

Masters Programme in Computer and Information Engineering

Anton Augustsson

June 2020



# Contents

<b>1 Basic Course in Mathematics</b>	<b>11</b>
1.1 Basic . . . . .	12
1.1.1 Mängder . . . . .	12
1.1.2 Potenslagar . . . . .	12
1.1.3 Logaritmer . . . . .	12
1.1.4 Tillvägagångs sätt . . . . .	12
1.1.5 Intervall . . . . .	13
1.1.6 Tillvägagångs sätt . . . . .	13
1.2 Komplexa tal . . . . .	14
1.2.1 Polärform . . . . .	14
1.3 Absolut Belopp . . . . .	16
1.3.1 Tillvägagångs sätt . . . . .	16
1.4 Summor . . . . .	17
1.4.1 Aritmetiska summor . . . . .	17
1.4.2 Geometriska summor . . . . .	17
1.4.3 Tillvägagångs sätt . . . . .	17
1.5 Kombinatorik . . . . .	18
1.5.1 Multiplikations principen . . . . .	18
1.5.2 Kombinationer . . . . .	18
1.5.3 Binomial satsen . . . . .	18
1.6 Funktioner och kordinatsystem . . . . .	19
1.6.1 Avståndsformeln . . . . .	19
1.6.2 Elipser . . . . .	19
1.7 Polynom division . . . . .	20
1.7.1 Tillvägagångs sätt . . . . .	20
1.8 Trigonometri . . . . .	21
1.8.1 Tillvägagångs sätt . . . . .	22
<b>2 Program Design and Data Structures</b>	<b>23</b>
2.1 Coding convention . . . . .	24
2.2 Design approach . . . . .	24
2.2.1 Process . . . . .	24
2.3 Recursion . . . . .	25
2.3.1 Recursion types . . . . .	25
2.4 Complexity . . . . .	25
2.4.1 Growth . . . . .	25
2.5 Recurrences . . . . .	25
2.5.1 Closed Form . . . . .	26
2.6 Higher-Order Function . . . . .	28
2.6.1 Higher-Order Functions on Lists . . . . .	28
2.7 Data types . . . . .	29
2.7.1 Basic . . . . .	29

2.7.2	Maybe Type . . . . .	29
2.7.3	New types and enumeration types . . . . .	29
2.7.4	Inductive Data Types . . . . .	32
2.7.5	Trees . . . . .	33
2.7.6	Other data types . . . . .	35
2.7.7	Graphs . . . . .	37
2.8	Important syntax . . . . .	40
2.8.1	Let . . . . .	40
2.8.2	IO . . . . .	40
2.9	Sorting Algorithms . . . . .	40
<b>3</b>	<b>Algebra 1</b>	<b>41</b>
3.1	logik . . . . .	42
3.1.1	värde tabeller . . . . .	42
3.1.2	Implikationer . . . . .	42
3.2	Mängder . . . . .	44
3.3	Bevis . . . . .	45
3.3.1	Induktions bevis . . . . .	45
3.3.2	Motsägelse bevis . . . . .	45
3.4	Delbarhet . . . . .	46
3.4.1	Största Gemensama Delaren (SGD) . . . . .	46
3.4.2	Primtal . . . . .	48
3.5	Diofantiska ekvationer . . . . .	49
3.6	Talbaser . . . . .	50
3.6.1	konvertera från decimal bass till annan bas . . . . .	50
3.6.2	konvertera från annan bas till decimal bas . . . . .	50
3.6.3	Andra exempel . . . . .	50
3.7	Functioner . . . . .	51
3.7.1	Inversen . . . . .	51
3.7.2	Relatioiner . . . . .	51
3.8	Summor . . . . .	52
3.8.1	Aritmetiska summor . . . . .	52
3.8.2	Geometriska summor . . . . .	52
3.9	Kongruensräkning . . . . .	53
3.10	Kardinalitet . . . . .	54
3.10.1	Uppräkneligamängder . . . . .	54
3.11	Polynom . . . . .	55
3.11.1	Polynom division . . . . .	55
3.11.2	Faktorsatsen . . . . .	56
<b>4</b>	<b>Single Variable Calculus</b>	<b>59</b>
4.1	Basic . . . . .	60
4.1.1	Mängder . . . . .	60
4.1.2	Intervall . . . . .	61
4.1.3	Funktion . . . . .	62
4.1.4	Trigonometri . . . . .	63
4.1.5	Exempel: Trigonometri . . . . .	64
4.2	Gränsvärden . . . . .	65
4.2.1	Kontinuitet . . . . .	66
4.3	Derivator . . . . .	67
4.3.1	Kjedje regeln . . . . .	67
4.3.2	L'Hôpital's rule . . . . .	68
4.3.3	Medelvärdessatsen . . . . .	68
4.3.4	Rolle . . . . .	68

4.3.5	växande funktioner . . . . .	69
4.3.6	Högreordnings deviator . . . . .	69
4.3.7	Impericit derivering . . . . .	69
4.3.8	invers funktioner . . . . .	69
4.3.9	exponetial och logaritm . . . . .	70
4.3.10	odefinerad form . . . . .	70
4.3.11	inversa trigometriska funktioner . . . . .	71
4.4	Grafritning . . . . .	72
4.5	Optemering . . . . .	74
4.6	Talfölder och serier . . . . .	75
4.6.1	Serier med varierande tecken . . . . .	77
4.6.2	Potensserier . . . . .	77
4.6.3	Taylor serier . . . . .	79
4.7	Integraler . . . . .	81
4.7.1	variabelsubstition . . . . .	83
4.7.2	Integration av rationella funktioner . . . . .	84
4.7.3	Partiell integration . . . . .	86
4.7.4	Generaliseringande integraler . . . . .	87
4.7.5	Volymberäkningar . . . . .	89
4.8	Differential ekvationer . . . . .	91
4.8.1	superabla ekvationer . . . . .	92
4.8.2	Linjära differentialekvationer av ordning 1 . . . . .	93
4.8.3	Linjära differentialekvationer av ordning 2 . . . . .	94
<b>5</b>	<b>Computer Architecture</b> . . . . .	<b>99</b>
5.1	ISA1 . . . . .	100
5.1.1	MIPS instructions . . . . .	100
5.1.2	Sequencing . . . . .	102
5.1.3	Converge to Assambly code . . . . .	103
5.2	ISA2 . . . . .	103
5.2.1	MIPS type format . . . . .	103
5.2.2	Procedure . . . . .	104
5.2.3	Other ISA . . . . .	106
5.3	Arithmetic . . . . .	108
5.3.1	Binary numbers . . . . .	108
5.3.2	Negative integers . . . . .	108
5.3.3	Oporations . . . . .	108
5.3.4	Non integer numbers (Floating and fixed point) . . . . .	108
5.3.5	Overflow . . . . .	109
5.4	Logic . . . . .	110
5.5	processor control and datapath . . . . .	112
5.5.1	Clock . . . . .	114
5.5.2	Critical path . . . . .	114
5.6	pipline . . . . .	115
5.7	Pipline hazards . . . . .	117
5.7.1	Data hazards . . . . .	117
5.7.2	Control hazards . . . . .	117
5.7.3	Structural hazards . . . . .	118
5.8	Predicting Branches and Exceptions . . . . .	119
5.8.1	Static pridictors . . . . .	119
5.8.2	Dynamic pridictors . . . . .	120
5.8.3	Exceptions . . . . .	122
5.9	Input/Output . . . . .	123
5.10	Cache . . . . .	125

5.10.1 Memory hierarchy . . . . .	125
5.10.2 Cache misses 3C's . . . . .	126
5.11 Virtual Memory . . . . .	127
5.12 Parallelism . . . . .	130
5.12.1 Multicore . . . . .	130
5.12.2 Parallel programming . . . . .	130
5.12.3 Synchronization . . . . .	130
5.12.4 Cache coherency . . . . .	131
5.12.5 ILP . . . . .	132
<b>6 Probability and Statistics DV</b>	<b>133</b>
6.1 Introduction . . . . .	134
6.1.1 Test . . . . .	134
6.2 Imperative programming i C . . . . .	134
6.2.1 Data structures . . . . .	134
6.3 Object-Oriented programming i Java . . . . .	135
<b>7 Linear Algebra and Geometry I</b>	<b>137</b>
7.1 Linjära ekvationssystem . . . . .	138
7.1.1 Total Matris . . . . .	139
7.2 Vektorer/koordinater i planet och rummet . . . . .	141
7.2.1 Bas . . . . .	141
7.3 Skalärprodukt och vektorprodukt . . . . .	143
7.3.1 Skalärprodukt . . . . .	143
7.3.2 Orthogonal projection . . . . .	145
7.3.3 Enhetsvektorer och ON-baser . . . . .	149
7.3.4 Vektorprodukten . . . . .	149
7.3.5 Area och Volym . . . . .	152
7.4 Linjer och plan . . . . .	153
7.4.1 orthogonal projection på plan . . . . .	155
7.5 Matrisräkning . . . . .	157
7.5.1 Transponat . . . . .	158
7.5.2 Matrisinvers . . . . .	158
7.6 Determinanter . . . . .	160
7.6.1 Ko-faktornar . . . . .	161
7.6.2 Geometri: parallellepiped . . . . .	161
7.7 Vektorer i $\mathbb{R}^n$ . . . . .	162
7.8 Linjära avbildningar $\mathbb{R}^n \rightarrow \mathbb{R}^m$ . . . . .	163
7.8.1 Matristransformationer och linjära funktioner . . . . .	163
7.8.2 Injektiv/Surjektiv/Bijektiv . . . . .	165
<b>8 Linear Algebra II</b>	<b>167</b>
8.1 Grundläggande teori . . . . .	168
8.1.1 Ekvationssystem och matris räkning . . . . .	168
8.1.2 determinanter . . . . .	169
8.1.3 flerdimensionell dvs $R^n$ . . . . .	169
8.1.4 Funktioner . . . . .	169
8.1.5 Linjer . . . . .	169
8.2 Vektorrum . . . . .	170
8.3 Underrum och linjära höljet . . . . .	171
8.4 Linjärt oberoende . . . . .	173
8.5 Bas . . . . .	174
8.6 Basomvandling . . . . .	175
8.7 Linjär avbildning . . . . .	177

8.8	Matrisen av en linjär avbildning . . . . .	178
8.9	Basbyte av linjära avbildningar . . . . .	179
8.10	Kärna och bild av en linjär avbildning . . . . .	180
8.11	Egenvärden och egenvektorer . . . . .	181
8.12	Diagonalisering . . . . .	183
8.13	Inre produkrum . . . . .	184
8.14	Symetriska och positiva definita matriser . . . . .	185
8.15	On-baser och Gram-Schmidt-ortonormalisering . . . . .	186
8.16	Isometriska avbildningar och spektralsatsen . . . . .	188
8.17	Andragradskurvor och andragradsytor . . . . .	189
8.18	System av linjära differentialekvationer . . . . .	190
<b>9</b>	<b>Computer System with Project Work</b>	<b>193</b>
9.1	M1 . . . . .	194
9.1.1	CPU . . . . .	194
9.1.2	Register . . . . .	194
9.1.3	Memory . . . . .	194
9.1.4	CPU context . . . . .	194
9.1.5	Memory allocation . . . . .	194
9.1.6	Kernal . . . . .	195
9.1.7	Multiprogramming . . . . .	195
9.2	M2 . . . . .	195
9.2.1	Network communication . . . . .	195
9.2.2	Processes . . . . .	196
9.3	M3 . . . . .	196
9.3.1	Dispatcher . . . . .	196
9.3.2	PSB . . . . .	197
9.3.3	IO bound/CPU bound . . . . .	197
9.3.4	Schedular algorithems . . . . .	198
9.4	M4 . . . . .	198
9.4.1	Concurrency . . . . .	198
9.4.2	Parallelism . . . . .	198
9.4.3	Client-Server modele . . . . .	198
9.4.4	Threads . . . . .	198
9.4.5	Locks . . . . .	199
9.4.6	Spinlock . . . . .	199
9.4.7	TestAndSet . . . . .	199
9.4.8	Swap . . . . .	199
9.4.9	Bounded waiting . . . . .	199
9.4.10	Semaphores . . . . .	199
9.4.11	Mutex lock . . . . .	199
9.4.12	Deadlock . . . . .	199
9.4.13	Mutual exclusion . . . . .	199
9.4.14	Deadlock prevention . . . . .	199
9.4.15	Deadlock avoidance . . . . .	200
9.4.16	Clock Synchronization . . . . .	200
9.4.17	Mutual exclusion . . . . .	200
9.4.18	Desired properties of Transactions . . . . .	200
9.4.19	WiFi . . . . .	200
9.4.20	Bluetooth . . . . .	200
9.4.21	ISM-band . . . . .	201
9.4.22	Properties of a medium . . . . .	201
9.4.23	Sampling . . . . .	201
9.4.24	Multithreading models . . . . .	201

9.4.25	Bounded buffer . . . . .	201
9.4.26	Priority inversion . . . . .	202
9.5	M5 . . . . .	202
9.5.1	Single contiguous . . . . .	202
9.5.2	Swapping . . . . .	202
9.5.3	Memory resident . . . . .	202
9.5.4	Partitioned allocation . . . . .	203
9.5.5	Logical address space . . . . .	203
9.5.6	Address binding . . . . .	203
9.5.7	Memory management unit (MMU) . . . . .	203
9.5.8	Fragmentation . . . . .	203
9.5.9	Compaction . . . . .	203
9.5.10	Memory protection . . . . .	203
9.5.11	Shared pages . . . . .	203
9.5.12	Translation lookaside buffer . . . . .	203
9.5.13	Operating System . . . . .	203
9.5.14	Access methods . . . . .	203
9.5.15	RAM . . . . .	203
9.5.16	Volatile memory . . . . .	204
9.5.17	Persistent data storage . . . . .	204
9.5.18	The file . . . . .	204
9.5.19	Block data storage . . . . .	204
9.5.20	File control block . . . . .	204
9.5.21	Directory . . . . .	204
9.5.22	Contiguous allocation . . . . .	204
9.5.23	Linked allocation . . . . .	204
9.5.24	Indexed allocation . . . . .	204
9.5.25	FAT . . . . .	204
9.5.26	iNode . . . . .	204
9.5.27	Directory . . . . .	204
9.5.28	M6 . . . . .	205
9.5.29	Classic historical ciphers . . . . .	205
9.5.30	Modern cryptography . . . . .	205
9.5.31	Public key exchange . . . . .	205
9.5.32	Certification Authority (CA) . . . . .	205
9.5.33	Chain of trust . . . . .	205
9.5.34	Self-issued certificates . . . . .	205
9.5.35	Usage of modern cryptography . . . . .	205
9.5.36	Man-in-middle attack . . . . .	206
9.5.37	Chosen-plaintext attack . . . . .	206
9.5.38	Side-channel attack . . . . .	206
9.5.39	Replay attack . . . . .	206
9.5.40	Security in the Internet stack . . . . .	207
9.5.41	Firewall . . . . .	207
<b>10</b>	<b>Basic Course in Mathematics</b>	<b>209</b>
10.1	Documentation . . . . .	210
10.1.1	Implicit documentation . . . . .	210
10.1.2	Implicit documentation . . . . .	210
10.2	Human factors . . . . .	210
10.2.1	Color combinations . . . . .	210
10.2.2	Personas . . . . .	210
10.2.3	Memory . . . . .	210
10.2.4	What is needed to design a system? . . . . .	210

10.3 Model-View-Control . . . . .	211
<b>11 Probability and Statistics DV</b>	<b>213</b>
11.1 Statistisk mått och begreppet sannolikhet . . . . .	214
11.1.1 Begrepp . . . . .	214
11.2 Sannolikheter och slumpvariabler . . . . .	216
11.2.1 Betingade sannolikheter . . . . .	216
11.2.2 Kedjer av händelse . . . . .	216
11.2.3 Oberonnde händelser . . . . .	216
11.3 Fördelningar . . . . .	217
11.3.1 Binomial-fördelningar . . . . .	217
11.3.2 Possion-fördelningar . . . . .	218
11.3.3 Likformig/rektangulär-fördelningar . . . . .	218
11.3.4 Exponential-fördelningar . . . . .	219
11.3.5 Normalfördelning-fördelningar . . . . .	219
11.3.6 Läges och spridningsmått . . . . .	219
11.4 Olikheter . . . . .	220
11.4.1 Markovs olikhet . . . . .	220
11.4.2 Thebysjövs olikhet (chebyshev) . . . . .	220
11.4.3 Fördelnings funktioner . . . . .	220
11.4.4 Oberonede slupvariabel . . . . .	221
11.4.5 Fördelning av summor . . . . .	222
11.4.6 Central gränsvärdesatsen (CGS) . . . . .	222
11.5 Simulering av slumptal . . . . .	222
11.5.1 äkta slummpmässiga tal . . . . .	222
11.5.2 Pseudoslumppmässiga tal . . . . .	223
11.6 Statistikens grunder . . . . .	224
11.6.1 Allmänt . . . . .	224
11.6.2 Medelfel . . . . .	225
11.6.3 Skattning av varians . . . . .	225
11.6.4 Väntevärdsriktig . . . . .	226
11.6.5 Konfidensintervall för $\mu$ från $N(\mu, \sigma^2)$ med känt $\sigma$ . . . . .	226
11.6.6 Example . . . . .	227
11.6.7 Konfidensintervall för $p$ från binomialfördelad . . . . .	227
11.6.8 Konfidensintervall för skillnad i väntevärde . . . . .	227
11.6.9 Ensidiga intervall . . . . .	228
11.6.10 Stickprov i par . . . . .	228
11.7 Regressions . . . . .	228
11.7.1 Modell . . . . .	228
11.7.2 Modellens giltighet . . . . .	229
11.7.3 Användning ag modellen . . . . .	229
11.8 Stokastiska processer . . . . .	230
11.8.1 Bornulli processer . . . . .	230
11.8.2 Poisson processer . . . . .	230
11.9 Markovkedjor . . . . .	231
11.9.1 Ehrenfestmodellen . . . . .	232
11.9.2 Google-kedjan . . . . .	233
11.9.3 Hashfunktioner . . . . .	233
11.9.4 Kollisionmodoll . . . . .	233
11.9.5 Markov Buffer/Markovkö . . . . .	234



# Chapter 1

## Basic Course in Mathematics

## 1.1 Basic

### 1.1.1 Mängder

Naturliga tal:  $\mathbb{N} = \{1, 2, 3, \dots\}$

Heltal:  $\mathbb{Z} = \{\dots - 2, 1, 0, 1, 2, \dots\}$

Rationella tal:  $\mathbb{Q} = \left\{ \frac{a}{b} \mid a, b \in \mathbb{Z}, b \neq 0 \right\}$

Irrationella tal:  $\mathbb{P} = \frac{\mathbb{R}}{\mathbb{Q}}$  eller  $\{x \mid x \in \mathbb{R}, x \notin \mathbb{Q}\}$

Reella tal:  $\mathbb{R} = \mathbb{P} \cup \mathbb{Q}$

### 1.1.2 Potenslagar

$$\begin{aligned} \left(\frac{a}{b}\right)^{-3} &= \left(\frac{b}{a}\right)^3 \\ \sqrt{a} &= a^{\frac{1}{2}} \end{aligned}$$

### 1.1.3 Logaritmer

#### Logaritmlagar:

Väksam över när  $a < 0$  då är logarytmen odefinerad.

$$(1): b = a^x \Leftrightarrow \log_a(b) = x \text{ för: } a > 0, b > 0, a \neq 1$$

$$(2): \log_a\left(\frac{b}{c}\right) = \log_a(b) - \log_a(c)$$

$$(3): \log_a(b * c) = \log_a(b) + \log_a(c)$$

$$(4): \log_a(b^d) = d \log_a(b)$$

$$(5): \log_a(b) = \frac{\log_f(b)}{\log_f(a)}$$

$$(6): \log_a(a) = 1$$

$$(7): \log_a(1) = 0$$

$$(8): a^{\log_a(x)} = x$$

$$(9): \log_{a^c}(b) = \frac{1}{c} \log_a(b)$$

### 1.1.4 Tillvägagångs sätt

$$\log_3(x) + \log_x\left(\frac{1}{27}\right) = 2 \tag{1.1}$$

**Lösning**

$$\begin{aligned}
 \log_3(x) - 3\log_x(3) &= 2 \\
 \log_3(x) - 3\frac{\log_3(3)}{\log_3(x)} &= 2 \\
 \log_3(x) - 3\frac{1}{\log_3(x)} &= 2 \\
 \log_3(x)^2 - 3 &= 2\log_3(x) \\
 y = \log_3(x)^2 & \\
 y^2 - 2y - 3 &= 0 \\
 \text{pq-formeln: } y &= 1 \pm \sqrt{4} = 1 \pm 2 \\
 \log_3(x) = 3 &\Leftrightarrow x = 3^3 = 27 \\
 \log_3(x) = -1 &\Leftrightarrow x = 3^{-1} = \frac{1}{3}
 \end{aligned}$$

**1.1.5 Intervall**

$$[a, b] = \{x | a \leq x \leq b\} \quad (1.2)$$

$$[a, \infty[ \quad (1.3)$$

$$]-\infty, \infty[ \quad (1.4)$$

**1.1.6 Tillvägagångs sätt**

$$\frac{2}{x-3} < \frac{5}{x} \quad (1.5)$$

**Lösning**

$$\begin{aligned}
 \frac{2}{x-3} - \frac{5}{x} &< 0 \\
 \frac{(x)2}{x(x-3)} - \frac{5(x-3)}{x(x-3)} &< 0 \\
 \frac{2x - 5x + 15}{x(x-3)} &< 0 \\
 \frac{-3x - 15}{x(x-3)} &< 0 \\
 \frac{-3(x+5)}{x(x-3)} &< 0 \\
 x \neq 0, x \neq 3
 \end{aligned}$$

Värde tabell:

	$x < 0$	$0 < x < 3$	$3 < x < 5$	$5 < x$
$x-5$	-	-	-	+
-3	-	-	-	-
x	-	+	+	+
$x-3$	-	-	+	+
hela	+	-	+	-

## 1.2 Komplexa tal

$$z = a + bi$$

$$\operatorname{Re}(z) = a$$

$$\operatorname{Im}(z) = b$$

$$\bar{z} = a - bi$$

$$|z| = \sqrt{a^2 + b^2}$$

Där  $b(\operatorname{Im}(z))$  är för y-axeln och  $a(\operatorname{Re}(z))$  är för x-axeln

Samma räkneregler för reella tal gäller för komplexa tal

### 1.2.1 Polärform

$$\arg(z) = \alpha + 2\pi * n$$

$\arg(z)$  är vinkeln mellan a (x-axeln) linjen  $|z|$ .  $n$  är heltal

$$\operatorname{Arg}(z) \in ]-\pi, \pi[$$

$$\operatorname{Arg}(z) \in \arg(z)$$

$$a = |z| \cos \alpha$$

$$b = |z| \sin \alpha$$

$$z = |z| \cos \alpha + i * |z| \sin \alpha$$

**Polärform:**

$$z = |z|(\cos \alpha + i * \sin \alpha) \quad (1.6)$$

**Eulers formel:**

$$z = |z|e^{i\alpha} \quad (1.7)$$

**Sats:**

$$|z * w| = |z| * |w|$$

$$\arg(z * w) = \arg(z) + \arg(w)$$

$$\arg\left(\frac{z}{w}\right) = \arg(z) - \arg(w)$$

**De Movre's formel:**

$$(\cos(\theta) + i * \sin(\theta))^n = \cos(n * \theta) + i * \sin(n * \theta)$$

### Binomisk ekvation

$$(\cos(\theta) + i * \sin(\theta))^n = \cos(n * \theta) + i * \sin(n * \theta)$$

### Tillvägagångs sätt

$$\text{Visa lösningarna i det komplexa tal planet } z^5 = \sqrt{3} + i \quad (1.8)$$

$$\begin{aligned} |\sqrt{3} + i| &= \sqrt{\sqrt{3}^2 + 1^2} = \sqrt{4} = 2 \\ \begin{cases} \sqrt{3} = 2 \cos(\alpha) \\ 1 = 2 \sin(\alpha) \end{cases} \\ \alpha &= \frac{\pi}{6} \\ |z|^5 (\cos(5 * \theta) + i * \sin(5 * \theta)) &= 2(\cos(\frac{\pi}{6}) + i * \sin(\frac{\pi}{6})) \end{aligned}$$

Vilket ger:

$$\begin{aligned} |z|^5 &= 2 \Leftrightarrow |z| = 2^{\frac{1}{5}} \\ 5 * \theta &= \frac{\pi}{6} + 2\pi n \Leftrightarrow \theta = \frac{\pi}{30} + \frac{2\pi n}{5} \quad n \in \mathbb{Z} \end{aligned}$$

I polärform blir det då:

$$z = 2^{\frac{1}{5}} \left( \cos \frac{\pi}{30} + \frac{2\pi n}{5} + i * \sin \frac{\pi}{30} + \frac{2\pi n}{5} \right)$$

Eulers formel:  $z = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30} + \frac{2\pi n}{5})}$

$$n = 0, 1, 2, 3, 4$$

$$z_1 = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30})}$$

$$z_2 = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30} + \frac{2\pi}{5})}$$

$$z_3 = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30} + \frac{4\pi}{5})}$$

$$z_4 = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30} + \frac{6\pi}{5})}$$

$$z_5 = 2^{\frac{1}{5}} e^{i(\frac{\pi}{30} + \frac{8\pi}{5})}$$

### 1.3 Absolut Belopp

#### 1.3.1 Tillvägagångs sätt

$$|2x - 8| + |1 - x| - 2|x - 3| = 8 + 3x \quad (1.9)$$

lösning

$$\left( \begin{array}{l} 2x - 8 = 0 \\ x = 4 \end{array} \right) \left( \begin{array}{l} 1 - x = 0 \\ x = 1 \end{array} \right) \left( \begin{array}{l} x - 3 = 0 \\ x = 3 \end{array} \right)$$

$$\text{I } \begin{cases} x \leq 1 \\ -(2x - 8) + (1 - x) + 2(x - 3) = 8 + 3x \end{cases}$$

$$\text{I } \begin{cases} x \leq 1 \\ -4x = 5 \end{cases} \quad \begin{cases} x \leq 1 \\ x = -\frac{5}{4} \end{cases} \text{ lösning}$$

$$\text{II } \begin{cases} 1 < x \leq 3 \\ -(2x - 8) - (1 - x) + 2(x - 3) = 8 + 3x \end{cases}$$

$$\text{II } \begin{cases} 1 < x \leq 3 \\ -2x = 7 \end{cases} \quad \begin{cases} 1 < x < 3 \\ x = -\frac{7}{2} \end{cases} \text{ Ej lösning}$$

$$\text{III } \begin{cases} 3 \leq x < 4 \\ -(2x - 8) - (1 - x) - 2(x - 3) = 8 + 3x \end{cases}$$

$$\text{III } \begin{cases} 3 \leq x \leq 4 \\ -(2x - 8) - (4x - 3) = 8 + 3x \end{cases} \quad \begin{cases} 3x < 4 \\ x = \frac{5}{6} \end{cases} \text{ Ej lösning}$$

$$\text{IV } \begin{cases} x \geq 4 \\ (2x - 8) - (1 - x) - 2(x - 3) = 8 + 3x \end{cases}$$

$$\text{IV } \begin{cases} x \geq 4 \\ x = -\frac{11}{2} \end{cases} \text{ Ej lösning}$$

## 1.4 Summor

### 1.4.1 Aritmetiska summor

$$s_n = a_1 + a_2 + a_3 + \dots + a_n = \frac{n(a_1 + a_n)}{2} \quad (1.10)$$

### 1.4.2 Geometriska summor

Börjar alltid med exponenten 0 och gör om summan så att den passar i följande talföljd:

$$s_n = a + ak + ak^2 + \dots + ak^{n-1} = \frac{a(k^n - 1)}{k - 1} \quad (1.11)$$

### 1.4.3 Tillvägagångs sätt

$$\sum_{k=n}^{2n} (2^k - k) \quad (1.12)$$

#### Lösning

sätter f = 0 = k - n

$$\begin{aligned} \sum_{f=0}^n (2^{f+n} - (f+n)) &= 2^n * \sum_{f=0}^n (2^f) - \sum_{f=0}^n (f+n) \\ \frac{2^n(2^{n+1} - 1)}{2 - 1} - \frac{3n(n+1)}{2} &= 2^{2n+1} - 2^n - \frac{3n(n+1)}{2} \end{aligned}$$

## 1.5 Kombinatorik

### 1.5.1 Multiplikations principen

permetationer  $p(a,b)$  då ordningen spelar roll.

Antal sätt: (exemplet: antal sätt av måltider 7 företer 5 varmrätter 4 efterätter ( $7 \cdot 5 \cdot 4$ )

$$n_1 \cdot n_2 \cdot \dots \cdot n_m \quad (1.13)$$

### 1.5.2 Kombinationer

Kombination  $c(a,b)$  då ordningen inte spelar roll.

Antal sätt: (exemplet: antal del-mängder två element ( $n =$ element  $k =$ antal element som kan väljas))

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (1.14)$$

$$\binom{n}{k} = \frac{n \cdot (n-1)(n-2) \cdot \dots \cdot (n-(k-1))}{k!} \quad (1.15)$$

### Pascal triangel

$n$							
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1
	0	1	2	3	4	5	6
					$k$		

### 1.5.3 Binomial satsen

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k} \quad (1.16)$$

## 1.6 Funktioner och kordinatsystem

Kom ihåg att när det stor  $(x-a)$  förflytas den i x-axeln a steg till höger  $\rightarrow$  medans  $(x+a)$  förflytas den a steg till vänster  $<-$

### 1.6.1 Avståndsformeln

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.17)$$

### 1.6.2 Elipser

Förenkla till denna formeln:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (1.18)$$

Där a är avståndet på x-axeln och b är avståndet på y-axeln

### Hyperbol

Ser väldigt anerlunda ut från elipser

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (1.19)$$

### Circkel

$$\frac{x^2}{a^2} + \frac{y^2}{a^2} = 1 \quad (1.20)$$

Där a = r

## 1.7 Polynom division

### 1.7.1 Tillvägagångs sätt

Man vet att ekvationen  $z^4 - 2z^3 - 7z^2 + 26z - 20 = 0$  har roten  $z = 2+i$ . Lös ekvationen fullständigt. (1.21)

$$\begin{aligned} z &= 2+i \text{ Är en läsning är också konjugatet en lösning enligt faktorsatsen } \bar{z} = a - bi \\ z &= 2 \pm i \end{aligned}$$

Vilket betyder att följande går att factorisera ut polynomet

$$(z - (2+i))(z - (2-i)) = z^2 - 4z + 5$$

**Långdivision (liggande stolen):**

$$\begin{array}{r} x^2 + 2x - 4 \\ \hline x^4 - 2x^3 - 7x^2 + 26x - 20 \\ - x^4 + 4x^3 - 5x^2 \\ \hline 2x^3 - 12x^2 + 26x \\ - 2x^3 + 8x^2 - 10x \\ \hline - 4x^2 + 16x - 20 \\ 4x^2 - 16x + 20 \\ \hline 0 \end{array}$$

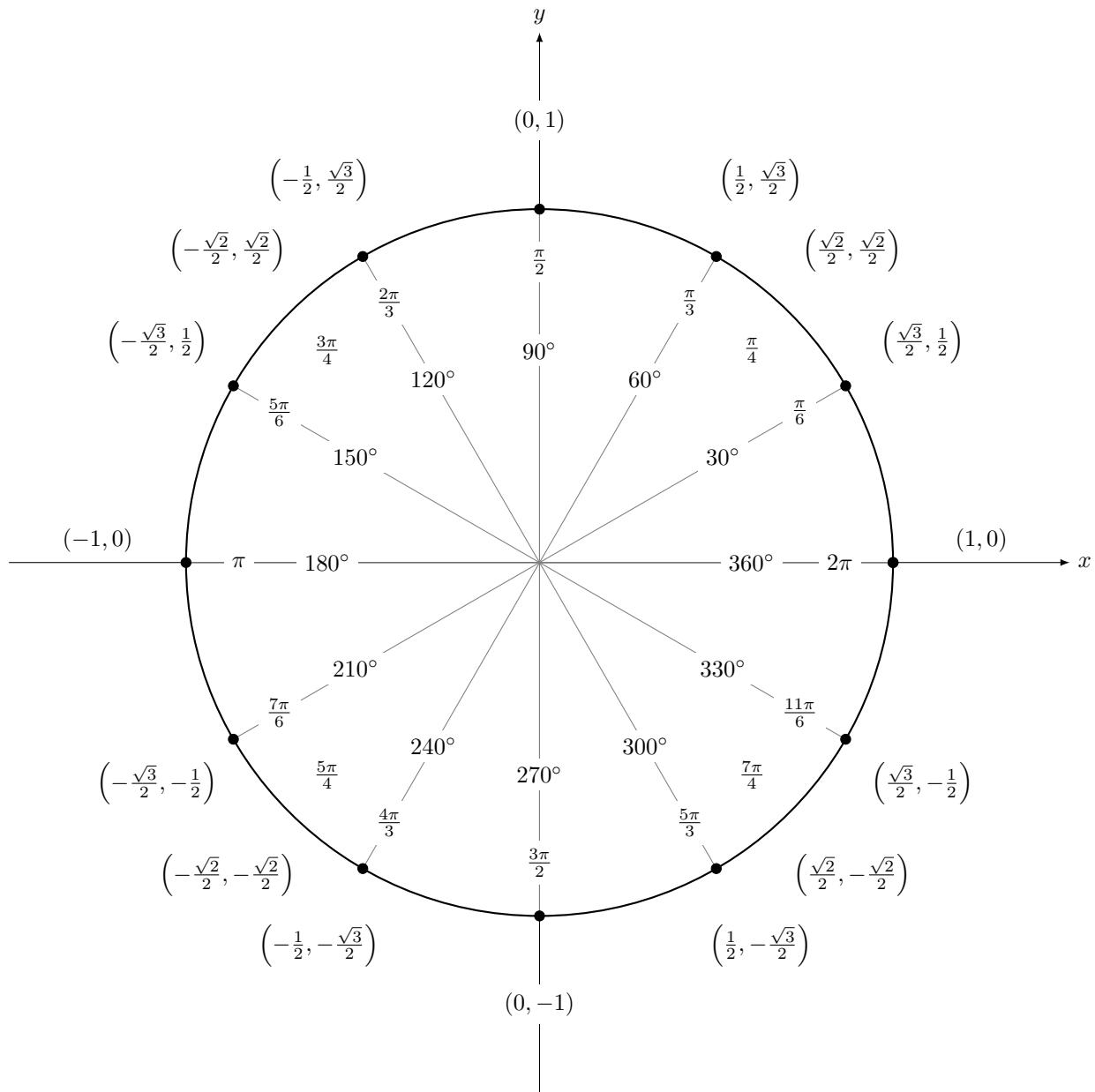
$z^2 + 2z - 4 = 0$  Är också en läsning som till slut ger följande

$$z = -1 \pm \sqrt{5}$$

$$z = 2 \pm i$$

Varge n grads polynom har alltid n stycken komplexa lösningar

## 1.8 Trigonometri



**Sats:**

$$360^\circ = 2\pi \text{rad}$$

$$v_g = v_r * \frac{180^\circ}{\pi}$$

$$v_r = v_g * \frac{\pi}{180^\circ}$$

**Sats:**

$$-1 \leq \sin t \leq 1$$

$$-1 \leq \cos t \leq 1$$

**Sats:**

$$\cos(-t) = \cos(t)$$

$$\sin(-t) = -\sin(t)$$

$$\tan(-t) = \frac{\sin(-t)}{\cos(-t)} = \frac{-\sin(t)}{\cos(t)}$$

**Additionsformlerna:**

$$\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \sin(\beta)\cos(\alpha)$$

$$\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \sin(\beta)\cos(\alpha)$$

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\beta)\sin(\alpha)$$

$$\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\beta)\sin(\alpha)$$

**Trigonometriska ettan:**

$$(\sin t)^2 + (\cos t)^2 = 1$$

$$\sin^2 t + \cos^2 t = 1$$

### 1.8.1 Tillvägagångs sätt

$$\begin{aligned} \cos \frac{\pi}{12} &= \cos \left( \frac{\pi}{3} - \frac{\pi}{4} \right) = \cos \frac{\pi}{3} \cos \frac{\pi}{4} + \sin \frac{\pi}{3} \sin \frac{\pi}{4} \\ &= \left( \frac{1}{2} \right) \left( \frac{1}{\sqrt{2}} \right) + \left( \frac{\sqrt{3}}{2} \right) \left( \frac{1}{\sqrt{2}} \right) = \frac{1 + \sqrt{3}}{2\sqrt{2}} \end{aligned}$$

## Chapter 2

# Program Design and Data Structures

## 2.1 Coding convention

### VARIANT

A (recursive) function terminates if it has a variant. The variant need to follow all of the flowing rules

- Needs to decrease every recursive call
- All ways positive

### Side effects

All IO functions have side effects in order to separate pure Haskell function with impure functions that changes the state with is commonly the case with imperative and object oriented programming. Every IO function has a side effects.

