die formale Logik für sie in der Regel ungeeignet ist. Zur Unterstützung der Nutzer wurden Spezifikationsmuster vorgeschlagen [1].

In diesem Bericht präsentieren wir die Ergebnisse der Evaluation der entwickelten Methode zur Muster-basierten Spezifikation von Konsistenzkriterien für Web-Dokumente. Wir möchten zeigen, dass die Anwendung unserer Spezifikationsmuster bei unerfahrenen Nutzern zu deutlich besseren Ergebnissen führt, als die direkte Anwendung der temporalen Logik CTL. Wir haben uns für die temporale Logik CTL statt für die temporale Beschreibungslogik \mathcal{ALCCTL} entschieden, weil CTL gebräuchlicher in anderen Anwendungsgebieten ist.

In Abschnitt 2 werden die Ziele der Evaluation kurz dargestellt. Der Test wird in Abschnitt 3 ausführlich in den folgenden Unterabschnitten beschrieben: Gruppenzusammensetzung (Abschnitt 3.1), Aufgabenzusammensetzung (Abschnitt 3.2), Testausführung (Abschnitt 3.3) und Testbewertung (Abschnitt 3.4). Die Ergebnisse werden in Abschnitt 4 präsentiert. Der Bericht endet mit einem Fazit und Ausblick (Abschnitt 5).

2 Ziele der Evaluation

Die Methode der Muster-basierten Spezifikation von Konsistenzkriterien für Web-Dokumente wurde im Vergleich zur Spezifikation mit der temporalen Logik CTL evaluiert. Die Testpersonen kannten vorher weder CTL noch die Spezifikationsmuster. Der Test wurde mit zwei Gruppen unter vergleichbaren Voraussetzungen durchgeführt. Die beiden Gruppen sollten die gleichen natürlichsprachlich formulierten Kriterien spezifizieren. Die erste Gruppe sollte die Kriterien mittels CTL formulieren, die zweite Gruppe mit Hilfe der Spezifikationsmuster.

Die Ziele der Evaluation waren:

- zu zeigen, dass bei vergleichbaren Voraussetzungen die Anwendung unserer Spezifikationsmuster bei unerfahrenen Nutzern zu deutlich besseren Ergebnissen führt, als die Anwendung der temporalen Logik CTL.
- festzustellen, ob und in welcher Hinsicht die natürlichsprachlich formulierten Kriterien unterschiedlich verstanden und interpretiert werden können.

3 Testbeschreibung

3.1 Gruppenzusammensetzung

Die Evaluation wurde mit 51 Studierenden der Universität Passau aus verschiedenen Studiengängen durchgeführt (Tabelle 1), die sich freiwillig zur Verfügung stellten. Die Teilnehmer wurden in zwei Testgruppen mit je 26 bzw. 25 Personen aufgeteilt und kannten vor dem Test weder CTL noch Spezifikationsmuster.

Studiengang	Gruppe 1	Gruppe 2
Bachelorstudiengang Business Administration and Economics	13	10
Bachelorstudiengang Kulturwirtschaft / International Cultural and Business Studies	4	10
Bachelorstudiengang Internet Computing	3	1
Bachelorstudiengang Wirtschaftsinformatik	4	0
Bachelorstudiengang Medien und Kommunikation	0	1
Masterstudiengang Informatik	1	0
Studiengang Rechtswissenschaft	0	1
Lehramt an Grundschulen	0	1
Lehramt an Realschulen	0	1
Lehramt an Gymnasien	1	0
Insgesamt	26	25

Tabelle 1: Die Anzahl der Teilnehmer pro Studiengang

3.2 Aufgabenzusammensetzung

Als Testdokument wurde eine Bedienungsanleitung für eine digitale Kamera, die auf der Web-Seite des Herstellers zu finden ist, ausgewählt. Der Name des Herstellers, der Markenname und die Abbildungen wurden verfremdet, um das Dokument zu anonymisieren. Das Dokument wurde auch um ca. ein Drittel gekürzt, so dass es für die Testpersonen überschaubar blieb.

