

Questo è l'indice del libro, in cui sono evidenziati i paragrafi corrispondenti agli argomenti trattati nel corso e che costituiranno il programma d'esame.
Si noti che la presentazione di alcuni argomenti e gli esempi forniti a lezione possono differire da quanto presentato nel libro.
Per ragioni didattiche, anche l'ordine con cui sono presentati gli argomenti a lezione non corrisponde esattamente a quello del testo.

Sommario

Ultimo aggiornamento: 5 giugno 2013

Prefazione all'edizione italiana	xv
Prefazione	xvii
1 Automi: metodo e follia	1
1.1 Perché studiare la teoria degli automi	2
1.1.1 Introduzione agli automi a stati finiti	2
1.1.2 Rappresentazioni strutturali	4
1.1.3 Automi e complessità	5
1.2 Introduzione alle dimostrazioni formali	5
1.2.1 Dimostrazioni deduttive	6
1.2.2 Riduzione a definizioni	8
1.2.3 Altre forme di teorema	9
1.2.4 Teoremi che non assomigliano a enunciati se-allora	13
1.3 Altre forme di dimostrazione	13
1.3.1 Dimostrazioni di uguaglianza tra insiemi	13
1.3.2 Il contronominale	15
1.3.3 Dimostrazioni per assurdo	16
1.3.4 Controesempi	17
1.4 Dimostrazioni induttive	18
1.4.1 Induzione sugli interi	18
1.4.2 Forme più generali di induzione sugli interi	21
1.4.3 Induzione strutturale	22
1.4.4 Induzione mutua	25
1.5 I concetti centrali della teoria degli automi	27
1.5.1 Alfabeto	27
1.5.2 Stringa	28
1.5.3 Linguaggio	29
1.5.4 Problema	30

1.6	Riepilogo	33
1.7	Bibliografia	34
2	Automi a stati finiti	35
2.1	Una descrizione informale degli automi a stati finiti	36
2.1.1	Le regole fondamentali	36
2.1.2	Il protocollo	37
2.1.3	Automi che possono ignorare azioni	39
2.1.4	L'intero sistema come automa	40
2.1.5	Validazione del protocollo mediante l'automato prodotto	42
2.2	Automi a stati finiti deterministici	43
2.2.1	Definizione di automa a stati finiti deterministico	43
2.2.2	Elaborazione di stringhe in un DFA	44
2.2.3	Notazioni più semplici per i DFA	45
2.2.4	Estensione della funzione di transizione alle stringhe	46
2.2.5	Il linguaggio di un DFA	50
2.2.6	Esercizi	50
2.3	Automi a stati finiti nondeterministici	53
2.3.1	Descrizione informale degli automi a stati finiti nondeterministici	53
2.3.2	Definizione di automa a stati finiti nondeterministico	54
2.3.3	La funzione di transizione estesa	55
2.3.4	Il linguaggio di un NFA	56
2.3.5	Equivalenza di automi a stati finiti deterministici e nondeterministici	58
2.3.6	Un caso sfavorevole di costruzione per sottoinsiemi	62
2.3.7	Esercizi	63
2.4	Un'applicazione: ricerche testuali	65
2.4.1	Ricerca di stringhe in un testo	66
2.4.2	Automi a stati finiti nondeterministici per ricerche testuali	66
2.4.3	Un DFA per riconoscere un insieme di parole chiave	67
2.4.4	Esercizi	69
2.5	Automi a stati finiti con epsilon-transizioni	69
2.5.1	Uso delle ϵ -transizioni	70
2.5.2	La notazione formale per gli ϵ -NFA	71
2.5.3	Epsilon-chiusure	71
2.5.4	Transizioni estese e linguaggi per gli ϵ -NFA	73
2.5.5	Eliminazione delle ϵ -transizioni	74
2.5.6	Esercizi	77
2.6	Riepilogo	77
2.7	Bibliografia	78