### INVARIANT

A data types invariant is what value are allowed for the data type to work. Similar to preconditions for a function. An example is integer data type that can only use positive integers therefor the invariant is positive integers.

## 2.2 Design approach

- top-down design (Cheating): Is to break down a complex system in to subsystems to solve the problem. Most often is to write everything by scratch.
- bottom-up design (Stacking): Is to piece existing system together to create a more complex system. little is programmed, most is copied.
- dodging: Get some code working more quickly, make progress with some part of the system and back-paddle to the dodged part later. The reason is to develop insight that will help solve the larger problem.

### 2.2.1 Process

1. Data Description
2. Data Examples
3. Function Description
4. Function Examples
5. Function Template
6. Code
7. Tests
8. Review and Refactor

**Programming to an Interface** More dynamic, can change models, less code to write and a layer of abstraction. ADT

## 2.3 Recursion

### 2.3.1 Recursion types

1. Simple recursion: There is at most one recursive call (in each branch) and the argument is decremented by one.
2. Complete recursion: Some argument becomes smaller in the recursive call, but not necessarily.
3. Multiple recursion: There are multiple recursive calls (in the same branch).
4. Mutual recursion: Two or more functions are defined in terms of each other.
5. Nested recursion: An argument to a recursive call is computed by a recursive call.
6. Recursion on a generalized problem: Sometimes, no suitable recursion scheme is obvious.

## 2.4 Complexity

### 2.4.1 Growth

1. Big  $\theta$  Notation: estimate of growth in intervals determine by constants Definition For non-negative functions  $f$  and  $g$ ,  $f(n) = \theta(g(n))$  if and only if there exist  $n_0 \geq 0$  and  $c_1, c_2 > 0$  such that for all  $n > n_0$   $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ .  $\theta(g(n))$  is the set of all functions  $f(n)$  that are bounded below and above
2. Big  $\Omega$  Notation: estimate of growth Lower bound
3. Big  $O$ : Notation: estimate of growth upper bound

### Relation

$$O(g(n)) \cap \omega(g(n)) = \theta(g(n))$$

## 2.5 Recurrences

### Example:

sumList [] = 0  
 sumList (x:xs) = x + sumList xs

1. pattern matching [] takes  $t_0$  time
2. pattern matching ( $x : xs$ ) takes  $t_1$  time
3. Adding  $x$  with recursive call takes  $t_{add}$
4. Then the recursive call takes  $T(n - 1)$

$$T(n) = \begin{cases} t_0 & \text{if } n = 0 \\ T(n - 1) + t_{add} + t_1 & \text{if } n > 0 \end{cases}$$

### 2.5.1 Closed Form

1. Use the substitution method to obtain a closed form for the following recurrence:

$$\begin{aligned}f(0) &= 5 \\f(n) &= f(n - 1) + n + 2, n > 0\end{aligned}$$

**Hint:**  $1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$  — you do not need to prove this fact.

#### Expansion Method

$$\begin{aligned}f(0) &= 5 \\f(1) &= f(0) + n + 2 = n + 5 + 2 \\f(2) &= f(1) + n + 2 = 2n + 5 + 2 + 2 \\f(3) &= f(2) + n + 2 = 3n + 5 + 2 + 2 + 2 \\f(n) &= +n^2 + 5 + n \cdot 2\end{aligned}$$

#### Induction proof

Step1: test with base case sense the base case is predefine for 0 we do  $n = 1$

$$\begin{aligned}f(1)_{VL} &= 1^2 + 5 + 1 \cdot 2 = 8, \\f(1)_{HL} &= f(0) + 1 + 2 = 5 + 3 = 8 \\f(1)_{VL} &= f(1)_{HL}\end{aligned}$$

Step2: assumption for p  $f(p) = p^2 + 5 + p \cdot 2$

$$\begin{aligned}f(p+1)_{VL} &= f(p) + (p+1) + 2 = (p^2 + 5 + p \cdot 2) + (p+1) + 2 = \\&= (p^2 + (p+1)) + 5 + (p+1) \cdot 2 = (p^2 + 2p + 1) + 5 + (p+1) \cdot 2 \\f(p+1)_{HL} &= (p+1)^2 + 5 + (p+1) \cdot 2 = (p^2 + 2p + 1) + 5 + (p+1) \cdot 2 \\f(p+1)_{VL} &= f(p+1)_{HL}\end{aligned}$$

Conclusion: according to induction hypothesis the recurrence of the function is equal to

$$2n + 5 + \frac{n(1+n)}{2}$$

### Substitution Method

$$\begin{aligned}
f(n) &= f(n-1) + n + 2 \\
&= (f(n-2) + (n-1) + 2) + 1n + 2 \\
&= f(n-2) + (n-1) + n + 2 \cdot 2 \\
&= (f(n-3) + (n-2) + 2)(n-1) + n + 2 \cdot 2 \\
&= f(n-3) + (n-2) + (n-1) + n + 3 \cdot 2 \\
&\quad \vdots \\
&= f(n-k) + (n-(k-1)) + (n-(k-2)) + (n-(k-3)) + \dots + n + k \cdot 2 \\
&\quad \vdots \\
&= f(n-n) + (n-(n-1)) + (n-(n-2)) + (n-(n-3)) + \dots + n + n \cdot 2 \\
&= f(0) + 1 + 2 + 3 + \dots + n + n \cdot 2 = 5 + 1 + 2 + 3 + \dots + n + n \cdot 2
\end{aligned}$$

We can see the following patterns

$$2n + 5 + \sum_{k=1}^n (k) = 2n + 5 + \frac{n(1+n)}{2}$$

### Induction proof

Step1: test with base case since the base case is predefine for 0 we do  $n = 1$

$$f(1)_{VL} = 2 \cdot 1 + 5 + \frac{1(1+1)}{2} = 2 + 5 + 1 = 8,$$

$$f(1)_{HL} = f(0) + 1 + 2 = 5 + 3 = 8$$

$$f(1)_{VL} = f(1)_{HL}$$

Step2: assumption for p  $f(p) = 2p + 5 + \frac{p(1+p)}{2}$

$$f(p+1)_{HL} = f(p) + (p+1) + 2 = (2p + 5 + \frac{p(1+p)}{2}) + (p+1) + 2 =$$

$$= 2(p+1) + 5 + \frac{p(1+p)}{2} + p = 2(p+1) + 5 + \frac{2(p+1) + p(1+p)}{2} =$$

$$= 2(p+1) + 5 + \frac{p^2 + 2p + p + 2}{2} =$$

$$f(p+1)_{HL} = 2(p+1) + 5 + \frac{(p+1)(p+2)}{2} = 2(p+1) + 5 + \frac{p^2 + 2p + p + 2}{2}$$

$$f(p+1)_{VL} = f(p+1)_{HL}$$

Conclusion: according to induction hypothesis the recurrence of the function is equal to

$$2n + 5 + \frac{n(1+n)}{2}$$

## 2.6 Higher-Order Function

### 2.6.1 Higher-Order Functions on Lists

```
map :: (a -> b) -> [a] -> [b]
filter :: (a -> Bool) -> [a] -> [a]
foldl :: (a -> b -> a) -> a -> [b] -> a
foldr :: (a -> b -> b) -> b -> [a] -> b
```

#### **map**

maps a function to each element in list.

#### **filter**

filters elements with a condision

```
filter :: (a -> Bool) -> [a] -> [a]
filter (<6) [6,3,0,1,8,5,9,3] = [3,0,1,5,3]
```

#### **foldl**

foldl recurses over a list “from the left,” i.e., it initially applies the given operation to the first list element and the given start value. starts from the left (first element) and apply the function to with each element. Similar to an accumulator. No one uses it since it is some what useless.

```
foldl :: (a -> b -> a) -> a -> [b] -> a
foldl (*) 1 [1,2,3,4] = 24
```

#### **foldr**

foldr recurses over a list “from the right,” i.e., it initially applies the given operation to the last list element and the given start value. starts from the right (last element) and apply the function to with each element. Similar to an accumulator.

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr (*) 1 [1,2,3,4] = 24
```

## 2.7 Data types

### 2.7.1 Basic

```
String :: ['char'] --list of characters
List :: []          --undefined element types and elements
Tuple :: ()         --Predefine elements
Char :: ''          --single character
Int :: 1            --Hole number with define size
Integer :: 1         --Hole number with undefined size
Float :: 1.1         --Real number with double-precision
Double :: 1.1        --Real number with single-precision
```

### 2.7.2 Maybe Type

If the return is maybe “nothing” then the “Maybe type” is used, since it does not have to return a specific value. It is not polymorphic since you have to specify the type, however “Just” is at its self polymorphic function. If a operation that requires a specific type one needs to remove “Just”, for instance by a let function.

### 2.7.3 New types and enumeration types

New types: One creates more relevant names and format of existing enumeration types. Overload is a problem with the use of the same operations that can not be used for the same data types. One needs to create new operations if it is not from the same type class.

Enumeration types: Instead of new types *type* enumeration types uses the operation call *data*. The difference is that enumeration types is independent from existing types therefor one becomes more flexible and precise.

#### example

```
data newTypeOfDataDerection = North | South | East | West
    deriving (Show) -- inorder to print

:t North
North :: newTypeOfDataDerection

-- We can use new types in pattern matching
oposit :: newTypeOfDataDerection -> newTypeOfDataDerection
oposit North = South
```

#### Type classes

A type class is a set of types that support certain related operations.

No function is applied by default to the new type, therefor you can write “deriving” the following functions are good to have

#### type classes to deriving for new data types

```
deriving (Show) -- in order to print in ghci and print normal
deriving (Eq) -- To test equality
deriving (Ord) -- the first one has the smallest value, order matter for comparison
```

#### New type classes

```
class Eq a where
  (==) :: a -> a -> Bool
```

**New type classes Instances**

```
instance Eq Colour where  
  (==) = eqColour
```

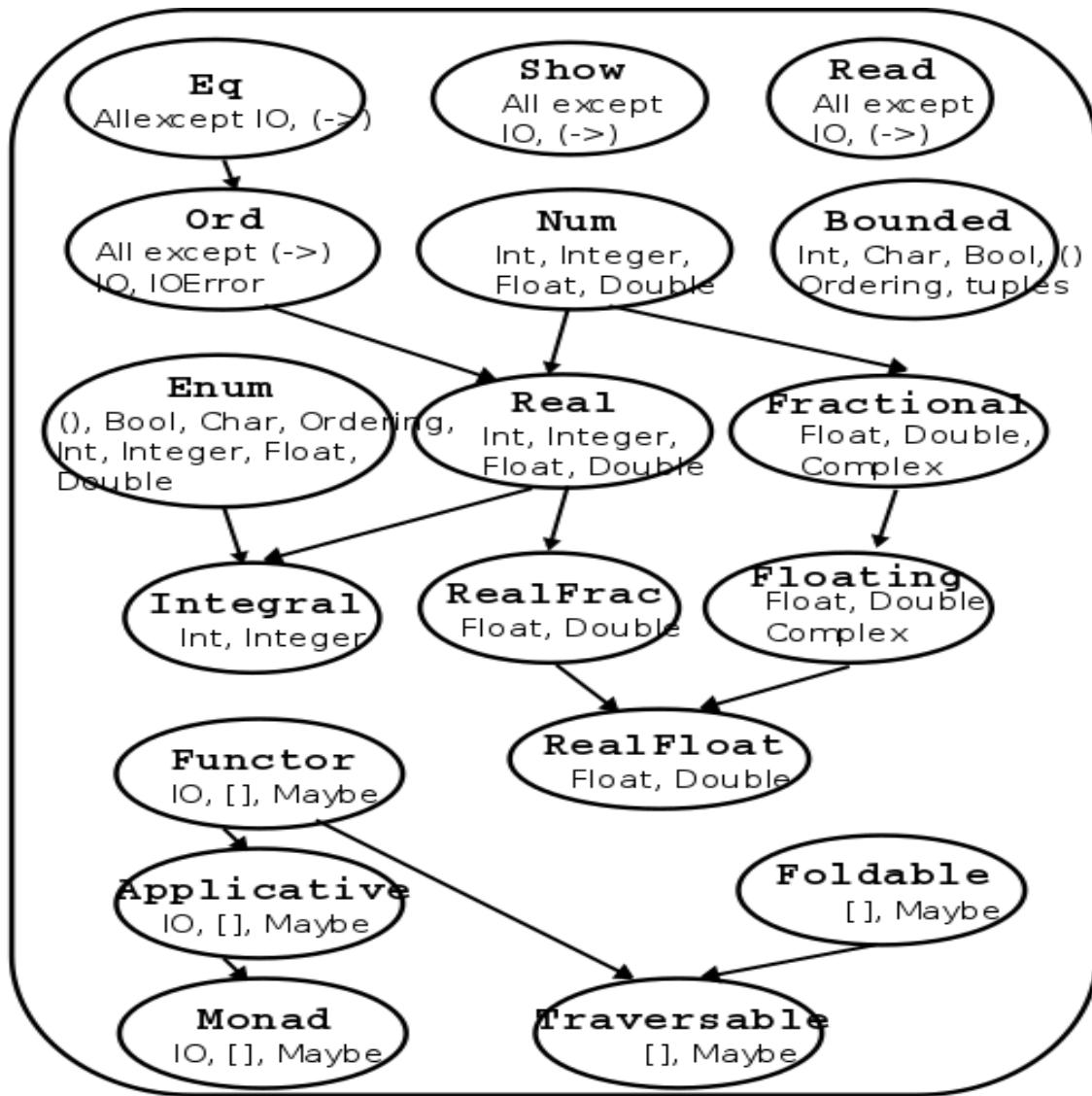


Figure 2.1: Type class

### 2.7.4 Inductive Data Types

Uses a base case then the inductive step as cases of. Multiple arguments. Type constructure

```
data AExp = Atom Int
| Plus AExp AExp
| Times AExp AExp

eval (Atom i)    = i
eval (Plus a b) = eval a + eval b
eval (Times a b) = eval a * eval b
```

### 2.7.5 Trees

#### Terminology

1. Search tree: All nodes on the left side is less then the parent and opposite on the right side
2. Out-degree: is how many children it has. Binary trees has out-degree 0 – 2
3. root Node: is a parent to any number of children at the top of the hierarchy
4. sub Node: is a parent to any number of children
5. Leaf: is a nod that has no children
6. Node: every element
7. Height: most steps from the root node to a leaf

#### Representation

1. Inorder: (Left, Root, Right)
2. Pre-order: (Root, Left, Right)
3. Post-order: (Left, Right, Root)

```
data FBTree a = Leaf a
| Node (FBTree a) a (FBTree a)
deriving (Show)

Node (Leaf 1) 5 (Leaf 21)
rootValue :: FBTree a -> a
rootValue (Node _ a _) = a
rootValue (Node x) = x

height :: FBTree a -> Int
height (Leaf _) = 0
height (Node b a c) = 1 + max (height b) (height c)
```

#### Binary tree

Insertion:  $O(1)$  ( $O(h)$  if search tree)  
 Delition:  $O(n)$  ( $O(h)$  if search tree)  
 Search:  $O(n)$  ( $O(h)$  if search tree)  
 Height:  $O(n)$  (n nodes)

1. Full binary tree: has node out-degree either 2 or 0 worst-case complexity of  $O(\log n)$ .
2. Binary tree: each node has up to an out-degree of 2

#### Binary search tree

Insertion:  $O(h)$  ( $O(n)$  if search tree)  
 Delition:  $O(h)$   
 Search:  $O(n)$   
 Height:  $O(n)$  (n nodes)

### Red and black trees

Insertion:  $O(\log n)$

Deletion:  $O(\log n)$

Search:  $O(\log n)$

Element:  $O(2^r)$  (rank r)

Height:  $O(2 \cdot \log_2(n + 1))$  (n nodes)

Better then a normal binary tree since it balance the tree, therefor it becomes smaller and more efficient to search in. One should use red and black tree when there is a large number of nodes, say 50.

Definition: A red-black tree is a binary search tree where every node is colored either red or black, with the following balancing invariants:

1. No red node has a red parent.
2. Every path from the root to an empty subtree contains the same.
3. A red-black tree with n nodes has height at most  $2 \cdot \log 2(n + 1)$ .
4. there are 4 cases to rebalance (1;4) is similar so is (2;3).

### Algorithm

1. Perform a standard binary-search-tree insertion.
2. Color the new node red.
3. Rebalance the tree, if there is a red node with a red parent.

### Binomial Trees and heaps

Insertion:  $O(\log n)$

Search:  $O(\log n)$  (number of trees n)

Element:  $2^r$  (rank r, n=1 then r=0)

A heap can be used to implement a priority queue, where elements are added to a pool and assigned a priority. In a min-priority queue, extraction of an element yields an element with minimum priority. The smallest node is the root and every child is equal or larger then its parent.

Binomial Trees is a data structure by linking trees of rank  $r - 1$  together. A binomial heap (Vuillemin, 1978) is a list of binomial trees such that each tree satisfies the min-heap property (hence the root of each tree contains its minimum key); and the trees have strictly increasing ranks.

### Terminology

1. Link: putting together two trees.
2. Merge: putting together two heaps
3. Binomial Heap: a list (forest!) of Binomial Trees
4. Binomial Trees have the largest subtree to the left, while Binomial

### Binomial heaps

1. Heaps, extracting minimum element in worst case  $O(\log |h|)$
2. Binomial Trees, The height (here: number of edges on the longest branch) is  $r$ .
3. Binomial Trees, There are  $2^r$  nodes in the tree.
4. Binomial Trees, There are  $\binom{r}{k}$  nodes at level  $k$ . (Hence its name!)
5. Binomial Trees, The root has  $r$  subtrees of ranks  $r - 1, r - 2, \dots, 1, 0$ .
6. A binomial heap  $h$  has at most  $\lceil \lg |h| \rceil + 1$  binomial trees.
7. Inserting a binomial tree into a binomial heap is like addition with base 2.
8. merging is made with two cases either (case 1) when one is smaller or (case 2) when they are equal

`BinoTree = Node Int Int [BinoTree] — Node rank key (subtrees with decreasing rank)`

### 2.7.6 Other data types

#### Tables, Stacking and queuing

1. Table: a list of key-value pairs
2. Stacks: elements accessed in Last-In First-Out (LIFO) order
3. Queues: elements accessed in First-In First-Out (FIFO) order

#### Table operations

```
empty :: Table k v
insert :: Eq k => Table k v -> k -> v -> Table k v
exists :: Eq k => Table k v -> k -> Bool
lookup :: Eq k => Table k v -> k -> Maybe v — value from key
delete :: Eq k => Table k v -> k -> Table k v
iterate :: Table k v -> (b -> (k, v) -> b) -> b -> b — Foldr
keys :: Table k v -> (b -> k -> b) -> b -> b — all keys
values :: Table k v -> (b -> v -> b) -> b -> b — all values
```

#### Stack operations

— interface  
`newtype Stack a = StackImpl [a] — opaque!`

```
empty :: Stack a
isEmpty :: Stack a -> Bool
push :: a -> Stack a -> Stack a — insert
top :: Stack a -> a — the first value
pop :: Stack a -> (a, Stack a) — take out
```

### Queue operations

```
— interface
newtype Queue a = Q [a] — opaque

empty :: Queue a
isEmpty :: Queue a -> Bool
head :: Queue a -> a
enqueue :: Queue a -> a -> Queue a — take out element
dequeue :: Queue a -> Queue a — insert element
toList :: Queue a -> [a]
```

### Hastables

- Key value lookup (an index)
- Array is only define for small index, hash has no limit on available keys.
- Typically we have  $n$  possible keys from set  $U$  for a hashtable (which is an array) with  $m$  slots, where  $n \geq m$ .
- since there is infinitely many elements and limited amount of key there will be element with the same key, therefor a coalition is created.
- Worst-Case Retrieval: time complexity of retrieving a element.
- Load Factor: How much data is in the table  $\frac{\text{elements}}{\text{slots}}$
- Rehashing: make the hastable more balanced.

### Collision Resolution by Chaining

- Most commonly used collision resolution
- Let each array slot (also called a bin) hold a list of elements (called a chain).
- In other words, When collision then add it to a list in that element

### Collision Resolution by Open Addressing

- Start with a table with each element is  $\perp$  previously used  $\Delta$ .
- Probing: is a function to insert items in a hastable therefore resolves collations
- Types of probing:
  - Linear probing:  $f(i) = i$ .
  - Quadratic probing:  $f(i) = c_2 \cdot i^2 + c_1 \cdot i$ , where  $c_2 \neq 0$ .
  - Double hashing:  $f(i) = i \cdot h''(i)$ , where  $h''$  is another hash function.
- Inserting with Linear Probing: Insert it to the next available key
- Deleting with Linear Probing: Ignores  $\perp$  and  $\Delta$  idex will change.

### 2.7.7 Graphs

#### Types of graphs

- list
- tree
- forest
- Directed Acyclic Graph (DAG)

#### Terminology

- Node, vertex (plural: vertices)
- Edge connects two nodes.
- Self-loop edge from node to itself
- Adjacent nodes connected by an edge
- Degree number of edges from or to a node
- In-degree number of edges to a node
- Out-degree number of edges from a node

#### Representation

- **Adjacency Matrix** — a 2-dimensional array  $A$  of 0/1 values, with  $A[i, j]$  containing the number of edges between nodes  $i$  and  $j$  (undirected graph), or from node  $i$  to node  $j$ 
  - + edge existence testing in  $\theta(1)$  time
  - finding next outgoing edge in  $O|V|$  time
  - + compact representation for dense graphs (when  $|E|$  is close to  $|V|^2$ )
- **Adjacency List** — a 1-dimensional array  $\text{Adj}$  of adjacency lists, with  $\text{Adj}[i]$  containing a list of the nodes adjacent to node  $i$ .
  - + finding next outgoing edge in  $\theta(1)$  time
  - edge existence testing in  $O(|V|)$  time
  - + compact representation for sparse graphs (when  $|E|$  is much smaller than  $|V|^2$ )
- **Edge List** — a list of tuples,  $(i, j)$ , for each edge  $(i, j)$  (plus a list of the nodes).
  - edge existence test in  $\theta(|E|)$ .
  - finding next outgoing edge in  $O(|E|)$  (unless appropriately sorted).
  - + compact representation for sparse graphs (when  $|E|$  is much smaller than  $|V|^2$ )

### topological sort

A topological sort is a linear ordering of all the nodes in a directed acyclic.

### Algorithm

1. Select a node with in-degree 0.
2. Output it.
3. Remove it.
4. Repeat (from 1) until no nodes are left.

Total running time is  $\theta(|V| + |E|)$ .

### Graph Traversals

- Breadth-first search (BFS). Uses a queue for each grey node.  
Time complexity:  $\theta(|V| + |E|)$  — linear in the size of the graph.
- Depth-first search (DFS). Uses a stack for each (grey) node and has a rest of nodes (white).  
Time complexity:  $\theta(|V| + |E|)$  — linear in the size of the graph.

#### Breadth-First Search: Algorithm

Input: Some node A.

1. Paint A gray. Paint other nodes white. Add A to an initially empty FIFO queue of gray nodes. All grey nodes is in the queue. It is a queue not a stack so first in first out.
2. Dequeue head node, X. Paint its undiscovered (white) adjacent nodes gray and enqueue them. Paint X black. Repeat until queue is empty. For every black node add it to BFS order (black)

#### Depth-First Search: Algorithm

Input: Some node A.

### DFS(G)

1. Paint all nodes white.
2. For each node v in G: if v is (still) white, DFS-Visit(G,v). Each subsequent call to DFS-Visit in line 2 is called a restart.

### DFS(G)

1. Colour v gray.
2. For each node u adjacent to v: if u is white, DFS-Visit(G,u).

**Strongly-Connected Components**

- Strongly connected component (SCC): maximal set of nodes where there is a path from each node to each other node.
- Many algorithms first divide a digraph into its SCCs, then process these SCCs separately, and finally combine the sub-solutions. (This is not divide & conquer, since a different algorithm is run on each SCC!)
- An undirected graph can be decomposed into its connected
  - 1. Enumerate the nodes of G in DFS finish order, starting from any node
  - 2. Compute the transpose  $G^T$  (that is, reverse all edges)
  - 3. Make a DFS in  $G^T$ , considering nodes in reverse finish order from original DFS
  - 4. Each tree in this depth-first forest is a strongly connected component

## 2.8 Important syntax

### 2.8.1 Let

```

let x = 1 in x * 2 == 2
case x of
  1 -> "Hello"
  2 -> "H"
  3 -> "Hel"

input: 3 == "Hel"

# other examples
let f x y = x + 3 >= y + 3.1 in f 1 1 == False

f x = let g z = z+1 in g (g x)
f 1 == 3

:t div
div :: Integral a => a -> a

:t (/)
(/) :: Fractional a => a -> a -> a

```

### 2.8.2 IO

**monads** Is used to return an IO and uses a operation ( $\gg=$ ) that replaces *do-notation*