Die temporale Struktur des Testdokuments (formal beschrieben in [1]) wird in Abbildung 1 dargestellt. Die einzelne Abschnitte des Dokuments (*Titelseite*, *Inhalt*, *Lieferumfang*, ..., *Index*), die sich in einzelnen HTML Dateien befinden, sind als Zustände ($s_0, s_1, s_2, \dots, s_{12}$) repräsentiert. Pfeile repräsentieren Links. Der Pfeil von s_0 zu s_1 bedeutet beispielsweise, dass ein Link zwischen *Titelseite* und *Inhalt* existiert. Jeder Zustand wird durch wichtige Informationen in Form von geltenden Aussagen annotiert. Zum Beispiel gilt in Zustand s_5 , dass es um den Abschnitt *Vorbereitungen* geht und die folgenden Themen vermittelt werden: *Einsetzen der Akkus*, *Einsetzen der Speicherkarte*, *Herausnehmen der Speicherkarte*, *Einstellung des Datums* und *Einstellung der Sprache*.

Die beiden Gruppen sollten die folgenden natürlichsprachlich formulierten Kriterien spezifizieren:

1. Auf allen Pfaden müssen *Wichtige Hinweise* vorkommen.
2. Jeder Begriff aus dem Index muss vorher beschrieben werden.
3. Für jede Aufnahmefunktion wird in der unmittelbaren Folge eine entsprechende Anzeigefunktion beschrieben.
4. Nach den *Vorbereitungen* dürfen keine Funktionen mehr beschrieben werden.
5. Jede Komponente aus dem *Überblick* muss später erwähnt werden, aber noch vor dem *Index*. Die Komponente darf nicht erst im *Index* erwähnt werden.

Die Kriterien 1 und 4 sind nicht erfüllt, während 2, 3 und 5 erfüllt sind. Die Kriterien 2 und 5 sind aber manuell nur mit großem Aufwand zu überprüfen.

3.3 Testausführung

Der Test wurde für jede Testgruppe separat durchgeführt. Die beiden Gruppen bekamen am Anfang eine kurze Einführung in das Verdikt-Projekt und die Testumgebung (ca. 10 Minuten). Die erste Gruppe bekam dann eine kompakte Schulung in CTL (ca. 45 Minuten). Der Kurs war eher praktisch als theoretisch orientiert. CTL -Syntax und -Semantik wurden nicht formal, sondern intuitiv (graphisch und beispielhaft) erläutert. Einige gezeigte Beispiele hatten eine ähnliche Struktur wie die Testaufgaben. Nach einer kurzen Pause bekamen die Testteilnehmer die fünf natürlichsprachlich formulierten Kriterien (siehe Abschnitt 3.2) als Testaufgaben. Die Testteilnehmer sollten selbstständig die gegebenen Kriterien mittels CTL Formeln ausdrücken. Als Hilfsmittel bekamen sie sowohl eine ausgedruckte Version des Testdokuments als auch eine Übersicht über die CTL Operatoren.

Die zweite Gruppe bekam nach der Einführung ins Verdikt-Projekt eine Unterweisung über die Struktur und Bedeutung der Spezifikationsmuster, sowie Beispiele zu deren Nutzung (ca. 30 Minuten). Einige gezeigte Beispiele für die Anwendung

