

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Klassifizierung	3
1.2.1	Sprachen mit impliziter Parallelität	4
1.2.2	Sprachen mit expliziter Parallelität	5
1.3	Prinzipien der parallelen Programmierung	6
1.4	Die wesentlichen Entwurfsziele von ASL	7
1.5	Aufbau	9
2	Grundlagen der Programmierung	11
2.1	Kriterien für die Programmerstellung	11
2.1.1	Adaptivität und Skalierbarkeit	11
2.1.2	Portabilität	11
2.1.3	Robustheit	12
2.2	Eigenschaften von Programmiersprachen	12
2.2.1	Lesbarkeit	12
2.2.2	Implementierbarkeit und Effizienz	13
2.2.3	Abstraktionsniveau	13
2.2.4	Umfang	14
2.3	Eigenschaften von Parallelrechnersprachen	14
2.3.1	Sequentialität	14
2.3.2	Parallelität	14
2.3.3	Problem-immanente Parallelität	14
2.3.4	Skalierbarkeit	15
2.3.5	Transformierbarkeit	15
2.3.6	Kommunikation	15
2.3.7	Granularität	15
3	Entwicklung der Sprache ASL	17
3.1	RGL – der Vorgänger von ASL	17
3.1.1	Entwurfsprinzipien	17
3.1.2	Bewertung	18
3.1.2.1	Anwendbarkeit	18
3.1.2.2	Darstellbare Probleme	18

3.1.2.3	Analysierbarkeit	18
3.1.2.4	Benutzerfreundlichkeit	19
3.1.2.5	Weiterentwicklung	19
3.2	Forderungen an die Sprache ASL	19
3.2.1	Von RGL übernommene Anforderungen	19
3.2.1.1	Verwendung einer mathematischen Notation	20
3.2.1.2	Natur- und ingenieurwissenschaftliche Probleme	20
3.2.1.3	Hohes Abstraktionsniveau	20
3.2.1.4	Automatische Transformation	20
3.2.1.5	Anlehnung an andere Sprachen	21
3.2.2	An ASL neu gestellte Anforderungen	21
3.2.2.1	Einheitliche syntaktische Darstellung	21
3.2.2.2	Erweiterbarkeit bezüglich der Zielrechnerarchitekturen	21
3.2.2.3	Modularität bezüglich der anzuwendenden Verfahren	22
3.2.2.4	Entscheidbarkeit bezüglich Terminierung und Komplexität	22
3.3	Forderungen an ASL-Spezifikationen	23
3.3.1	Von RGL übernommene Anforderungen	23
3.3.2	Von ASL-Spezifikationen erfüllte Anforderungen	23
3.3.2.1	Architekturunabhängigkeit	23
3.3.2.2	Konfigurationsunabhängigkeit	24
3.3.2.3	Problemgrößenunabhängigkeit	24
3.4	Entwurfskriterien für ASL	24
3.4.1	Single-Assignment-Prinzip	24
3.4.2	Erstellung von größenunabhängigen Spezifikationen	25
3.4.2.1	Das INIT-Konzept	25
3.4.2.2	Problemgrößenbestimmende Parameter	25
3.4.2.3	Größenbestimmende Parameter und Eingabedaten	25
3.4.3	Spezifikationsaufrufe	26
3.4.3.1	Interpretation	26
3.4.3.2	Vererbung der größenbestimmenden Parameter	26
3.4.4	Klasse der beschreibbaren Probleme	26
3.4.5	Analysierbarkeit	27
3.4.6	Automatische Transformation von Spezifikationen	27
3.5	Beschränkung auf affine Indexausdrücke	27
3.5.1	Begriffe	28
3.5.1.1	Formale Definitionen	28
3.5.1.2	Indexausdruck und Indexfunktion	28
3.5.1.3	Beispiele	29
3.5.2	Vorarbeiten zu RGL und ASL	30
3.5.2.1	Entwicklung von RGL	30
3.5.2.2	Eindeutigkeit- und Vollständigkeit	31
3.5.2.3	Transformation von RGL-Spezifikationen	32
3.5.2.4	Entwicklung von ASL	32