```

class Monad m where
  ( $\gg=$ ) :: m a -> (a -> m b) -> m b
  return :: a -> m a

```

## 2.9 Sorting Algorithms

1. Insertion Sort: One at a time
2. Bubble Sort: sort in order two a time starting from left and work to the right side.
3. Merge Sort: Divide and Conquer, split into even pieces and sort them and then merge/sort again.
4. Quicksort: Divide and Conquer, takes a pivot to split smaller and larger.

## Chapter 3

### Algebra 1

### 3.1 logik

Utsaga: ett påstående som erhåller antigen värderna sann(S) eller falsk(F) vilket ge en stluten utsaga eller ej ett sannings värde vilket kallas för öppna utsagor

Kombinerade utsagor:

Kunjuktion utsagor: består av två utsagor som vi kallar A och B

där and:  $\wedge (A \wedge B)$

Dissjunktion utsagor: består av två utsagor som vi kallar A eller B

or:  $\vee (A \vee B)$

icke utsaga A (motsatsen):  $\neg A$

Alla:  $\forall$

Minst en:  $\exists$

$\neg(\forall x : A) \Leftrightarrow \exists x : \neg A$

$\neg(\exists x : A) \Leftrightarrow \forall x : \neg A$

**Exempel:**

A: Alla reala tal x gäller att  $(x + 1)^2 = 0$

$\forall x : (x + 1)^2 = 0$

$\neg(\forall x : (x + 1)^2 = 0) = \exists x : (x + 1)^2 \neq 0$

$\neg A$ : Det finns reala tal x gäller att  $(x + 1)^2 \neq 0$

#### 3.1.1 värde tabeller

Kunjuktions värdetabel:

A	B	$A \wedge B$
S	S	S
S	F	F
F	S	F
F	F	F

Dissjunktions värdetabel:

A	B	$A \vee B$
S	S	S
S	F	S
F	S	S
F	F	F

#### 3.1.2 Implikationer

A medför B:  $A \Rightarrow B$  (implikation)

A och B medför varandra:  $A \Leftrightarrow B$  (ekvivalens)

**Implication värdetabell:**

A	B	$A \Rightarrow B$
S	S	S
S	F	F
F	S	S
F	F	S

**Ekvivalens värdetabel:**

A	B	$A \Leftrightarrow B$
S	S	S
S	F	F
F	S	F
F	F	S

**Exempel1:**

$$x = \sqrt{6 - x} \quad (3.1)$$

$$x = \sqrt{6 - x} \Rightarrow x^2 = 6 - x \Leftrightarrow x^2 + x - 6 = 0$$

$$\text{pq-formeln: } x = -\frac{1}{2} \pm \sqrt{\frac{1}{4} + \frac{24}{4}} \Leftrightarrow (x = -3) \vee (x = 2)$$

Eftersom det är en implikation är höger och inte ekvivalens så behöver inte rötterna vara sanna

Testar för falska rötter:

$$2 = \sqrt{6 - 2} \text{ Sann}$$

$$2 = \sqrt{6 - (-3)} \text{ Falsk } 2 \neq \sqrt{6 - (-3)}$$

**Exempel2:**

$$(x + 2)(x + 1) = 2x(x + 1) \quad (3.2)$$

$$(x + 2)(x + 1) = 2x(x + 1) \text{ is not } \Rightarrow (x + 2) = 2x \text{ (x=0 is not allowed)}$$

Insted do ass following:

$$\begin{aligned} (x + 2)(x + 1) = 2x(x + 1) &\Leftrightarrow (x + 2)(x + 1) - 2x(x + 1) = 0 \Leftrightarrow (x + 1)(x + 2 - 2x) = 0 \Leftrightarrow \\ &\Leftrightarrow (x + 1 = 0) \vee (2 - x = 0) \end{aligned}$$

svar ej: x=1 och x=2

svar: x=1 eller x=2

svar:  $x = 1 \vee x = 2$

## 3.2 Mängder

Naturliga tal:  $\mathbb{N} = \{0, 1, 2, 3, \dots\}$

Heltal:  $\mathbb{Z} = \{\dots - 2, 1, 0, 1, 2, \dots\}$

Rationella tal:  $\mathbb{Q} = \left\{ \frac{a}{b} \mid a, b \in \mathbb{Z}, b \neq 0 \right\}$

Irrationella tal:  $\mathbb{P} = \mathbb{R} \setminus \mathbb{Q}$  eller  $\{x \mid x \in \mathbb{R}, x \notin \mathbb{Q}\}$

Reella tal:  $\mathbb{R} = \mathbb{P} \cup \mathbb{Q}$

$$A \cup B = \{x : (x \in A) \vee (x \in B)\}$$

$$A \cap B = \{x : (x \in A) \wedge (x \in B)\}$$

$$A \setminus B = \{x : (x \in A) \wedge (x \notin B)\}$$

$$A^\# = \{x : (x \in X) \wedge (x \notin A)\}$$

**Exempel:**

$$\text{Bevisa: } X \setminus (A \cup B) = (X \setminus A) \cap (X \setminus B) \quad (3.3)$$

$$\begin{aligned} x \in (X \setminus (A \cup B)) &\Rightarrow (x \in X) \wedge (x \notin (A \cup B)) \Rightarrow \\ &\Rightarrow (x \in X) \wedge (x \notin A) \wedge (x \notin B) \Rightarrow \\ &\Rightarrow (x \in X \setminus A) \wedge (x \in X \setminus B) \Rightarrow \\ &\Rightarrow x \in (X \setminus A) \cap (X \setminus B) \Rightarrow \\ &\Rightarrow X \setminus (A \cup B) \subseteq (X \setminus A) \cap (X \setminus B) \end{aligned}$$

### 3.3 Bevis

#### 3.3.1 Induktions bevis

steg 1: bevisa att det gäller för basfallet

Steg 2: bevisar att  $p \Rightarrow p$  a. Antar att det stämmer för  $p$  b. Vissar att med att stoppa in antagandet i  $p+1$  så blir  $hl = vl$  Tips: Förenkla  $hl$  först och sedan  $vl$  på klad papper fram och tillbaka **Exempel Recursion:**

$$a_1 = 2, a_{n+1} = \frac{7a_n}{7 - a_n}, n \in \mathbb{N}$$

$$\text{Vissa med induktion att } a_{n+1} = \frac{14}{7 - 2n}$$

Bevis med induktions

steg 1: visar att påståendet som vi kallar  $p$  gäller för basfallet ( $n=1$ )

$$VL_1 : a_{1+1} = a_2 = \frac{7 \cdot 2}{7 - 2} = \frac{14}{5}, HL_1 : a_{1+1} = a_2 = \frac{14}{7 - 2} = \frac{14}{5}$$

steg 2: visar att  $p_m \Rightarrow p_{m+1}$

steg 2a: antar att  $p_m$  gäller

$$a_{m+1} = \frac{14}{7 - 2m}$$

steg 2b: bevisar attt  $p_m \Rightarrow p_{m+1}$  genom att använda antagandet

$$\begin{aligned} VL_{m+1}a_{m+2} &= \frac{7a_{m+1}}{7 - a_{m+1}} = \frac{7 \frac{14}{7 - 2m}}{7 - \frac{14}{7 - 2m}} = \frac{\frac{7 \cdot 14}{7 - 2m}}{\frac{7(7 - 2m) - 14}{7 - 2m}} = \\ &= \frac{7 \cdot 14}{7(7 - 2m + 2)} = \frac{14}{7 - 2(m + 1)} \\ HL_{m+1} &: \frac{14}{7 - 2(m + 1)} \end{aligned}$$

Enlight induktionsprincipen är  $p_m$  sann för alla  $n = 1, 2, 3, \dots$  VSB

#### 3.3.2 Motsägelse bevis

Steg 1: Formulera utsagan och icke utsagan Steg 2: Hitta en motsägelse med utsagan Tips: Förenka båda led, tänk på teorin vi har, bättre att gå vaga moteveringar en inga alls

Bevis med motsägelse

Antar att motsatsen är sann

### 3.4 Delbarhet

a är delbar med b, altså kvoten ger ingen rest. Vi följand:  $a \mid b$  (3.4)

**Divitions algoritmen:**

$$\begin{aligned} a, b &\in \mathbb{Z} \\ a &\geq 0 \wedge b \geq 0 \\ a \mid b &\Rightarrow (q \in \mathbb{Z} : q \geq 0) \wedge (r \in \mathbb{Z} : 0 \leq r \leq a) \text{ Sådant att} \\ b &= qa + r \\ q &= \text{kvoten}, r = \text{resten} \end{aligned}$$

#### 3.4.1 Största Gemensama Delaren (SGD)

$$\begin{aligned} SGD(a, b) \\ a &= bq + r \\ 0 \leq r &\leq b \end{aligned}$$

**Euklides algoritm:**

$$\begin{aligned} SGD(a, b) \\ a &= bq_1 + r_1 \\ b &= r_1q_2 + r_2 \\ r_1 &= r_2q_3 + r_3 \\ r_2 &= r_3q_4 + r_4 \\ &\vdots \\ &\vdots \\ r_{k-3} &= r_{k-2}q_{k-1} + r_k \\ r_{k-2} &= r_{k-1}q_k + 0 \\ SGD(a, b) &= r_k \\ \text{Om } r_k &= 1 \Rightarrow a \in \text{primtal} \vee b \in \text{primtal} \end{aligned}$$

**Exempel:**

$$\text{Förenkla } \frac{114}{96}$$

$$\begin{aligned} SGD(114, 96) : \\ 114 &= 1 * 96 + 18 \\ 96 &= 5 * 18 + 6 \\ 18 &= 3 * 6 + 0 \\ \frac{114}{6} &= 19 \\ \frac{96}{6} &= 16 \\ \frac{19}{16} \end{aligned}$$

**Lemma:**

$$\begin{aligned} a, b \in \mathbb{Z} \\ x, y \in \mathbb{Z} \\ SGD(a, b) = ax + by \end{aligned}$$

**Aritmetiska fundamentalsatsen:**

$$\begin{aligned} a \in \mathbb{Z} \wedge a \geq 2 \Rightarrow \\ \Rightarrow a \text{ kan endast primtalsfaktoriseras på ETT SÄTT} \end{aligned}$$

**Lemma 2.7:**

$$\begin{aligned} a \geq 2 \wedge a \notin \text{Primatal} \Rightarrow \\ \Rightarrow q \in \text{Primatal} \wedge q \mid a \wedge \text{a går att primtals faktoriera} \end{aligned}$$

### 3.4.2 Primtal

**Sats:**

För att bestäma om tal a är ett primtal

$$a \geq 1 \wedge a \in \text{Primatal}$$

om ett tall p finns som delar a gäller följande

$$2 \leq p \leq \sqrt{a} \leq a$$

**Exempel:**

Bestäm om 211 är ett primtal (3.5)

Om 211 är ett primtal så finns det inte en äkta delare a

$$2 \leq a \leq \sqrt{211}$$

$$\sqrt{211} \approx 16$$

$$a : \{\emptyset, \beta, \emptyset, \pi, \alpha, \alpha\}$$

a kan inte vara en äktadelare

**Euklides algoritm:**

Det finns oändligt många primtal

**Bevis:**

Motsägelsebevis

Antar att det finns ändligt många primtal

$$p_1, p_2, P_3, \dots, p_n$$

$$M = \prod_{k=1}^{n+1} n + 1 \Rightarrow M > p_j, j = 1, 2, 3, \dots, n$$

Vissar att M är ett primtal

$1 < b < M \wedge b \mid M$  Där b är minsta äkta delaren av M  $\Rightarrow$

$$\Rightarrow M = p_1 * b \Rightarrow p_1 \mid 1 \wedge p \geq 2$$

Detta är falskt eftersom båda utsägorna kan inte vara samtidigt

Eftersom motsatsen inte fungerar resulterar det i att satsen är sann

### 3.5 Diofantiska ekvationer

Sats:

$$\begin{aligned} ax + by = c \wedge a, b, c \in \mathbb{Z} & \quad a \neq 0, b \neq b \\ \Rightarrow & \\ ax + by = c \text{ Där } SGD(a, b) = 1 & \\ \text{Har den anmäla lösningen:} & \\ x = Cx_0 - nb \wedge y = Cy_0 + na & \end{aligned}$$

**Exempel:**

En lastbil lastas med 12kg packet och 20kg paket. Totalt väger lasten 296, hur många av varge packet? (3.6)

$$\begin{aligned} ax + by = C & \Leftrightarrow 12x + 20y = 296 \\ \text{steg1: Testar om } SGD(a, b) \mid c & \\ SGD(20, 12) = 4 \Rightarrow 4 \mid 296 & \\ \text{steg2: delar SGD med HL och VL} & \\ \frac{12x - 20y}{4} = \frac{296}{4} & \Leftrightarrow 3x - 5y = 74 \\ SGD(5, 3) : & \\ 5 = 1 \cdot 3 + 2 & \\ 3 = 1 \cdot 2 + 1 & \\ 2 = 2 \cdot 1 + 0 & \\ \text{steg3: Hjälp ekvation för att hittax}_0, y_0 & \\ 3x_0 - 5y_0 = 1 & \\ 1 = 3 - 1 * 2 & \\ 1 = 3 - (5 - 3) & \\ 1 = 2 \cdot 3 - 1 \cdot 5 & \\ x_0 = 2, y_0 = -1 & \\ \text{steg4: almäna lösningen } x = Cx_0 - bn, y = Cy_0 + an & \\ x = 74 \cdot 2 - 5n = 148 - 5n & \\ y = 74 \cdot (-1) + 3n = -74 + 3n & \\ \text{steg5: hittar godtyckliga lösningar} & \\ \text{Intervallet som n ligger i för x-termen:} & \\ n = 29, 28, 27, \dots, x = 148 - 5 \cdot 28 = 148 - 140 = 8 & \\ \text{Intervallet som n ligger i för y-termen:} & \\ n = 27, 28, 29, \dots, x = -74 + 3 \cdot 28 = 74 - 84 = 10 & \\ n = 28, x = 8, y = 10 & \\ 12 \cdot 8 + 20 \cdot 10 = 296 & \end{aligned}$$

## 3.6 Talbaser

### 3.6.1 konvertera från decimal bass till annan bas

$$175_8 = 1 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0$$

### 3.6.2 konvertera från annan bas till decimal bas

$$1609_{10} = (3 \cdot 8^3 + 1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0) = 3111_8$$

eller så kan man använda euklides algoritm

skriv 517 i talbas 3

$$517 = 172 \cdot 3 + 1$$

$$172 = 57 \cdot 3 + 1$$

$$57 = 20 \cdot 3 + 0$$

$$20 = 6 \cdot 3 + 2$$

$$6 = 2 \cdot 3 + 0$$

$$2 = 0 \cdot 3 + 2$$

Svar:  $517_{tio} = 202011_{tre}$

### 3.6.3 Andra exempel

**Exempel:** Skriv  $137_{nio}$  i bass tre

$$\begin{aligned} 137_{nio} &= 1 \cdot 9^2 + 3 \cdot 9^1 + 7 \cdot 9^0 = 1 \cdot 3^4 + 3 \cdot 3^2 + 7 \cdot 3^0 = \\ &= 1 \cdot 3^4 + 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0 = 11021_{tre} \end{aligned}$$

## 3.7 Functioner

**Typer av funktioner:**

Injectio: alla element x har olika värden y  $f : A \rightarrow B, \{\forall x \in A : x_1 \neq x_2, f(x_1) \neq f(x_2)\}$

Surjekcio: mängd D är definitions mängden  $\{g : C \rightarrow D, g(x) = y, (\forall y \in D \wedge \exists x \in C)\}$

Bijekcio: Injectio  $\wedge$  Surjekcio

**Kareskapprodukten:**

$$A \times B = \{(a, b) : a \in A \wedge b \in B\}$$

Låt  $A = \{1, 2, 3\} \wedge B = \{x, y, z, w\}$

$A \times B :$

$\{(1, x), (1, y), (1, z), (1, w)\}$

$\{(2, x), (2, y), (2, z), (2, w)\}$

$\{(3, x), (3, y), (3, z), (3, w)\}$

### 3.7.1 Inversen

En funktions invers kan enda

$f : A \rightarrow B \wedge \text{Bijektiv} \Rightarrow f^{-1}(x)$  Finns, där

(1)  $x = f^{-1}(y) \Leftrightarrow y = f(x)$

(2)  $D_{f^{-1}} = V_f \Leftrightarrow D_f = V_{f^{-1}}$

(3)  $x = f^{-1}(f(x)), x \in D_f = V_{f^{-1}}$

(3)  $y = f^{-1}(f(y)), y \in D_f = V_{f^{-1}}$

### 3.7.2 Relatioiner

Relation:  $xRy$

Reflexiv:  $\forall x \in X : xRx$

Symetrisk:  $xRy \Rightarrow yRx, x \in X \wedge y \in X$

Transitiv:  $(xRy) \wedge (yRz) \Rightarrow xRz, \forall x, y, z \in X$

Ekvivalensrelation: Reflexiv och Symetrisk Transitiv

## 3.8 Summor

### 3.8.1 Aritmetiska summor

$$s_n = a_1 + a_2 + a_3 + \dots + a_n = \frac{n(a_1 + a_n)}{2} \quad (3.7)$$

### 3.8.2 Geometriska summor

Börjar alltid med exponenten 0 och gör om summan så att den passar i följande talföljd:

$$s_n = a + ak + ak^2 + \dots + ak^{n-1} = \frac{a(k^n - 1)}{k - 1} \quad (3.8)$$

**Exempel:**

$$\sum_{k=n}^{2n} (2^k - k) \quad (3.9)$$

sätter f = 0 = k - n

$$\begin{aligned} \sum_{f=0}^n (2^{f+n} - (f + n)) &= 2^n * \sum_{f=0}^n (2^f) - \sum_{f=0}^n (f + n) \\ \frac{2^n(2^{n+1} - 1)}{2 - 1} - \frac{3n(n + 1)}{2} &= 2^{2n+1} - 2^n - \frac{3n(n + 1)}{2} \end{aligned}$$

### 3.9 Kongruensräkning

Räkneregler:

$$\begin{aligned}a + b \pmod{n} &\equiv a \pmod{n} + b \pmod{n} \\a \cdot b \pmod{n} &\equiv a \pmod{n} \cdot b \pmod{n} \\a^b \pmod{n} &\equiv (a \pmod{n})^b\end{aligned}$$

**Exempel:** Vilket är det minsta positiva rest som kan erhållas vid division av  $19^{18}$  med 17?

$$19^{18} \equiv 2^{18} \pmod{17} \equiv 2^4 \cdot 2^4 \cdot 2^4 \cdot 2^4 \cdot 2^2 \pmod{17} \equiv (-1)^4 \cdot (-1)^4 \cdot (-1)^4 \cdot (-1)^4 \cdot 2^2 \pmod{17}$$

svar: resten blir 4

## 3.10 Kardinalitet

Kardinalitet eller "mäktighet" är ett sett att räkna med mängders sorlek och alla oändliga mängder har samma kardinalitet fast det är en delmängd. Naturliga tall har samma kardinalitet som reala tal trots att naturliga tal är en del mängd av reala talen

Låt A och B vara mängder. Vi sägger att A och B har samma kardinalitet då det finns en bijektion  
 $\exists f : A \rightarrow B, A \sim B$

Vi säger att A står i relation med B omm A och B har samma kardinalitet  $ARB$

### 3.10.1 Uppräkneligamängder

En mängd X sägs vara uppräknerlig omm X har samma kardinalitet som  $\mathbb{N}$

$\exists g : \mathbb{N} \rightarrow X$  där g är bijektiv

Exempel på uppräknerliga mängder är  $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \{1, 5, 78\}$

Exempel på ej uppräknerliga mängder  $\mathbb{R}, (0, 1)$

## 3.11 Polynom

### 3.11.1 Polynom division

Triviala delare: ej heltals kvot med delaren, har en kostant sådant  $\lambda \cdot f, \lambda \notin \mathbb{Z}$

Äkta delare: heltals kvot med delaren  $f(x), \exists a \in \text{polynom} : a | f(x)$

Irreducible: om polynomet endast har triviala delare det finns lösningar heltaslösningar  $f(x) = 0$

Reducible: om polynomet har äkta delare

Multiplisitet: vilken grad polynomet har

**Exempel:**

Man vet att ekvationen  $z^4 - 2z^3 - 7z^2 + 26z - 20 = 0$  har roten  $z = 2+i$ . Lös ekvationen fullständigt. (3.10)

$z = 2 + i$  Är en lösning är också konjugatet en lösning enligt faktorsatsen  $\bar{z} = a - bi$

$$z = 2 \pm i$$

Vilket betyder att följande går att factorisera ut polynomet

$$(z - (2 + i))(z - (2 - i)) = z^2 - 4z + 5$$

**Långdivision (liggande stolen):**

$$\begin{array}{r} x^2 + 2x - 4 \\ \hline x^2 - 4x + 5 ) \overline{x^4 - 2x^3 - 7x^2 + 26x - 20} \\ \quad - x^4 + 4x^3 - 5x^2 \\ \hline \quad 2x^3 - 12x^2 + 26x \\ \quad - 2x^3 + 8x^2 - 10x \\ \hline \quad - 4x^2 + 16x - 20 \\ \quad 4x^2 - 16x + 20 \\ \hline 0 \end{array}$$

$z^2 + 2z - 4 = 0$  Är också en lösning som till slut ger följande

$$z = -1 \pm \sqrt{5}$$

$$z = 2 \pm i$$

Varge n grads polynom har alltid n stycken komplexa lösningar

### 3.11.2 Faktorsatsen

$$f(x) = (x - \alpha)p(x)$$

**Exempel, Gemensam root hos två polynom:**

$p(x) = x^4 - x^3 + x^2 + 2 = 0$ ,  $g(x) = x^3 + 4x^2 + 4x + 3 = 0$  har en Gemensam root  
 Eftersom polynomen har en Gemensam root vet vi att det finns en gemensam  $(x - \alpha)$

$$f(x) = (x - \alpha)p_1(x)$$

$$g(x) = (x - \alpha)p_2(x)$$

euklides algoritm:

$$f(x) = (x - 5)g(x) + 17(x^2 + x + 1)$$

$$g(x) = \left(\frac{1}{17}(x + 3)\right)(17(x^2 + x + 1)) + 17(x^2 + x + 1) + 0$$

$$h(x) = x^2 + x + 1$$

$$f(x) = (x - 5)(x + 3)h(x) + 17h(x) = h(x)((x - 5)(x + 3) + 17) = h(x)(x^2 - 2x + 2)$$

$$g(x) = (x + 3)h(x)$$

$$f(x) = 0 \Leftrightarrow (h(x) = 0 \vee x^2 - 2x + 2 = 0)$$

$$h(x) = 0 \Rightarrow x = -\frac{1}{2} \pm \frac{\sqrt{3}}{2}i$$

$$x^2 + 2x + 2 = 0 \Rightarrow x = 1 \pm \sqrt{1 - 2} = 1 \pm i$$

**Exempel, Heltalslösning med okänd konstant:**

$$x^3 + bx^2 - 7x - 7 = 0 \text{ har en heltals lösning}$$

Eftersom ekvationen har en heltals lösning vet vi att  $\alpha \mid 7$

Kandidaterna är  $\{1, -1, 7, -7\}$

$$I(x = 1) 1 + b \cdot 1 - 7 - 7 = 0 \Rightarrow b = 13 \in \mathbb{Z}$$

$$II(x = -1) -1 + b \cdot -1 + 7 - 7 = 0 \Rightarrow b = 1 \in \mathbb{Z}$$

$$III(x = 7) 7 + b \cdot 7 - 49 - 7 = 0 \Rightarrow b = \underline{\quad} \notin \mathbb{Z}$$

$$IV(x = -7) 7 + b \cdot -7 + 49 - 7 = 0 \Rightarrow b = \underline{\quad} \notin \mathbb{Z}$$

svar:  $b=13$ ,  $b=1$

**Exempel, Rationell root:**

$g(x) = 2x^3 + 2x^2 + 2x + 3 = 0$  har en rationell root

Eftersom polynomet har en ratiotonal root vet vi att:

$$\frac{p}{q} \quad p, q \in \mathbb{Z}, \quad SGD(p, q) = 1, \quad p \mid 3 \wedge q \mid 2 \quad (a_0)$$

Det möjliga delarna är  $p = \pm 1, \pm 3, q = \pm 1, \pm 2$

$x = -\frac{3}{2}$  är den enda av kandidaterna som ger en sann root

Vilket vi löser genom polynom division

**Exempel, Renimaginär root:**

$z^4 + 4z^3 + 8z^2 + 12z + 15 = 0$  har en renimaginär root

Eftersom polynomet har en rentimaginär root vet vi att  $z = bi, b \in \mathbb{R}, b \neq 0$

$$0 = b^4 z^4 + 4b^3 z^3 + 8b^2 z^2 + 12bz + 15 = b^4 - 4b^3 i - 8b^2 + 12bi + 15 = (b^4 - 8b^2 + 15) + (12b - 4b^3)i =$$

$$0 = b^4 - 8b^2 + 15$$

$$0 = 12b - 4b^3 = 4b(3 - b^2) \Rightarrow b = 0, b = \sqrt{3}, b = -\sqrt{3}$$

$b = 0$  är ej en giltig lösning. Enligt faktorsatsen får vi följande

$$p(z) = (z - \sqrt{3}i)(z + \sqrt{3}i)Q(z) = (z^2 + 3)Q(z)$$

Vilket vi löser genom polynom division

**Exempel, Dubbel root:**

$p(t) = t^3 - 5t^2 + 3t + 9 = 0$  har en dubbel root

Eftersom polynomet har en dubbel root vet vi att derivatan ger os en root  $p'(t) = 0$

Att räkna ut derivatan gör man genm formeln  $ax^b \Rightarrow b \cdot ax^{b-1}$

$$p'(t) = 3t^2 - 10t + 3 = 0 \Rightarrow t_1 = 3, t_2 = \frac{1}{3} \text{ dock är sista en falsk root}$$

$$p(t) = x - 3^2 Q(t)$$

Vilket vi löser genom polynom division

**Exempel, SGD polynom:**

$$\text{Förenkla } \frac{x^4 + x^3 + 2x - 4}{x^4 - x^3 - 2x - 4}$$

$$SGD(x^4 + x^3 + 2x - 4, x^4 - x^3 - 2x - 4) :$$

$$x^4 + x^3 + 2x - 4 = 1 \cdot x^4 - x^3 - 2x - 4 + (2x^3 + 4x)$$

$$x^4 - x^3 - 2x - 4 = \frac{x}{2} - \frac{1}{2} \cdot (2x^3 + 4x) + (-2x^2 - 4) \text{ med polynom divition}$$

$$2x^3 + 4x = -x \cdot (-2x^2 - 4) + 0$$

Med SGD så kan vi ta ett polynom som är hjämt delbart ex:  $-2(x^2 + x)$

$$\text{polynom divition: } \frac{x^4 + x^3 + 2x - 4}{x^2 + x} = x^2 + x - 2$$

$$\text{polynom divition: } \frac{x^4 - x^3 - 2x - 4}{x^2 + x} = x^2 - x - 2$$

$$\frac{x^4 + x^3 + 2x - 4}{x^4 - x^3 - 2x - 4} = \frac{x^2 + x - 2}{x^2 - x - 2}$$

## Chapter 4

# Single Variable Calculus

## 4.1 Basic

### 4.1.1 Mängder

Naturliga tal:  $\mathbb{N} = \{0, 1, 2, 3..\}$  or  $\{1, 2, 3..\}$

Heltal:  $\mathbb{Z} = \{.. - 2, 1, 0, 1, 2..\}$

Rationella tal:  $\mathbb{Q} = \left\{ \frac{a}{b} \mid a, b \in \mathbb{Z}, b \neq 0 \right\}$

Irrationella tal:  $\mathbb{P} = \mathbb{R} \setminus \mathbb{Q}$  eller  $\{x \mid x \in \mathbb{R}, x \notin \mathbb{Q}\}$

Reella tal:  $\mathbb{R} = \mathbb{P} \cup \mathbb{Q}$

$$A \cup B = \{x : (x \in A) \vee (x \in B)\}$$

$$A \cap B = \{x : (x \in A) \wedge (x \in B)\}$$

$$A \setminus B = \{x : (x \in A) \wedge (x \notin B)\}$$

$$A^\# = \{x : (x \in X) \wedge (x \notin A)\}$$

### 4.1.2 Intervall

Open intervall =  $(1, 4)$

Closed intervall =  $[1, 4]$

$$[a, b] = \{x | a \leq x \leq b\} \quad (4.1)$$

$$[a, \infty[ \quad (4.2)$$

$$]-\infty, \infty[ \quad (4.3)$$

**Exempel: Olikheter och intervall**

$$\frac{2}{x-3} < \frac{5}{x} \quad (4.4)$$

$$\begin{aligned} \frac{2}{x-3} - \frac{5}{x} &< 0 \\ \frac{(x)2}{x(x-3)} - \frac{5(x-3)}{x(x-3)} &< 0 \\ \frac{2x - 5x + 15}{x(x-3)} &< 0 \\ \frac{-3x + 15}{x(x-3)} &< 0 \\ \frac{-3(x-5)}{x(x-3)} &< 0 \\ x \neq 0, x \neq 3 \end{aligned}$$

Värde tabell:

	$x < 0$	$0 < x < 3$	$3 < x < 5$	$5 < x$
$x-5$	-	-	-	+
-3	-	-	-	-
x	-	+	+	+
$x-3$	-	-	+	+
hela	+	-	+	-

### 4.1.3 Funktion

$$f : A \rightarrow B$$

A: domain/definitionsmängd, B: Målmängd/kodomängd/värdemängd

Injektion: alla element x har olika värden  $y : f : A \rightarrow B, \{\forall x \in A : x_1 \neq x_2, f(x_1) \neq f(x_2)\}$

Surjektion: mängd D är definitions mängden  $\{g : C \rightarrow D, g(x) = y, (\forall y \in D \wedge \exists x \in C)\}$

Bijektion: Injektion  $\wedge$  Surjektion

$$f \circ g = f(g(x))$$

Begränsad: functionen är inom ett intervall, altså nedåt och uppåt

Begränsat nedåt: functionen är endast begrensad nedåt

Begränsat uppåt: functionen är endast begrensad uppåt

Jämn function: altså är den symetrisk, ex:  $x^2, f(-x) = f(x)$

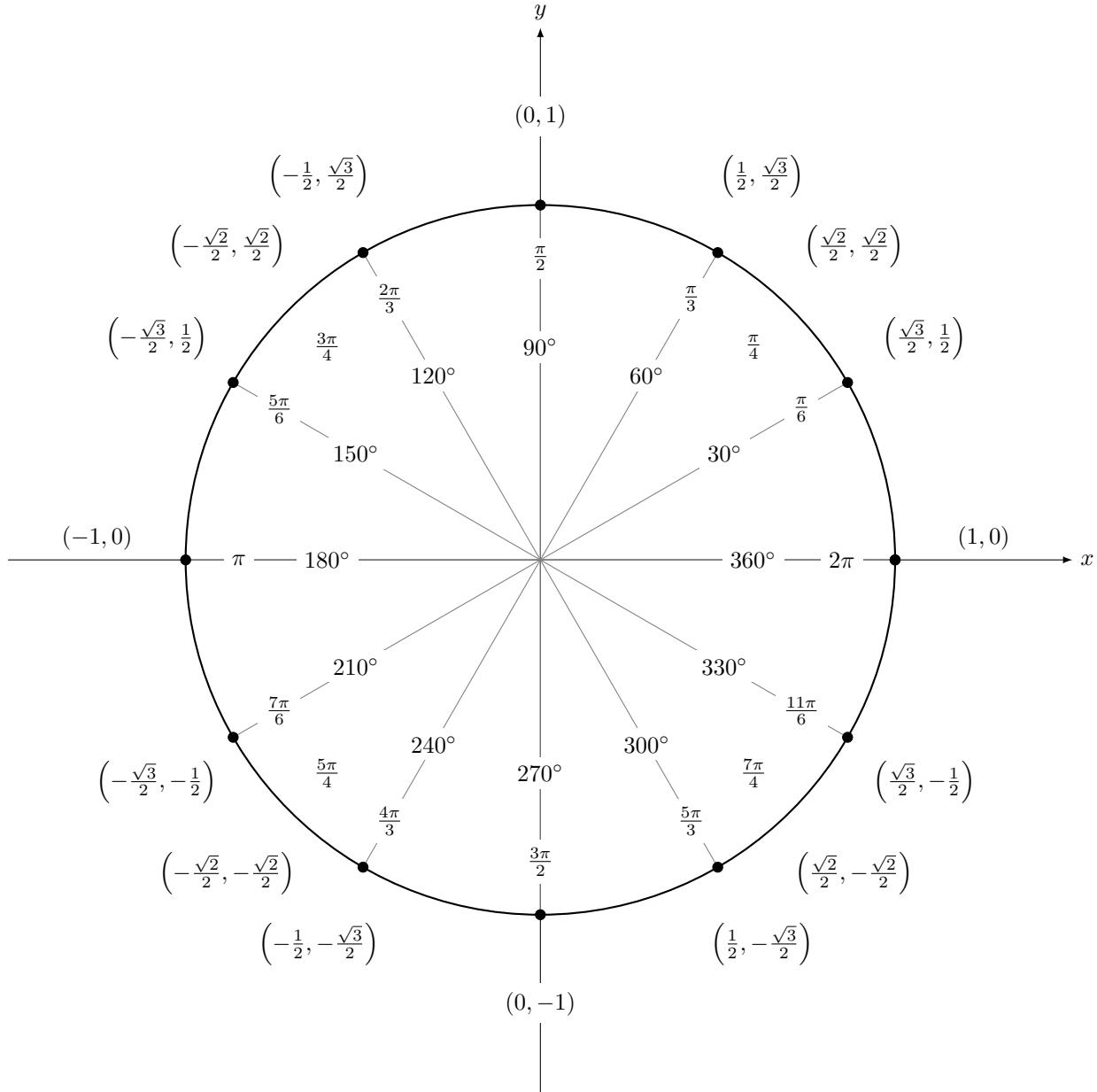
Udda function: altså är den spegelvänt symetrisk, ex:  $x^3, f(-x) = -f(x)$

$$\text{Polynom } p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = \sum_{i=0}^n (a_i x^i)$$

Rationell funktion:  $R(x) = \frac{P(x)}{Q(x)}$ , ex:  $x^{-1} = \frac{1}{x}$  ej polynom men rationell funktion

#### 4.1.4 Trigonometri

$\sin$	0	$\frac{\sqrt{0}}{2}$	$\frac{\sqrt{1}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{4}}{2}$
$\cos$	$\frac{\sqrt{4}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{1}}{2}$	$\frac{\sqrt{0}}{2}$	



**Sats:**

$$360^\circ = 2\pi \text{rad}$$

$$v_g = v_r * \frac{180^\circ}{\pi}$$

$$v_r = v_g * \frac{\pi}{180^\circ}$$

**Sats:**

$$-1 \leq \sin t \leq 1$$

$$-1 \leq \cos t \leq 1$$

**Sats:**

$$\cos(-t) = \cos(t)$$

$$\sin(-t) = -\sin(t)$$

$$\tan(-t) = \frac{\sin(-t)}{\cos(-t)} = \frac{-\sin(t)}{\cos(t)}$$

**Additionsformlerna:**

$$\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \sin(\beta)\cos(\alpha)$$

$$\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \sin(\beta)\cos(\alpha)$$

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\beta)\sin(\alpha)$$

$$\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\beta)\sin(\alpha)$$

**Trigonometriska ettan:**

$$(\sin t)^2 + (\cos t)^2 = 1$$

$$\sin^2 t + \cos^2 t = 1$$

#### 4.1.5 Exempel: Trigonometri

$$\begin{aligned} \cos \frac{\pi}{12} &= \cos \left( \frac{\pi}{3} - \frac{\pi}{4} \right) = \cos \frac{\pi}{3} \cos \frac{\pi}{4} + \sin \frac{\pi}{3} \sin \frac{\pi}{4} \\ &= \left( \frac{1}{2} \right) \left( \frac{1}{\sqrt{2}} \right) + \left( \frac{\sqrt{3}}{2} \right) \left( \frac{1}{\sqrt{2}} \right) = \frac{1 + \sqrt{3}}{2\sqrt{2}} \end{aligned}$$

## 4.2 Gränsvärden

$$\lim_{x \rightarrow x_0} f = L \wedge \lim_{x \rightarrow x_0} g = M \Rightarrow \lim_{x \rightarrow x_0} (f + g) = L + M$$

Squeeze theorem:  $h(x) \leq f(x) \leq g(x)$ ,  $\lim_{x \rightarrow x_0} h(x) = \lim_{x \rightarrow x_0} g(x_0) = L \Rightarrow \lim_{x \rightarrow x_0} f(x) = L$

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} = 1$$

$\lim_{x \rightarrow \infty} \sin(x)$  är ej definierad

**Example: Limit för sin**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow 0} \frac{\sin(2x^2)}{x^2} \\ & \lim_{x \rightarrow 0} \frac{2 \cdot \sin(2x^2)}{2x^2} = 2 \cdot \lim_{x \rightarrow 0} \frac{\sin(2x^2)}{2x^2} = 2 \end{aligned}$$

**Example: 0/0**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow 0} \frac{x \cdot \cos(x) + \frac{\sin(x)}{x}}{x + x^2} \\ & \lim_{x \rightarrow 0} \frac{x \cdot \cos(x) + \frac{\sin(x)}{x}}{x(1 + x)} = \frac{1 + 1}{1} = 2 \end{aligned}$$

**Example: roten ur i nämnaren**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow 0} \frac{x}{\sqrt{x+9}-3} \\ & \lim_{x \rightarrow 0} \frac{x(\sqrt{x+9}+3)}{(\sqrt{x+9}-3)(\sqrt{x+9}+3)} = \lim_{x \rightarrow 0} \frac{x(\sqrt{x+9}+3)}{x+9-9} = \lim_{x \rightarrow 0} \sqrt{x+9} + 3 = 6 \end{aligned}$$

**Example: roten ur i nämnaren**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow -\infty} \frac{2x-1}{\sqrt{3x^2+x+1}} \\ & \lim_{x \rightarrow -\infty} \frac{x(2-\frac{1}{x})}{\sqrt{x^2(3+\frac{1}{x}+\frac{1}{x^2})}} = \lim_{x \rightarrow -\infty} \frac{x(2-\frac{1}{x})}{x\sqrt{3+\frac{1}{x}+\frac{1}{x^2}}} = \frac{-2}{\sqrt{3}} \end{aligned}$$

**Example: infinity sin and squeeze**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow \infty} \frac{\sin(x)}{x} \\ & \left| \frac{\sin(x)}{x} \right| = \frac{|\sin(x)|}{|x|} \leq \frac{1}{|x|} \Rightarrow \text{squeeze } \lim_{x \rightarrow \infty} \left| \frac{\sin(x)}{x} \right| = 0 \\ & \Rightarrow \lim_{x \rightarrow \infty} \frac{\sin(x)}{x} = 0 \end{aligned}$$

### 4.2.1 Kontinuitet

en funktion är kontinuerlig i punkt  $x_0$ ,  $\forall \varepsilon \wedge \exists \delta : |x - x_0| < \delta \Leftrightarrow |f(x) - f(x_0)| < \varepsilon$

Altså så finns det inga hopp i funktionen. Man kan dra penan på grafen utan att släppa om funktionen bestor av flera uttryck är funktionen kontinuerlig om alla uttrycken är det

## 4.3 Derivator

Tangent: linjen som har samma lutning i en punkt som functionens derivata

$$y - f(x_0) = f'(x_0)(x - x_0) \Rightarrow y = ax + b$$

Samband: *Deriverbar  $\Rightarrow$  Kontinuerlig*

### Räkneregler

$f$	$f'$
$c$	0
$x$	1
$x^n$	$nx^{n-1}$
$a^x$	$a^x \ln a$
$e^{kx}$	$ke^x$
$\sin x$	$\cos x$
$\cos x$	$-\sin x$
$\tan x$	$1 + \tan^2 x \frac{1}{\cos^2 x}$
$\ln x$	$\frac{1}{x}$

$$(f + g)' = f' + g'$$

$$(cf)' = cf'$$

$$(fg)' = f'g + fg' \text{ Produktregeln}$$

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2} \text{ Kvotregeln, } g \neq 0$$

#### 4.3.1 Kjedje regeln

om functionen  $g$  är deriverbar i  $x$  och functionen är deriverbar i  $g(x)$

$$\Rightarrow h(x) = f \circ g = f(g(x)) \text{ deriverbar i } x$$

$$h'(x) = (f(g(x)))' = f'(g(x))g'(x)$$

#### Example: Kjedje regeln

Lös:  $(x^x)'$

$$(x^x)' = (e^{x \ln x})' = e^{(x \ln x)} (\ln x + x \frac{1}{x}) = e^{x \ln x} (1 + \ln x) = x^x (1 + \ln x)$$

### 4.3.2 L'Hôpital's rule

$$f, g : \mathbb{R} \rightarrow \mathbb{R} \wedge (\lim_{x \rightarrow x_0} \frac{f}{g} = \frac{0}{0} \vee \text{"} \lim_{x \rightarrow x_0} \frac{f}{g} = \frac{\pm\infty}{\pm\infty} \text{"}) \Rightarrow \lim_{x \rightarrow x_0} \frac{f}{g} = \lim_{x \rightarrow x_0} \frac{f'}{g'}$$

**Example: L'Hôpital's rule**

$$\begin{aligned} \text{Lös: } & \lim_{x \rightarrow 1} \frac{x^4 - 6x^3 + 5x^2}{x^3 - 8x^2 + 7x} \\ & \lim_{x \rightarrow 1} \frac{x^4 - 6x^3 + 5x^2}{x^3 - 8x^2 + 7x} = \text{"} \frac{0}{0} \text{"} \\ \text{L'Hôpital's rule} \Rightarrow & \lim_{x \rightarrow 1} \frac{4x^3 - 18x^2 + 10x}{3x^2 - 16x + 7} \\ & = \frac{-4}{-6} = \frac{2}{3} \end{aligned}$$

### 4.3.3 Medelvärdessatsen

Anta att  $f$  är kontinuerlig på  $[a, b]$  och deriverbar på  $(a, b) \Rightarrow \exists c \in (a, b)$

$$\text{så att } \frac{f(b) - f(a)}{b - a} = f'(c)$$

**Example: Medelvärdessatsen**

Bevisa att för varge  $a > b$  gäller följande  $|\cos a - \cos b| \leq |a - b|$

$f(x) = \cos x$  är kont och deriverbar i  $\mathbb{R} \wedge [a, b]$

$$\text{MVS } \Rightarrow \exists c \in (a, b) : f'(c) = \frac{f(b) - f(a)}{b - a} = \frac{\cos(b) - \cos(a)}{b - a} = -\sin(c)$$

$$\Rightarrow \left| \frac{\cos(b) - \cos(a)}{b - a} \right| = |- \sin(c)| \Rightarrow \frac{|\cos(b) - \cos(a)|}{|b - a|} = |- \sin(c)| \leq 1$$

$$\frac{|\cos(b) - \cos(a)|}{|b - a|} \leq 1 \Rightarrow |\cos(b) - \cos(a)| \leq |a - b|$$

### 4.3.4 Rolle

Anta att  $f$  är kontinuerlig på  $[a, b]$  och deriverbar på  $(a, b)$ . Om  $f(a) = f(b) \Rightarrow \exists c \in (a, b) : f'(c) = 0$

### 4.3.5 växande funktioner

- växande:  $x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2)$
- stänkt växande:  $x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$
- avtagande:  $x_1 < x_2 \Rightarrow f(x_1) \geq f(x_2)$
- stänkt avtagande:  $x_1 < x_2 \Rightarrow f(x_1) > f(x_2)$

### 4.3.6 Högreordnings derivator

$$\begin{aligned} \text{andragrads derivator: } (f')' &= f'' = \frac{d^2 f}{dx^2} \\ \text{tredjegrads derivator: } (f'')' &= f''' = \frac{d^3 f}{dx^3} \\ \text{ntrigrads derivator: } f^{(n)} &= f^n = \frac{d^n f}{dx^n} \end{aligned}$$

### 4.3.7 Impericit derivering

deriverar båda sidorna om modifierat

#### Example: Impericit derivering

$$\begin{aligned} &\text{Bestäm tangenten till } x^3 + y^3 = 6xy \text{ i } (3,3) \\ &(3^3 + 3^3 = 6 \cdot 3 \cdot 3) \text{-sant} \\ &y - y_0 = f'(x_0)(x - x_0) \Rightarrow y - 3 = f'(3)(x - 3) \\ &\text{Implicit derivering: } 3x^2 + 3y^2 y' = 6y + 6xy' \\ &\Rightarrow 3x^2 y' - 6xy' = 6y - 6x^2 \Rightarrow y'(3x^2 - 6x) = 6y - 3x^2 \Rightarrow y' = \frac{6y - 3x^2}{3x^2 - 6x} \\ &\Rightarrow y'(3) = \frac{63 - 3 \cdot 3^2}{3 \cdot 3^2 - 6 \cdot 3} = -1 \\ &\Rightarrow y - 3 = -(x - 3) \Rightarrow x + y = 6 \vee y = -x + 6 \end{aligned}$$

### 4.3.8 invers funktioner

$$\begin{aligned} f : A \rightarrow B \wedge \text{Bijektiv} &\Rightarrow f^{-1}(x) \text{ Finns, där} \\ (1) \quad x &= f^{-1}(y) \Leftrightarrow y = f(x) \\ (2) \quad D_{f^{-1}} &= V_f \Leftrightarrow D_f = V_{f^{-1}} \\ (3) \quad x &= f^{-1}(f(x)), x \in D_f = V_{f^{-1}} \\ (3) \quad y &= f^{-1}(f(y)), y \in D_f = V_{f^{-1}} \end{aligned}$$

### 4.3.9 exponential och logaritm

$$\begin{aligned}\frac{a^{-3}}{b} &= \frac{b^3}{a} \\ \sqrt{a} &= a^{\frac{1}{2}} \\ a^{x+y} &= a^x a^y \\ (a^x)^y &= a^x a^y \\ a^{-x} &= \frac{1}{a^x} \\ a^{\frac{m}{n}} &= \sqrt[n]{a^m}\end{aligned}$$

(1):  $b = a^x \Leftrightarrow \log_a(b) = x$  för:  $a > 0, b > 0, a \neq 1$

(2):  $\log_a\left(\frac{b}{c}\right) = \log_a(b) - \log_a(c)$

(3):  $\log_a(b * c) = \log_a(b) + \log_a(c)$

(4):  $\log_a(b^d) = d \log_a(b)$

(5):  $\log_a(b) = \frac{\log_f(b)}{\log_f(a)}$

(6):  $\log_a(a) = 1$

(7):  $\log_a(1) = 0$

(8):  $a^{\log_a(x)} = x$

(9):  $\log_{a^c}(b) = \frac{1}{c} \log_a(b)$

### 4.3.10 odefinerad form

$$\frac{0}{0}, \infty \cdot 0, 1^\infty, \infty^0, 0^0$$

### 4.3.11 inversa trigometriska funktioner

#### arcsin

$$\cos : \left[ \frac{-\pi}{2}, \frac{\pi}{2} \right] \rightarrow [-1, 1]$$

$$\arcsin 1 = \frac{\pi}{2}$$

$$\arcsin 0 = 0$$

$\arcsin \pi$  = odefinerad

$$\sin(\arcsin(x)) = x$$

$$\arcsin(\sin(x)) = x$$

$$\begin{aligned} (\arcsin(x))' &= \frac{1}{\sin'(\arcsin(x))} = \frac{1}{\cos(\arcsin(x))} = \\ &\frac{1}{\sqrt{\cos^2(\arcsin(x))}} = \frac{1}{\sqrt{1 - \sin^2(\arcsin(x))}} = \frac{1}{\sqrt{1 - x^2}} \end{aligned}$$

#### arccos

$$\cos : [0, \pi] \rightarrow [-1, 1]$$

$$\arccos 1 = 0$$

$$\arccos 0 = \frac{\pi}{2}$$

$\arccos \pi$  = odefinerad

$$\cos(\arccos(x)) = x$$

$$\arccos(\cos(x)) = x$$

$$(\arccos(x))' = \frac{1}{\sqrt{1 - x^2}}$$

#### arctan

$$\cos : \left[ \frac{-\pi}{2}, \frac{\pi}{2} \right] \rightarrow \mathbb{R}$$

$$\tan(\arctan(x)) = x$$

$$\arctan(\tan(x)) = x$$

$$(\arctan(x))' = \frac{1}{1 + x^2}$$

**exempel**

$$\begin{aligned}\tan(\arccos x) &= \frac{\sin(\arccos x)}{\cos \arccos x} = \frac{\sqrt{1-x^2}}{x} \\ \cos(\arctan x) &= \cos \frac{\arcsin x}{\arccos x} = \frac{1}{1+x^2} \\ \sin(\arccos x) &= \sqrt{1-x^2} \\ \cos(\arcsin x) &= \sqrt{1-x^2}\end{aligned}$$

## 4.4 Grafritning

Ta reda på extrem punkterna och rita utifrån det

**Extremvärden**

- global minimum punkt:  $x_0, \forall x : f(x) \geq f(x_0)$
- lokal minimum punkt:  $x_0, \forall x \text{ nära } x_0 : f(x) \geq f(x_0)$
- global maximum punkt:  $x_0, \forall x : f(x) \leq f(x_0)$
- lokal maximum punkt:  $x_0, \forall x \text{ nära } x_0 : f(x) \leq f(x_0)$
- Kritisk punkt:  $f'(x) = 0$

**Komplexity**

- konvex: Tangent liger altid under funktionen  $y'' > 0$
- konkav: Tangent liger altid över funktionen  $y'' < 0$
- Inflektionspunkt: då funktionen byter från konvex till konkav eller tvärtom

**Exempel: Daten som behövs beräknas vid grafritning**

Rita funktionen  $y = (x^2 - 1)^3$

(1) Kollar om funktionen är Konternuelig:

Eftersom funkrionen består av polynom är den konternuerlig

(2) Extrem punkter:

(I) Derivatan: funktionen är deriverbar  $y' = 3(x^2 - 1)^2 \cdot 2x = 0$

$$\Rightarrow x = 0$$

(II) Singulär punkter:

eftersom funktionen är deriverbar i alla punkter i definitions mängden

Så finns det inga singulär punkter

(III) End värdet: Eftersom funktionen är ej definerad i ett intervall så finns det inga

(3) Komplexitet:

Andra derivatan avgör om funktionen är knvex eller konkav

$$y'' = (6x(x^2 - 1)^2)' = (6x(x^4 - 2x^2 + 1))' = (6x^5 - 12x^3 + 6x)' = 30x^4 - 36x^2 + 6 = 0$$

$$\Rightarrow t^2 - \frac{6}{5}t + \frac{1}{5} = 0 \Rightarrow t = \frac{6}{10} \pm \sqrt{\frac{36}{100} - \frac{20}{100}} = \frac{6}{10} \pm \frac{4}{10}$$

$$t = 1 \vee t = \frac{1}{5} \Rightarrow x = \pm 1 \vee x = \pm \frac{1}{\sqrt{5}}$$

(4) Asymptoter:

(I) lodräta asymptoter:

$$\lim_{x \rightarrow +\infty} y = +\infty$$

$$\lim_{x \rightarrow -\infty} y = +\infty$$

(II) vågräta asymptoter:

Eftersom funktionen inte är en kvot finns det inga sådana asymptoter

$f'$	$x < -1$	$x = -1$	$-1 < x < -\frac{1}{\sqrt{5}}$	$x = -\frac{1}{\sqrt{5}}$	$\dots$
$f''$	—	—	—	—	...
$f$	avtagande kokav	0 avtagande inflektions punkt	+ avtagande konvex	0 avtagande inflektions punkt	...

## 4.5 Optemering

Sats: om funktionen  $f(x)$  är konternuerlig på det slutna intervallet  $[a, b]$   
 så antar det sitt största värde och minsta värde där  $\exists x_1, x_2 \in [a, b]$   
 så att  $f(x_1) \leq f(x) \leq f(x_2)$