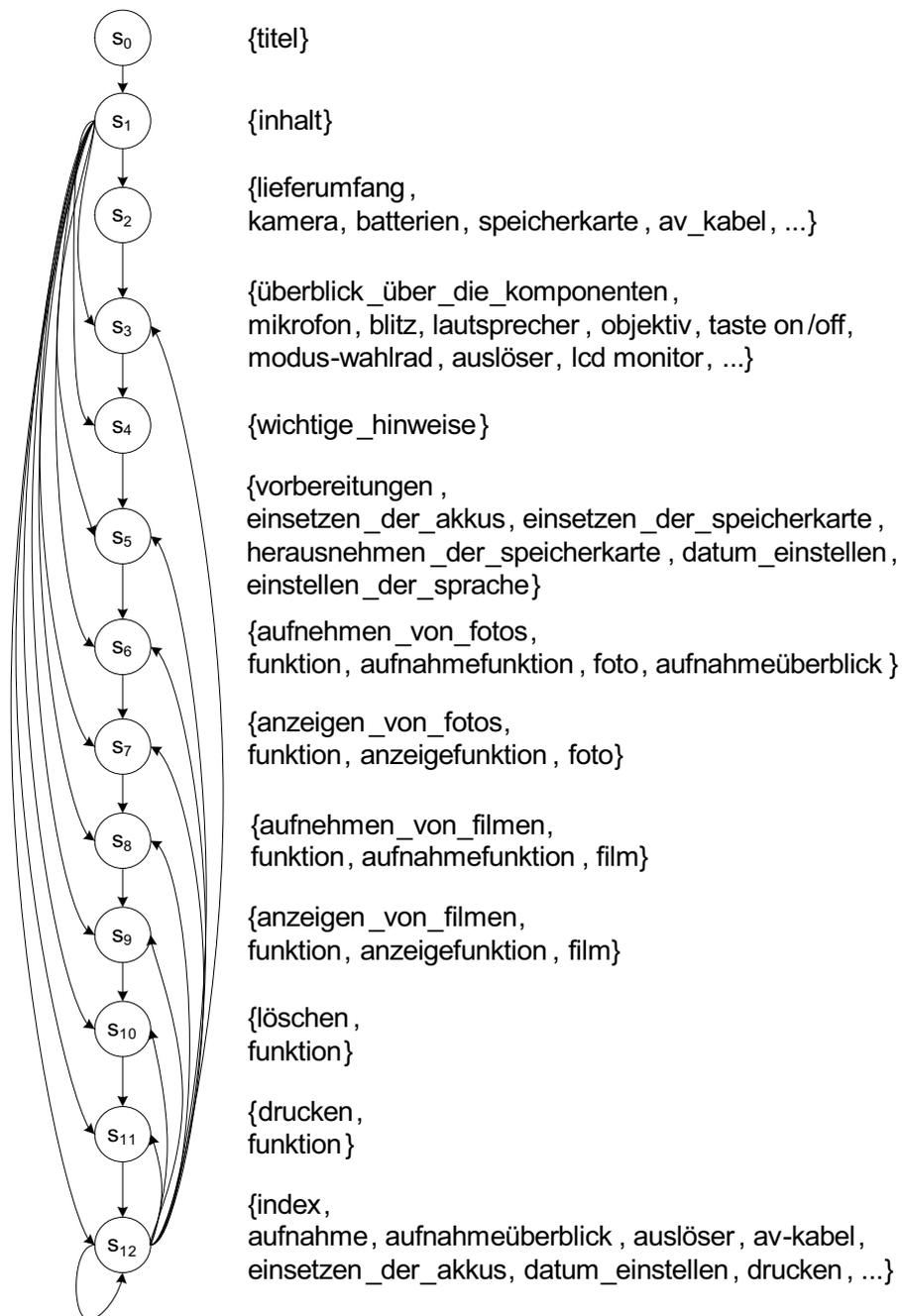


Abbildung 1: Temporale Struktur des Testdokuments

der Spezifikationsmuster waren ähnlich zu den Testaufgaben. Nach einer kurzen Pause bekamen die Testteilnehmer die fünf natürlichsprachlich formulierten Kriterien (siehe Abschnitt 3.2) zum Spezifizieren. Als Hilfsmittel bekamen sie eine

ausgedruckte Version des Testdokuments und eine Liste mit allen Spezifikationsmustern und deren Bedeutung (siehe Anhang A). Die Testteilnehmer sollten erkennen, welche der angebotenen Muster (siehe Anhang A) am besten zu dem jeweiligen natürlichsprachlich formulierten Kriterium passt und mit welchen Aussagen die Spezifikationsmuster-Parameter instanziiert werden sollten. Zum Beispiel passt zu Kriterium 1 das Muster Nummer 19 am besten, und der einzige Parameter P dieses Musters sollte mit der Aussage *wichtige_hinweise* instanziiert werden. Die beiden Gruppen bekamen auch auf Wunsch eine persönliche Hilfestellung bei der Instanziierung von Variablen (bei CTL) und Parametern (bei den Spezifikationsmustern), weil die Instanziierung nicht der Schwerpunkt der Evaluation war. Beide Gruppen hatten 60 Minuten Zeit, um die Aufgaben zu lösen.

3.4 Testbewertung

Wir haben die Antworten als entweder brauchbar oder unbrauchbar bewertet. Eine Spezifikation ist brauchbar, wenn bei der Verifikation keine "false positive" Ergebnisse heraus kommen. Gleichzeitig sind "false negative" Ergebnisse bei brauchbaren Spezifikationen erlaubt. D.h. zu strenge Kriterien sind brauchbar und zu wenig strenge Kriterien sind unbrauchbar.

Bei der Bewertung der Ergebnisse der zweiten Gruppe haben sich folgende Besonderheiten gezeigt. Da die natürlichsprachlich formulierten Kriterien nicht präzise genug ausdrückbar sind, gibt es bei Kriterien 1, 2 und 4 mehrere richtige Lösungen. Bei Kriterium 5 haben wir einige Lösungen als brauchbar bewertet, die im Bezug zu dem Testdokument richtige Verifikationsergebnisse ergeben, obwohl sie im allgemeinen nicht mit dem genannten Kriterium korrespondieren würden.