3	Espressioni e linguaggi regolari	79
3.1	Espressioni regolari	79
3.1.1	Gli operatori delle espressioni regolari	80
3.1.2	Costruzione di espressioni regolari	82
3.1.3	Precedenza degli operatori delle espressioni regolari	84
3.1.4	Esercizi	85
3.2	Automi a stati finiti ed espressioni regolari	86
3.2.1	Dai DFA alle espressioni regolari	86
3.2.2	Conversione di DFA in espressioni regolari per eliminazione di stati	91
3.2.3	Conversione di espressioni regolari in automi	97
3.2.4	Esercizi	101
3.3	Applicazioni delle espressioni regolari	103
3.3.1	Le espressioni regolari in UNIX	103
3.3.2	Analisi lessicale	104
3.3.3	Ricerca di pattern in un testo	106
3.3.4	Esercizi	108
3.4	Proprietà algebriche per le espressioni regolari	109
3.4.1	Associatività e commutatività	109
3.4.2	Identità e annichilatori	110
3.4.3	Distributività	110
3.4.4	Idempotenza	111
3.4.5	Proprietà relative alla chiusura	112
3.4.6	Alla ricerca di proprietà per le espressioni regolari	112
3.4.7	Verifica di proprietà algebriche sulle espressioni regolari	114
3.4.8	Esercizi	116
3.5	Riepilogo	117
3.6	Bibliografia	117
4	Proprietà dei linguaggi regolari	119
4.1	Dimostrare che un linguaggio non è regolare	119
4.1.1	Il pumping lemma per i linguaggi regolari	120
4.1.2	Applicazioni del pumping lemma	121
4.1.3	Esercizi	123
4.2	Proprietà di chiusura dei linguaggi regolari	124
4.2.1	Chiusura dei linguaggi regolari rispetto a operazioni booleane	125
4.2.2	Inversione	130
4.2.3	Omomorfismi	132
4.2.4	Omomorfismi inversi	133
4.2.5	Esercizi	138
4.3	Problemi di decisione per i linguaggi regolari	141
4.3.1	Conversioni	142
4.3.2	Verificare se un linguaggio regolare è vuoto	144
4.3.3	Appartenenza a un linguaggio regolare	145

Verificare se un linguaggio è vuoto, finito o infinito. A lezione è stato presentato in maniera differente rispetto al libro, utilizzando il pumping lemma.



4.3.4	Esercizi	146
4.4	Equivalenza e minimizzazione di automi	146
4.4.1	Verifica dell'equivalenza di stati	146
4.4.2	Equivalenza di linguaggi regolari	149
4.4.3	Minimizzazione di DFA	151
4.4.4	Perché il DFA minimo non può essere migliorato	154
4.4.5	Esercizi	156
4.5	Riepilogo	156
4.6	Bibliografia	157
5	Grammatiche e linguaggi liberi dal contesto	159
5.1	Grammatiche libere dal contesto	159
5.1.1	Un esempio informale	160
5.1.2	Definizione delle grammatiche libere dal contesto	161
5.1.3	Derivazioni per mezzo di una grammatica	163
5.1.4	Derivazioni a sinistra e a destra	165
5.1.5	Il linguaggio di una grammatica	167
5.1.6	Forme sentenziali	168
5.1.7	Esercizi	169
5.2	Alberi sintattici	171
5.2.1	Costruzione di alberi sintattici	171
5.2.2	Il prodotto di un albero sintattico	173
5.2.3	Inferenza, derivazioni e alberi sintattici	174
5.2.4	Dalle inferenze agli alberi	175
5.2.5	Dagli alberi alle derivazioni	176
5.2.6	Dalle derivazioni alle inferenze ricorsive	180
5.2.7	Esercizi	181
5.3	Applicazioni delle grammatiche libere dal contesto	182
5.3.1	Parser	182
5.3.2	YACC: un generatore di parser	184
5.3.3	Linguaggi di markup	185
5.3.4	XML e DTD	188
5.3.5	Esercizi	194
5.4	Ambiguità nelle grammatiche e nei linguaggi	195
5.4.1	Grammatiche ambigue	195
5.4.2	Eliminare le ambiguità da una grammatica	197
5.4.3	Derivazioni a sinistra come modo per esprimere l'ambiguità	200
5.4.4	Ambiguità inerente	201
5.4.5	Esercizi	203
5.5	Riepilogo	204
5.6	Bibliografia	205