### Exempel: Max/Min värde

Låt  $f(x) : [-1, 2] \rightarrow \mathbb{R}$ ,  $f(x) = x^3 - 3|x|$

- (1) Konternuelig:  
 $f$  är konternuerlig i intervallet

- (2) Extrem punkter:

(I) Derivatan: funktionen är deriverbar i intervallet förutom då  $x = 0$

låt  $f$  bestå av  $f_1(x) = x^3 - 3x, x \geq 0 \wedge f_1(x) = x^3 + 3x, x < 0$

$$\Rightarrow f'_1(x) = 3x^2 - 3, x \geq 0 \wedge f'_2(x) = 3x^2 + 3, x < 0$$

$$f'_1(x) = 0 \Rightarrow x = 1 \in [1, 2], f(1) = -2$$

$$f'_2(x) = 0 \Rightarrow x = \text{Odefinerad}$$

(II) Singulär punkter:

$f$  är inte deriverbar då  $x = 0, f(0) = 0$

(III) End punkterna: Eftersom funktionen är ej definerad i ett intervall så finns det inga

$$f(-1) = -4$$

$$f(2) = 2$$

- (3) Största och minsta värdet:

$$f(-1) < f(1) < f(0) < f(2)$$

Svar: Max är 2, min är -4

## 4.6 Talfölder och serier

**Def:** Talföljder

En talföjd är en funktion  $a : \mathbb{N} \rightarrow \mathbb{N}$

Vi skriver  $a_n$  istället för  $a(n)$ ,  $a_2 = a(2)$

Vi säger att  $a_n \rightarrow a \in \mathbb{N}$  så är den **konvergent**

Vi säger att  $a_n \rightarrow \infty$  så är den **divergent** eller ej existerande

Konvergent kan vara begränsad uppåt eller nedåt

Talföljed kan vara växande eller avtagande

$$a_n \rightarrow a \vee b_n \rightarrow b \text{ Omm } a \text{ och } b \text{ existerar } a_n + b_n \rightarrow a + b$$

**Exempel:** Derivatan av serier

$$\text{Ange } f'(2), \quad f(x) = \sum_{n=1}^{\infty} \frac{(x-2)^n}{n^2 2^{2n}}$$

$$\text{Anger den första termen } (\frac{x-2}{4})' = \frac{1}{4}$$

$$f'(x) = \frac{1}{4} + \sum_{n=2}^{\infty} \frac{n(x-2)^{n-1}}{n^2 2^{2n}} \text{ Inre och yttre derivatan}$$

$$f'(2) = \frac{1}{4} + \sum_{n=2}^{\infty} \frac{n(2-2)^{n-1}}{n^2 2^{2n}} = \frac{1}{4} + 0 = \frac{1}{4}$$

**Sats:**

Om  $(a_n)^\infty$  är Konvergent  $\Rightarrow (a_n)$ ,  $n = 1$  är begränsad

**Sats:**

Låt  $a > 0$

(I)  $a^n \rightarrow 0$  om  $a < 1$

(II)  $a^n \rightarrow +\infty$  om  $a > 1$

$$(a_n)_{n=1}^{\infty} = \sum_{k=0}^n a_n = a_1 + a_2 + a_3 + \dots$$

**Sats:** Geometrisk serie

$$s_n = a + ak + ak^2 + \dots + ak^{n-1} = \frac{a(k^n - 1)}{k - 1}$$

$$\sum_{n=0}^{\infty} r^n \Rightarrow$$

Konvergent om  $|r| < 1$

Divergent om  $|r| > 1$

**Sats:**

$$\sum_{n=0}^{\infty} a_n \text{ konvergent} \Rightarrow a_n \rightarrow 0$$

**Sats: p-serie**

$$\sum_{n=0}^{\infty} \frac{1}{n^p} \Rightarrow$$

konvergent om  $\Rightarrow p > 1$   
divergent om  $\Rightarrow p \leq 1$

**Sats:**

Antag att  $0 \leq a_n \leq b_n$  för varje  $n \in \mathbb{N}$

$$\sum_{n=1}^{\infty} a_n \text{ konvergent} \Rightarrow a_n \rightarrow 0$$

(I)  $\sum_{n=1}^{\infty} b_n$  konvergerar  $\Rightarrow \sum_{n=1}^{\infty} a_n$  konvergerar

(II)  $\sum_{n=1}^{\infty} a_n$  divergerar  $\Rightarrow \sum_{n=1}^{\infty} b_n$  divergerar

**Sats:**

Antag att  $a_n > 0, b_n > 0$  för varje  $n \in \mathbb{N}$  och antag att  $\frac{a_n}{b_n} \rightarrow L \neq 0 \wedge L < \pm\infty$

Då gäller att serien  $\sum_{n=1}^{\infty} a_n$  och  $\sum_{n=1}^{\infty} b_n$

är båda divergenta

**Sats: kvotkriterium**

$$\text{Låt } a_n > 0 \wedge \lim_{n \rightarrow \infty} \frac{a_{n+1}}{a_n} \rightarrow L \Rightarrow$$

(I)  $0 \leq L \leq 1 \Rightarrow$  Konvergerar  $\sum_{n=1}^{\infty} a_n (a_n \rightarrow 0)$

(II)  $L > 1 \Rightarrow$   $\sum_{n=1}^{\infty} a_n$  divergerar ( $a_n \rightarrow +\infty$ )

**Exempel:** Måste kunna

$$\sum_{n=1}^{\infty} \frac{5^n n!}{n^n}$$

Låt  $a_n = \frac{5^n n!}{n^n} \Rightarrow \frac{a_{n+1}}{a_n} = \frac{5^{n+1}(n+1)!/n^n}{5^n n!/n^n} = 5 \frac{(n+1)n^n}{(n+1)^{n+1}} = 5 \left(\frac{n}{n+1}\right)^n = 5 \frac{1}{\left(1 + 1/n\right)^n} \rightarrow \frac{5}{e} > 1 \Rightarrow$  derigerar

**Sats:** Rotkriteriet (Ovanlig)

$$\text{Låt } a_n > 0 \wedge \lim_{n \rightarrow \infty} \sqrt[n]{a_n} \rightarrow L \Rightarrow$$

(I)  $0 \leq L \leq 1 \Rightarrow$  Konvergerar  $\sum_{n=1}^{\infty} a_n (a_n \rightarrow 0)$

(II)  $L > 1 \Rightarrow \sum_{n=1}^{\infty} a_n$  derigerar ( $a_n \rightarrow +\infty$ )

#### 4.6.1 Serier med varierande tecken

**Sats:** leibinz

Antag att  $\sum_{n=1}^{\infty} a_n$  är en alternerande serie  
och att om  $|a_n| \rightarrow 0 \wedge |a_n| \geq |a_{n+1}| \Rightarrow$  konvergent

**Def:** absolut konvergent

Endast om  $\sum_{n=1}^{\infty} |a_n|$  konvagerar  $\Rightarrow$  då är den absolutkonvergent  
om seriern är absolutkonvergent så är den också konvergent behöver inte vara tvärtom

#### 4.6.2 Potensserier

**Def:** Potensserier

En potensserier kring  $x_0$  är en serie på formen  $\sum_{n=1}^{\infty} a_n (x - x_0)^n$ ,  $a_n \in \mathbb{R}$  (koefficienten)  
 $x$  är en variabel

**Def: Konvergens radie**

$$T(x) = \sum_{n=1}^{\infty} \frac{f(x_0)^{(n)}}{n!} (x - x_0)^n$$

vid exempelvis grad 2 så blir det andra derivatan man räknar ut

$$R = L^{-1} = \frac{1}{L} \text{ (L är där serien konvergerar, R är konvergens radie)} \sum_{n=1}^{\infty} a_n (x - x_0)^n, a_n \in \mathbb{R} \text{ (koefficienten)}$$

$x$  är en variabel

Om konvergens radien är noll så finns det inte några intervar för konvergens eller divergens

**Sats: Potensserier**

En potensserie kring  $x_0$  är en serie på formen  $\sum_{n=1}^{\infty} a_n (x - x_0)^n$   $a_n \in \mathbb{R}$   $a_n$  koficienten  $x$  är en variabel

När det står ex  $(3x + 2)^n$  så måste det stå med i  $a_n, 3^n$

**Exempel: Potensserier**

$\sum_{n=1}^{\infty} \frac{(x - 7)^n}{n(n+1)}$  är en potensserie där  $x_0 = 7$ ,  $a_n = \frac{1}{n(n+1)}$

$$\left| \frac{a_{n+1}}{a_n} \right| = \left| \frac{1/((n+1)(n+2))}{1/(n(n+1))} \right| = \left| \frac{n}{n+2} \right| = \left| \frac{n}{n(1+2/n)} \right|$$

$$\Rightarrow \left| \frac{1}{1} \right| = 1, \text{ då } n \rightarrow \infty$$

$$\Rightarrow R = \frac{1}{1} = 1 \Rightarrow \text{vi får följande intervall av potenserien}$$

Absolutkonverget för  $|x - 7| < 1 \Leftrightarrow 6 < x < 8$

Divergent för  $|x - 7| > 1 \Leftrightarrow 8 < x \vee x < 6$

Testar edge fallen  $x = 6, x = 8$

$$\sum_{n=1}^{\infty} \frac{(8 - 7)^n}{n(n+1)} \Rightarrow \left| \sum_{n=1}^{\infty} \frac{(1)^n}{n(n+1)} \right|, \left| \frac{(1)^n}{n(n+1)} \right| \rightarrow 0 \Rightarrow \text{absolutkonvergerar}$$

$$\sum_{n=1}^{\infty} \frac{(6 - 7)^n}{n(n+1)} \Rightarrow \left| \sum_{n=1}^{\infty} \frac{(-1)^n}{n(n+1)} \right|, \left| \frac{(-1)^n}{n(n+1)} \right| \rightarrow 0 \Rightarrow \text{absolutkonvergerar}$$

**Svar:**

Absolutkonverget för  $6 \leq x \leq 8$

Divergent för  $8 < x \vee x < 6$

### 4.6.3 Taylor serier

**Def:** Talorpolynom

Talorpolynom av grad  $n \in \mathbb{N}$ , kring  $x = x_0$

$$f(x) = P_n(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k, \quad 0! = 1, f^{(0)} = f$$

När  $x \approx x_0 \Rightarrow f(x) \approx p_n(x)$

**Sats:** Talor sats

Antag att  $f \in C^{n+1}$ .

$f(x) = p_n(x) + R_{n+1}(x)$  Där felet är  $R_{n+1}(x)$

$$R_{n+1}(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x - x_0)^{n+1}, \quad x \vee x_0 \leq c \leq x_0 \vee x$$

**exempel:** felet

Upskata felet hos  $f(x) = \sin x$ , med grad 5

Sista termen är  $\frac{x^5}{5!} \Rightarrow$  Felet  $\frac{f^{(6)}(c)}{6!} x^6, (0 < c < 1)$

$$\left| \frac{f^{(6)}(c)}{6!} x^6 \right| \leq \frac{|f^{(6)}(c)|}{6!} = \frac{|\sin c|}{6!} \leq \frac{1}{6!} = \frac{1}{720}$$

**Exempel:** grad n vid specificerad punkt

Lös: Hitta Taylorpolynomet, ordning  $2n - 1$ , för  $\sin 2x$  vid  $x = \frac{\pi}{2}$

$$f(x) = \sin(2x)$$

$$f\left(\frac{\pi}{2}\right) = 0$$

$$f'(x) = -2 \cos(2x)$$

$$f\left(\frac{\pi}{2}\right) = -2$$

$$f''(x) = -2^2 \sin(2x)$$

$$f\left(\frac{\pi}{2}\right) = 0$$

$$f^{(3)}(x) = -2 \cos(2x)$$

$$f\left(\frac{\pi}{2}\right) = 2^3$$

$$f^{(4)}(x) = -2^2 \sin(2x)$$

$$f\left(\frac{\pi}{2}\right) = 0$$

$$f^{(5)}(x) = -2^4 f'(x)$$

$$f\left(\frac{\pi}{2}\right) = -2^5$$

$$p_{2n-1}(x) = -2\left(x - \frac{\pi}{2}\right) + \frac{2^3}{3!}\left(x - \frac{\pi}{2}\right)^3 - \frac{2^5}{5!}\left(x - \frac{\pi}{2}\right)^5 + \dots + \frac{(-1)^n \cdot 2^{2n-1}}{(2n-1)!}\left(x - \frac{\pi}{2}\right)^{(2n-1)}$$

**Def: maclaurin polynomial**

Taylorpolynom av grad  $n \in \mathbb{N}$ , då  $x_0 = 0$

Big-O notation: Vi säger att  $g(x) = O(f(x))$  när  $x \approx x_0$ , värsta fall

Man kan också säga istället för Big-O notation en rest  $R_{n+1}(x) = x^{n+1}H(x)$  där  $H(x)$  begränsad i intervallet

$$e^x = \sum_{k=0}^n \frac{x^k}{k!} + O(x^{n+1})$$

$$\sin x = \sum_{k=0}^n \frac{(-1)^{k-1}}{(2k-1)!} x^{2k-1} + O(x^{2n})$$

$$\cos x = \sum_{k=0}^n \frac{(-1)^k}{(2k)!} x^{2k} + O(x^{2n+1})$$

$$\ln(1+x) = \sum_{k=0}^n \frac{x^k}{k} + O(x^{n+1})$$

## 4.7 Integraler

**Regler no.1**

$$\int_a^a f(x)dx = 0$$

**Regler no.2**

$$\int_a^b f(x)dx = - \int_b^a f(x)dx$$

**Regler no.3 (Linjearitet)**

$$A, B \in \mathbb{R}$$

$$\int_a^b (A \cdot f(x)dx + B \cdot g(x)) = A(\int_a^b f(x)dx) + B(\int_a^b g(x)dx)$$

Fungerar på subtraction men INTE multiplicatione eller divition!

**Regler no.4**

$$c \in [a, b]$$

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx$$

**Regler no.5**

$$f(-x) = -f(x) \text{ är udda}$$

$$\int_{-a}^a f(x)dx = 0$$

**Regler no.6**

$$f(-x) = f(x) \text{ är jämn}$$

$$\int_{-a}^a f(x)dx = 2 \int_0^a f(x)dx$$

**Regler no.7**

$$\left| \int_a^b f(x)dx \right| \leq \int_a^b |f(x)|dx$$

**Regler no.8**

$$f(x) \leq g(x) \Rightarrow \int_a^b f(x)dx \leq \int_a^b g(x)dx$$

**Sats:** Medelvärdessatsen för integraler

$$f : [a, b] \rightarrow \mathbb{R} \text{ konternuelig } \wedge \exists c \in (a, b) \Rightarrow \int_a^b f(x)dx = f(c)(b - a)$$

**Sats:** Fanalysens huvudsats, Fundemetal theorem of calculus

Satsen är den vi använder för att lösa integraler utan geometrisk tolkning

$$f : [a, b] \rightarrow \mathbb{R} \text{ konternuelig } \Rightarrow F(x) = \int_a^x f(t)dt, a \leq x \leq b$$

**Sats:**

$$\begin{aligned} f : [a, b] \rightarrow \mathbb{R} \text{ konternuelig } &\Rightarrow F \text{ är e primitiv funktion till } f(F'(x) = f(x)) \\ \int_a^b f(x)dx &= F(b) - F(a) \\ \text{Vi skriver } \int_a^b f(x)dx &= [F(x)]_a^b = F(b) - F(a) \end{aligned}$$

**Exempel:** arean mellan två grafer

$$\begin{aligned} \int_a^b (f(x) - g(x))dx &\text{ där f är översta funnktionen och g är understa} \\ \int_0^1 (x - x^2)dx &= [x^2/2 - x^3/3]_0^1 = 1/2 - 1/3 = 1/6 \end{aligned}$$

**Primitiva funktioner (Obestämda integralen)**

$\int f dx$	$F$
$\int 1 dx$	$x + c$
$\int n^n dx$	$x^{n+1}/(n+1) + c$
$\int 1/xdx$	$\ln x  + c$
$\int \sin x dx$	$-\cos x + c$
$\int \cos x dx$	$\sin x + c$
$\int e^x dx$	$e^x + c$
$\int 1/\sqrt{1-x^2} dx$	$\arcsin x + c$
$\int 1/(x^2+1) dx$	$\arctan x + c$
$\int 1/\cos^2 x dx$	$\tan x + c$

**Exempel:**

$$\begin{aligned} \int_a^b e^x dx &= [e^x]_a^b = e^b - e^a \\ \int_e^x dx &= e^x + c, c \in \mathbb{R} \end{aligned}$$

**Def:** medelvärdet

$$Avgf = \frac{1}{b-a} \int_a^b f dx$$

### 4.7.1 variabelsubstitution

$$\int f'(g(x))g'(x)dx = f(g(x)) + c$$

$$\int_e^x dx = e^x + c, \quad c \in \mathbb{R}$$

**Exempel:**

$$\int e^{x^2} 2x dx$$

Låt  $u = x^2 \Rightarrow du = 2x dx \Rightarrow \int e^{x^2} 2x dx = \int e^u du =$

$$= e^u + c = e^{x^2} + c, \quad c \in \mathbb{R}$$

### Integraler som följer mönster

Integralles som inehåller  $\sqrt{x^2 + a^2}, \quad a > 0$

$$x = a \tan u \Rightarrow u = \arctan x, \quad -\pi/2 < u < \pi/2$$

$$\Rightarrow \sqrt{x^2 + a^2} = \sqrt{a^2 + a^2 \tan^2 u} = \sqrt{a^2(1 + \tan^2 u)} = a \sqrt{\frac{\sin^2 u + \cos^2 u}{\cos^2 u}} = a \frac{1}{\cos u}$$

Exempel:  $\int \frac{1}{(1 + 25x^2)^{3/2}} dx$

$$\int \frac{dx}{\sqrt{1 + 25x^2}^3} = \int \frac{dx}{\sqrt{25(1/25 + x^2)}^3} = \frac{1}{5^3} \int \frac{dx}{\sqrt{1/205x^2}^3}$$

Låt  $x = 1/5 \tan u \Leftrightarrow u = \arctan 5x$

$$\Rightarrow dx = \frac{1}{5 \cos^2 u} du$$

$$\Rightarrow \frac{1}{5^3} \int \frac{dx}{\sqrt{1/205x^2}^3} = \frac{1}{5^3} \int \frac{\cos^3 u}{1/5^3} \frac{1}{5 \cos^2 u} du = \frac{1}{5} \int \cos u du =$$

$$= \frac{1}{5} \sin u + c = \frac{1}{5} \sin \arctan 5x + c \quad \text{Kan nu förenkla med trigonometriska regler}$$

och få:  $\frac{x}{\sqrt{1 + 25x^2}} + c$

Integralles som inehåller  $\sqrt{x^2 - a^2}, \quad a > 0$

$$x = \frac{a}{\cos u} \Rightarrow u = \arccos \frac{a}{x}$$

$$\Rightarrow \sqrt{x^2 - a^2} = \sqrt{\frac{a^2}{\cos^2 u} - a^2} = \sqrt{\frac{a^2 - a^2 \cos^2 u}{\cos^2 u}} = \sqrt{\frac{a^2 \sin^2 u - a^2 \cos^2 u}{a^2 \cos^2 u}} = a |\tan u|$$

Integralles som inehåller  $\sqrt{ax + b}$

$$ax + b = u^2$$

Exempel:  $\int \frac{dx}{2 + \sqrt{x}}$

Låt:  $u^2 = x \Rightarrow 2udu = dx$

$$\Rightarrow \int \frac{2udu}{2 + u} = 2 \int \frac{u+2-2}{2+u} du = 2\left(\int du - 2 \int \frac{du}{2+u}\right) = 2u - 4(\ln 2 + u) + c$$

### 4.7.2 Integration av rationella funktioner

Om det står beräkna generaliseringen så ska man beräkna med detta

Rationella funktioner:  $R(x) = \frac{P(x)}{Q(x)}$ ,  $P, Q$  är polynom

Algoritm

1. Polynom division om grad av täljare är större än grad av nämnare
2. Faktorisera nämnaren i reala faktorer
3. Partiella bråk och sedan integrera dem, (dela upp i mindre lösbara integraler)

#### Partiella bråk

$x - a$	$\frac{A}{x-a}$
$(x - a)^k$	$\frac{A_1}{x-a} + \frac{A_2}{x-a^2} + \dots + \frac{A_k}{x-a^k}$
$x^2 + bx + c, (b^2 - 4c < 0)$	$\frac{Ax+B}{x^2+bx+c}$
$x^2 + bx + c, (b^2 - 4c < 0)$	$\frac{A_1x+B_1}{x^2+bx+c} + \frac{A_2x+B_2}{(x^2+bx+c)^2} + \dots + \frac{A_kx+B_k}{(x^2+bx+c)^k}$

$$\int \frac{1}{ax+b} dx = \frac{1}{a} \ln |ax+b| + c$$

$$\int \frac{x}{x^2+a^2} dx = \frac{1}{2} \ln |x^2+a^2| + c$$

$$\int \frac{x}{x^2-a^2} dx = \frac{1}{2} \ln |x^2-a^2| + c$$

$$\int \frac{dx}{x^2+a^2} = \frac{1}{a} \tan^{-1} \frac{x}{a} + c$$

$$\int \frac{dx}{x^2-a^2} = \frac{1}{2a} \ln \frac{x-a}{x+a} + c$$

$$\int \frac{x^3 + 2}{x^2 - 5x + 4} dx$$

1. Polynomdivision

$$P(x) = x^3 + 2, Q(x) = x^2 - 5x + 4$$

Med polynom division så får vi kvot  $x + 5$  och rest  $21x - 18$

$$x^3 + 2 = (x + 5)(x^2 - 5x + 4) + 21x - 18$$

2. Faktorisera nämnaren i reala faktorer

$$\Rightarrow \int \frac{x^3 + 2}{x^2 - 5x + 4} dx = \int \frac{(x + 5)(x^2 - 5x + 4) + 21x - 18}{x^2 - 5x + 4} dx = \int (x + 5)dx + \int \frac{21x - 18}{x^2 - 5x + 4}$$

3. Integrerar partiella bråk

$$\frac{21x - 18}{x^2 - 5x + 4} = \frac{21x - 18}{(x - 4)(x - 1)} = \frac{A}{x - 4} + \frac{B}{x - 1} \Rightarrow \frac{21x - 18}{(x - 4)(x - 1)} = \frac{Ax - A + Bx - 4B}{(x - 4)(x - 1)}$$

$$\Rightarrow A + B = 21 \wedge -A - 4B = -18 \Rightarrow A = 22 \wedge B = -1$$

$$\int (x + 5)dx + \int \frac{22}{x - 4} dx - \int \frac{1}{x - 1} dx$$

Svar:  $x^2/2 + 5x + 22 \ln|x - 4| - \ln|x - 1| + c, c \in \mathbb{R}$

### 4.7.3 Partiell integration

Om det är integraler med två funktioner så hjälper oftast partiell integration

#### Formel

$$\int_a^b f' g dx = [fg]_a^b - \int_a^b fg' dx$$

$$\int f' g dx = fg - \int fg' dx$$

#### Bevis

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x) \Rightarrow$$

$$\int_a^b (fg)' dx = \int_a^b f' g dx + \int_a^b fg' dx \Rightarrow \int_a^b f' g dx = \int_a^b (fg)' dx - \int_a^b fg' dx$$

$$\int_a^b f' g dx = [fg]_a^b - \int_a^b fg' dx$$

#### Exempel: polynom \* trigonomotrik

$$\int_0^{\pi/2} x \sin x dx$$

$$\int_0^{\pi/2} (-\cos x)' x dx = [-x \cos x]_0^{\pi/2} + \int_0^{\pi/2} \cos x (x)' dx =$$

$$[-x \cos x]_0^{\pi/2} + [\sin x]_0^{\pi/2} = 1$$

#### Exempel: polynom \* exponensiel

$$\int e^x x^2 dx$$

$$\int (e^x)' x^2 dx = [e^x x^2] - \int e^x 2x dx = e^x x^2 - 2 \int (e^x)' x dx = e^x x^2 - 2e^x x + 2 \int (e^x)' dx = e^x x^2 - 2e^x x + 2e^x + c$$

#### Exempel: exponensiel \* trigonomotrik

$$\int e^x \cos x dx$$

Låt  $I = \int e^x \cos x dx = \int (e^x)' \cos x = e^x \cos x - \int e^x \sin x dx = e^x \cos x + e^x \sin x - \int e^x \cos x dx \Rightarrow$

$$I = e^x \cos x + e^x \sin x - I \Rightarrow I = \frac{e^x \cos x + e^x \sin x}{2} + c$$

#### Exempel: bonus ln

$$\int_1^2 \ln x dx$$

$$\int_1^2 x' \ln x dx = \dots = 2 \ln 2 - 1$$

#### 4.7.4 Generaliseringar av integraler

**Def:**

Antag att  $f$  är kontinuerlig i  $a, b$  och att  $\lim_{x \rightarrow a^+} f(x) = \infty$

Vi definierar den generaliserade integralen  $\int_a^b f(x)dx = \lim_{\varepsilon \rightarrow 0^+} \int_{a+\varepsilon}^b f(x)dx, \dots, konstingst$

Om gränsvärdet existerar och är ändlig säger vi att integralen är konvergent, annars är den divergent

#### Beteckning

$\varepsilon$  Andvänds för små tal  $\lim_{\varepsilon \rightarrow 0^+}$

$M$  Andvänds för stora tal  $\lim_{M \rightarrow +\infty}$

$c$  Andvänds för konstanter  $+c$

#### Sats:

$$\int_a^{+\infty} \frac{dp}{x^p}, \quad a > 0$$

$p > 1 \Rightarrow$  Konvergerar

$p \leq 1 \Rightarrow$  Divergerar

#### Sats: Jämförelsesatsen

Anta att  $f$  och  $g$  är konternuerliga och  $0 \leq f(x) \leq g(x), (a \in [-\infty, +\infty), b \in (-\infty, \infty])$

(I) om integralen är konvergent  $\int_a^b g(x)dx \Rightarrow \int_a^b f(x)dx$  är också konvergent

(II) om integralen är divergent  $\int_a^b f(x)dx \Rightarrow \int_a^b g(x)dx$  är också divergent

#### Sats:

Om  $f(x)$  är positiv konternuerlig och avtagande i intervallet  $x \geq N$ ,

så är serien  $\sum_{n=N}^{\infty} f(n)$  konvergent

precis när  $\int_N^{\infty} f(x)dx$  är konvergent

**Exempel:**

Betämm om serien konvergerar eller divergerar  $\sum_{n=10}^{\infty} \frac{1}{n \ln n (\ln(\ln n))^2}$

$$f(x) = \frac{1}{x \ln x (\ln(\ln x))^2}$$

är positiv, konternuerlig och avtagande då nämnaren är stängt positivtökande för  $x \geq 10$

$$\int_{10}^{+\infty} f(x) dx = \lim_{M \rightarrow +\infty} \int_{10}^M f(x) dx, (u = \ln \ln x \Rightarrow du = 1/\ln x \cdot 1/x dx) \Rightarrow \lim_{M \rightarrow +\infty} \int_{\ln \ln 10}^{\ln \ln M} \frac{1}{u^2}$$

$$\lim_{M \rightarrow +\infty} [-1/u]_{\ln \ln 10}^{\ln \ln M} = \lim_{M \rightarrow +\infty} (-1/\ln \ln M + 1/\ln \ln 10) = 1/\ln \ln M$$

$$\sum_{n=10}^{\infty} \frac{1}{n \ln n (\ln(\ln n))^2} \text{ konvergerar}$$

**Sats:**

Antag att  $a_n > 0, b_n > 0$  för varje  $n \in \mathbb{N}$  och antag att  $\frac{a_n}{b_n} \rightarrow L \neq 0 \wedge L < \pm\infty$

Då gäller att serien  $\int_{n=1}^{\infty} a_n$  och  $\int_{n=1}^{\infty} b_n$  är båda divergenta

### 4.7.5 Volymberäkningar

$$V = \int_a^b A(x)dx \text{ Rotationsvolymer runt x-axeln} \Leftrightarrow \pi \int_a^b ((g(x))^2 - (f(x))^2) dx \text{ g är övre f är undre}$$

$$V = 2\pi \int_a^b xf(x)dx \text{ Rotationsvolymer runt y-axeln} \Leftrightarrow 2\pi \int_a^b x(g(x) - f(x))dx \text{ g är övre f är undre}$$

**exempel**

Beräkna volymen av den kropp som uppstår när området som begränsas av kurvan  $y = 4x - x^2 - 3$  och x-axeln roteras kring y-axeln

$$\begin{aligned} y = 0 &\Leftrightarrow 4x - 3 - x^2 = 0 \Leftrightarrow x = 1 \vee x = 3 \\ &\wedge y > 0, x \in [1, 3] \\ \Rightarrow V &= 2\pi \int_1^3 x(4x - 3 - x^2)dx = 2\pi \int_1^3 (4x^2 - 3x - x^3)dx \\ &= 2\pi [4/3x^3 - 3/2x^2 - x^4/4]_1^3 = 16\pi/3 \end{aligned}$$

### Kurvängd

$$\begin{aligned} L &= \|\vec{x}_0 - \vec{x}_1\| = \sqrt{(a-c)^2 + (b-d)^2} \\ \Rightarrow L &= \int_a^b \sqrt{1 + (f'(x))^2} dx \end{aligned}$$

**exempel**

Find the lenght of curve  $y = x^3/12 + 1/x$  from  $x = 1$  to  $x = 4$

$$\begin{aligned} f(x) &= x^3/12 + 1/x \\ L &= \int_1^4 \sqrt{1 + (f'(x))^2} dx = \int_1^4 \sqrt{1 + (x^2/4 - 1/x^2)^2} dx \\ &= \int_1^4 \sqrt{1 + x^4/16 - 1/2 + 1/x^4} dx = \int_1^4 \sqrt{x^4/16 + 1/x^4 + 1/2} dx \\ &= \int_1^4 \sqrt{(x^2/4 + 1/x^2)^2} dx = \int_1^4 (x^2/4 + 1/x^2) dx \\ &= [x^2/4 + 1/x^2]_1^4 = 6 \end{aligned}$$

### Rotationsarea

$$A = \int_a^b 2\pi|f(x)|\sqrt{1 + (f'(x))^2}dx$$

#### exempel: Gabriel's Horn/Torricell's trumpet

Bestäm volymen och arean av den kropp som uppstår när området som begränsas av kurvan  $y = 1/x, x \geq 1$  och x-axeln roteras kring x-axeln

$$V = \pi \lim_{M \rightarrow +\infty} \int_1^M \left(\frac{1}{x}\right)^2 dx = \pi \lim_{M \rightarrow +\infty} [-1/x]_1^M = \pi \cdot 1 = \pi$$

$$\begin{aligned} A &= \lim_{M \rightarrow +\infty} \int_1^M 2\pi|1/x|\sqrt{1 + (-1/x^2)^2}dx = 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{\sqrt{1 + 1/x^4}}{x} dx \\ &= 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{x^2 \sqrt{1 + 1/x^4}}{x^3} dx = 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{\sqrt{x^4 + 1}}{x^3} dx \\ &> 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{\sqrt{x^4}}{x^3} dx = 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{x^2}{x^3} dx \\ &= 2\pi \lim_{M \rightarrow +\infty} \int_1^M \frac{1}{x} dx \text{ Diveigerar enligt p-satsen till } \infty \end{aligned}$$

Eftersom integralen har en undre begränsning som divergerar även divergerar också integralen till  $+\infty$

## 4.8 Differential ekvationer

Tangent plan (Slope field): Andvänds för att se hur kurvan ser ut utefrån olika start värden.

grad: högsta graden på derivatan  $ty'''(t) - 4y'(t) + 5t^2y(t) = e^t$  har grad 3

Linjär diffirential ekvation: diff funktionerna har ingen upphöjning

$y'' - 4t^2t' + e^t y = 0$  är linjär

$t^2y''' + 5ty' - 4y^2 = 5$  är inte linjär

Homogen:  $omh(t) = 0 \Rightarrow$  ODE är homogen

Inhomogen:  $omh(t) \neq 0 \Rightarrow$  ODE är inhomogen

$y''' - \sin^2(t)y' = 5y$  är homogen

$e^{t^2}y^{(5)} - y'' + 4ty(t) = e^t + t^3$  är inhomogen

**Sats: superposition principle**

Låt  $a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y = 0$

Om  $y_1(x), y_2(x)$  är lösningar till differentials ekvanationen

$\Rightarrow Ay_1(x) + By_2(x)$ ,  $A, b \in \mathbb{R}$  är en lösning till differentials ekvanationen

### 4.8.1 superabla ekvationer

**Def:** superabla ekvationer

En differentialekvation är separabel om den kan skrivas på formeln

$$\frac{dy}{dx} = f(x)g(y)$$

**Exempel: Vilka är separabel**

(I)  $y' = x + y$  är inte separabel

(II)  $\frac{dy}{dx} = 1 + e^y$  är separabel

**Exempel: beräkning**

$$\text{Lös } y'(x) = (1 + e^{-x})(y^2 - 1)$$

$$y = \pm 1$$

$$\begin{aligned} \frac{dy}{y^2 - 1} &= (1 + e^{-1})dx \Rightarrow \int \frac{dy}{y^2 - 1} = \int (1 + e^{-1})dx (*) \\ \int \frac{1}{(y+1)(y-1)} dy &= \int \frac{((y+1) - (y-1))}{2(y+1)(y-1)} dy = \int \frac{1}{2(y-1)} dy - \int \frac{1}{2(y+1)} dy \\ (*) \Rightarrow \int \frac{1}{2(y-1)} dy - \int \frac{1}{2(y+1)} dy &= \int (1 + e^{-1})dx \\ \Rightarrow 1/2 \ln |y-1| - 1/2 \ln |y+1| &= x - e^{-x} + c \Rightarrow \ln \left| \frac{y-1}{y+1} \right| = 2x - 2e^{-x} + 2c \\ \Rightarrow e^{1/2 \ln \left| \frac{y-1}{y+1} \right|} &= e^{2x - 2e^{-x} + 2c} \\ \Rightarrow y = \frac{1 + e^{2x - 2e^{-x}} + 2c}{1 - e^{2x - 2e^{-x}} + 2c} \vee y &= \frac{1 - e^{2x - 2e^{-x}} + 2c}{1 + e^{2x - 2e^{-x}} + 2c} \vee y = \pm 1 \end{aligned}$$

### 4.8.2 Linjära differentialekvationer av ordning 1

#### Methodology

$$y' + p(x)y = q(x)$$

$\text{Om } g(x) = 0 \Rightarrow \text{homogen och därmed separable}$

$\text{Om } g(x) \equiv 0 \text{ Multiplisera med } e^{M(x)}$  För att kuna använda produktregeln så vi kan slå ihop  $y, y'$

$$M(x) = \int p(x)dx \text{ är en primitiv funktion till } p(x) \Rightarrow c = 0$$

$$\Rightarrow e^{M(x)}y' + e^{M(x)}p(x)y(x) = e^{M(x)}q(x) \text{ Antifunktionen slår ut } p(x)$$

$$\Rightarrow e^{M(x)}y' + (e^{M(x)})'y(x) = e^{M(x)}q(x) \text{ VL kan vi använda produktregeln bakvänt}$$

$$\Rightarrow (e^{M(x)}y(x))' = e^{M(x)}q(x) \text{ Integrerar båda led}$$

$$\Rightarrow \int (e^{M(x)}y(x))' dx = \int (e^{M(x)}q(x)) dx \text{ Antiderivatan slår ut derivatan}$$

$$\Rightarrow e^{M(x)}y(x) = \int (e^{M(x)}q(x)) dx \text{ Får } y \text{ ensamt}$$

$$\Rightarrow y(x) = e^{-M(x)} \int (e^{M(x)}q(x)) dx$$

#### Exempel

$$\text{Lös } (1+t^2)y' + ty = \frac{1}{\sqrt{1+t^2}}$$

$$y' + \frac{t}{(1+t^2)y} = \frac{1}{(1+t^2)\sqrt{1+t^2}} \text{ Linjär ode, ordning 1}$$

$$p(t) = \frac{t}{(1+t^2)y}, q(t) = \frac{1}{(1+t^2)\sqrt{1+t^2}}$$

$$M(t) = \int \frac{t}{(1+t^2)y} dt, \text{ Låt } u = 1+t^2 \Rightarrow dt = du/2t$$

$$\Rightarrow M(t) = \int \frac{1}{2u} du = \frac{1}{2} \ln |u| = \frac{1}{2} \ln t^2 + 1$$

$$e^{M(t)} = e^{\frac{1}{2} \ln t^2 + 1} = \sqrt{t^2 + 1}$$

$$\Rightarrow \sqrt{t^2 + 1}y' + \frac{t\sqrt{t^2 + 1}}{t^2 + 1}y = \frac{1}{1+t^2}$$

$$\Rightarrow \sqrt{t^2 + 1}y' + \frac{t}{\sqrt{t^2 + 1}}y = \frac{1}{1+t^2}$$

$$\Rightarrow (\sqrt{t^2 + 1}y)' = \frac{1}{1+t^2} \Rightarrow \sqrt{t^2 + 1}y = \int \frac{1}{1+t^2}$$

$$\Rightarrow \sqrt{t^2 + 1}y = \arctan(t) + c$$

$$\Rightarrow y = \frac{\arctan(t)}{\sqrt{t^2 + 1}} + \frac{c}{\sqrt{t^2 + 1}}, c \in \mathbb{R}$$

### 4.8.3 Linjära differentialekvationer av ordning 2

#### Methodology

Det finns 3 metoder för att lösa Linjära differentialekvationer av ordning 2

$ay'' + by' + cy = 0$  vilket är den generella formeln. Antag att lösningen är på formen

$$y = e^{rx}, \quad y' = re^{rx}, \quad y'' = r^2e^{rx} \Rightarrow ar^2e^{rx} + bre^{rx} + ce^{rx} = 0$$

$(ar^2 + br + c)e^{rx} = 0 \Rightarrow ar^2 + br + c = 0$  användar pq-formeln och får följande

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(i) Om  $b^2 - 4ac > 0 \Rightarrow r_1, r_2 \in \mathbb{R} \wedge r_1 \neq r_2 \Rightarrow y_1 = e^{r_1 x}, y_2 = e^{r_2 x}$

är lösningen till  $ay'' + by' + cy = 0 \Rightarrow y_k = c_1 e^{r_1 x} + c_2 e^{r_2 x}, c_1, c_2 \in \mathbb{R}$  superposition

(ii)  $b^2 - 4ac = 0 \Rightarrow r_1 = r_2 \in \mathbb{R}$  dubbelrot  $y_1 = e^{r_1 x}, y_2 = xe^{r_1 x}$

(iii) Om  $b^2 - 4ac < 0 \Rightarrow r_1 \neq r_2 \in \mathbb{C}$

$r_1 = k + li, r_2 = k - li$  Koplextals konjugat enlight euler formel, alla lösningar är

$$y = c_1 e^{r_1 x} + c_2 e^{r_2 x}, c_1, c_2 \in \mathbb{C}$$

$$y = c_1 e^{r_1 x} + c_2 e^{r_2 x} = c_1 e^{(k+li)x} + c_2 e^{(k-li)x} = c_1 e^{kx} e^{lix} + c_2 e^{kx} e^{-lix}$$

$$c_1 e^{kx} (\cos lx + i \sin lx) + c_2 e^{kx} (\cos -lx + i \sin -lx)$$

$$c_1 e^{kx} (\cos lx + i \sin lx) + c_2 e^{kx} (\cos lx - i \sin lx) = e^{kx} ((c_1 + c_2) \cos lx + (c_1 - c_2)i \sin lx)$$

$$e^{kx} (\vec{c}_1 \cos lx + \vec{c}_2 \sin lx)$$

$$y = e^{kx} (c_1 \cos lx + c_2 \sin lx), c_1, c_2 \in \mathbb{R}$$

Vi för ett svar som är realt, men vi använder genväg med komplexa tal

#### Exempel Positiv kvot

Lös  $y''(x) - 4y'(x) + 3y(x) = 0$

$$r^2(x) - 4r(x) + 3 = 0 \Rightarrow r_1 = 1, r_2 = 3 \Rightarrow y(x) = c_1 e^x + c_2 e^{3x}, c_1, c_2 \in \mathbb{R}$$

#### Exempel noll kvot

Lös  $y''(t) - 4y'(t) + 4$

$$r^2 - 4r + 4 = 0 \Rightarrow (r - 2)^2 = 0 \Rightarrow r = 2$$
 dubbel rot

$$y(t) = c_1 e^{2t} + c_2 t e^{2t}, c_1, c_2 \in \mathbb{R}$$

#### Exempel koplex kvot

Lös  $y''(t) + 25y(t) = 0$

$$r^2 + 25 = 0 \Rightarrow r^2 = -25 \Rightarrow r = \pm 5i$$

$$y(t) = c_1 \cos(5t) + c_2 \sin(5t), c_1, c_2 \in \mathbb{R}$$

## Partikulär lösning

### Methodology

Partikulär lösning är för att gissa lösningen till det ike homogena lösningarna  
 $ay'' + by' + cy = h(t)$ ,  $y(t) = y_h(t) + y_p(t)$

If  $f(x) = P_n(x)$ , try  $y_p = x^m A_n(x)$

If  $f(x) = P_n(x)e^{rx}$ , try  $y_p = x^m A_n(x)e^{rx}$

If  $f(x) = P_n(x)e^{rx} \cos(kx)$ , try  $y_p = x^m e^{rx}(A_n(x) \cos(kx) + B_n(x) \sin(kx))$

If  $f(x) = P_n(x)e^{rx} \sin(kx)$ , try  $y_p = x^m e^{rx}(A_n(x) \cos(kx) + B_n(x) \sin(kx))$

För att ta reda på t eller k så kan tänka att den homogena lösningen är ej en lösning  
 så börja med k=0 sen gå up tills det är sant

### Exempel trigonometrisk

Lös  $y'' + 9y = \sin 3x$

Sats: diff ekv kan skrivas på följande formel  $y = y_h + y_p$

Homogena lösningen

$$r^2 + 9 = 0 \Rightarrow r = \pm 3i \Rightarrow y_h = A \sin 3x + B \cos 3x$$

Particulöra lösningen

$$\text{Sats } y_p = x^m(A_1 \sin 3x + A_2 \cos 3x), m = 1$$

är godtaglig då det är första ekv som inte kan skrivas på homogen formel

$$y_p = A_1 x \sin 3x + A_2 x \cos 3x$$

$$y'_p = A_1(\sin 3x + 3x \cos 3x) + A_2(\cos 3x - 3x \sin 3x)$$

$$y''_p = A_1(3 \cos 3x + 3 \cos 3x - 9x \sin 3x) + A_2(-3 \sin 3x - 3 \sin 3x - 9x \cos 3x)$$

$$\Rightarrow y''_p = A_1(6 \cos 3x - 9x \sin 3x) + A_2(-6 \sin 3x - 9x \cos 3x) + 9(A_1 x \sin 3x + A_2 x \cos 3x) = \sin 3x$$

$$6A_1 \cos 3x - 6A_2 \sin 3x = \sin 3x \Rightarrow A_1 = 0, A_2 = -1/6$$

$$\Rightarrow y_p = -x/6 \cos 3x$$

Den almäna lösningen blir på följande

$$y(x) = A \sin 3x + B \cos 3x - x/6 \cos 3x$$

**Exempel polynom och Exponensiel**

Lös  $y'' + y' - 2y = 4e^2x + x^2$

Sats: diff ekv kan skrivas på följande formel  $y = y_h + y_p$

Homogena lösningen

$$r^2 + r - 2 = 0 \Rightarrow r_1 = 1, r_2 = -2$$

$$y_h = c_1 e^x + c_2 e^{-2x}, c_1, c_2 \in \mathbb{R}$$

Particulöra lösningen

$$\text{Sats } y_p = y_{p1} + y_{p2}$$

$$y'' + y' - 2y = 4e^{2x}$$

$$y_{p1} = x^m(Ae^{2x}), m = 0 \Rightarrow y'_{p1} = 2Ae^{2x} \Rightarrow y''_{p1} = 4Ae^{2x}$$

$$\Rightarrow 4Ae^{2x} + 2Ae^{2x} - 2Ae^{2x} = 4e^{2x} \Rightarrow A = 1 \Rightarrow y_{p1} = e^{2x}$$

$$y_{p2} = Ax^2 + Bx + C \Rightarrow y'_{p2} = 2Ax + B \Rightarrow y''_{p2} = 2A$$

$$\Rightarrow (2A) + (2Ax + B) - (2(Ax^2 + Bx + C)) = x^2$$

$$\Rightarrow -2A = 1, 2A - 2B = 0, 2A + B - 2C = 0 \Rightarrow A = -1/2, B = -1/2, C = -3/4$$

$$\Rightarrow y_{p2} = -1/2x^2 - 1/2x - 3/4$$

$$\Rightarrow y(x) = c_1 e^x + c_2 e^{-2x} + e^{2x} - 1/2x^2 - 1/2x - 3/4$$

**Resonans****Exempel Resonans**

$$\text{Lösy}'' + y = \sin 2t$$

$$r^2 + 1 = 0 \Rightarrow r = \pm i$$

$$y_h = c_1 \cos t + c_2 \sin t$$

$$y_p = A \sin t + B \cos t$$

$$y'_p = 2A \cos 2t - 2B \sin 2t$$

$$y''_p = -4A \sin 2t - 4B \cos 2t$$

$$y''_p + y_p = \sin 2t \Rightarrow -4A \sin 2t - 4B \cos 2t + A \sin t + B \cos t = \sin 2t$$

$$\Rightarrow A = -1/3, B = 0 \Rightarrow y_p = -1/3 \sin 2t \Rightarrow y(t) = c_1 \cos t + c_2 \cos t - 1/2 \sin 2t$$

**Serielösningar**

Antag att serien  $\sum_{n=0}^{\infty} c_n(x - x_0)^n$  konvergerar för alla  $n = 0$

$x \in x_0 - R, x_0 + R$  Då är funktionen  $f(x) = \sum_{n=0}^{\infty} c_n(x - x_0)^n$  deriverbar i

$(x_0 - R, x_0 + R) \wedge f'(x) \sum_{n=0}^{\infty} n c_n (x - x_0)^{n-1}$  Den sista serien konvergerar också i intervallet  $(x_0 - R, x_0 + R)$



## Chapter 5

# Computer Architecture

## 5.1 ISA1

### 5.1.1 MIPS instructions

The instructions that a assembly language use for the MIPS proses.

#### Terminology

- Register file: 32 registers from R0-R31, each is 32 bits.
- Memory: Large each adders stores a word (4 x Registers). Address is always increments of 4. Memory is segmented into 4 8-bits of data to create a word.
- PC (Project counter): Increase with 4 each time to go to the next memory address.
- Instruction Registers: Retrieve the current instructions.
- Control: Inserts data to the register file and add the operation to ALU (Compute).
- ALU (Compute): Calculates the value.
- Memory Address: Call the value from a specific address. From ex “lw”.
- Data Register: Retries and directs value from memory to register file.

## The operations most used

Function	Instruction	Effect
<b>add</b>	add R1, R2, R3	R1 = R2 + R3
<b>sub</b>	sub R1, R2, R3	R1 = R2 - R3
<b>add immediate</b>	addi R1, R2, 145	R1 = R2 + 145
<b>multiply</b>	mult R2, R3	hi, lo = R2 * R3
<b>divide</b>	div R2, R3	low = R2/R3, hi = remainder
<b>and</b>	and R1, R2, R3	R1 = R2 & R3
<b>or</b>	or R1, R2, R3	R1 = R2   R3
<b>and immediate</b>	andi R1, R2, 145	R1 = R2 & 14
<b>or immediate</b>	ori R1, R2, 145	R1 = R2   145
<b>shift left logical</b>	sll R1, R2, 7	R1 = R2 << 7
<b>shift right logical</b>	srl R1, R2, 7	R1 = R2 >> 7
<b>load word</b>	lw R1, 145(R2)	R1 = memory[R2 + 145]
<b>store word</b>	sw R1, 145(R2)	memory[R2 + 145] = R1
<b>load upper immediate</b>	lui R1, 145	R1 = 145 << 16
<b>branch on equal</b>	beq R1, R2, 145	if (R1 == R2) go to PC + 4 + 145*4
<b>branch on not equal</b>	bne R1, R2, 145	if (R1 != R2) go to PC + 4 + 145*4
<b>set on less than</b>	slt R1, R2, R3	if (R2 < R3) R1 = 1, else R1 = 0
<b>set less than immediate</b>	slti R1, R2, 145	if (R2 < 145) R1 = 1, else R1 = 0
<b>jump</b>	j 145	go to 145
<b>jump register</b>	jr R31	go to R31
<b>jump and link</b>	jal 145	R31 = PC + 4; go to 145

Figure 5.1: MIPS operation tabel

*note:* memory operations like (sw R1, 0(R2)) stores the word in position R2+0 where 0 is the increment since if it is a loop it is needed. The value at that position is R1.

### 5.1.2 Sequencing

## Sequencing in detail

46

1. ALU compares registers
2. Result tells the Control whether to branch
3. If the branch is taken, then the Control adds a constant from the instruction to the Program Counter
4. The Control always adds 4 to the Program Counter

For unconditional jumps the Control replaces the Program Counter with the constant from the instruction.

The label constant is in instruction words, so it needs to be multiplied by 4 to convert to byte address.

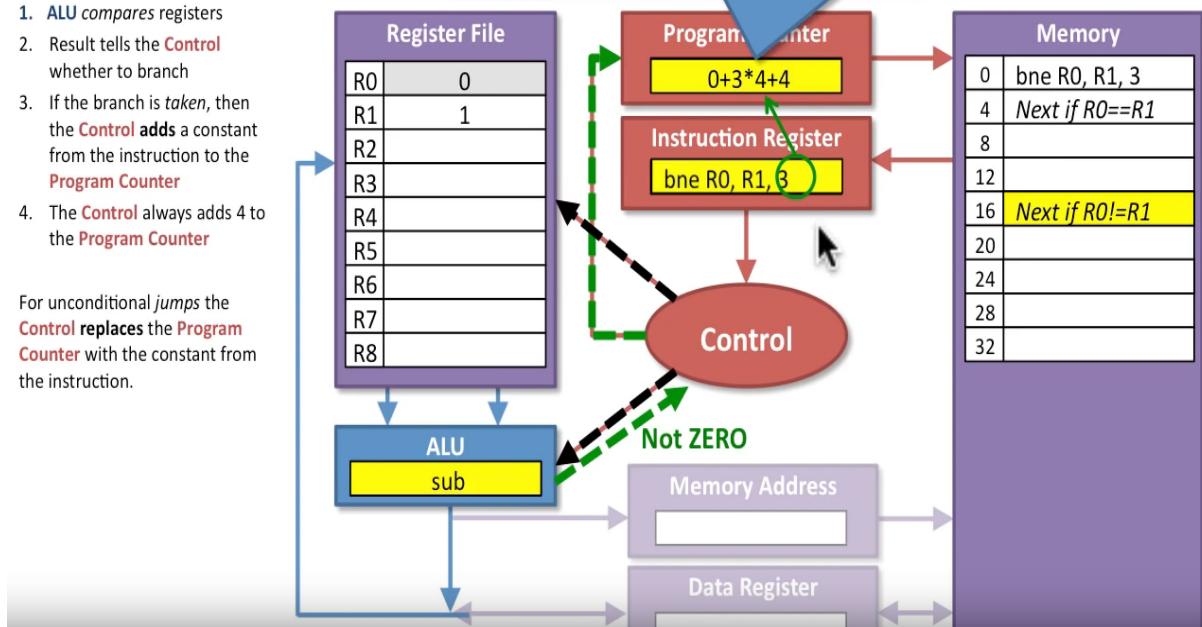


Figure 5.2: Sequencing

### 5.1.3 Converge to Assambly code

If Else example

