

Inhaltsverzeichnis

1	Einführung und Motivation	1
1.1	Anforderungen an persistente Objektsysteme	1
1.2	Ansätze zur Verbesserung der Qualität persistenter Objektsysteme	2
2	Sprachen und Systeme zur Programmierung persistenter Objektsysteme	7
2.1	Datenbankprogrammierung: Sprachen und Konzepte	7
2.1.1	Typorientierte Datenbankprogrammiersprachen	8
2.1.2	Logikbasierte und funktionale Datenbanksprachen	10
2.1.3	Objekt-orientierte Sprachen und Systeme	12
2.2	Programmierumgebungen für persistente Objektsysteme	12
2.2.1	Lose Kopplung durch Wortspracheneinbettung	13
2.2.2	Systemintegration in Datenbankprogrammiersprachen . .	14
2.2.3	Systemtechnische Vereinfachungen in persistenten Systemen	15
3	Tycoon: Ein Modell- und Systemüberblick	19
3.1	Das Tycoon System und seine Komponenten	20
3.2	Zur sprachlichen Ausgestaltung der Sprache TL	23
3.3	Das Tycoon Programmierszenario	27
4	Ein Sprachkern zur generischen Benennung, Bindung und Typisierung	33
4.1	Lexikalische und syntaktische Regeln	33
4.2	Benennung und Typisierung vordefinierter Werte und Funktionen	34
4.3	Benennung, Bindung und Typisierung benutzerdefinierter Werte	37
4.3.1	Statische Bindungen	37
4.3.2	Dynamische Bindungen	41
4.3.3	Funktionen höherer Ordnung	42
4.4	Vordefinierte Wertkonstruktoren und Wertselektoren	43
4.4.1	Tupel und Feldselektion	44
4.4.2	Varianten und Variantenprojektion	46
4.4.3	Records, Recorderweiterung und Recordfeldselektion .	48
4.5	Typkompatibilität und Subtypbeziehungen	49
4.5.1	Subtypisierung zwischen Tupeltypen	51
4.5.2	Subtypisierung zwischen Funktionstypen	53
4.5.3	Subtypisierung zwischen Recordtypen	54

4.6 Benennung, Bindung und Metatypisierung benutzerdefinierter Typen	57
4.6.1 Statische Bindungen und Typabstraktion	57
4.6.2 Dynamische Bindungen	61
4.6.2.1 Polymorphe Funktionen und generische Module	62
4.6.2.2 Typoperatoren	63
4.6.2.3 Dynamische Typisierung	68
4.6.3 Rekursive Typbindungen: Korrektheit und Entscheidbarkeit	71
4.6.3.1 Beispiele rekursiver Typbindungen	71
4.6.3.2 Nicht-parametrisierte rekursive Typen	73
4.6.3.3 Rekursive Typoperatoren	77
5 Generische Programmierung persistenter Objektsysteme	81
5.1 Imperative Programmierung	81
5.1.1 Mutabilität und Zuweisungen	81
5.1.2 Kontrollstrukturen und Ausnahmebehandlung	85
5.1.3 Subtypisierungsregeln für veränderliche Bindungen	88
5.1.4 Felder und Feldindizierung	90
5.2 Objekt-orientierte Programmierung	91
5.2.1 Von modularer zu objekt-orientierter Programmierung	92
5.2.2 Inkrementelle Softwareentwicklung, Subtypisierung und Vererbung	96
5.2.3 Objektidentität	99
5.2.4 Dynamische Bindung von Selbstdnachrichten	100
5.3 Logikbasierte Programmierung	102
5.4 Programmierung im Großen	105
5.4.1 Module, Schnittstellen und Importbeziehungen	106
5.4.2 Typbindungen in Signaturen	109
5.4.3 Bindung an externe Funktionen	111
5.4.4 Ein Persistenzmodell für TL	113
6 Definition der statischen TL Sprachsemantik	117
6.1 Zur Formalisierung der TL Sprachdefinition	117
6.2 Konsistenzbedingungen auf TL Programmen	120
6.3 Normalisierung von TL Programmen	121
6.4 Die abstrakte Syntax für TL	123
6.5 Überblick über die verwendeten Notationen	127
6.5.1 Substitutionen	130
6.5.2 Qualifizierte Typvariablen	131
6.5.3 Kontraktive Typen	131
6.6 Ausgewählte Beispiele und Diskussion der TL Typregeln	132
6.7 Sichtbarkeitsregeln in de Bruijn Notation	138
6.8 Ein Algorithmus zur Typüberprüfung	142

7	Eine portable untypisierte Zwischensprache	147
7.1	Überblick über das TML Maschinenmodell	148
7.2	Syntaktische Objekte in TML	150
7.3	Semantische Objekte und Objektspeicherstrukturen	152
7.4	Strukturelle operationale Semantik von TML	154
7.5	Transformation von TL Termen in TML Terme	159
7.5.1	Variablenallokation	159
7.5.2	Übersetzung von Ausdrücken	163
7.6	Eine portable Abstrakte Maschine für TML	164
7.7	Übersetzung von TML Termen in Maschinenkode	167
8	Eine modellunabhängige Objektspeicherschnittstelle	171
8.1	Portable Datenrepräsentation	171
8.2	Portable Programmrepräsentation	174
8.3	Interaktion zwischen TML Evaluatoren und dem Objektspeicher	174
9	Innovative Tycoon Bibliotheksabstraktionen	177
9.1	Iterationsabstraktion über generalisierte Massendaten	177
9.2	Transaktionale Datenbankprogrammierung	180
9.3	Direkte graphische Manipulation komplexer Objekte	182
9.4	Eine strikt typisierte generische SQL Schnittstelle	184
9.5	Benutzerdefinierte Grammatiken	185
9.6	Implementierung der Tycoon Sprachprozessoren	187
A	Die Programmiersprache TL	191
A.1	Syntaktische Konventionen	191
A.2	Symbole	191
A.3	Reservierte Schlüsselworte	193
A.4	Produktionen	193
A.4.1	Kompilationseinheiten	193
A.4.2	Bindungen	193
A.4.3	Werte	194
A.4.4	Signaturen	195
A.4.5	Typen	195
A.4.6	Bezeichner	195
A.5	Die TL Typregeln	196
A.5.1	Wohlgeformte Signaturen	196
A.5.2	Wohlgeformte Typen	197
A.5.3	Wert- und Typsignaturen	197
A.5.4	Subsignaturen	198
A.5.5	Subtypen	198
A.5.6	Signaturen von Bindungen	200
A.5.7	Typen von Werten	201
A.5.8	Restriktionen	203
B	Ausgewählte Schnittstellen der Tycoon Standardbibliothek	205

C Die Tycoon Maschine	217
C.1 Datenrepräsentation (Modul “data”)	217
C.2 Programmrepräsentation (Modul “code”)	218
C.3 Objektspeicherschnittstelle (Modul “store”)	221
C.4 Evaluatoren und Evaluationszustände (Modul “thread”)	225
Literaturverzeichnis	227