4 Ergebnisse

Nach der Testbewertung hat es sich bestätigt, dass es für unerfahrene Benutzer deutlich leichter ist, die Kriterien mit Spezifikationsmustern, als mit CTL Formeln zu formulieren. Trotz vergleichbarer Voraussetzungen konnten die Teilnehmer aus der ersten Gruppe (CTL) nur ~~ca. 40%~~ der Aufgaben richtig lösen, während bei der zweiten Gruppe die Erfolgsquote bei über 70% lag. Die präzisen Angaben für beiden Gruppen befinden sich in Tabelle 2 und im Diagramm in der Abbildung 2. Die Bewertung hat auch gezeigt, wie natürlichsprachlich formulierte Kriterien unterschiedlich interpretiert und verstanden werden können. Ein Problem wurde bei Kriterium 3 erkannt: Viele Testpersonen aus der ersten Gruppe haben die Formulierung "in der unmittelbare Folge" als die Formulierung "im gleichen Abschnitt" interpretiert.

Die Bewertung hat auch unsere Vermutung bestätigt, dass die Instanziierung von Variablen, d.h. Spezifikationsmuster-Parameter zusätzlich unterstützt werden muss. Diese Frage wurde schon teilweise im Rahmen des Verdikt-Projekts beantwortet [2].

31%

Anzahl der richtig spezifizierten Kriterien	Anzahl der Teilnehmer aus	
	Gruppe 1 (CTL)	Gruppe 2 (Muster)
5	0	9
4	3	6
3	5	4
2	8	4
1	10	0
0	0	2
Insgesamt	26	25

Tabelle 2: Ergebnisse

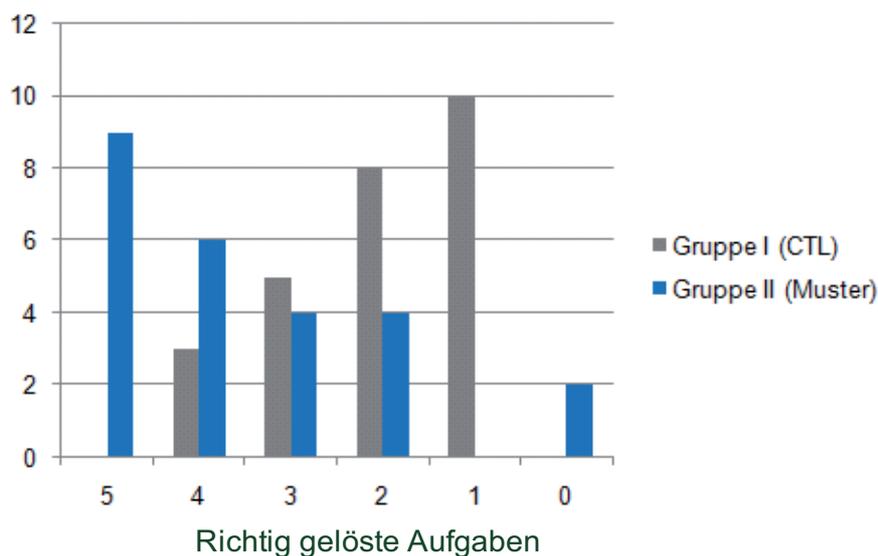


Abbildung 2: Die Anzahl der richtig spezifizierten Kriterien

Die Teilnehmer aus der ersten Gruppen benötigten die gesamte Bearbeitungszeit (60 Minuten), während die meisten Teilnehmer aus der zweiten Gruppe nur bis zu 30 Minuten benötigten.

5 Fazit und Ausblick

In diesem Bericht haben wir die Evaluationsergebnisse über die Methode der Musterbasierten Spezifikation von Konsistenzkriterien für Web-Dokumente im Vergleich zur Spezifikation mit der temporalen Logik CTL dargestellt. Es wurde gezeigt, dass trotz vergleichbarer Voraussetzungen die Teilnehmer aus der ersten Gruppe (CTL) nur ca. 40% der Aufgaben richtig lösen konnten, während bei der zweiten Grup-

pe (Spezifikationsmuster) die Erfolgsquote bei über 70% lag. Wir konnten auch wichtige Erkenntnisse über die Formulierung der natürlichsprachlich ausgedrückten Kriterien und über die Verständlichkeit der getesteten Muster gewinnen.