3.5.3	Betrachtung weiterer Literatur	33
3.5.3.1	Ansätze für das Time-Space-Mapping	33
3.5.3.2	Ansätze für das Lösen von linearen Gleichungssystemen	35
3.5.3.3	Berechnung von Zeit- und Prozessorkomplexitäten	36
3.5.3.4	Durchführung von Abhängigkeitstests	36
3.5.3.5	Statistische Angaben	37
3.5.3.6	Behandlung von nicht linearen Systemen	37
3.5.3.7	Ergebnisse	38
3.5.4	Literaturunabhängige Überlegungen	38
3.5.4.1	RGL und ASL	38
3.5.4.2	Eigenschaften von affinen Ausdrücken	40
3.6	Bearbeitung einer ASL-Spezifikation	42
3.6.1	Anwendungsphase	43
3.6.1.1	Bestimmung der aktuellen Problemgröße	43
3.6.1.2	Syntaktische Korrektheit der Spezifikation	43
3.6.1.3	Behandlung der Konstantenbezeichner	43
3.6.1.4	Semantische Korrektheit der Spezifikation	43
3.6.2	Parallelisierungsphase	44
3.6.2.1	Erzeugung der monolithischen Form	44
3.6.2.2	Erzeugen der Normalform	44
3.6.2.3	Abhängigkeitsanalyse	44
3.6.3	Architekturphase	45
4	Elementare Konstrukte von ASL	47
4.1	Aufbau einer ASL-Spezifikation	47
4.1.1	Der Spezifikationskopf	48
4.1.2	Der Definitionsteil	49
4.1.3	Der Gleichungsteil	50
4.1.4	Der Eingabeteil	51
4.1.5	Ausgabeteil	51
4.2	Bezeichner	51
4.2.1	Syntax	51
4.2.2	Semantik	52
4.2.2.1	Einfache Bezeichner	53
4.2.2.2	Indizierte Bezeichner	53
4.2.2.3	Wertbezeichner	54
4.2.2.4	Gebundene Bezeichner	54
4.2.2.5	Konstantenbezeichner	54
4.2.2.6	Funktionsbezeichner	55
4.2.2.7	Bereichsbezeichner	55
4.2.2.8	Verwendung von Bezeichnern	55
4.3	Ausdrücke	57
4.3.1	Syntax	57

4.3.2	Semantik	57
4.3.2.1	Gleichung	57
4.3.2.2	Funktionsdefinition	59
4.3.2.3	Spezifikationsaufruf	60
4.3.2.4	Indexausdruck	60
4.4	Bereichsangaben	61
4.4.1	Syntax	61
4.4.2	Semantik	62
4.5	Gleichungen	63
4.5.1	Historische Entwicklung	63
4.5.2	Aufbau einer Gleichung	64
4.5.2.1	Linke Gleichungsseite	64
4.5.2.2	Rechte Gleichungsseite	64
4.5.2.3	Bereichsangabe einer Gleichung	64
4.5.3	Typen von Gleichungen	65
4.5.3.1	Einfache Gleichungen	66
4.5.3.2	Gleichungsschablonen	66
4.5.4	Instantiierung	66
5	Semantik von ASL	69
5.1	ALGORITHM	69
5.2	DEFINE	70
5.2.1	Typdefinitionen	70
5.2.1.1	Situation in RGL	71
5.2.1.2	Typen in ASL	71
5.2.1.3	Elementare Datentypen	73
5.2.1.4	Datenstrukturen	73
5.2.1.4.1	Einfache Datenstrukturen	73
5.2.1.4.2	Geschachtelte Datenstrukturen	74
5.2.1.4.3	Verwendung von Datenstrukturen	74
5.2.1.4.4	Beispiele für Datenstrukturen	75
5.2.1.4.5	Beschreibung von Netzwerken	75
5.2.1.5	Verträglichkeit von Datentypen	75
5.2.1.5.1	Verträglichkeit der elementaren Datentypen	76
5.2.1.5.2	Verträglichkeit von Datenstrukturen	76
5.2.1.5.3	Datentypen und Operatoren	77
5.2.1.6	Schlüsselwort DEFAULT	77
5.2.1.7	Zusammenfassung und Bewertung	77
5.2.2	DECLARE	79
5.2.2.1	Situation	79
5.2.2.2	Benutzerdefinierte Konstanten	80
5.2.2.3	Größenbestimmende Parameter	80
5.2.2.4	Aufbau einer Konstantendefinition	82
5.2.2.4.1	Operatoren	82