Come decidere l'equivalenza di due linguaggi regolari. A lezione è stato presentato sfruttando le proprietà di chiusura rispetto alle operazioni booleane e la possibilità di decidere se un linguaggio regolare è vuoto.

6	Automi a pila	207
6.1	Definizione di automa a pila	207
6.1.1	Introduzione informale	207
6.1.2	Definizione formale di automa a pila	209
6.1.3	Una notazione grafica per i PDA	211
6.1.4	Descrizioni istantanee di un PDA	212
6.1.5	Esercizi	215
6.2	I linguaggi di un PDA	216
6.2.1	Accettazione per stato finale	217
6.2.2	Accettazione per stack vuoto	218
6.2.3	Da stack vuoto a stato finale	218
6.2.4	Da stato finale a stack vuoto	221
6.2.5	Esercizi	223
6.3	Equivalenza di PDA e CFG	224
6.3.1	Dalle grammatiche agli automi a pila	225
6.3.2	Dai PDA alle grammatiche	228
6.3.3	Esercizi	232
6.4	Automi a pila deterministici	233
6.4.1	Definizione di PDA deterministico	234
6.4.2	Linguaggi regolari e PDA deterministici	234
6.4.3	DPDA e linguaggi liberi dal contesto	236
6.4.4	DPDA e grammatiche ambigue	236
6.4.5	Esercizi	237
6.5	Riepilogo	238
6.6	Bibliografia	239
7	Proprietà dei linguaggi liberi dal contesto	241
7.1	Forme normali per grammatiche libere dal contesto	241
7.1.1	Eliminazione di simboli inutili	242
7.1.2	Calcolo dei simboli generatori e raggiungibili	243
7.1.3	Eliminazione di ϵ -produzioni	244
7.1.4	Eliminazione delle produzioni unitarie	248
7.1.5	Forma normale di Chomsky	252
7.1.6	Esercizi	255
7.2	Il pumping lemma per i linguaggi liberi dal contesto	259
7.2.1	Dimensione degli alberi sintattici	259
7.2.2	Enunciato del pumping lemma	260
7.2.3	Applicazioni del pumping lemma per i CFL	262
7.2.4	Esercizi	264
7.3	Proprietà di chiusura dei linguaggi liberi dal contesto	266
7.3.1	Sostituzione	266
7.3.2	Applicazioni del teorema di sostituzione	268
7.3.3	Inversione	269

7.3.4	Intersezione con un linguaggio regolare	269
7.3.5	Omomorfismo inverso	273
7.3.6	Esercizi	275
7.4	Proprietà di decisione dei CFL	277
7.4.1	Complessità delle conversioni fra CFG e PDA	277
7.4.2	Tempo di esecuzione della conversione in forma normale di Chomsky	279
7.4.3	Verificare se un CFL è vuoto	280
7.4.4	Appartenenza a un CFL	281
7.4.5	Anteprima di problemi indecidibili per i CFL	285
7.4.6	Esercizi	285
7.5	Riepilogo	286
7.6	Bibliografia	287

Verificare se un linguaggio è vuoto, finito o infinito. A lezione è stato presentato in maniera differente rispetto al libro, utilizzando il pumping lemma.