```

R1 = i; R2 = j; R3 = h

if (i==j)      bne R1, R2, DoElse
    h = i+j;   add R3, R1, R2
                j SkipElse
else           DoElse:
    h = i-j;   sub R3, R1, R2
                SkipElse:

```

Figure 5.3: Assambly exampel

## 5.2 ISA2

### 5.2.1 MIPS type format

- R-format, Aretmetic and lodgical (add)
- I-format, Load/store, branch and intemidiats (addi, beq)
- J-format, Jump (j skipelse)

Name	Bit Fields						Notes (32 bits total)
	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	
R-Format	op	rs	rt	rd	shmt	funct	Arithmetic, logic
I-format	op	rs	rt	address/immediate (16)			Load/store, branch, immediate
J-format	op	target address (26)					

Figure 5.4: MIPS Types

A instruction allwas ends with 00 it means that when a instruction is done we update the program couner to by 4 or more it changes the posistion of the 16bit intermidjet for i-format and can. (bne, beq)

### 5.2.2 Procedure

Procedure is how a function call is handled. It is divided into two parts Caller and Callee, the caller calls the procedure (callee). The operation for controlling procedure is called jump-and-link (jal). It stores the return address ( $PC+4$ ) in \$ra (R31). Where ( $ra = \text{register address (updates automatically)}$ ) ( $PC = \text{program counter}$ ) ( $R31 = \text{value}$ ). jr \$ra returns the value.

### Stacking

To avoid conflicts with writing over the callers register files, one uses staking in predefined ranges to allow store data independent from the callers. (R29 = store stack data \$sp) When a register file is added the stack pointer is moved and when it is done it returns with (jr) and then reset the previously used register files. when it has used what it needs it removes them when leaded one value at a time

```

integer_array_sum:
    addi $sp, $sp, -4    # increase stack-pointer by one word
    sw $s0, 0($sp)      # save array index i to stack
    addi $sp, $sp, -4    # increase stack-pointer by one word
    sw $s1, 0($sp)      # save element n to stack

    addi $v0, $zero, 0    # Initialize Sum to zero.
    add $s0, $zero, $zero # Initialize i to zero.

ia_loop:
    beq $s0, $a1, ia_done    # Done if i == N
    lw $s1, 0($a0)          # n = A[i]
    add $v0, $v0, $s1        # Sum = Sum + n
    addi $a0, $a0, 4         # address = ARRAY + 4*i
    addi $s0, $s0, 1         # i++
    j ia_loop               # next element

ia_done:
    lw $s1, 0($sp)          # loads i from stack-pointer
    lw $s0, 4($sp)          # loads n from stack-pointer
    addi $sp, $sp, 8         # restore stack-pointer

