

Inhaltsverzeichnis

1 Einleitung	1
1.1 Traditionelle Software-Entwicklung vs. Prototyping	1
1.2 Rapid Prototyping von Datenbankanwendungen	2
1.3 Übersicht über verwandte Ansätze	4
1.3.1 Testdatengenerierung auf der Basis von Schemainformationen	4
1.3.2 Testdatengenerierung für Benchmarktests	5
1.4 Aufbau der Arbeit	7
2 Integritätsbedingungen im Umfeld relationaler Datenbankanwendungen	9
2.1 Definitionen	9
2.2 Syntax und Semantik der betrachteten Integritätsbedingungen	12
2.3 Beispiel eines Relationenschemas	16
3 Testdatengenerierung auf der Basis von Integritätsbedingungen	21
3.1 Beweisbarkeit prädikatenlogischer Ausdrücke	21
3.2 Heuristische Datengenerierung vs. compilative Erzeugung von Programmcode	22
3.3 Vorgehensweise im Überblick	24
3.3.1 Auswirkungen der Integritätsbedingungen auf den Generierungsprozeß	24
3.3.2 Ablauf der Testdatengenerierung	26
3.4 Überlegungen zur Anwendungssituation in der Praxis	30
4 Deduktion in Aussagen- und Prädikatenlogik	37
4.1 Deduktion in der Aussagenlogik	37
4.1.1 Sprachdefinition der Aussagenlogik	38
4.1.2 Semantik der Aussagenlogik	41
4.2 Deduktion in der Prädikatenlogik	45
4.2.1 Sprachdefinition der Prädikatenlogik	45
4.2.2 Semantik der Prädikatenlogik	47
5 Ein Kalkül zur Erzeugung von Generierungsparametern aus Integritätsbedingungen	49
5.1 Berechnung der Betrachtungsreihenfolge für Relationen des Datenbankschemas	50
5.1.1 Ableitungsregeln für 1:n-Beziehungen	50
5.1.2 Ableitungsregeln für n:m-Beziehungen	55
5.1.3 Berechnung der Relationenfolge	58
5.1.4 Betrachtung von Zyklen in Relationenschemata	62
5.2 Aufstellen des Prädikatengraphen	67
5.3 Berechnung zulässiger Wertebereiche	67

5.4 Berechnung zulässiger Untermengenkardinalitäten	72
5.4.1 Betrachtung unabhängiger Wertmengenbedingungen	74
5.4.2 Betrachtung von Wertmengenbedingungen mit gemeinsamem Summenattribut	79
5.4.3 Betrachtung von Wertmengenbedingungen mit gemeinsamem Gruppenattribut	81
5.4.4 Betrachtung verketteter Wertmengenbedingungen	82
5.4.5 Betrachtung sich überkreuzender Wertmengenbedingungen	88
5.5 Berechnung der Gesamtkardinalitäten	94
5.6 Integration der Kardinalitätsverhältnisse über referentielle Integritätsbedingungen	97
6 Implementierung eines Testdatengenerators auf der Basis von Integritäts- bedingungen	103
6.1 Ein PROLOG-Interpreter zur Berechnung von Generierungsparametern	103
6.1.1 Transformation der Integritätsbedingungen auf PROLOG-Klauseln	104
6.1.2 Berechnung der Relationenfolge und Aufstellen des Prädikatengraphen	106
6.1.3 Berechnung zulässiger Generierungsparameter	110
6.1.4 Berechnung der Untermengenkardinalitäten aus Wertmengen- und Wertebereichsbedingungen	113
6.2 Ein parametergesteuertes Datenbank-Programm zur Wertegenerierung	116
6.2.1 Verwaltung der Schemainformationen	117
6.2.2 Wertegenerierung für Nichtschlüsselattribute	122
6.2.3 Wertegenerierung für Schlüsselattribute	125
6.2.3.1 Wertegenerierung für 1:n-Beziehungen	126
6.2.3.2 Wertegenerierung für 1:n-Beziehungen	127
7 Zusammenfassung	131
Literatur	133
Anhang	