8	Macchine di Turing: introduzione	289
8.1	Problemi che i calcolatori non possono risolvere	289
8.1.1	Programmi che stampano “Ciao, mondo”	290
8.1.2	Un ipotetico verificatore di ciao-mondo	292
8.1.3	Ridurre un problema a un altro	295
8.1.4	Esercizi	298
8.2	La macchina di Turing	298
8.2.1	La ricerca della soluzione a tutte le domande matematiche	299
8.2.2	Notazione per la macchina di Turing	300
8.2.3	Descrizioni istantanee delle macchine di Turing	301
8.2.4	Diagrammi di transizione per le macchine di Turing	304
8.2.5	Il linguaggio di una macchina di Turing	308
8.2.6	Le macchine di Turing e l’arresto	308
8.2.7	Esercizi	309
8.3	Tecniche di programmazione per le macchine di Turing	311
8.3.1	Memoria nello stato	311
8.3.2	Tracce multiple	312
8.3.3	Subroutine	314
8.3.4	Esercizi	315
8.4	Estensioni alla macchina di Turing semplice	317
8.4.1	Macchine di Turing multinastro	317
8.4.2	Equivalenza di macchine di Turing mononastro e multinastro	318
8.4.3	Tempo di esecuzione e costruzione da n a un nastro	319
8.4.4	Macchine di Turing nondeterministiche	321
8.4.5	Esercizi	322
8.5	Macchine di Turing ristrette	325
8.5.1	Macchine di Turing con nastri semi-infiniti	325
8.5.2	Macchine multistack	328
8.5.3	Macchine a contatori	330
8.5.4	La potenza delle macchine a contatori	331

8.5.5	Esercizi	333
8.6	Le macchine di Turing e i computer	334
8.6.1	Simulazione di una macchina di Turing da parte di un computer	334
8.6.2	Simulazione di un computer da parte di una macchina di Turing	336
8.6.3	Confronto dei tempi di esecuzione dei computer e delle macchine di Turing	340
8.7	Riepilogo	342
8.8	Bibliografia	344
9	Indecidibilità	347
9.1	Un linguaggio non ricorsivamente enumerabile	348
9.1.1	Enumerazione delle stringhe binarie	349
9.1.2	Codici per le macchine di Turing	349
9.1.3	Il linguaggio di diagonalizzazione	350
9.1.4	Dimostrazione che L_d non è ricorsivamente enumerabile	351
9.1.5	Esercizi	352
9.2	Un problema indecidibile ma ricorsivamente enumerabile	353
9.2.1	Linguaggi ricorsivi	353
9.2.2	Complementi di linguaggi ricorsivi e RE	354
9.2.3	Il linguaggio universale	357
9.2.4	Indecidibilità del linguaggio universale	359
9.2.5	Esercizi	360
9.3	Problemi indecidibili relativi alle macchine di Turing	362
9.3.1	Riduzioni	362
9.3.2	Macchine di Turing che accettano il linguaggio vuoto	364
9.3.3	Il teorema di Rice e le proprietà dei linguaggi RE	366
9.3.4	Problemi sulle specifiche di macchine di Turing	369
9.3.5	Esercizi	369
9.4	Il problema di corrispondenza di Post	371
9.4.1	Definizione del problema di corrispondenza di Post	371
9.4.2	PCP modificato	373
9.4.3	Dimostrazione di indecidibilità di PCP: conclusione	376
9.4.4	Esercizi	381
9.5	Altri problemi indecidibili	382
9.5.1	Problemi relativi a programmi	382
9.5.2	Indecidibilità dell'ambiguità delle CFG	382
9.5.3	Il complemento di un linguaggio associato a una lista	384
9.5.4	Esercizi	387
9.6	Riepilogo	388
9.7	Bibliografia	389