    jr $ra                  # return to caller

```

Figure 5.5: code ex

Notice that the register files for the callee save is \$sx and for the caller \$tx.

## MIPS register names and conventions

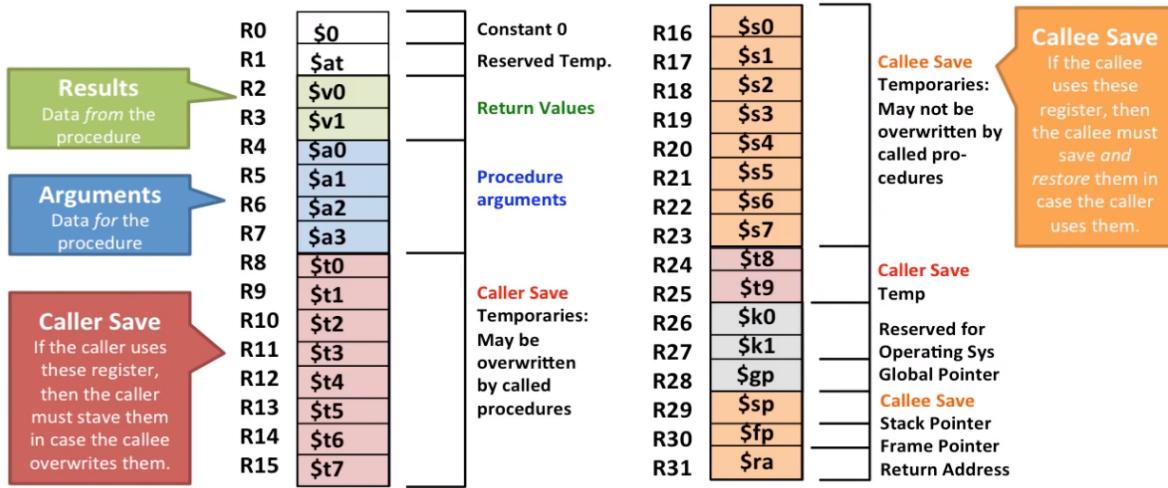


Figure 5.6: MIPS register

### 5.2.3 Other ISA

- **Accumulator** (1 register)
  - 1 address      **add A**       $acc \leftarrow acc + mem[A]$
- **General purpose register file** (load/store)
  - 3 addresses      **add Ra Rb Rc**       $Ra \leftarrow Rb + Rc$   
**load Ra Rb**       $Ra \leftarrow Mem[Rb]$
- **General purpose register file** (Register-Memory)
  - 2 address      **add Ra addressB**       $Ra \leftarrow Mem[addressB]$
- **Stack** (not a register file but an operand stack)
  - 0 address      **add**       $tos \leftarrow tos + next$   
 $tos = top\ of\ stack$

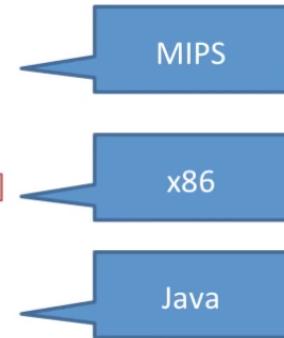


Figure 5.7: Other ISA

Stack	Accumulator	Register (Register-memory)	Register (Load-store)
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R1, B	Load R2, B
Add	Store C	Store C, R1	Add R3, R2, R1
Pop C			Store C, R3

JVM	PDP-8, 8008, 8051	x86	MIPS, PPC, ARM, SPARC
-----	-------------------	-----	-----------------------

Figure 5.8: Other ISA instructions

## 5.3 Arithmetic

### 5.3.1 Binary numbers

- Binary numbers reduces noise and disregard the exact voltage to result in on/off mode.
- In computer science octal and hexadecimals are also often used because of its properties. every octal digit represents 3 in decimal every hex digit represents 4 in decimal
- msb=most significant bit
- lsb=least significant bit
- Optimization:  $0010 * 0101$  2 operations  $\rightarrow 0101 * 0010$  1 operation
- Multiplication: (fixed point need to shift decimal point)  $0010 * 0101 = 0010 + (0010 \text{ shifting}[00]) = 1010$   
 $0011.010 * 0001.110 =$
- Addition: (Same for fixed points and integers)  $1110 + 1000 = \text{carry}[1]0110$  Overflow  $0101 + 0001 = 0110$   
 Not overflow

### 5.3.2 Negative integers

- *Signed magnitude*: msb is the sign  $1011_2 = -(2+1)_{10} = -3_{10}$
- *twos complement*: if msb is 1 then negative number every other digit after is positive  $1011_2 = -8 + 2 + 1_{10} = -5_{10}$

### 5.3.3 Operations

#### 5.3.4 Non integer numbers (Floating and fixed point)

- Converting binary to decimal:  $0.111 = 1/2 + 1/4 + 1/8 = 0.875$
- Fixed point: for defined ranges 1.001 and 0.100 one can choose large and small representations
- Floating point: IEEE standard, see following image
- Standard floating point is in binary so FFFF is at most 1111 and with two's complement 0111



$$(-1)^s \times (1.f) \times 2^{(e-127)}$$

Figure 5.9: Floating Point

### 5.3.5 Overflow

- input: msb = 1 for both numbers and output: msb = 0 (overflow)
- input: msb = 0 for both numbers and output: msb = 1 (overflow)
- input: msb = 0 for both numbers and output: msb = 0 (Not overflow)
- input: msb = 1 for both numbers and output: msb = 1 (Not overflow)
- input: msb = 1 and msb = 0 (Maby overflow)

## 5.4 Logic

# Logic Gates

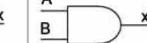
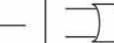
Name	NOT	AND	NAND	OR	NOR	XOR	XNOR																																																																																																
Alg. Expr.	$\bar{A}$	$AB$	$\bar{AB}$	$A+B$	$\bar{A+B}$	$A \oplus B$	$\bar{A \oplus B}$																																																																																																
Symbol																																																																																																							
Truth Table	<table border="1"><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	1	1	1	0	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
A	X																																																																																																						
0	1																																																																																																						
1	0																																																																																																						
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					

Figure 5.10: Floating Point

- truth table, is used to represent all possible inputs and then the corresponding output.
- To determined the possible schematics one can write for the input A and B “!A and B” not A and B. One often get a unnecessary complexity, one can often simplify the schematics. It is only done with the inputs that have an output of one.
- De Morgan’s Law  $!(A + B) = !A \cdot !B$  + and · works the same way but are different.
- karno map is used to easily determine the schematics. It is a matrix. It is possible to use more inputs then two. In the matrix one looks at when output is one.
- A cabal with 4 wires can take 4 bits. Overflow needs one extra wire on output

### Building blocks

- Building blocks are the first layer of abstraction since one can not see the logical gates constructed of. For example add is a building block.
- Two types of logic, combinational and sequential. combinational: output just depends on input sequential: output depends on the state. State is stored in memory update on clock.
- MUXes (Multiplexers): choose input from a bus/ routing signal. 2-bit decision for input 4-bit selection. Starts from position 0.
- DEMUXes (Demultiplexers): opposite, take on signal and decides where it wants to be sent to Bus is a multi-bit signal “wire”
- Decoder: binary to hot, can choose position in array.
- Encoder: hot to binary.

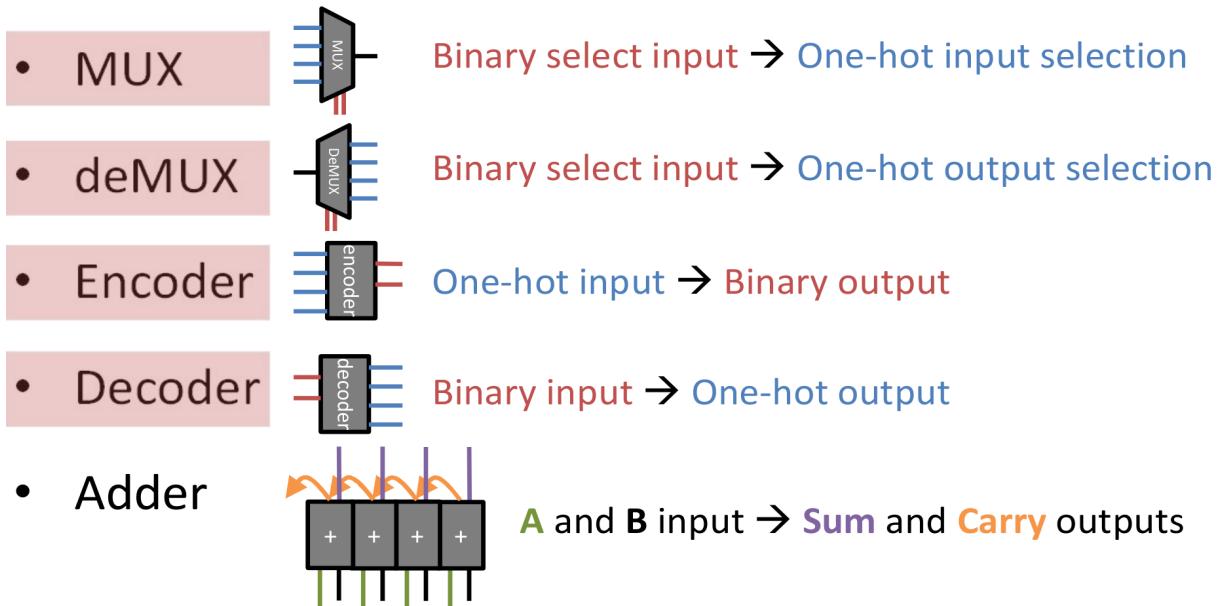


Figure 5.11: Floating Point

- Adders: adds carries on extra bit to handle overflow

### Latches

- One use is for counters with a clock stateless triggers the count. So when input is one the output is 0 and when the clock clicks it is one then when the input is 0 the output is still 1.
- flip flop: is a edge triggered latch has a clock trigger to open or close latch and then can save it to SRAM cell

### Memory

- SRAM (Static Random Access Memory) Big, fast and expensive. Loops through value in order to store it. To write to memory one uses a switch to update value.
- DRAM (Dynamic Random Access Memory) Smaller, Slower and cheaper. Uses a transistor to store value and a capacitor to store the charge so the data does not disappear. That is why it is slower because the capacitor needs to be recharged

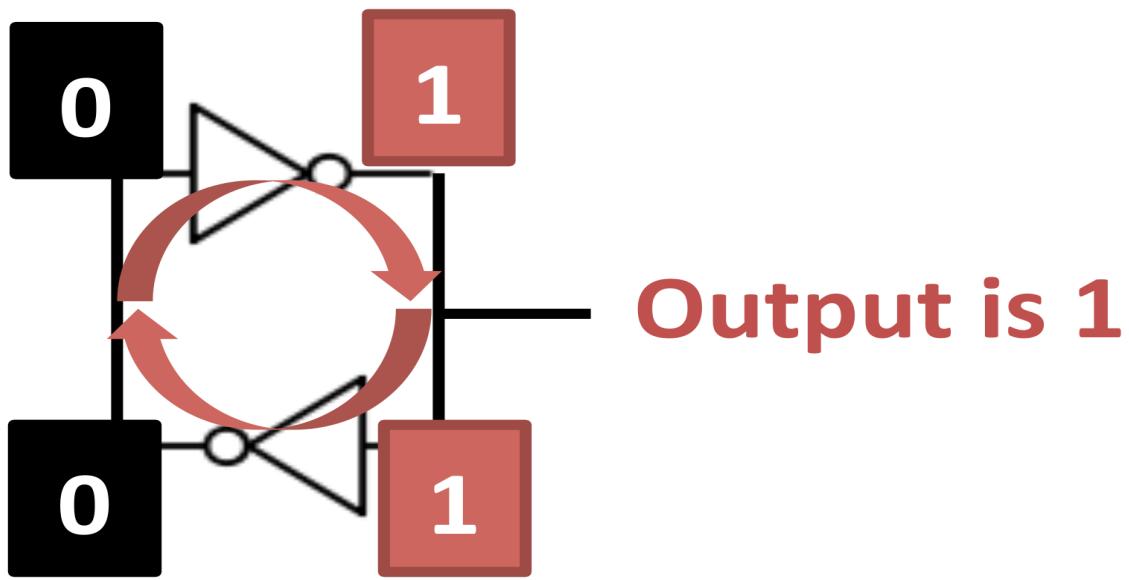


Figure 5.12: latche

## 5.5 processor control and datapath

There are 3 main parts of the MIPS datapath as seen in the following image:

- **ALU operations (add, or, etc.)**
  - Load the instruction
  - Calculate the next PC
  - Read the register file
  - Do the ALU operation
  - Write data back to the register file
- **Memory access (load/store)**
  - Load the instruction
  - Calculate the next PC
  - Read the register file
  - Calculate the address
  - Read/Write the data memory
  - Write data back to the register file
- **Instruction fetch (branch)**
  - Load the instruction
  - Calculate the next PC
  - Read the register file
  - Calculate the branch address
  - Do a branch comparison
  - Update the PC

Figure 5.13: mips datapath 3 parts

An overview of the mips datapath with j-format instruction:

Think about the controller like a decoder that decodes the opt code from the different formats.

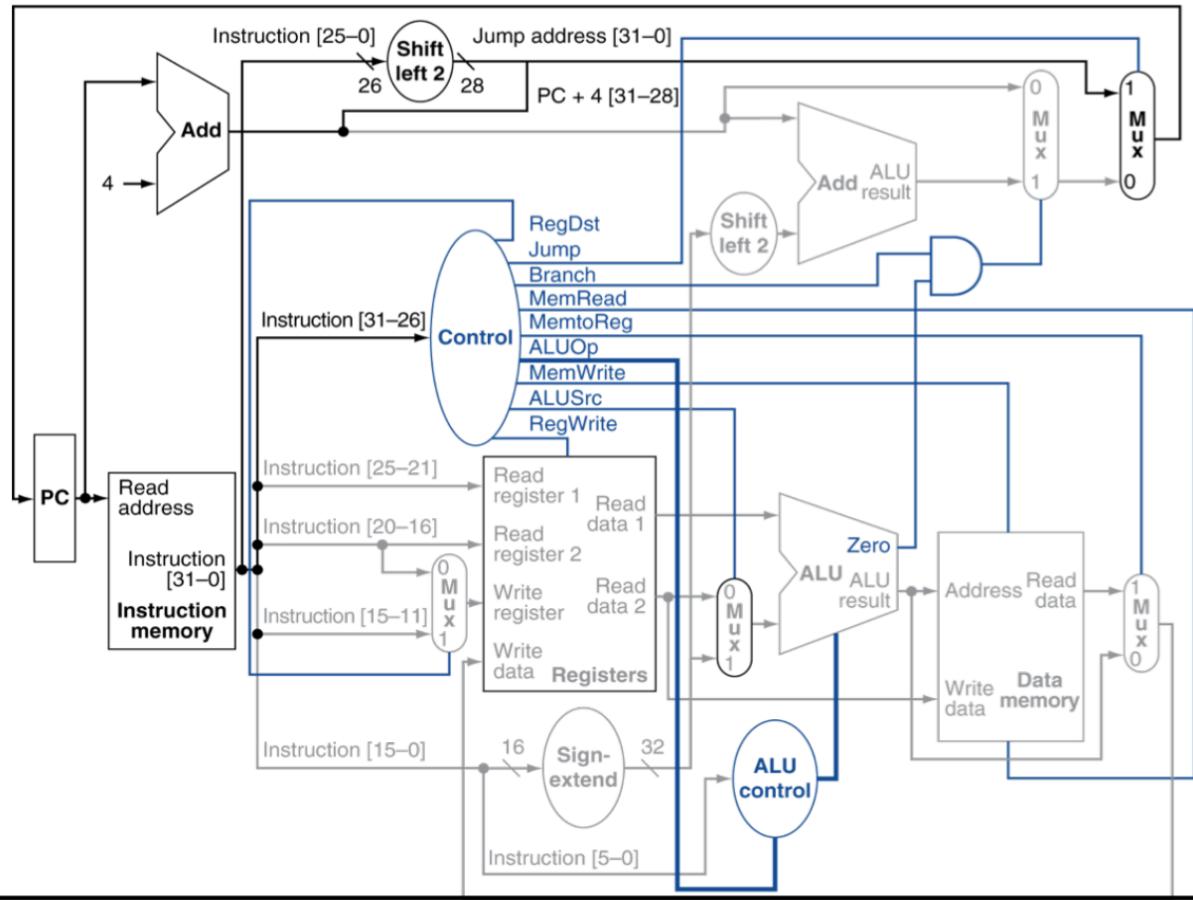


Figure 5.14: mips datapath

### 5.5.1 Clock

A clock is used to update the state and continue with the other instructions. Every state element uses a clock. In mips processor clock goes to memory, pc, rf and dm. The clock unit is in (MHz) converting from (ns) is simple. Ex 10ns:  $1/10\text{ns} = 0.1\text{MHz}$ .

### 5.5.2 Critical path

The longest path of the datapath is the critical path. Often PC is the fastest so one then calculates the amount of time it takes for the instruction to travel the data path and refer to know what the critical path is. The longest part is data memory size it is big and slow. One often say read and write takes constant time regardless of the amount of different read and write.

## 5.6 pipeline

The purpose is to split a instruction cycle into multiple stages to run instructions in parallel to improve performance. Instruction stage has its own pipeline register file for storing the necessary registers. Pipeline registers have a performance reduction since it takes time, therefore more pipeline stages do not equal greater performance over a certain amount. One issue of performance is balance the stages inorder to get a good clock frequency. Not all stages are needed for each operation or instruction

### MIPS stages

- IF = Instruction fetch
- ID = Decode and RF Read
- EX = ALU Execute
- MEM = Memory
- WB = RF Write back

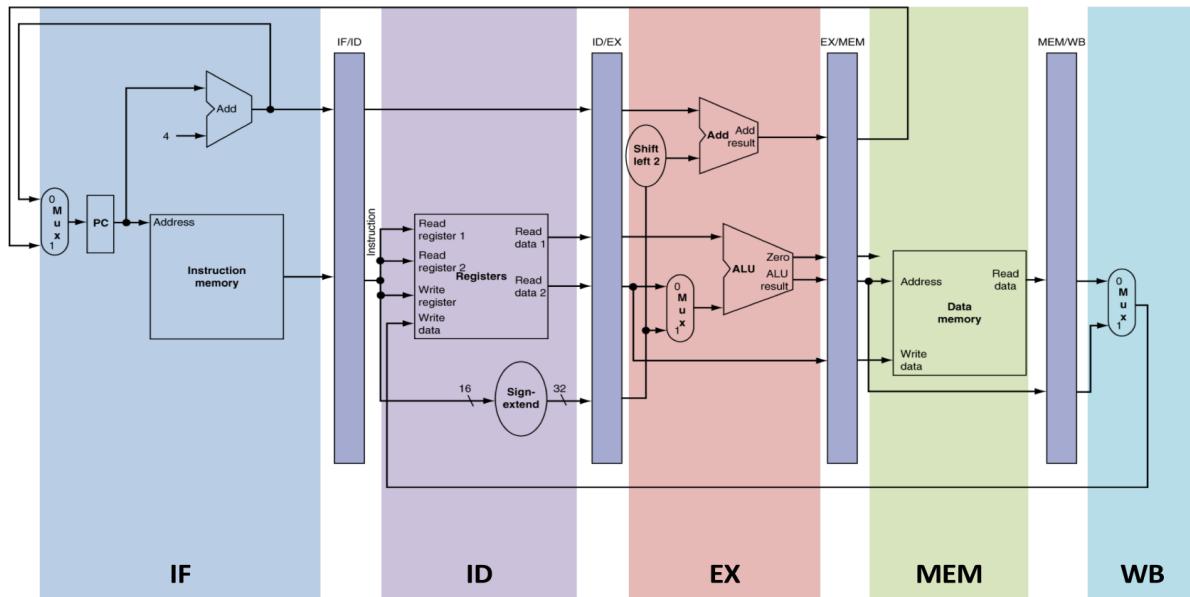


Figure 5.15: Pipeline

**Terminology**

- Bubbles= Is detected by the hardware where there is no instructions
- Nop= Is a sudo instruction for a stall type instruction (do not do anything)
- Delay slots= a stall type for branches. Try to fill in those slots with useful instructions.
- Interference= Can not read and write at the same time.
- Double pumping= Spilling write to first clock cycle and the second cycle is read.
- Forwarding= Getting data from a different pipeline stage from a register file.

**Calculating time complexity**

- Latency: (stages\*(penalty per stage))  
overhead: (instructions time (ex 100ns)) / (stages (ex 1000)) + (register overhead (1ns)) = 1.1ns Total time:  $1.1 * 1000$

## 5.7 Pipeline hazards

### 5.7.1 Data hazards

#### The issue

- Data is not available where we need it (later in the pipeline; not written back yet).
- Data is not available when we need it (need to read memory first).

#### Fixing the issue

- Forward the data to where we need it.
- Stall if data is not ready yet (NOPs or Bubbles).

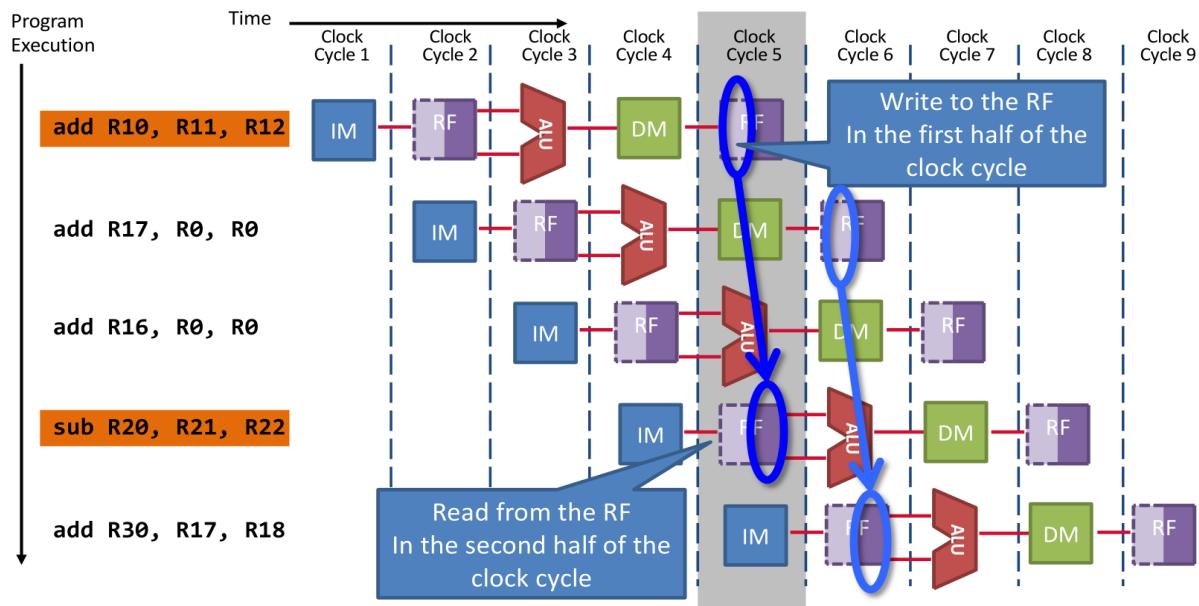


Figure 5.16: Bubble pump

### 5.7.2 Control hazards

#### The issue

- Don't know which instruction is next when we need to fetch it.

#### Fixing the issue

- Calculate branch as early as possible.
- Stall with a branch delay slot.

### 5.7.3 Structural hazards

#### The issue

- Can't do the instruction because the hardware is busy.

#### Fixing the issue

- Build more hardware (double-pumped register file). Double-pumped write at the first half cycle and read at the other half

#### Calculating time complexity

- Branch delay  
(How many branches (ex 20%)) / (How many useful instructions can be filled in (ex 50%)) = 20%/0.5  
= 10%
- Instructions per cycle

## 5.8 Predicting Branches and Exceptions

When the prediction is wrong, clean up is needed. – “Kill” or “Squash” so they don’t execute (they were wrong) – Prevent them from writing: disable RegWrite, MemWrite in the pipeline – Turn them into NOPs: change opcode to add R0, R0, R0 in the pipeline

### 5.8.1 Static predictors

- Predict always not taken
- Predict always taken
- Backwards-Taken, Forward-Not-Taken (BTFTNT)

### 5.8.2 Dynamic predictors

The implementation of branch predictors look something like this:

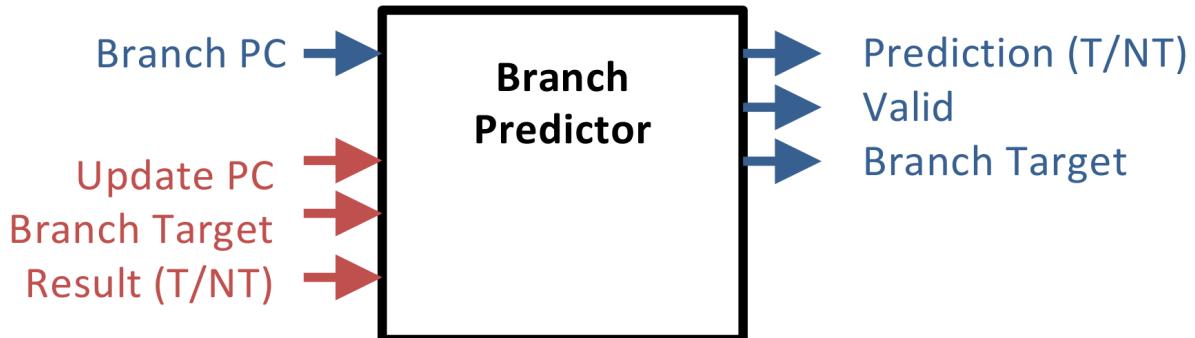


Figure 5.17: branch-predictor

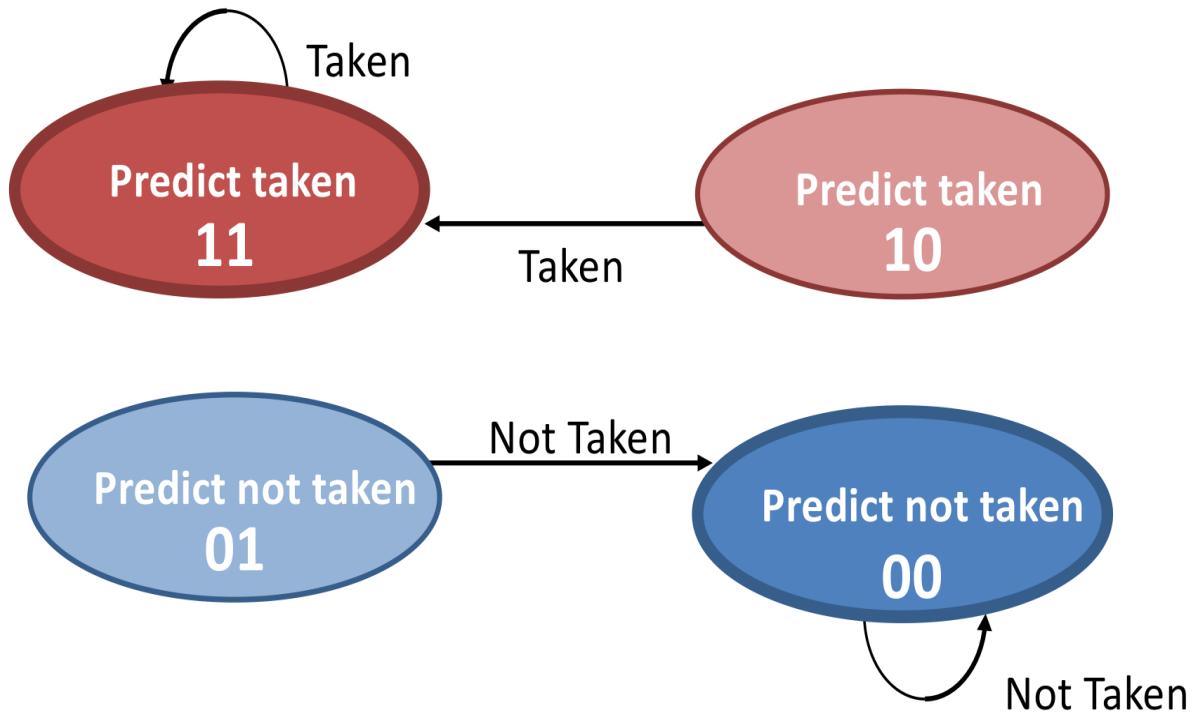


Figure 5.18: 2-bit predict

**BTB**

- Branch Target Buffer (BTB)
- Save a table with PC in order to have history that we can predict on

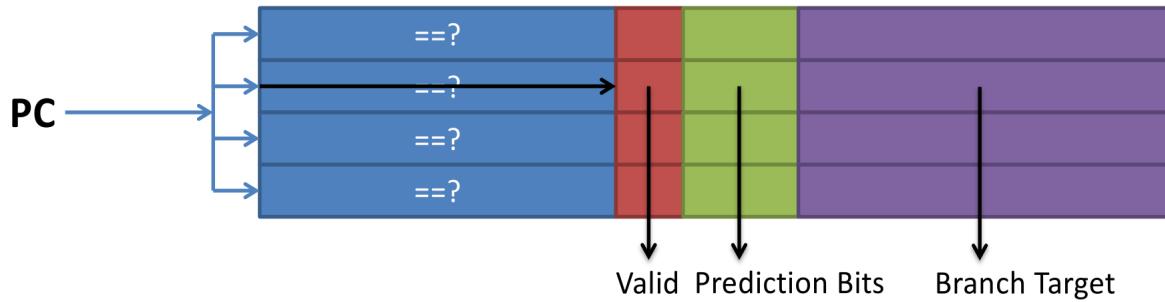


Figure 5.19: btb

### 5.8.3 Exceptions

Exceptions are non-normal events that interrupt the normal flow of instructions

- Divide by zero
- Misaligned memory access
- Page fault
- Memory protection violation

Interrupts are external events that interrupt the normal flow of

- **Synchronous vs. asynchronous**
  - Synchronous: occur at the same place every time a program executes
  - Asynchronous: caused by external devices and happen at different times
- **User requested vs. coerced**
  - Coerced are hardware events the user can't control
  - User requested are from the user
- **User maskable vs. nonmaskable**
  - Can the user disable the exception/interrupt?
- **Within vs. between instructions**
  - Does the event prevent the current instruction from completing, or interrupt after it?
- **Resume vs. terminate**
  - Can the event be handled (corrected) by the OS or program, or must the program be terminated?

Figure 5.20: exceptions

## 5.9 Input/Output

### Terminology

- nvrrom (flash): Similar to ram but it saves data when there is no power
- Busses: Parallel: many bits at once (e.g., 32 bits together in one clock) Used to be used everywhere Still used inside chips
- Serial links: Serial: one bit at a time (e.g., 32 clock cycles to send 32 bits) Used to be used only where distances were long (e.g., networks) Now used for most off-chip communications
- memory-mapped I/O: Manual handling of I/O devices Map portions of the address space to I/O devices Read and write to those addresses to access the
- direct memory access (DMA): Hardware who manage I/O devices (dynamically)
- Polling: The device puts its status somewhere The OS repeatedly checks for it to change
- Interrupt: When the device is ready, it gets the processor's attention by signaling an interrupt The OS then jumps to an interrupt handler to handle the event
- Throughput: x/s The read and write speed
- Latency: s/x Accessing time.
- Overhead: any combination of excess or indirect computation time, memory, bandwidth, or other resources that are required to perform a specific task

**Data transfers: manual copy**

```
add R2, R0, R0      // Counter starts at 0
loop:
    lw R4, 0x12f0(R0) // Read the I/O device
    sw R4, 0xfe00(R2) // Store the results
    addi R2, R2, 4     // Next location
    bne R3, R2, loop
done:
```

**Data transfers: DMA**

```
addi R2, R0, 0xfe00 // destination
addi R3, R0, 230400 // number of words to copy
sw R1, 0x100b(R0) // set the device address
sw R2, 0x1008(R0) // set the destination address
sw R3, 0x1004(R0) // set the length and start
wait:
    lw R2, R0(0x1000) // Wait for it to be done
    bne R2, R0, wait
done:
```

## 5.10 Cache

### 5.10.1 Memory hierarchy

• Registers	3 accesses/cycle	32-64
• Cache	1-10 cycles	8kB-256kB
• Cache	40 cycles	4-20MB
• DRAM	200 cycles	4-16GB
• Flash	1000+ cycles	64-512GB
• Hard Disk	1M+ cycles	2-4TB

Figure 5.21: memory-hierarchy

#### 3-types of cache types

- Fully-associative: Have to search all blocks, but very flexible
- Direct-mapped: Only one place for each block, no flexibility
- Set-associative: Only have to search one set for each block, flexible because each set can store multiple blocks in its ways

#### Write policies

- Write-through: slow, simple
- Write-back: fast (keeps the data just in the cache), more complex

#### Terminology

- Data: What is being stored
- Tag: What address the data has
- Index: What we use to determine where we want to put the data
- Cache line (block) size: How many tag's there are
- Valid bit: If the data is correct
- Dirty bit: The data has been changed and we need to write it to memory
- Type of cache: Fully-associative (FA), Set-associative (SA), Direct-mapped (DM)
- Replacement policy: Write-through, Write-back

### Cache blocks

- tag has 30bits and the 2 other bits is to determine what data 63 bits for each entry 32bit data 30bit tag 1bit valid-bit
- larger block of data -> fewer tags -> more efficient storage
- 1 word cache block we have 1 32bit data
- 2 word cache block we have 2 32bit data
- 4 word cache block we have 4 32bit data
- we load more data when we have space for it we will then have spatial locality
- Last 2: determine the byte within the word
- Next N: determine which word in the cache block
- Remaining 32-N-2: tag

### Calculate overhead

- Data= Cache-Lines \* Byte-Lines
- Overhead= Cache-Lines \* (Tags-per-line + valid-bit)
- % data= Overhead / Data

### Calculate Address (Tag Index Offset)

- Direct mapped:  
Offset= log<sub>2</sub> (number of data blocks) (+ 2-bit (4-bit cache block size if 64 then log 64))  
Index= log<sub>2</sub> (number of cache lines)  
Tags= 32(bit processor)- (Index+Offset)
- x-set associative:  
Offset= log<sub>2</sub> (number of data blocks) (+ 2-bit determines how the address looks like)  
Index= log<sub>2</sub> ((number of cache lines)/x)  
Tags= 32(bit processor)- (Index+Offset)
- fully associative:  
Offset= log<sub>2</sub> (number of data blocks) (+ 2-bit determines how the address looks like)  
Index= 0  
Tags= 32(bit processor)- (Offset)

### Calculate Average memory access time

- Average-cycles= CacheL1\*Cycles + CacheL2\*Cycles + Dram\*Cycles

### 5.10.2 Cache misses 3C's

- cold/compulsory miss: When the program first begins it doesn't have anything in the cache
- conflict miss: too small to store the necessary data
- capacity miss: maps at the same block and then overwrites unnecessarily

- Determine which bits in a 32-bit address are used for selecting the **byte (B)**, selecting the **word (W)**, indexing the **cache (I)**, and the cache **tag (T)**, for each of the following caches:
- 64-line, direct-mapped, write-through, 8 byte line**
  - 8 byte line = 2 words per line, need 1 word bit
  - 64 lines, need 6 bits for indexing
  - TTTT TTTT TTTT TTTT TTTI IIII IWBB**
- 256-line, fully-associative, write-back, 16 byte line**
  - 16 byte line = 4 words per line, need 2 word bits
  - 256 lines, but fully associative! 0 bits for index (data can go anywhere)
  - TTTT TTTT TTTT TTTT TTTT TTTT TTTT WWBB**
- 4096-line, 4-way set-associative, write-back, 64 byte line**
  - 64 byte line = 16 words per line, need 4 word bits
  - 4096 lines/4-ways = 1024 sets, need 10 bits for indexing
  - TTTT TTTT TTTT TTTT IIII IIII IIWW WWBB**

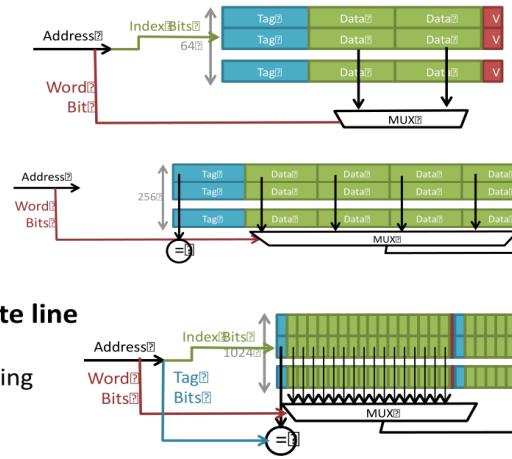


Figure 5.22: cache-format

## 5.11 Virtual Memory

### Why use VM

- Map memory to disk (“unlimited” memory)
- Keep programs from accessing each other’s memory (security)
- Fill holes in the RAM address space (efficiency)

### Terminology

- Translation= map address
- Page tables= for each program keep track of all translations
- Fine grain= page table with specific address
- Coarse grain= page table with address mapped ranges
- Page Table Entries (PTEs)= number of translation
- Translation Lookaside Buffer TLB= All of the pages, fast translation via hardware
- VA= Virtual program addresses
- PA= Physical RAM addresses
- Page offset= point to a range and then use the offset to determine where
- Translation Lookaside Buffer (TLB)= page table cache (Faster)
- Multilevel page table translation= page table points to other page tables (inception)

### Combining TLB and cache

- Physical caches: slow and needs the translation first and then save get the PA also known as Physical-Index, Physical-Tagged (PIPT)
- Virtual caches: fast uses only virtual addresses, no translation, difficult for protection also known as Virtual-Index, Virtual-Tagged (VIVT):
- Virtual-Index, Physically-Tagged (VIPT): VA for index PA for tags, fast does translation and fetch from cache at the same time, most commonly used. We need a comparator to see if the PA tags from cache matches the TLB PA only then we can say if we had a hit or a miss. The VA offset is what the PA tag is selected and the VA tag (number of virtual pages) is what the TLB uses. Mostly used for L1 cache not L2.

### TLB

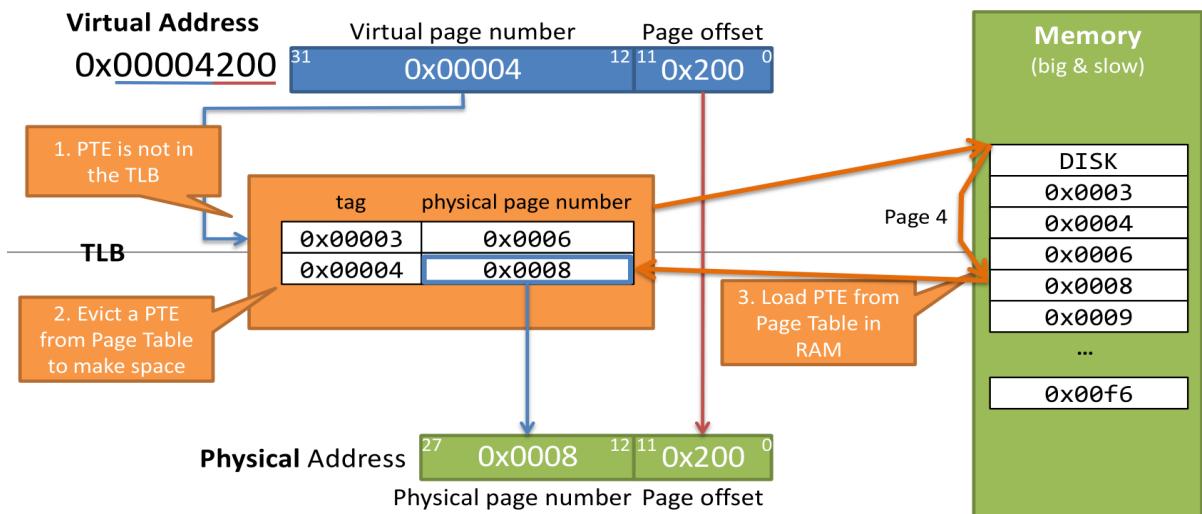


Figure 5.23: tlb

**useful conversion:**  $2^n = xB$

0		10	1kB	20	1MB	30	1GB
1	2B	11	2kB	21	2MB	31	2GB
2	4B	12	4kB	22	4MB	32	4GB
3	8B	13	8kB	23	8MB		
4	16B	14	16kB	24	16MB		
5	32B	15	32kB	25	32MB		
6	64B	16	64kB	26	64MB		
7	128B	17	128kB	27	128MB		
8	256B	18	256kB	28	256MB		
9	512B	19	512kB	29	512MB		

Figure 5.24: conversion

#### Calculating Page sizing

- Number-of-Virtual-Pages=  $2^{32}/\text{pages}$
- Bits-used-for-Page-Offset=  $\log(\text{pages})$
- Bits-used-for-VPN= 32(bit processor)- Bits-used-for-Page-Offset

#### Calculating TLB size

- TLB-size=Entries\*Pages

## 5.12 Parallism

### 5.12.1 Multicore

**powerwall**

$$P = CV^2f$$

$C$  = capasiter, smaller transistors smaller capasiters

$V$  = voltage, decreasing makes it slower

$f$  = frequency, reducing clock speed

### 5.12.2 Paralel programming

**Avrage processors**

Calculate how many cores are used in avrage we need to know

how chunks (work) there is in total

how many time units (cycles think of a reverse pyramid)

there is 16 input data and we have 8-cores, we want to calculate the total sum

what is the avrage prossesor being used

15chunks, 4timeunits  $\Rightarrow 15/4 = 3.75$

**parallel issues**

- Most programs can not utilize parallelism, need to devide the program to diffrent executions.
- We also need to share cache and therefore have performance issue since we cant use the entire cache.

**How much faster**

75%parallel, 25%nonparallel we have hundred thousand cores

$\Rightarrow$  Parallelism takes  $(0.75/100000 + 0.25 * 1)$

Singular takes  $(0.75 * 1 + 0.25 * 1) \Rightarrow 4 * \text{ faster}$

$$\text{Speedup with Amdahl's law } \text{Speedup} = \frac{1}{(1 - P) + P/S}$$

$P$  = Parallel aktion

$S$  = Speed up of the parrallel part

$$\Rightarrow \frac{1}{(1 - 0.75) + 0.75/100000} = \frac{1}{0.25 + 0} = 4$$

### 5.12.3 Synchronization

- Fix the issue with 2 processors accessing the same value at the same time.
- We can use atomic swap to first set lock to 1 then check the data.

## locks

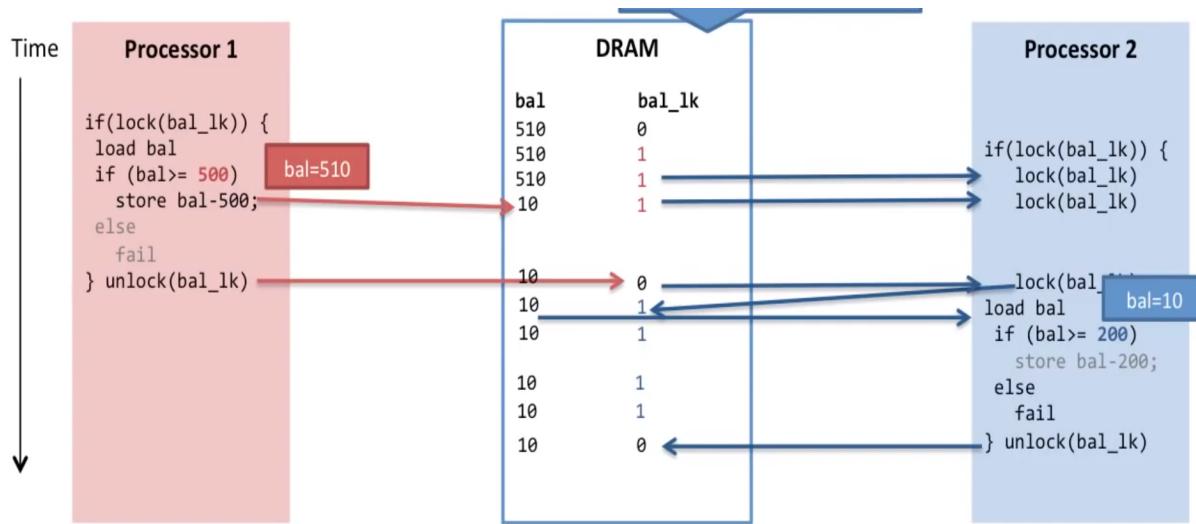


Figure 5.25: locks

## 5.12.4 Cache coherency

- How we use caches to save memory
- Locks: if the data has been accessed. (not a guarantee of protection)
- Snooping: Look what the other processors are storing in their private cache

## snooping

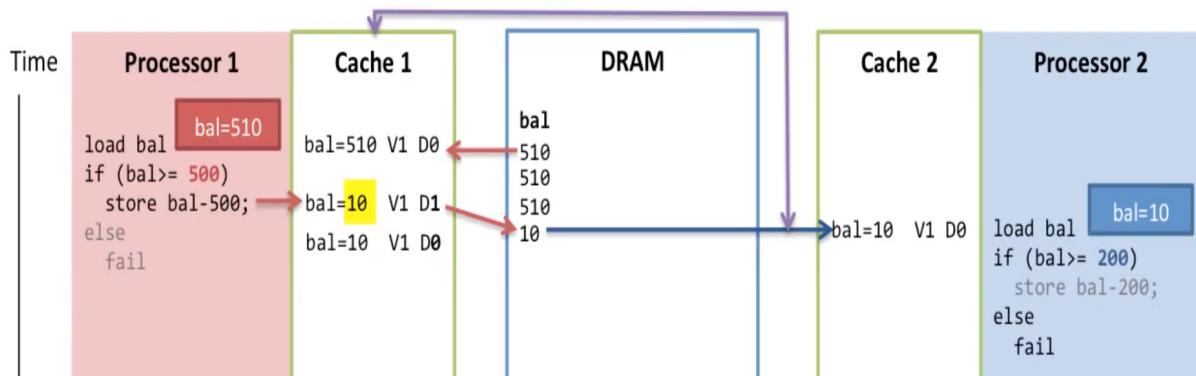


Figure 5.26: locks

### 5.12.5 ILP

- Instruction level parallelism (ILP)
- better to have out-of-order execution since we can find independent instructions
- We use Dual-issue pipeline so we can have one for memory instructions and one for non memory instruction
- It makes ISA promise with in order execution
- Issue with data hazards, more complexity

**dual issue pipeline**

- **Regular Path**

- **Ld/St Path**

- Added:
  - More RF ports
  - 2<sup>nd</sup> instruction fetch
  - 2<sup>nd</sup> sign-extension
  - 2<sup>nd</sup> ALU
  - More forwarding logic and paths
- Now we can issue both a ld/st and any other instruction at the same time!

```

1: add r1, r2, r3
1: ld r4, r5
2: sub r7, r1, r4
2: st r8, r9
3: or r5, r8, r9
3: nop
  
```

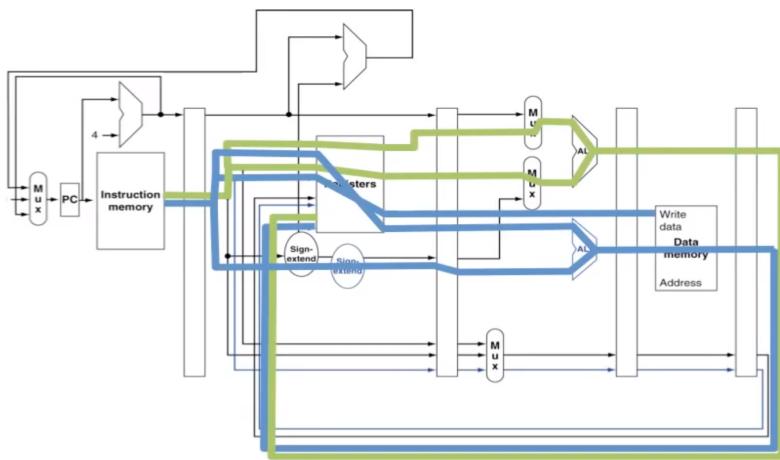


Figure 5.27: dual-issue-pipeline

## Chapter 6

# Probability and Statistics DV

## 6.1 Introduction

### 6.1.1 Test

Innan:

Läs igenom standard instruktionerna.

Sätt upp mjukvaru miljö på skollans datorer.

Under:

Det är mycket tid.

Läs igenom instruktionerna på uppgiften noggrant.

## 6.2 Imperative programming i C

C is a strict programming language, we must therefore write every type of variables. Like any other imperative programming language it execute sequentially and every argument is evaluated in order. It also allow side effects like changing the state freely.

### Vilkorsatser

```
if (expr) { expr; }
if (age > 100) { puts("Very\_old"); }

if (expr) { expr; } else { expr; }
if (age % 2 == 0) { puts("Even"); } else { puts("Odd"); }

expr ? expr : expr
a < b ? b : a;
```

### Shortcuts loop operations

```
n_fakultet *= n--;
```

### 6.2.1 Data structurer

#### stack och heap

I kortids minnet som c kompilatorn hanterar skjälv så sappar datan i en stack. Stack funkar som talrikar som man staplar på varandra och tar den som du la senast ut.

Heap är för långtids mine och störe data som arrayer. Det fungerar som ett rutat papper där varge ruta är en viss data size.

**Shortcuts loop operations** För att använda heapen, (rita på det rutade papret) måste man följa 4 steg.

*steg 1: räkna ut hur många rutor vi behöver*

Använd plattformsberoende hjälpmödel och vanlig aritmetik

```
sizeof(T) * antal
int size = sizeof(int) * 1024
```

*steg 2: reservera motsvarande yta*

Här kan det gå fel — det kanske inte finns plats på papperet

```
T *namn = malloc(antal bytes);
int *skonummer = malloc(size);
```

*steg 3: använd ytan hejvilt*

Men sudda först! (Beror på datastrukturen)

*steg 4: lämna tillbaka platsen när du är klar*

Annars kommer det gå dåligt i ett framtida steg 2

```
free(namn);
free(skonummer);
```

### Structar

För att skapa egna data structurer som används man sig av "struct" operatorn. Om man vill gömma fler data typer i en så kan man använda en "union". För att göra en typ så används man "typedef" och avslutar med

```
namnet_{t}
```

### Pekare

Pekare pekar var datan finns så man kan skicka information utan att sätta stora mängder data.