Der gleiche Test soll in der nahen Zukunft noch einmal mit wieder ca. 50 Testpersonen durchgeführt werden, so dass die schon erzielten Ergebnisse eine höhere statistische Bedeutung bekommen.

Auch im Verdikt-Projekt wird in diesem Bereich weiter geforscht. Eine Beispielsorientierte Spezifikationsmethodik wird entwickelt, die auf Spezifikationsmustern basiert. Diese Methode soll im Vergleich mit den Spezifikationsmustern evaluiert werden.

6 Danksagung

Frau Dr. P. Genkova, Universität Passau, hat mit wertvollen Anregungen zu dieser Evaluation beigetragen.

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft (DFG) unter Fördernummer FR 1021/7-1 finanziert.

Literatur

- [1] M. Jakšić and B. Freitag. Temporal Patterns for Document Verification. Technical Report MIP-0805, University of Passau, Germany, 2008.
- [2] B. Stadler. Dokumentmodellierung mit Ontologien. Bachelorarbeit, Lehrstuhl für Informationsmanagement, Universität Passau, 2008.
- [3] F. Weitzl. *Document Verification with Temporal Description Logics*. PhD thesis, University of Passau, 2008.
- [4] F. Weitzl, M. Jakšić, and B. Freitag. Towards the Automated Verification of Semi-structured Documents. *Data & Knowledge Engineering*, 68:292–317, 2009.

A Hilfestellung für die Spezifikation mit Spezifikationsmuster

No.	Beschreibung
1	Auf allen Pfaden gilt P in jeder Texteinheit im ganzen Dokument. (<i>universally p, -, globally, -</i>)
2	Auf allen Pfaden gilt nicht P in jeder Texteinheit im ganzen Dokument. (<i>universally p, absence, globally, -</i>)
3	Auf mindestens einem Pfad gilt P in jeder Texteinheit im ganzen Dokument. (<i>universally p, some_path, globally, -</i>)
4	Auf allen Pfaden gilt P in jeder Texteinheit, bevor S zum ersten Mal gilt. (<i>universally p, -, before s, -</i>)
5	Auf allen Pfaden gilt nicht P in jeder Texteinheit, bevor S zum ersten Mal gilt. (<i>universally p, absence, before s, -</i>)
6	Auf mindestens einem Pfad gilt P in jeder Texteinheit, bevor S zum ersten Mal gilt. (<i>universally p, some_path, before s, -</i>)
7	Auf allen Pfaden gilt P in jeder Texteinheit, bevor S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, -, before s, real_before</i>)
8	Auf allen Pfaden gilt nicht P in jeder Texteinheit, bevor S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, absence, before s, real_before</i>)
9	Auf mindestens einem Pfad gilt P in jeder Texteinheit, bevor S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, some_path, before s, real_before</i>)
10	Auf allen Pfaden gilt P in jeder Texteinheit, nachdem S zum ersten Mal gilt. (<i>universally p, -, after s, -</i>)
11	Auf allen Pfaden gilt nicht P in jeder Texteinheit, nachdem S zum ersten Mal gilt. (<i>universally p, absence, after s, -</i>)

Tabelle 3: Spezifikationsmuster und deren Bedeutung, Teil 1

No.	Beschreibung
12	Auf mindestens einem Pfad gilt P in jeder Texteinheit, nachdem S zum ersten Mal gilt. (<i>universally p, some_path, after s, -</i>)
13	Auf allen Pfaden gilt P in jeder Texteinheit, nachdem S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, -, after s, real_after</i>)
14	Auf allen Pfaden gilt nicht P in jeder Texteinheit, nachdem S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, absence, after s, real_after</i>)
15	Auf mindestens einem Pfad gilt P in jeder Texteinheit, nachdem S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>universally p, some_path, after s, real_after</i>)
16	Auf allen Pfaden gilt P in jeder Texteinheit zwischen S und R. (<i>universally p, -, between s and r, -</i>)
17	Auf allen Pfaden gilt nicht P in jeder Texteinheit zwischen S und R. (<i>universally p, absence, between s and r, -</i>)
18	Auf mindestens einem Pfad gilt P in jeder Texteinheit zwischen S und R. (<i>universally p, some_path, between s and r, -</i>)
19	Auf allen Pfaden gilt P in einer Texteinheit. (<i>exists p, -, globally, -</i>)
20	Auf mindestens einem Pfad gilt P in einer Texteinheit. (<i>exists p, some_path, globally, -</i>)
21	Auf allen Pfaden gilt P in einer Texteinheit, bevor S zum ersten Mal gilt. (<i>exists p, -, before s, -</i>)
22	Auf mindestens einem Pfad gilt P in einer Texteinheit, bevor S zum ersten Mal gilt. (<i>exists p, some_path, before s, -</i>)
23	Auf allen Pfaden gilt P in einer Texteinheit, bevor S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>exists p, -, before s, real_before</i>)
24	Auf mindestens einem Pfad gilt P in einer Texteinheit, bevor S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>exists p, some_path, before s, real_before</i>)