10 Problemi intrattabili	391
10.1 Le classi \mathcal{P} e \mathcal{NP}	392
10.1.1 Problemi risolvibili in tempo polinomiale	393
10.1.2 Un esempio: l'algoritmo di Kruskal	393
10.1.3 Tempo polinomiale nondeterministico	396
10.1.4 Un esempio in \mathcal{NP} : il problema del commesso viaggiatore	397
10.1.5 Riduzioni polinomiali	398
10.1.6 Problemi NP-completi	399
10.1.7 Esercizi	401
10.2 Un problema NP-completo	403
10.2.1 Il problema della soddisfacibilità	403
10.2.2 Rappresentazione di istanze di SAT	405
10.2.3 NP-completezza del problema SAT	406
10.2.4 Esercizi	411
10.3 Una versione ristretta di soddisfacibilità	412
10.3.1 Forme normali di espressioni booleane	413
10.3.2 Conversione in CNF di espressioni booleane	414
10.3.3 NP-completezza di CSAT	416
10.3.4 NP-completezza di 3SAT	421
10.3.5 Esercizi	422
10.4 Altri problemi NP-completi	423
10.4.1 Descrivere problemi NP-completi	423
10.4.2 Il problema dell'insieme indipendente	424
10.4.3 Il problema della copertura per nodi	427
10.4.4 Il problema del circuito hamiltoniano orientato	429
10.4.5 Circuiti hamiltoniani non orientati e TSP	435
10.4.6 Riepilogo dei problemi NP-completi	436
10.4.7 Esercizi	437
10.5 Riepilogo	441
10.6 Bibliografia	441
11 Altre classi di problemi	443
11.1 Complementi dei linguaggi in \mathcal{NP}	444
11.1.1 La classe di linguaggi co- \mathcal{NP}	444
11.1.2 Problemi NP-completi e co- \mathcal{NP}	445
11.1.3 Esercizi	446
11.2 Problemi risolvibili in spazio polinomiale	447
11.2.1 Macchine di Turing in spazio polinomiale	447
11.2.2 Le relazioni di \mathcal{PS} e \mathcal{NPS} con altre classi	448
11.2.3 Spazio polinomiale deterministico e nondeterministico	449
11.3 Un problema completo per \mathcal{PS}	451
11.3.1 PS-completezza	451
11.3.2 Formule booleane con quantificatori	453

11.3.3	Valutazione di formule booleane con quantificatori	454
11.3.4	PS-completezza del problema QBF	456
11.3.5	Esercizi	460
11.4	Classi di linguaggi basate sulla randomizzazione	461
11.4.1	Quicksort: un esempio di algoritmo randomizzato	461
11.4.2	Un modello di macchina di Turing con randomizzazione	462
11.4.3	Il linguaggio di una macchina di Turing randomizzata	463
11.4.4	La classe \mathcal{RP}	465
11.4.5	Riconoscimento di linguaggi in \mathcal{RP}	467
11.4.6	La classe \mathcal{ZPP}	468
11.4.7	Relazioni tra \mathcal{RP} e \mathcal{ZPP}	469
11.4.8	Relazioni con le classi \mathcal{P} e \mathcal{NP}	470
11.5	Complessità e numeri primi	470
11.5.1	L'importanza della verifica di primalità	471
11.5.2	Introduzione all'aritmetica modulare	473
11.5.3	La complessità del calcolo in aritmetica modulare	475
11.5.4	Verifica di primalità in tempo polinomiale randomizzato	476
11.5.5	Verifiche di primalità nondeterministiche	477
11.5.6	Esercizi	480
11.6	Riepilogo	480
11.7	Bibliografia	482
A	La gerarchia di Chomsky	485
A.1	Grammatiche	485
A.1.1	Definizione di grammatica	486
A.1.2	Derivazioni e linguaggi	487
A.2	Tipi di grammatiche e gerarchia di Chomsky	488
A.2.1	Linguaggi di tipo 0	490
A.2.2	Linguaggi di tipo 1 o dipendenti dal contesto	490
A.2.3	Linguaggi di tipo 2 e di tipo 3	491
A.3	Bibliografia	492
	Indice analitico	493