```
int a
int *b
```

Syntaxen som används är: pekartypen

```
int *
, adresstagningsoperatorn
&
och avrefereringsoperatorn
*
. Skillnaden mellan arrayer och pointers är väldigt liten i c. Därför är samma operation för string (
char *
) som en pointer (
int *
). Pekaren behöver inte peka på en data värde utan kan peka på flertalet värden.
```

### Linked list

Varge element har data av någon typ samt en pekare till nästa element. Head och tail kan beskriva en sådan lista, då head är första elementet och tail är rästen.

Tid komplexiteten är: *On*. Linjär

### Hash table

Hash table är ett mer effektivt sätt att hantera data istället för en array. Då man inte behöver källa varelement vad den finns utan använder en hash function för att skapa ett "nummer" som pekar till vad datan finns någonstans exempelvis kan det vara mod element i hash table då vet vi har vi ska ska söka efter. Ett problem som kommer uppstå är att data mappas till samma plats, då får man använda exempelvis *linuer probing* som säger att om det är fult på platsen gå till nästa tills det finns och om det inte finns så protesterar programmet. Varge element innehåller inget värde eller deleted värde om det är tomta.

Tid komplexiteten är: *O1*. Konstant men inte i verkligheten

## 6.3 Object-Oriented programming i Java

kör den mest specifika implementationen av metoden single dispage, vi bryr os om objectet vi kör metoden på men inte argumentet



## Chapter 7

# Linear Algebra and Geometry I

## 7.1 Linjära ekvationssystem

### 3 gundläggande operationer för att lösa linjära ekvationer

- (1) (Tvåpilar) Byt om på två ekvationer
- (2) (Labbda) Multiplicera båda sidorna av en ekvation med  $\lambda \neq 0$
- (3) (Labda pil) addera  $\lambda$  gånga en ekvation till en annan ekvation

#### Generell lösning

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

Lambda pil upp  $\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$

Ger nya ekvationssystemet

Lambda pil upp  $\begin{cases} (a_1 + \lambda a_2)x + (b_1 + \lambda b_2)y = c_1 + \lambda c_2 \\ a_2x + b_2y = c_2 \end{cases}$

-Lambda pil upp  $\begin{cases} (a_1 + \lambda a_2)x + (b_1 + \lambda b_2)y = c_1 + \lambda c_2 \\ a_2x + b_2y = c_2 \end{cases}$

#### Exempel linjär ekvationssystem

$$\begin{cases} 2x + 3y = 4 \\ 3x - y = 6 \end{cases}$$

Andvänder rad operationer för att lösa ekv systemet

$$\begin{aligned} (-\frac{3}{2})\text{pil ned } & \begin{cases} 2x + 3y = 4 \\ 3x - y = 6 \end{cases} \\ (3x - y) - \frac{3}{2}(2x + 3y) &= 6 - \frac{3}{2} \cdot 4 \Leftrightarrow \\ (3 - \frac{3}{2} \cdot 2)x + (-1 - \frac{9}{2})y &= 6 - 6 \Leftrightarrow -\frac{11}{2}y = 0 \\ (-\frac{2}{11})\text{ned } & \begin{cases} 2x + 3y = 4 \\ -\frac{11}{2}y = 0 \end{cases} \\ (-3)\text{pil upp } & \begin{cases} 2x + 3y = 4 \\ y = 0 \end{cases} \\ (\frac{1}{2})\text{upp } & \begin{cases} 2x = 4 \\ y = 0 \end{cases} \\ & \begin{cases} x = 2 \\ y = 0 \end{cases} \end{aligned}$$

**Kontrol:** stoppar in  $x$  och  $y$  i ekvationerna

$$2 \cdot 2 + 3 \cdot 0 = 4 \text{ (stämmer)}$$

$$3 \cdot 2 - \cdot 0 = 6 \text{ (stämmer)}$$

### 7.1.1 Total Matris

#### Termonologi

- Rader och Kolonner: Rader är vågräta delen av matrisen (ekvationen) Kolonner är lodräta delen (koeficenterna)
- ledande ekvivalent:
- trappstegs matris: När ledande ekvivalent är i trapp form går ned max ett steg
- rad kanonisk: När alla av de ledande ekvivalent inte har någon i samma kolonn

**Exempel matriser**

$$\left\{ \begin{array}{l} x + 2y + z = -1 \\ 2x + (a+3)y + 3z = -4 \\ x + (3-a)y + (a-2)z = a-1 \end{array} \right.$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 2 & a+3 & 3 & -4 \\ 1 & 3-a & a-2 & a-1 \end{array} \right)$$

(-1 rad1 till rad2), (-2 rad1 till rad3)

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 1-a & a-3 & a \end{array} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 0 & a-2 & a-2 \end{array} \right)$$

$a \neq 1 \wedge a \neq 2$

$$(\frac{1}{a-2} \text{rad } 3)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

(-1rad 3 till rad 2), (-1rad 3 till rad 1)

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & a-1 & 0 & -3 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$(\frac{1}{a-1} \text{rad } 2)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & a-1 & 0 & -3 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & 1 & 0 & \frac{3}{1-a} \\ 0 & 0 & 1 & 1 \end{array} \right)$$

(-2rad 2 till rad 3)

$$\left( \begin{array}{ccc|c} 1 & 0 & 0 & -2 - \frac{6}{1-a} \\ 0 & 1 & 0 & \frac{3}{1-a} \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$\left\{ \begin{array}{l} x = -2 - \frac{6}{1-a} \\ y = \frac{3}{1-a} \\ z = 1 \end{array} \right.$$

**Kontroll:** stoppar in x,y,z i ekvationerna

$$\left\{ \begin{array}{l} \left( \frac{2a-8}{1-a} \right) + 2 \cdot \left( \frac{3}{1-a} \right) + 1 = -1 \\ 2 \cdot \left( \frac{2a-8}{1-a} \right) + (a+3) \cdot \left( \frac{3}{1-a} \right) + 3 \cdot 1 = -4 \\ \left( \frac{2a-8}{1-a} \right) + (3-a) \cdot \left( \frac{3}{1-a} \right) + (a-2) \cdot 1 = a-1 \end{array} \right.$$

## 7.2 Vektorer/koordinater i planet och rummet

### Räkne regler vektorer

$$A1 \vec{u} + \vec{v} = \vec{v} + \vec{u}$$

$$A2 \vec{u} + (\vec{v} + \vec{w}) = (\vec{v} + \vec{u}) + \vec{w}$$

$$A3 \vec{u} + \vec{0} = \vec{u}$$

$$A4 \vec{u} + \vec{v} = \vec{0} \Leftrightarrow \vec{v} = -\vec{u}$$

$$M1 1\vec{u} = \vec{u}$$

$$M2 k(l\vec{u}) = (kl)\vec{u}$$

$$M3 (k + l)\vec{u} = k\vec{u} + l\vec{u}$$

$$M4 k(\vec{u} + \vec{v}) = k\vec{u} + k\vec{v}$$

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ för } \mathbb{R}^2$$

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \text{ för } \mathbb{R}^3$$

### Definition: Parallelala vektorer

$$\text{Parallelala omm } \exists k : \vec{u} = k\vec{v} \vee k\vec{u} = \vec{v}$$

### 7.2.1 Bas

Standard bas är välbikant då i planet är x och y axeln medans i ett rum är x, y och z. Baser som inte är standard är vektorer som ej är parralella som då skappar axlar som ej behöver vara vinkelräta.

#### Definition: Bas

$$\text{Bas i plan } \forall \vec{x}, \exists k_1, k_2 : \vec{x} = k_1\vec{u} \vee k_2\vec{v}$$

$$\text{Bas generell } \vec{x} = k_1\vec{u}_1 + k_2\vec{u}_2 + \dots + k_n\vec{u}_n$$

$$\underline{u} = (\vec{u}_1 \vec{u}_2 \dots \vec{u}_n)$$

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \vec{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

**Exempel: kordenatar för vektor i bas**

$$\text{Låt: } \vec{u} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \vec{v} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

$$\text{Hitta kordenarterna för vektorn } \vec{x} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

Lösning: vi måste hittar reala tal  $k_1$  och  $k_2$  så att

$$\begin{pmatrix} 0 \\ 6 \end{pmatrix} = k_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + k_2 \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2k_1 - 2k_2 \\ k_1 + k_2 \end{pmatrix}$$

$$\begin{cases} 2k_1 - 2k_2 = 0 \\ k_1 + k_2 = 6 \end{cases} \Rightarrow \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

**Exempel: Om det är en bas**

För vilken värde på  $a$  är vektorerna en bas för  $\mathbb{R}^3$

$$\vec{u}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \vec{u}_2 = \begin{pmatrix} 1 \\ 2 \\ a \end{pmatrix}, \vec{u}_3 = \begin{pmatrix} 1 \\ a \\ 3 \end{pmatrix}$$

Vi måste hitta ett  $a$  sådant att

$$\begin{aligned} k_1 \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, k_2 \begin{pmatrix} 1 \\ 2 \\ a \end{pmatrix}, k_3 \begin{pmatrix} 1 \\ a \\ 3 \end{pmatrix} &= \begin{pmatrix} ? \\ ? \\ ? \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & a \\ 3 & a & 3 \end{pmatrix} \xrightarrow[-2]{+} \xrightarrow[+]^{-3} &\sim \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & (a-2) \\ 0 & (a-3) & 0 \end{pmatrix} \end{aligned}$$

Om  $a = 2$ : då är sista ekvationen  $0 = ?$  och vi ser att lösningarna får parametrar och därför: antingen ingen lösningar eller oändligt många lösningar.

Oavsett -ej bas

Om  $a = 3$ : då blir det samma problem som  $a=2$

Svar: de tre vektorer är en bas om  $a$  inte är 2 eller 3

**Definition: Punkter i planet**

Från origo: alla  $P = (x, y)$  och  $\overrightarrow{OP} = \begin{pmatrix} x \\ y \end{pmatrix}$

Om  $A = (a_1, a_2)$  och  $B = (b_1, b_2)$  då är  $\overrightarrow{AB} = \begin{pmatrix} b_1 - a_1 \\ b_2 - a_2 \end{pmatrix}$

**Definition: Längd**

Längd vektor i plan  $|\vec{v}| = \sqrt{a^2 + b^2}$

Längd vektor i rum  $|\vec{v}| = \sqrt{a^2 + b^2 + c^2}$

## 7.3 Skalärprodukt och vektorprodukt

### 7.3.1 Skalärprodukt

$$\vec{u} \bullet \vec{v} = |\vec{u}| |\vec{v}| \cos \theta$$

$$\vec{u} \bullet \vec{v} = u_1 v_1 + u_2 v_2 + \dots + u_3 v_3$$

#### Räkneregler

$$\begin{aligned}\vec{u} \bullet \vec{v} &= \vec{v} \bullet \vec{u} \\ \vec{u} \bullet (\vec{v} + \vec{w}) &= \vec{u} \bullet \vec{v} + \vec{u} \bullet \vec{w} \\ \lambda(\vec{u} \bullet \vec{v}) &= (\lambda \vec{u}) \bullet \vec{v} = \vec{u} \bullet (\lambda \vec{v}) \\ \vec{u} \bullet \vec{u} &= |\vec{v}|^2 \\ \vec{u} \bullet \vec{u} &= 0 \Leftrightarrow \vec{u} = \vec{0}\end{aligned}$$

#### Exempel: parrallel och ortogonal

För vilka värden på a och b är vektorerna i  $\mathbb{R}^3$   $\begin{pmatrix} 1 \\ a \\ 2 \end{pmatrix}$  och  $\begin{pmatrix} b \\ 8 \\ a \end{pmatrix}$

(a) parallel?, (b) ortognala?

(a)

$$\begin{cases} k = b \\ ak = 8 \Rightarrow \\ 2k = a \end{cases}$$

$$\begin{cases} (2k)k = 8 \Rightarrow k^2 = 4 \Rightarrow k = 2 \\ 2k = a \end{cases}$$

$$k = b = 2, a = 4$$

(b)

$$\begin{pmatrix} 1 \\ a \\ 2 \end{pmatrix} \bullet \begin{pmatrix} b \\ 8 \\ a \end{pmatrix} = 1b + a2 + 2a = b + 4a = 0$$

#### Exempel: Längd-formeln

$$|\vec{v}| = \sqrt{|\vec{v}|^2} = \sqrt{|\vec{v}| \bullet |\vec{v}|}$$

Beräkna längden av (ON-bas):  $\vec{v} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$

$$\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} \bullet \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} = 2^2 + 1^2 + 2^2 = 9 \Rightarrow$$

$$|\vec{v}| = \sqrt{9} = 3$$

Därmed är längden: 3

**Exempel: Vinkel-formeln**

Låt  $\vec{u}$  och  $\vec{v}$  vara vektorer och vinkel blir då:

$$\theta = \arccos \frac{\vec{u} \bullet \vec{v}}{|\vec{u}| |\vec{v}|}$$

Beräkna vinkeln av (ON-bas):  $\vec{u} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$  och  $\vec{v} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$

$$\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \bullet \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = 1 \cdot 2 + 2^2 + 2 \cdot 1 = 8$$

$$|\vec{u}| = \sqrt{1^2 + 2^2 + 2^2} = \sqrt{9} = 3$$

$$|\vec{v}| = \sqrt{2^2 + 2^2 + 1^2} = \sqrt{9} = 3$$

$$\arccos \frac{\vec{u} \bullet \vec{v}}{|\vec{u}| |\vec{v}|} = \arccos \frac{8}{3 \cdot 3} = \arccos \frac{8}{9} \approx 27.27^\circ$$

**Exempel: Längd av två vektorer**

Låt  $u$  och  $v$  vara två vektorer sådana att

$$|\vec{u}| = 4, |\vec{v}| = 2 \text{ och vinkeln mellan } \vec{u} \text{ och } \vec{v} \text{ är } \frac{2\pi}{3}$$

Bestäm längden av  $3\vec{u} - 2\vec{v}$

$$\begin{aligned} \sqrt{|3\vec{u} - 2\vec{v}|^2} &= \sqrt{9|\vec{u}|^2 + 4|\vec{v}|^2 - 2|\vec{u}||\vec{v}| \cos \frac{2\pi}{3}} \\ &= \sqrt{9 \cdot 16 + 4 \cdot 4 - 4 \cdot 3 \cdot 8 \cdot \frac{-1}{2}} = \sqrt{13 \cdot 16} = 4\sqrt{13} \end{aligned}$$

**Exempel: beräkna skalärprodukten**

$$\underline{u} \begin{pmatrix} 9 \\ 9 \end{pmatrix} \bullet \underline{u} \begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

$$\underline{u} = (\vec{u}_1, \vec{u}_2), \vec{u}_1 \bullet \vec{u}_1 = 9, \vec{u}_1 \bullet \vec{u}_2 = 6, \vec{u}_2 \bullet \vec{u}_2 = 8$$

$$\begin{aligned} \underline{u} \begin{pmatrix} 9 \\ 9 \end{pmatrix} \bullet \underline{u} \begin{pmatrix} -2 \\ -1 \end{pmatrix} &= (9\vec{u}_1 - 2\vec{u}_2) \bullet (9\vec{u}_1 - 1\vec{u}_2) = -18|\vec{u}_1|^2 + (-9 - 18)\vec{u}_1 \bullet \vec{u}_1 - 9|\vec{u}_2|^2 \\ &= -18 \cdot 9 - 27 \cdot (6) - 9 \cdot 8 = -396 \end{aligned}$$

### 7.3.2 Ortogonal projektion

Hitta punkt på linjen som är närmast en punkt. Punkten är ortogonala (vinkelrätta)  
Parallell koposant skrivs  $\vec{v}_{||l}$  och ortogonal skrivs  $\vec{v}_{\perp l}$ .

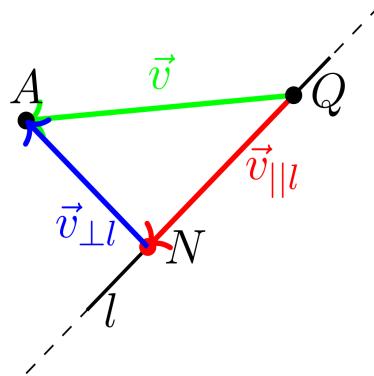


Figure 7.1: Ortogonal projektion

#### Ortogonal projektion Formeln

$$\vec{v}_{||l} = \frac{\vec{u} \bullet \vec{v}}{|\vec{u}|^2} \vec{u}$$

#### Exempel: Hitta parallell och ortogonal komposanten

$$\vec{v} = \vec{v}_{\perp l} + \vec{v}_{||l}$$

$$\vec{v}_{||l} = \frac{\vec{u} \bullet \vec{v}}{|\vec{u}|^2} \vec{u}$$

Beräkna ortogonal komposanten av:

$$\vec{v} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} \text{ och } \vec{u} = \begin{pmatrix} -4 \\ 4 \\ 2 \end{pmatrix}$$

Beräkna först parallell komposanten:

$$\vec{v}_{||\vec{u}} = \frac{\vec{u} \bullet \vec{v}}{|\vec{u}|^2} \vec{u} = \frac{-18}{36} \vec{u} = \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix}$$

Ortogonal komposanten blir då:

$$\vec{v}_{\perp \vec{u}} = \vec{v} - \vec{v}_{||\vec{u}} = \begin{pmatrix} 1 - 2 \\ -2 + 2 \\ 3 - 1 \end{pmatrix}$$

Svar:  $\begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$

**Exempel: Närmaste punkt och avståndet**

Bestäm avståndet från punkten  $P = (-2, 4, 3)$  till linjen

$$\begin{cases} x + 2y - z = 1 \\ x - y + 5z = 4 \end{cases}$$

Finn även den punkt på linjen  $l$  som ligger närmast punkten  $P$

Skriver ekvationen på parameter form

$$\begin{array}{ccc} \left( \begin{array}{ccc|c} 1 & 2 & -1 & 1 \\ 1 & -1 & 5 & 4 \end{array} \right) & \xrightarrow[-1]{+} & \left( \begin{array}{ccc|c} 1 & 2 & -1 & 1 \\ 0 & 1 & -2 & -1 \end{array} \right) \xrightarrow[-2]{+} & \sim \\ \left( \begin{array}{ccc|c} 1 & 0 & 3 & 3 \\ 0 & 1 & -2 & -1 \end{array} \right) & \xrightarrow[-1]{+} & & \sim \\ l: \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix} + t \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix} & & & \text{dvs } (p_0 + t\vec{v}) \end{array}$$

Hittar en godtycklig punkt ex  $t = 1 \Rightarrow Q = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

(1.) beräknar vektor från godtykliga punkten till den givna

$$\overrightarrow{PQ} = \begin{pmatrix} 0 - (-2) \\ 1 - 4 \\ 1 - 3 \end{pmatrix} = \begin{pmatrix} 2 \\ -3 \\ -2 \end{pmatrix}$$

(2.) Beräknar ortogonala projectionen

$$\overrightarrow{NQ} = \overrightarrow{PQ}_{\parallel v} = \frac{\begin{pmatrix} 2 \\ -3 \\ -2 \end{pmatrix} \bullet \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix}}{9 + 4 + 1} \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix} = \frac{-14}{14} \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix}$$

(3.) Beräknar punkten från närmaste punkt till

$$\overrightarrow{ON} = \overrightarrow{OQ} + \overrightarrow{QN} = \overrightarrow{OQ} - \overrightarrow{NQ} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -3 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ 3 \\ 2 \end{pmatrix}$$

$$(4.) \text{ Beräknar längden } \left| \begin{pmatrix} -2 \\ 4 \\ 3 \end{pmatrix} - \begin{pmatrix} -3 \\ 3 \\ 2 \end{pmatrix} \right| = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3}$$

Svar: avståndet är  $\sqrt{3}$  och punten är  $(-3, 3, 2)$

**Exempel: Spegling**

Bestäm speglingen av punkten  $A : (1, 3, -3)$  i planet  $\pi$

som går genom origo och innehåller punkterna  
 $(-1, 1, 1)$  och  $(3, 3, 1)$

Beräknar planets ekvation

$$\vec{n} = \overrightarrow{OQ} \times \overrightarrow{OP} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 4 \\ 6 \end{pmatrix} = 2 \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}$$

$-x + 2y + 3z = 0$  Eftersom den går egenom origo

$$\overrightarrow{NA} = \overrightarrow{OA}_{||\vec{n}} = \frac{\begin{pmatrix} 1 \\ 3 \\ -3 \end{pmatrix} \times \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}}{1+4+9} \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} = \frac{14}{14} \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}$$

Beräkna speglingen rita då är det uppenbart

$$\overrightarrow{OA'} = \overrightarrow{OA} - 2\overrightarrow{NA} = \begin{pmatrix} 1 \\ 2 \\ -3 \end{pmatrix} - 2 \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 3 \end{pmatrix}$$

**Exempel Hitta punkt ortogonal projektion**

Låt  $l$  vara linjen genom punkten  $Q = (1, 2, 3)$  parallell med vektorn

$$\vec{u} = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$$

Hitta den punkt  $N$  på  $l$  som är närmast  $A = (1, 7, 4)$

Lösning: Rita figur och tolka. Vi ortogonalt projeicerar

$$\vec{v} = \overrightarrow{QA} = \begin{pmatrix} 1-1 \\ 7-2 \\ 4-3 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \\ 1 \end{pmatrix} \text{ på vektorn } \vec{u}$$

$$\vec{v}_{||l} = \vec{v}_{||\vec{u}} = \frac{\begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \bullet \begin{pmatrix} 0 \\ 5 \\ 1 \end{pmatrix}}{\begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \bullet \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}} \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \frac{25}{50} \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$$

Detta är ju igen vektorn som pekar frpn  $Q$  till närmaste punkten som därför blir

$$N = \left( 1 - \frac{1}{2}3, 2 - \frac{1}{2}4, 3 - \frac{1}{2}5 \right) = \left( -\frac{1}{2}, 0, \frac{1}{2} \right)$$

**Kontroll:** kollar om  $\overrightarrow{NQ} = k\vec{u}$ ,  $k \in \mathbb{R}$

$$\begin{pmatrix} 1 - (-1/2) \\ 2 - 0 \\ 3 - 1/2 \end{pmatrix} = \begin{pmatrix} 3/2 \\ 2 \\ 5/2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \text{ (stämmer)}$$

**Räkneregler ortogonal**

För vektorer  $\vec{u}$ ,  $\vec{v}$  och  $\vec{w}$  och skalär  $\lambda \in \mathbb{R}$

$$(\lambda\vec{v})_{||\vec{u}} = \lambda(\vec{v}_{||\vec{u}})$$

$$(\vec{v} + \vec{w})_{||\vec{u}} = \vec{v}_{||\vec{u}} + \vec{w}_{||\vec{u}}$$

$$(\lambda\vec{v})_{\perp\vec{u}} = \lambda(\vec{v}_{\perp\vec{u}})$$

$$(\vec{v} + \vec{w})_{\perp\vec{u}} = \vec{v}_{\perp\vec{u}} + \vec{w}_{\perp\vec{u}}$$

### 7.3.3 Enhetsvektorer och ON-baser

En enhetsvektor är en vektor med längd 1.

Om man har en vektor  $\vec{u}$  kan man skala om den med ett positivt tal så den får längd 1.

En bas  $\underline{\vec{u}} = (\vec{u}_1 \vec{u}_2 \dots \vec{u}_n)$  kallas en ortonormal bas (ON-bas) omm:

(1) Alla vektorerna är enhetsvektorer  $|\vec{u}_i|^2 = \vec{u}_i \bullet \vec{u}_i = 1$

(2) Varje par av vektorer  $\vec{u}_i$  och  $\vec{u}_j$  för  $i \neq j$  är ortogonal:  $\vec{u}_i \bullet \vec{u}_j = 0$

$$\hat{\vec{u}} = \frac{1}{|\vec{u}|} \vec{u}$$

Skalärproducten har samma räkneregler i ON-baser som i standard bas

### 7.3.4 Vektorprodukten

I högersystem kan representeras av en höger hand där tumen är vektorn 1 pekfingret är 2 och långfingret är 3. **Definition: högersystem**

En bas  $\underline{\vec{u}} = (\vec{u}_1 \vec{u}_2 \vec{u}_3)$  kallas ett högersystem omm:

set ifrån spetsen av  $\vec{u}_3$  vridas  $\vec{u}_1$  moturs till  $\vec{u}_2$

#### Definition: Vektorprodukten

Givet två icke-parallelala vektorer  $\vec{u}$  och  $\vec{v}$  med vinkel  $\theta$

mellan dem definieras  $\vec{u} \times \vec{v}$  som den entydiga vektor som uppfyller

(a)  $\vec{u} \times \vec{v}$  är ortogonal mot båda  $\vec{u}$  och  $\vec{v}$

(b) längden av  $\vec{u} \times \vec{v}$  är  $|\vec{u} \vec{v} \sin \theta|$

(c)  $(\vec{u} \vec{v} \vec{u} \times \vec{v})$  är ett högersystem

I fallet där  $\vec{u}$  och  $\vec{v}$  är parallella är  $\vec{u} \times \vec{v} = \vec{0}$

#### Definition: produkter

Sala om: tal  $\cdot$  vektor = vektor

Skalär produkt: vektor  $\bullet$  vektor = tal

Vektor produkt: vektor  $\times$  vektor = vektor

#### Räkneregler: Vektorprodukten

För vektorer  $\vec{u}, \vec{v}, \vec{w}$  och ett tal  $\lambda$  gäller

$$\vec{u} \times \vec{v} = -\vec{v} \times \vec{u}$$

$$\vec{u} \times (\vec{v} + \vec{w}) = \vec{u} \times \vec{v} + \vec{u} \times \vec{w}$$

$$\lambda(\vec{u} \times \vec{v}) = \lambda(\vec{u}) \times \vec{v} = \vec{u} \times (\lambda \vec{v})$$

### Vektorprodukten metologi

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} v_2 \cdot w_3 - v_3 \cdot w_2 \\ v_3 \cdot w_1 - v_1 \cdot w_3 \\ v_1 \cdot w_2 - v_2 \cdot w_1 \end{pmatrix}$$

Figure 7.2: Vektorprodukten

### Exempel: Hitta basen

Låt  $\vec{u}_1 = \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix}$  och  $\vec{u}_2 = \begin{pmatrix} 4 \\ -1 \\ -8 \end{pmatrix}$

Verifiera att dessa är ortogonala, och hitta en vektor  $\vec{u}_3$

Så att  $\underline{u} = (\hat{\vec{u}}_1 \hat{\vec{u}}_2 \hat{\vec{u}}_3)$  är en ON-bas (och skriv ut denna)

Lösning: Först kollar vi att skalärprodukten är 0 :

$$\begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix} \bullet \begin{pmatrix} 4 \\ -1 \\ -8 \end{pmatrix} = 16 - 8 - 8 = 0$$

Därmed ortogonala. Nu måste vi hitta en vektor som är ortogonal mot båda vektorprodukten är just det

$$\vec{u}_3 = \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix} \times \begin{pmatrix} 4 \\ -1 \\ -8 \end{pmatrix} \begin{pmatrix} -64 + 1 \\ 4 + 32 \\ -4 - 32 \end{pmatrix} = \begin{pmatrix} -63 \\ 36 \\ -36 \end{pmatrix}$$

Längderna av dessa är (räkar vara lika)

$$|\vec{u}_2|^2 = |\vec{u}_1|^2 = 4^2 + (\pm 8)^2 + (\pm 1)^2 = 81$$

och vi kan göra liknade beräkning för  $\vec{u}_3$ , men vi vet redan att:

$$|\vec{u}_3| = |\vec{u}_1||\vec{u}_2| \sin \theta = 9 \cdot 9 \cdot 1 = 81$$

Desnas tre normering blir därför:

$$\hat{\vec{u}}_1 = \frac{1}{9} \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix}, \hat{\vec{u}}_2 = \frac{1}{9} \begin{pmatrix} 4 \\ -1 \\ -8 \end{pmatrix}, \hat{\vec{u}}_1 = \frac{1}{81} \begin{pmatrix} -63 \\ 36 \\ -36 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} -7 \\ 4 \\ -4 \end{pmatrix}$$

$$\text{Så basen är: } \underline{u} = \left( \frac{1}{9} \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix}, \frac{1}{9} \begin{pmatrix} 4 \\ -1 \\ -8 \end{pmatrix}, \frac{1}{81} \begin{pmatrix} -63 \\ 36 \\ -36 \end{pmatrix} \right)$$

**Exempel: Uttryck vektorer i varandra (2.5.8.b)**

Uttryck  $w$  i  $u$  och  $v$

$$|u| = 6, |v| = 8, |w| = 7, u \text{ och } v \text{ bildar vinkeln } \frac{\pi}{6},$$

$$u \text{ och } w \text{ vinkeln } \frac{\pi}{2} \text{ och } v \text{ och } w \text{ vinkeln } \frac{2\pi}{3}$$

Altså ska hitta  $\vec{w} = k_1 \vec{u} + k_2 \vec{v}$ ,  $k_1, k_2 \in \mathbb{R}$

Vi ser att  $k_2 < 0$  enligt bild, därmed får vi följande

$$-k_2 \cdot \cos \frac{\pi}{3} = 7 \Rightarrow -k_2 \cdot \frac{1}{2} \cdot 8 = 7 \Rightarrow k_2 = \frac{7}{4}$$

Beräknar  $k_1$

$$6k_1 = \frac{7}{4} \cdot 8 \cdot \sin \frac{\pi}{3} \Rightarrow k_1 = \frac{7}{6}\sqrt{3}$$

$$\vec{w} = \frac{7\sqrt{3}}{6}\vec{u} - \frac{7}{4}\vec{v}$$

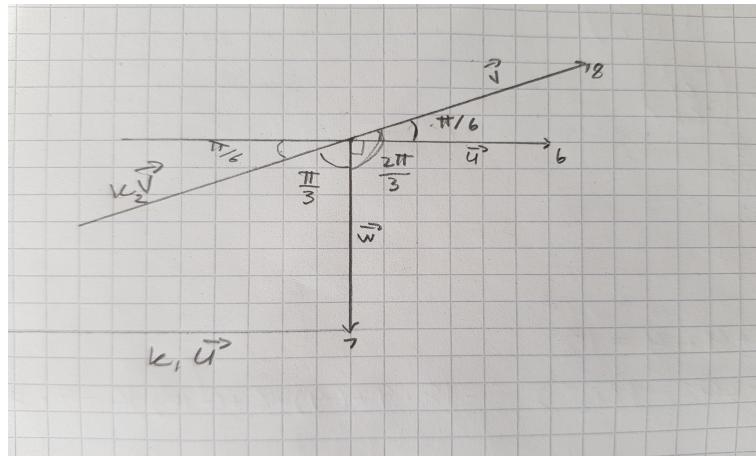


Figure 7.3: 2.5.8.b

### 7.3.5 Area och Volym

Area parralelogram:  $|\vec{u}||\vec{v}| \sin \theta = |\vec{u} \times \vec{v}|$

Volym parralelogram:  $|(\vec{u} \times \vec{v}) \bullet \vec{w}|$

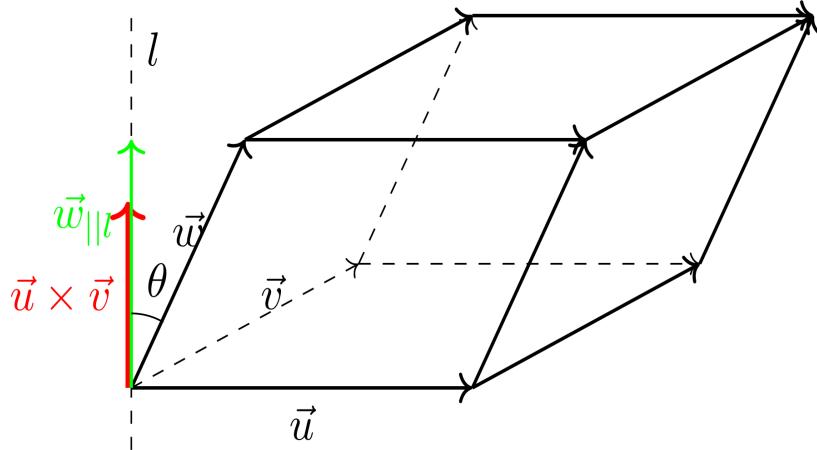


Figure 7.4: Ortogonal projektion

#### Exempel: Hitta volymen

Hitta volymen av parallellipipeden med sidorna

$$\vec{u} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \vec{v} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \vec{w} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

och avgör om  $(\vec{u} \vec{v} \vec{w})$  är ett högersystem, ett vänstersystem, eller ej en bas.

Lösning: Vi beräknar

$$\vec{u} \times \vec{v} = \begin{pmatrix} 1 \cdot 1 - 1 \cdot 1 \\ 1 \cdot 2 - 1 \cdot 1 \\ 1 \cdot 1 - 1 \cdot 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

$$(\vec{u} \times \vec{v}) \bullet \vec{w} = 0 + 1 - 2 = -1$$

Så det är ett vänstersystem och volymen är 1

## 7.4 Linjer och plan

Parameterform:  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + t \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, t \in \mathbb{R}$

Normalform:  $ax + by = c \Rightarrow \vec{n} = \begin{pmatrix} a \\ b \end{pmatrix}$

Plan:  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} + t \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + s \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$

### Exempel: Skriv i parameterform

Om vi löser ekvationen (vars lösningar är en linje):  $x + 3y = 4$ ;

$$y = t \Rightarrow x = 4 - 3t \Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix} + t \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

Hitta parameterform för linje genom givna punkter

$$A = \begin{pmatrix} -21 \\ 20 \end{pmatrix}, B = \begin{pmatrix} -24 \\ -22 \end{pmatrix}$$

$$\overrightarrow{AB} = \overrightarrow{OB} - \overrightarrow{OA} = \begin{pmatrix} -24 - (-21) \\ -22 - 20 \end{pmatrix} = \begin{pmatrix} -3 \\ -42 \end{pmatrix} \Rightarrow$$

$$L : \underline{e} \begin{pmatrix} x \\ y \end{pmatrix} = \underline{e} \begin{pmatrix} -21 \\ 20 \end{pmatrix} + t \underline{e} \begin{pmatrix} -3 \\ -42 \end{pmatrix}$$

### Exempel: Hitta planets ekvation

Uppgift: Hitta en ekvation för planet som går genom  $(1, 2, 3)$  och är parallell med vektorerna

$$\vec{u} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}, \vec{v} = \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

Lösnig: Vi måste hitta en vektor  $\vec{n}$  som är ortogonal mot planet

Inte så lätt i 3D som för linje i 2D. Men vi har en formel.

$$\vec{n} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \times \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 45 - 48 \\ 42 - 36 \\ 32 - 35 \end{pmatrix} = \begin{pmatrix} -3 \\ 6 \\ -3 \end{pmatrix}$$

Det kan vi skriva på normal formen:  $-3x + 6y - 3z = c$

För att hitta  $c$  så insätter vi det kända punkten  $(1, 2, 3)$

$$c = -3 \cdot 1 + 6 \cdot 2 - 3 \cdot 3 = 0$$

Vi får formen:  $-3x + 6y - 3z = 0$

**Exempel: Beskriva linje på normalform**

Uppgift: Hitta normalvektorn för linje  $A = (16, 4)$ ,  $B = (9, 12)$

$$\overline{AB} = \begin{pmatrix} 9 - 16 \\ 12 - 4 \end{pmatrix} = \begin{pmatrix} -7 \\ 8 \end{pmatrix}$$

$$L : \overline{OA} + t\overline{AB} = \begin{pmatrix} 16 \\ 4 \end{pmatrix} + t \begin{pmatrix} -7 \\ 8 \end{pmatrix}$$

$$\begin{cases} x = 16 - 7t \\ y = 4 + 8t \end{cases}$$

$$t = \frac{16 - x}{7} = \frac{y - 4}{8} \Leftrightarrow 128 - 8x = 7y - 21$$

$$8x + 7y = 149$$

**Exempel: Skärning mellan plan**

$$\begin{cases} -3x + y + 4z = 4 \\ x - 4y - 5z = -5 \end{cases}$$

$$\begin{cases} 0 - 11y - 11z = -11 \\ x - 4y - 5z = -5 \end{cases}$$

$$\begin{cases} 0 + y + z = 1 \\ x - 4y - 5z = -5 \end{cases}$$

$$\begin{cases} x + 0 - z = -1 \\ 0 + y + z = 1 \end{cases}$$

$$\begin{pmatrix} t - 1 \\ 1 - t \\ t \end{pmatrix}$$

**Exempel: Skärning mellan plan och linjen**

$$3x + 3y + 4z = -7$$

$$\begin{cases} x = 2 - 3t \\ y = 1 - 3t \\ z = 3t \end{cases}$$

$$3(2 - 3t) + 3(1 - 3t) + 4(3t) = -7 \Rightarrow -6t = -16 \Rightarrow t = \frac{8}{3}$$

$$\begin{cases} x = 2 - 3t = -6 \\ y = 1 - 3t = -7 \\ z = 3t = 8 \end{cases}$$

### 7.4.1 ortogonal projektion på plan

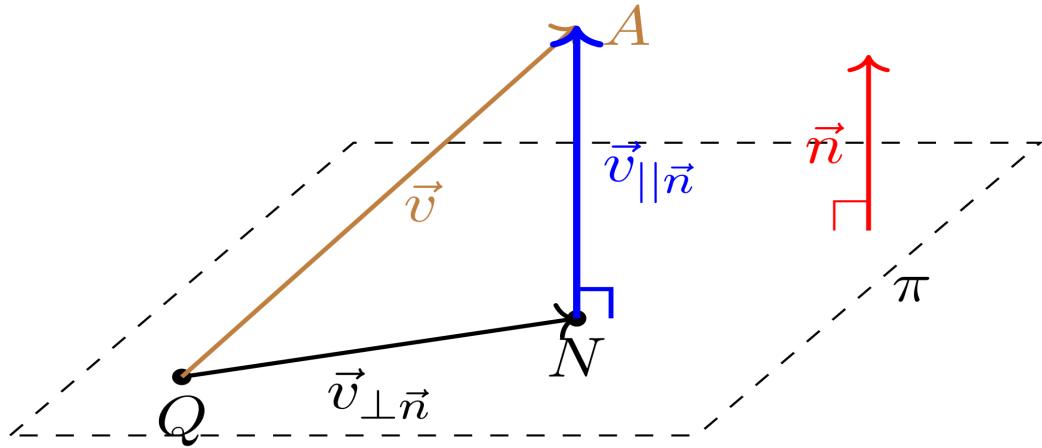


Figure 7.5: Ortogonal projektion på plan

#### Exempel: Hitta närmaste punkten i planet

Uppgift: hitta närmaste punkten till punkten  $A = (1, -5, 2)$   