Tabelle 4: Spezifikationsmuster und deren Bedeutung, Teil 2

No.	Beschreibung
25	Auf allen Pfaden gilt P in einer Texteinheit, nachdem S zum ersten Mal gilt. (<i>exists p, -, after s, -</i>)
26	Auf mindestens einem Pfad gilt P in einer Texteinheit, nachdem S zum ersten Mal gilt. (<i>exists p, some_path, after s, -</i>)
27	Auf allen Pfaden gilt P in einer Texteinheit, nachdem S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>exists p, -, after s, real_after</i>)
28	Auf mindestens einem Pfad gilt P in einer Texteinheit, nachdem S zum ersten Mal gilt. P darf nicht in der gleichen Texteinheit wie S gelten. (<i>exists p, some_path, after s, real_after</i>)
29	Auf allen Pfaden gilt P in einer Texteinheit zwischen S und R. (<i>exists p, -, between s and r, -</i>)
30	Auf mindestens einem Pfad gilt P in einer Texteinheit zwischen S und R. (<i>exists p, some_path, between s and r, -</i>)
31	Auf allen Pfaden gilt: Vor jeder Texteinheit, in welcher Q gilt, gibt es eine Texteinheit, in welcher P gilt. (<i>p precedes q, -, globally, -</i>)
32	Auf allen Pfaden vor dem ersten Auftreten von S gilt: Vor jeder Texteinheit, in welcher Q gilt, gibt es eine Texteinheit, in welcher P gilt. (<i>p precedes q, -, before s, -</i>)
33	Auf allen Pfaden vor dem ersten Auftreten von S gilt: Vor jeder Texteinheit, in welcher Q gilt, gibt es eine Texteinheit, in welcher P gilt. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>p precedes q, -, before s, real_before</i>)
34	Auf allen Pfaden nach dem ersten Auftreten von S gilt: Vor jeder Texteinheit, in welcher Q gilt, gibt es eine Texteinheit, in welcher P gilt. (<i>p precedes q, -, after s, -</i>)
35	Auf allen Pfaden nach dem ersten Auftreten von S gilt: Vor jeder Texteinheit, in welcher Q gilt, gibt es eine Texteinheit, in welcher P gilt. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>p precedes q, -, after s, real_after</i>)
36	Auf allen Pfaden gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus auf allen Pfaden irgendwann auch Q. (<i>q follows p, -, globally, -</i>)

Tabelle 5: Spezifikationsmuster und deren Bedeutung, Teil 3

No.	Beschreibung
37	Auf allen Pfaden gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus in allen direkten Nachfolgern Q. (<i>q follows p, immediate, globally, -</i>)
38	Auf allen Pfaden vor dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus auf allen Pfaden irgendwann auch Q. (<i>q follows p, -, before s, -</i>)
39	Auf allen Pfaden vor dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus in allen direkten Nachfolgern Q. (<i>q follows p, immediate, before s, -</i>)
40	Auf allen Pfaden vor dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus auf allen Pfaden irgendwann auch Q. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>q follows p, -, before s, real_before</i>)
41	Auf allen Pfaden vor dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus in allen direkten Nachfolgern Q. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>q follows p, immediate, before s, real_before</i>)
42	Auf allen Pfaden nach dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus auf allen Pfaden irgendwann auch Q. (<i>q follows p, -, after s, -</i>)
43	Auf allen Pfaden nach dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus in allen direkten Nachfolgern Q. (<i>q follows p, immediate, after s, -</i>)
44	Auf allen Pfaden nach dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus auf allen Pfaden irgendwann auch Q. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>q follows p, -, after s, real_after</i>)
45	Auf allen Pfaden nach dem ersten Auftreten von S gilt: In allen Texteinheiten, in denen P gilt, gilt von dort aus in allen direkten Nachfolgern Q. Q darf nicht in der gleichen Texteinheit wie S gelten. (<i>q follows p, immediate, after s, real_after</i>)

Tabelle 6: Spezifikationsmuster und deren Bedeutung, Teil 4