5.2.2.4.2	Bezeichner	83
5.2.2.5	Zusammenfassung und Bewertung	85
5.2.3	RANGES	86
5.2.3.1	Situation	86
5.2.3.2	Benutzerdefinierte Bereichsangaben	86
5.2.3.3	Aufbau einer Bereichsdefinition	88
5.2.3.3.1	Affine Ausdrücke für Unter- und Obergrenzen	90
5.2.3.3.2	Indizierte Bezeichner in Unter- und Ober- grenze	90
5.2.3.3.3	Nichtleere Bereichsangaben	90
5.2.3.3.4	Überlappende Teilbereiche	90
5.2.3.4	Zusammenfassung und Bewertung	90
5.3	EQUATIONS	91
5.3.1	Übersicht	92
5.3.2	„Normale“ Gleichungsschablonen	93
5.4	FUNCTIONS	94
5.4.1	Übersicht	94
5.4.2	Aufbau einer Funktionsdefinition	94
5.4.2.1	Gestaltungsregeln	95
5.4.2.2	Parameter einer Funktion	96
5.4.2.3	Aufruf anderer Funktionen	97
5.4.2.4	Funktionsbezeichner als INPUT	98
5.4.3	Verwendung von Funktionsbezeichnern	99
5.4.4	Zusammenfassung und Bewertung	100
5.5	OBTAIN	101
5.5.1	Bedeutung eines OBTAIN-Konstrukts	101
5.5.1.1	OBTAIN-Konstrukte in RGL	101
5.5.1.2	OBTAIN-Konstrukte in ASL	102
5.5.2	Verwendung eines OBTAIN-Konstrukts	102
5.5.3	Aufbau eines OBTAIN-Konstrukts	102
5.5.3.1	OBTAIN-Teil	103
5.5.3.2	FROM-Teil	104
5.5.3.3	WITH-Teil	106
5.5.4	Syntaktische Nachvollziehbarkeit	107
5.5.4.1	Definition	108
5.5.4.2	Prüfkriterien	108
5.5.5	Übergabeparameter	110
5.5.6	Zusammenfassung und Bewertung	112
5.6	Mathematische Anweisungen	113
5.6.1	Numerische Operatoren	114
5.6.2	Vergleichende Operatoren	115
5.6.3	Logische Operatoren	116
5.6.4	Transzendente Operatoren	117
5.7	IF THEN ELSE FI	118

5.7.1	Bedingungsteil	118
5.7.1.1	Operatoren	118
5.7.1.2	Bezeichner	120
5.7.1.3	Funktionsbezeichner	120
5.7.1.4	ASL-spezifische Konstrukte	120
5.7.2	Zusammenfassung und Bewertung	121
5.8	FIRST	122
5.8.1	Funktionalität eines FIRST-Konstrukts	122
5.8.2	Aufbau eines FIRST-Konstrukts	124
5.8.2.1	Prädikat eines FIRST-Konstrukts	124
5.8.2.1.1	Operatoren	124
5.8.2.1.2	Bezeichner	125
5.8.2.2	Kriterien für die Gestaltung von FIRST-Konstrukten	125
5.8.3	Zusammenfassung und Bewertung	126
5.9	ASSOC	126
5.9.1	Aufbau eines ASSOC-Konstrukts	127
5.9.2	Zusammenfassung und Bewertung	128
5.10	Hardwarespezifische Konstrukte	129
5.10.1	PSI	129
5.10.2	CON	129
5.10.3	BUS	130
5.10.4	ADR	130
5.10.5	Zusammenfassung und Bewertung	131
5.11	INPUT	132
5.12	COMPUTE	133
5.13	Kommentare	133
6	Transformation einer ASL-Spezifikation	135
6.1	Erzeugung der monolithischen Form	135
6.1.1	Benutzerdefinierte Funktionen	136
6.1.1.1	Situation	136
6.1.1.2	Auflösungsverfahren	136
6.1.2	Datenstrukturen	138
6.1.2.1	Situation	138
6.1.2.2	Auflösungsverfahren	139
6.1.2.3	Bedeutung der Selektoren	140
6.1.3	FIRST-Konstrukte	140
6.1.3.1	Situation	140
6.1.3.2	Auflösungsverfahren	140
6.1.4	OBTAIN-Konstrukte	143
6.1.4.1	Situation	143
6.1.4.2	Aufbau der Inputschnittstelle	144
6.1.4.3	Übernahme des Gleichungsteils	145
6.1.4.4	Aufbau der Computeschnittstelle	145

6.2	Erzeugung der Normalform	147
6.2.1	Behandlung von Datenumordnungen	147
6.2.1.1	Datenvervielfachungen	148
6.2.1.1.1	Umwandlungsverfahren	148
6.2.1.1.2	Optimierungsansätze	149
6.2.1.2	Datenreduzierungen	150
6.2.2	Elementare Transformationen	151
6.2.2.1	Einführung neuer Bezeichner	151
6.2.2.2	Ersetzung von linken Seiten durch rechte Seiten . . .	152
6.2.2.3	Ersetzung von rechten Seiten durch linke Seiten . . .	153
6.2.3	Indexausdrücke in Gleichungen	154
6.2.4	Indexausdrücke in mehreren Gleichungen	155
7	Abstrakt spezifizierte Algorithmen	157
7.1	Matrizenmultiplikation	159
7.1.1	Datenvervielfachung (Broadcasting)	159
7.1.2	Staggering (Algorithmus von Cannon)	162
7.2	Matrixtransposition	165
7.3	Matrixinversion	166
7.4	Gleichungssysteme: Direkte Verfahren	170
7.5	Behandlung von Tridiagonalmatrizen	173
7.6	Gleichungssysteme: Iterative Verfahren	176
7.6.1	Jacobi-Relaxation (Gesamtschrittverfahren)	176
7.6.2	Gauß-Seidel-Relaxation (Einzelschrittverfahren)	179
7.6.3	CG-Verfahren (Conjugate Gradient)	181
7.7	Mehrgitterverfahren	185
7.7.1	Restriktion	185
7.7.2	Glättung	189
7.7.3	Prolongation	193
7.8	Sortiervverfahren	198
7.8.1	Odd-Even Transposition Sort	198
7.8.2	Batcher's Bitonisches Sortieren	201
7.9	Fast Fourier Transformation (FFT)	204
7.10	Chi-Quadrat-Methode	212
8	Schlußbemerkungen	215
8.1	Transformierte Algorithmen	215
8.1.1	Algorithmenspezifische Anforderungen	215
8.1.2	Bemerkungen zur Partitionierung	216
8.1.3	Beispiel Matrizenmultiplikation	217
8.1.3.1	Datenfeld gleich Prozessorfeld	218
8.1.3.2	Datenfeld größer als Prozessorfeld	219
8.1.3.3	Datenfeld kleiner als Prozessorfeld	221
8.1.4	Batcher's Bitonisches Sortieren	221

8.2 Zusammenfassung	224
8.3 Derzeitiger Stand und Ausblick	225
Spezifikationsverzeichnis	227
Produktionsverzeichnis	229
Tabellenverzeichnis	230
Literaturverzeichnis	232
Indexverzeichnis	239