i planet  $P : x + 2y - z = 1$  Hitta även avståndet mellan  $A$  och planet

Lösning: Först väljer vi godtycklig punkt  $Q = (1, 0, 0)$  i planet, och beräknar

$$\vec{v} = \overrightarrow{QA} = \begin{pmatrix} 1 - 1 \\ -5 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ -5 \\ 2 \end{pmatrix}$$

Normalvektor till planet  $x + 2y - z = 1$  är  $\vec{n} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$

$$\vec{v}_{||\vec{n}} = \frac{0 \cdot 1 + (-5) \cdot 2 + 2 \cdot (-1)}{1^2 + 2^2 + (-1)^2} \vec{n} = -2\vec{n}$$

Så vi får koordinaterna:

$$\overrightarrow{ON} = \overrightarrow{OA} - \overrightarrow{AN} = \overrightarrow{OA} - \vec{v}_{||\vec{n}} = \begin{pmatrix} 1 \\ -5 \\ 2 \end{pmatrix} - (-2)\vec{n} = \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}$$

så närmaste punkten är  $N = (3, -1, 0)$ . Avståndet är  $| -2\vec{n} | = 2\sqrt{6}$

**Exempel: Hitta närmaste punkten på linje till linje**

Hitta den punkt på linjen

$$l : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} + t \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, t \in \mathbb{R}$$

som är närmast linjen

$$k : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} + s \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, s \in \mathbb{R}$$

$$A = (-1 + t, t, t), B = (-1 + s, 2s, -1 + s)$$

$$\overrightarrow{AB} = \begin{pmatrix} (-1 + s) - (-1 + t) \\ (2s) - (t) \\ (-1 + s) - (t) \end{pmatrix} = \begin{pmatrix} s - t \\ 2s - t \\ -1 + s - t \end{pmatrix}$$

Vektorn ska vara ortogonal mot:  $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \wedge \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$

$$\begin{cases} 1(s - t) + 1(2s - t) + 1(-1 + s - t) = 0 \\ 1(s - t) + 2(2s - t) + 1(-1 + s - t) = 0 \end{cases} = \begin{cases} -1 + 4s + -3t = 0 \\ -1 + 6s + -4t = 0 \end{cases}$$

$$\Rightarrow s = \frac{3t + 1}{4} \Rightarrow -1 + \frac{3}{2}(3t + 1) - 4t = 0$$

$$\begin{cases} t = -1 \\ s = -1/2 \end{cases} \text{ (Kontrollera)} -1 - 2 + 3 = 0, -1 - 3 + 4 = 0$$

$$A = (-2, -1, -1), B = (-3/2, -1, -3/2) \Rightarrow \overrightarrow{AB} = \begin{pmatrix} 1/2 \\ 0 \\ -1/2 \end{pmatrix}$$

Svar: punkten på linjen  $l$  är  $A = (-2, -1, -1)$

## 7.5 Matrisräkning

**Begräpp:** matriser

diagonal  
huvuddiagonalen  
Kummuterar:  $AB = BA$   
 $A = (a_{ij})_{r \times k}$   
Rang: antalet ledande kofisenter i trappsteksmatris

**Räkneregler:** matriser

Addition: endast i sama form  
Multiplication:  $(r \times k)(k \times m) \Rightarrow r \times m$   
 $(AB)C = A(BC)$   
 $\lambda(AB) = (\lambda A)B = A(\lambda B)$   
 $A(B + C) = AB + AC$   
 $(B + C)A = BA + CA$

Villkor: Kvadratisk	$\overbrace{\begin{pmatrix} 1 & 1 & 5 \\ 1 & 0 & 2 \\ 2 & -1 & 3 \end{pmatrix}}^B$
	$\underbrace{\begin{pmatrix} 1 & 2 & 0 \\ 3 & -1 & 5 \end{pmatrix}}_A \quad \underbrace{\begin{pmatrix} 3 & 1 & 9 \\ 12 & -2 & 28 \end{pmatrix}}_{AB}$

Figure 7.6: Ortogonal projektion på plan

**Exempel:** Multiplication av matriser

$$\begin{pmatrix} 2 & 3 & 4 \\ -1 & 2 & 2 \\ -5 & -2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 2a & 3b & 4c \\ -a & 2b & 2c \\ -5a & -2b & c \end{pmatrix}$$

**Definition: Enhetsmatrisen**Enhetsmatrisen  $I_n$ 

$$A^0 I_n = I_n$$

$$A : 2 \times 2 A I_2 = A$$

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

**7.5.1 Transponat****Definition: Transponaten** $A^t$  betyder inte  $A$  upphöjt till  $t$ 

$$A = (a_{ij})_{r \times k} \Rightarrow A^t = A = (\alpha_{ij})_{k \times r}$$

$$A = (\alpha_{ij}) = (a_{ij})$$

**Exempel: Transponaten**Symmetrisk  $A = A^t$ 

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}^t = \begin{pmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{pmatrix}$$

**Räkneregler: Transponaten**

$$(A + B)^t = A^t + B^t$$

$$(\lambda A)^t = \lambda A^t$$

$$(A^t)^t = A$$

$$(AB)^t = B^t A^t$$

**7.5.2 Matrisinvers****Räkneregler: Matrisinvers**

$$AB = BA = I$$

Inversen finns endast om matrisen är kvadratisk

 $A$  är inventerba $AX = B$  har entydiga lösningar för alla  $B$  $AX = 0$  har enbart lösningen  $X = 0$  $A$  har rang $A \sim I$  (är radekvivalent med)  $I$

**Exempel:** Matrisinvers 3x3

$$\left( \begin{array}{ccc} 1 & 2 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & -1 \end{array} \right) \sim \left( \begin{array}{ccc|ccc} 1 & 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \end{array} \right) \sim \left( \begin{array}{ccc|ccc} 1 & 2 & 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \end{array} \right) \sim$$

$$\left( \begin{array}{ccc|ccc} 1 & 0 & -1 & -1 & 2 & 0 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & 1 & 1 & 1/2 & -1/2 & -1/2 \end{array} \right) \sim \left( \begin{array}{ccc|ccc} 1 & 0 & 0 & -1/2 & 3/2 & -1/2 \\ 0 & 1 & 0 & 1/2 & -1/2 & 1/2 \\ 0 & 0 & 1 & 1/2 & -1/2 & -1/2 \end{array} \right) \sim$$

$$A^{-1} = \left( \begin{array}{ccc} -1/2 & 3/2 & -1/2 \\ 1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 \end{array} \right) = \frac{1}{2} \left( \begin{array}{ccc} -1 & 3 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{array} \right)$$

**Räkneregler:** Matrisinvers

$$A = \left( \begin{array}{cc} a & b \\ c & d \end{array} \right)$$

$$A^{-1} = \frac{1}{ad - bc} \left( \begin{array}{cc} d & -b \\ -c & a \end{array} \right)$$

## 7.6 Determinanter

Ekvations system har unika lösningar då koefficient matrisen är inventerbar. Som är ekvivalent med att determinanten är nollskild.

**Exempel: determinant många led genväg**

$$\begin{pmatrix} 3 & 2 & -3 & 24 & 1005 \\ 0 & 1 & 23 & 14 & 15 \\ 0 & 0 & 3 & 7 & -5 \\ 0 & 0 & 2 & 4 & 3 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix}$$

Vi det finns endast två produkter som inte innehåller en nol faktor

$$\begin{pmatrix} (3) & 2 & -3 & 24 & 1005 \\ 0 & (1) & 23 & 14 & 15 \\ 0 & 0 & (3) & (7) & -5 \\ 0 & 0 & (2) & (4) & 3 \\ 0 & 0 & 0 & 0 & (5) \end{pmatrix} = 3 \cdot 1 \cdot 3 \cdot 4 \cdot 5 - 3 \cdot 1 \cdot 6 \cdot 2 \cdot 5 = 0$$

**Exempel: determinant 3x3**

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} &= \\ &= 1 \cdot 5 \cdot 9 - 1 \cdot 6 \cdot 8 - 2 \cdot 4 \cdot 9 + \\ &\quad + 2 \cdot 6 \cdot 7 + 3 \cdot 4 \cdot 8 - 3 \cdot 5 \cdot 7 = \\ &= 45 - 48 - 72 + 84 + 96 - 105 = 0 \end{aligned}$$

**Sats:**

Om  $B$  är matrisen  $A$  där man har bytt om på rad  $i$  och  $j$  är  $\det A = -\det B$

- (1) Bytt två rader om i  $A$  ändras determinaten sitt teken
- (2) Skalas en rad om med  $\lambda$  skalas också determinaten om med  $\lambda$
- (3) Addera en rad något på en annan rad i  $A$  ändrar inte denna determinant
- (4)  $\det(A) = \det(A^t)$
- $\det(AB) = \det(A)\det(B)$

**Exempel: determinant 3x3**

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 3 & 8 & 7 \\ 1 & 3 & 6 & 5 \\ 0 & 0 & 2 & 3 \end{pmatrix} &= - \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 1 \\ 0 & 3 & 8 & 7 \\ 0 & 0 & 2 & 3 \end{pmatrix} = - \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 2 & 3 \end{pmatrix} \\ &- \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 11 \end{pmatrix} = -1 \cdot 1 \cdot (-1) \cdot 11 = 11 \end{aligned}$$

**Exempel:** okännt x (determinant)

$$\begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ x & x^2 & x^4 & x^5 \\ x^2 & x^3 & x^4 & x^6 \end{pmatrix}$$

$$\begin{aligned} x \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ x^2 & x^3 & x^4 & x^6 \end{pmatrix} &= \text{rad 4 -rad 1} x^3 \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ 1 & x & x^2 & x^4 \end{pmatrix} = x^3 \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ 0 & 0 & 0 & x \end{pmatrix}^{R4} = \\ \text{rad 3 -rad 1} x^3(x^4 - x^3) \begin{pmatrix} 1 & x & x^2 \\ 1 & x^2 & x^3 \\ 1 & x & x^3 \end{pmatrix} &= x^6(x-1)^2 \begin{pmatrix} 1 & x & x^2 \\ 1 & x^2 & x^3 \\ 0 & 0 & x \end{pmatrix} = \\ x^6(x-1)(x^3 - x^2) \begin{pmatrix} 1 & x \\ 1 & x^2 \end{pmatrix} &= x^9(x-1)^3 = 0 \end{aligned}$$

### 7.6.1 Ko-faktorna

$\det(A) = a_{i1}C_{i1} + a_{i2}C_{i2} + \dots + a_{in}C_{in}$  där  $a_{ij}$  är teknet o  $C_{ij}$  är kofisienten

$\tilde{A}^t$  är alla kofaktorererna från  $A$

$\det(a^{-1}) = \det(1/\det(a))$

$$\begin{aligned} \begin{pmatrix} 2 & 1 & 1 & -1 \\ (1) & (0) & (2) & (0) \\ 1 & -2 & 1 & 2 \\ 3 & 1 & -1 & 5 \end{pmatrix}^{R2} &= (-1)^{2+1} \cdot 1 \cdot \begin{pmatrix} 1 & 1 & -1 \\ -2 & 1 & 2 \\ 1 & -1 & 5 \end{pmatrix} \\ + (-1)^{2+2} \cdot 0 \cdot \begin{pmatrix} 2 & 1 & -1 \\ 1 & 1 & 2 \\ 3 & -1 & 5 \end{pmatrix} + (-1)^{2+3} \cdot 2 \cdot \begin{pmatrix} 2 & 1 & -1 \\ 1 & -2 & 2 \\ 3 & 1 & 5 \end{pmatrix} + (-1)^{2+4} \cdot 0 \cdot \begin{pmatrix} 2 & 1 & 1 \\ 1 & -2 & 1 \\ 3 & 1 & -1 \end{pmatrix} \\ - 1(5 + 2 - 2 - (-1 - 2 - 10)) + 0 - 2(-20 + 6 - 1 - (6 + 4 + 5)) + 0 = -18 - 2 \cdot (-30) = 42 \end{aligned}$$

### 7.6.2 Geometri: parallelepiped

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \bullet \left( \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \right)$$

## 7.7 Vektorer i $\mathbb{R}^n$

Skallär produkt och ärmad längd är samma regler som innan

Vektor mellan punkter är också samma regler

Vinkeln är samma som innan (orthogonal projection)

$$\text{Vinkel formel: } \theta = \arccos \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|}$$

(Cauchy-Schwarz olikheten) logist!  $|\vec{v} \bullet \vec{w}| \leq |\vec{v}| |\vec{w}| (|3 - 1| \leq |3| - 1)$

Linjärt oberoende: inga parametrar vid lösning av  $c_1 \vec{v}_1 + c_2 \vec{v}_2 + \dots + c_n \vec{v}_n = \vec{0}$

Linjära Höljet (spannet): mängden av alla möjliga linjärakombinationer av vektorerna

Det linjära höljet av vektorerna är hela  $\mathbb{R}^n$  omm  $\text{Rang } V = n$

Bas definnerar lika dant i fler dimensioner, för att vara en bas  $\mathbb{R}^n$

måste de vara linjärt oberoende och det linjära höljet är hela  $\mathbb{R}^n$

## 7.8 Linjära avbildningar $\mathbb{R}^n \rightarrow \mathbb{R}^m$

### 7.8.1 Matristransformationer och linjära funktioner

#### Termenologi

Standardmatris: En matris som multipliseras med argumentet för att få svaret

Sammansättning: Låt  $f : A \rightarrow B, g : B \rightarrow C$

$$(g \circ f)(x) = g(f(x))$$

Vektor produkt:  $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

$$F(\vec{x}) = \vec{a} \times \vec{x}$$

**Definition:** En funktion  $T : \mathbb{R}^k \rightarrow \mathbb{R}^n$  kallas linjär om den uppfyller:

För alla  $\vec{v}, \vec{w} \in \mathbb{R}^k$  och  $\lambda \in \mathbb{R}$  gäller

$$(i) \quad T(\vec{v} + \vec{w}) = T(\vec{v}) + T(\vec{w})$$

$$(ii) \quad T(\lambda \vec{v}) = \lambda T(\vec{v})$$

**Formel standard matris**

$$[T] \begin{pmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_k \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ T(\vec{v}_1) & T(\vec{v}_2) & \dots & T(\vec{v}_k) \\ | & | & & | \end{pmatrix}$$

**Exempel: hitta standard matris**

Hitta standardmatrisen  $[T]$  för den linjära funktionen  $T : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  som uppfyller att

$$T \begin{pmatrix} 0 \\ -4 \end{pmatrix} = \begin{pmatrix} 8 \\ 8 \\ -8 \end{pmatrix} \wedge T \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ -4 \end{pmatrix}$$

$$[T] \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 8 & -1 \\ 8 & 0 \\ -8 & -4 \end{pmatrix}$$

$$[T] = \begin{pmatrix} 8 & -1 \\ 8 & 0 \\ -8 & -4 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}^{-1} = \begin{pmatrix} 3 & -2 \\ 2 & -2 \\ 2 & 2 \end{pmatrix}$$

Kontrol: multiplisera standard matrisen med input och få output

**Exempel: hitta standard matris ortogonal projection**

Hitta standardmatrisen för  $P : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

där  $P$  är den ortogonala projektionen på planet  $\pi : 2x + y + 3z = 0$

$$T(\vec{n}) = T \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Hittar två godtykliga punkter på planet:  $A = (1, 1, -1)$ ,  $B = (-1, 2, 0)$

$$\overrightarrow{OA} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \overrightarrow{OB} = \begin{pmatrix} -1 \\ 2 \\ 0 \end{pmatrix}$$

$$[T] \begin{pmatrix} 3 & 1 & -1 \\ 1 & 1 & 1 \\ 2 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \Rightarrow$$

$$[T] = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 1 & -1 \\ 1 & 1 & 1 \\ 2 & -1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 5/7 & -1/7 & -3/7 \\ -1/7 & 13/14 & -3/14 \\ -3/7 & -3/14 & 5/14 \end{pmatrix}$$

Kontrol: multiplisera standard matrisen med input och få output

**Exempel: hitta standard matris spegling**

Hitta standardmatrisen för  $P : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

där  $P$  är speglingen i planet  $\pi : -x + 2y - 2z = 0$

$$T(\vec{n}) = T \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix}$$

Hittar två godtykliga punkter på planet:  $A = (0, 1, 1)$ ,  $B = (2, 1, 0)$

$$\overrightarrow{OA} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \overrightarrow{OB} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

$$[T] \begin{pmatrix} -1 & 0 & 1 \\ 2 & 1 & 2 \\ -2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix} \Rightarrow$$

$$[T] = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \\ 2 & 1 & 2 \\ -2 & 1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 7/9 & 4/9 & -4/9 \\ 4/9 & 1/9 & 8/9 \\ -4/9 & 8/9 & 1/9 \end{pmatrix}$$

Kontrol: multiplisera standard matrisen med input och få output

### 7.8.2 Injektiv/Surjektiv/Bijektiv

Termenologi

[T]	kolonnvektorerna i [T]	funktionen T
$\text{rang}([T])=k$	linjär oberoende	injektiv
$\text{rang}([T])=n$	spannet är hela $\mathbb{R}^n$	surjektiv
$n=\text{rang}([T])=k$	är en bas för $\mathbb{R}^n = \mathbb{R}^k$	bijektiv



## Chapter 8

# Linear Algebra II

## 8.1 Grudläggande teori

### 8.1.1 Ekvationssystem och matris räkning

$$\left\{ \begin{array}{l} x + 2y + z = -1 \\ 2x + (a+3)y + 3z = -4 \\ x + (3-a)y + (a-2)z = a-1 \end{array} \right.$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 2 & a+3 & 3 & -4 \\ 1 & 3-a & a-2 & a-1 \end{array} \right)$$

(-1 rad1 till rad2), (-2 rad1 till rad3)

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 1-a & a-3 & a \end{array} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 0 & a-2 & a-2 \end{array} \right)$$

$a \neq 1 \wedge a \neq 2$

$$\left( \frac{1}{a-2} \text{rad 3} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & a-1 & 1 & -2 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

(-1rad 3 till rad 2), (-1rad 3 till rad 1)

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & a-1 & 0 & -3 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$\left( \frac{1}{a-1} \text{rad 2} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & a-1 & 0 & -3 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$\left( \begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & 1 & 0 & \frac{3}{1-a} \\ 0 & 0 & 1 & 1 \end{array} \right)$$

(-2rad 2 till rad 3)

$$\left( \begin{array}{ccc|c} 1 & 0 & 0 & -2 - \frac{6}{1-a} \\ 0 & 1 & 0 & \frac{3}{1-a} \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$\left\{ \begin{array}{l} x = -2 - \frac{6}{1-a} \\ y = \frac{3}{1-a} \\ z = 1 \end{array} \right.$$

**Kontroll:** stoppar in x,y,z i ekvationerna

$$\left\{ \begin{array}{l} \left( \frac{2a-8}{1-a} \right) + 2 \cdot \left( \frac{3}{1-a} \right) + 1 = -1 \\ 2 \cdot \left( \frac{2a-8}{1-a} \right) + (a+3) \cdot \left( \frac{3}{1-a} \right) + 3 \cdot 1 = -4 \\ \left( \frac{2a-8}{1-a} \right) + (3-a) \cdot \left( \frac{3}{1-a} \right) + (a-2) \cdot 1 = a-1 \end{array} \right.$$

### 8.1.2 determinanter

$$\begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ x & x^2 & x^4 & x^5 \\ x^2 & x^3 & x^4 & x^6 \end{pmatrix}$$

$$\begin{aligned} x \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ x^2 & x^3 & x^4 & x^6 \end{pmatrix} &= \text{rad 4 -rad 1} x^3 \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ 1 & x & x^2 & x^4 \end{pmatrix} = x^3 \begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & x^2 & x^3 & x^4 \\ 1 & x & x^3 & x^4 \\ 0 & 0 & 0 & x \end{pmatrix}^{R4} = \\ \text{rad 3 -rad 1} x^3(x^4 - x^3) \begin{pmatrix} 1 & x & x^2 \\ 1 & x^2 & x^3 \\ 1 & x & x^3 \end{pmatrix} &= x^6(x-1)^2 \begin{pmatrix} 1 & x & x^2 \\ 1 & x^2 & x^3 \\ 0 & 0 & x \end{pmatrix} = \\ x^6(x-1)(x^3 - x^2) \begin{pmatrix} 1 & x \\ 1 & x^2 \end{pmatrix} &= x^9(x-1)^3 = 0 \end{aligned}$$

### 8.1.3 flerdimisionel dvs $R^n$

räkneregler

### 8.1.4 Funktioner

polynom funktioner vid en viss grad också

### 8.1.5 Linjer

$$l : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 5 \end{pmatrix} + t \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

Då är riktnings vektorn  $\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$

Och går genom  $\begin{pmatrix} 3 \\ 2 \\ 5 \end{pmatrix}$

## 8.2 Vektorrum

### Definition: vektor rum

En mängd  $\mathbb{V}$  kallas för en reellt vektorrum om

1. Det finns en operator på  $\mathbb{V}$  som kallas addition och beräknas med  $+$ , sådant att om  $\vec{u}, \vec{v} \in \mathbb{V}$  så gäller  
 $\vec{u} + \vec{v} \in \mathbb{V}$
2. Det finns en poeration på  $\mathbb{V}$  som kallas skalning eller multipliseras med reella tal, som betecknas med  $\cdot$ , sådan att om  $\lambda \in \mathbb{R} \wedge \vec{v} \in \mathbb{V}$  så gäller  $\lambda \cdot \vec{v} \in \mathbb{V}$

räknereglerna gäller som i la1

### Axiomen: vektor rum

1.  $\vec{u} + \vec{v} = \vec{v} + \vec{u}$  (Kommutativ lag)
  2.  $\vec{u}(\vec{v} + \vec{w}) = (\vec{u} + \vec{v}) + \vec{w}$  (Associativ lag)
  3. Det finns ett nollement  $\vec{0}$  så att  $\vec{v} + \vec{0} = \vec{v}$
  4. Till varje  $\vec{v} \in \mathbb{V}$  finns ett element  $-\vec{v}$  så att  
 $\vec{v} + (-\vec{v}) = 0$
  5.  $1 \cdot \vec{v} = \vec{v}$
  6.  $\lambda \cdot (\mu \cdot \vec{v}) = (\lambda \cdot \mu) \cdot \vec{v}$  (Associativ lag)
  7.  $(\lambda + \mu) \cdot \vec{v} = \lambda \cdot \vec{v} + \mu \cdot \vec{v}$  (Distributiv lag)
  8.  $\lambda \cdot (\vec{u} + \vec{v}) = \lambda \cdot \vec{u} + \lambda \cdot \vec{v}$  (Distributiv lag)
- $\forall \lambda, \mu \in \mathbb{R} \wedge \vec{u}, \vec{v}, \vec{w} \in \mathbb{V}$

## 8.3 Underrum och linjära höljet

Ett underrum är en delmängd  $U$  ej  $\emptyset$  som är sluten under addition och skalning

**Definition: Underrum**

En delmängd  $U$  av ett vektorrum  $V$  kallas för  
ett underrum eller delrum av  $V$  om  $U$  är ett vektorrum  
med den addition och den multiplikation med reella tal som definierats i  $V$ .

**Sats: Underrum**

En icketom mängd  $U$  av ett vektorrum  $V$  är ett  
underrum om och endast om följande gäller  
 1. Om  $\vec{u} \in U$  och  $\vec{v} \in U$ , Så är  $\vec{u} + \vec{v} \in U$   
 2. Om  $\vec{u} \in U$  och  $\lambda \in \mathbb{R}$ , Så är  $\lambda\vec{u} \in U$

**Regler: Underrum**

1. Det snabbaste sättet att testa om det gäller är om nollvektorn finns  
om den inte gör det så bryter det mot 2-lagen
2. Homogena ekvationer och linjer/plan genom origo gäller det alltid för
3. Kontunerligt deriverbara, så gäller det att det är ett underrum

**Exempel: Underrum (Ex 1)**

är  $\left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mid x + y + z = 5 \right\}$  ett underrum?

Nej, då:  $0 + 0 + 0 \neq 5$

**Definition: linjära hörjet**

En delmängd  $U$  av ett vektorrum  $V$  kallas för ett underrum eller delrum av  $V$  om  $U$  är ett vektorrum med den addition och den multiplikation med reella tal som definierats i  $V$ .

**Exempel: Underrum (Ex 4)**

Vilka vektorer i  $P$  tillhör  $u = [1, 1+x, 1+x+x^2]$

Svar:  $u = P_2$  (eftersom)  $1 \in u, x \in u$ , och  $x^2 \in u$

Då alla  $a \cdot 1 + b \cdot x + c \cdot x^2 \in u$

Inga polynom av grad  $> 2$  tillhör  $u$ .

Vi försöker kombinera de olika elementen i  $u$  med addition och multiplication

## 8.4 Linjärt oberoende

**Ide:** linjärt beroende

Om vektorerna kan skrivas om en linjär kombination av det andra vektorerna  
så är vektorn linjärt beroende då det inte behöver vektorn

vi får reda på beroende genom att ställa upp ekv

$$x_1 \vec{v}_1 + x_2 \vec{v}_2 + \dots + x_n \vec{v}_n = \vec{0}$$

Om denna ekvation har icke triviala lösningar (ej noll) då är den beroende

**Exempel: om linjärt oberonnde**

Är mängden  $\left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}$  i  $\mathbb{R}^2$  linjärt oberoende

$$x_1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} + x_2 \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \vec{0} \Leftrightarrow \left( \begin{array}{cc|c} 1 & 2 & 0 \\ 2 & 1 & 0 \end{array} \right)$$

$x_1 = 0, x_2 = 0$  Endast triviala lösningar

## 8.5 Bas

### Ide: Bas och Dimention

Bas omm om vektorerna spanner upp hella spannet och det linjärt oberonde  
Standard basen  $e$

Vi behöver tree vektorer för att utgöra en bas i  $\mathbb{R}^3$  är  $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in \mathbb{R}^3$

Dimentioner är antallet oberonde vektorer som spänner up "rummet"  
"dimentioner blir då samma som antallet element som en bas av polynom"

$\mathbb{P}_3 = a + bx^1 + cx^2 + dx^3$  har 4 dimentioner

$\mathbb{R}^3$  har 3 dimentioner

Dimentioner för matriser är  $n \times m$  ( $2 \times 2 - matris = dim4$ )

En bas med tre vektorer där vektorerna  $\mathbb{R}^4$  ger 3 dim

### Exempel: Ta fram bas från plan

låt  $2x - yz = 0$

Lösning:  $x = y/2 - z/2$

Låt  $y = s, z = t$  vi får då lösningen på parameter form

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} s/2 - t/2 \\ s \\ t \end{pmatrix} = s/2 \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + t/2 \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$$

$$\text{Basen blir då } \left( \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} \right)$$

## 8.6 Basomvandling

Vi kan ta inversen av matrisen av vektorerna som utger en bas och få matrisen som kan multipliseras med vektor för att få omvandlingen.

**Sats: basbyte**

$$\vec{v}_e = T \vec{v}_f \text{ Där } T \text{ är en matris}$$

$$T_{\underline{e}}^{\underline{f}} = (T_{\underline{f}}^{\underline{e}})^{-1}$$

$$\vec{v}_f = T_{\underline{f}}^{\underline{e}} \vec{v}_e$$

**Exempel: Från standar till annan bas**

$$\text{Låt } \underline{f} = (f_1 \ f_2) = \left( \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \right), \vec{v} = \begin{pmatrix} -2 \\ 5 \\ 1 \end{pmatrix}$$

Utryck  $\vec{v}$  i basen  $\underline{f}$

$$\begin{aligned} \vec{v}_e &= \begin{pmatrix} -2 \\ 5 \\ 1 \end{pmatrix} = x_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + x_2 \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \\ \left\{ \begin{array}{l} 2x_1 + x_2 = -2 \\ x_1 + 2x_2 = 5 \\ x_1 + x_2 = 1 \end{array} \right. &\Rightarrow \left( \begin{array}{cc|c} 2 & 1 & -2 \\ 1 & 2 & 5 \\ 1 & 1 & 1 \end{array} \right) \sim \left( \begin{array}{cc|c} 2 & 1 & -2 \\ 0 & 1 & 4 \\ 0 & 0 & 0 \end{array} \right) \\ x_2 &= 4, \quad x_1 = 1/2(-2 - 4) = -3 \Rightarrow \vec{v}_f = \begin{pmatrix} -3 \\ 4 \end{pmatrix} \end{aligned}$$

**Exempel: Finn matrisen för bas byte vektorer**

$$\text{Låt } \underline{f} = \left( \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right) \text{ Bestäm bas omvandling matrisen}$$

$$\text{Lös } T_{\underline{f}}^{\underline{e}} = \left( \begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array} \right) \sim \left( \begin{array}{cc|cc} 1 & 0 & -1 & 2 \\ 0 & 1 & 1 & -1 \end{array} \right)$$

**Exempel: Finn matrisen för bas byte polynom**

$$\text{Låt } \underline{e} = (1 \ x) \text{ och } \underline{f} = (1 \ 1-x)$$

$$\vec{f}_1 = 1 = \vec{e}_1 \Rightarrow \vec{f}_{1e} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\vec{f}_2 = 1-x = \vec{e}_1 - \vec{e}_2 \Rightarrow \vec{f}_{2e} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$T_{\underline{e}}^{\underline{f}} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \Rightarrow (T_{\underline{e}}^{\underline{f}})^{-1} T_{\underline{f}}^{\underline{e}} =$$

$$\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$$

**Exempel:** Finn matrisen för bas byte mellan olika matriser

Basbyte mellan  $\underline{f} = \left( \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right)$  och  $\left( \begin{pmatrix} -1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right)$

Lös  $T_{\underline{f}}^g$ :  $\left( \begin{array}{cc|cc} 1 & 2 & -1 & 3 \\ 1 & 1 & 1 & 2 \end{array} \right) \sim$

$$\left( \begin{array}{cc|cc} 1 & 0 & 3 & 1 \\ 0 & 1 & -2 & 1 \end{array} \right) \Rightarrow \left( \begin{array}{cc} 3 & 1 \\ -2 & 1 \end{array} \right)$$

**Definition:** Nollrum, Kolonrum, radrum.

låt A vara en  $m \times n$ -matris

1.  $\dim(A:\text{s kolonrum}) = \dim(A:\text{s radrum}) = \text{rang } A$
2.  $\dim(A:\text{s kolonrum}) + \dim(A:\text{s nollrum}) = n$

**Exempel:** Finn Nollrum, Kolonrum, radrum.

Hitta en bas för kolonrummet, radrummet och nollrumet till följande matris. Bestäm även rummens dimensioner

$$A = \begin{pmatrix} 1 & 2 & 3 & 3 \\ 1 & 3 & 4 & 5 \\ -1 & 0 & -1 & 1 \end{pmatrix}$$

Låt kolonernana vara vektorer, då får vi följande gäller

$$x_1v_1 + x_2v_2 + x_4v_3 + x_4v_4 = 0 \Leftrightarrow (v_1 \ v_2 \ v_3 \ v_4) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0 \Leftrightarrow Ax = 0$$

Gausselimination ger oss  $A \sim \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -s+t \\ -s-2t \\ s \\ t \end{pmatrix} = s \begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \end{pmatrix} + t \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

Basen för nollrummet blir då  $\left( \begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix} \right)$

Basen för kolonrummet blir då det vektorer i A som har Pivot element dvs  $v_1, v_2$

$$\left( \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix} \right)$$

Basen för radrummet blir  $\left( \begin{pmatrix} 1 \\ 0 \\ 1 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 2 \end{pmatrix} \right)$

## 8.7 Linjär avbildning

**Definition:** Linjär avbildning

låt  $\mathbb{V}$  och  $\mathbb{W}$  vara vektorrum. En funktion  $F : \mathbb{V} \rightarrow \mathbb{W}$  kallas för linjär avbildning omm

1.  $F(\vec{v} + \vec{w}) = F(\vec{v}) + F(\vec{w})$
2.  $F(\lambda \cdot \vec{v}) = \lambda \cdot F(\vec{v})$

**Exempel:** Derivatan av polynom

Är följande en linjär avbildning

$$F : C^1(a, b) \rightarrow C(a, b) \text{ definierad genom } F(f) = \frac{df}{dx}$$

$$\left( \frac{d}{dx}(af(x) + bg(x)) \right) = a \frac{df}{dx} + b \frac{dg}{dx}$$

$$\left( \frac{d}{dx}(\lambda af(x)) \right) = \lambda \left( \frac{d}{dx}(af(x)) \right)$$

Svar: ja  $F$  är en linjär avbildning

## 8.8 Matrisen av en linjär avbildning

$$\begin{array}{ccc}
 \mathbb{V} & \xrightarrow{F} & \mathbb{W} \\
 \underline{\mathbf{v}} \cdot ? \uparrow & & \downarrow (?)_{\underline{\mathbf{w}}} \\
 \mathbb{R}^n & \xrightarrow{(F)_{\underline{\mathbf{w}}}^{\underline{\mathbf{v}}} } & \mathbb{R}^m
 \end{array}$$

$$(F)_{\underline{\mathbf{w}}}^{\underline{\mathbf{v}}}(\mathbf{x})_{\underline{\mathbf{v}}} = (F(\underline{\mathbf{v}}(\mathbf{x})_{\underline{\mathbf{v}}})_{\underline{\mathbf{w}}} = (F(\mathbf{x}))_{\underline{\mathbf{w}}}$$

Figure 8.1: Matris av en linjär avbildning

**Exempel: Rotations matrisen**

$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

**Exempel: Hitta matrisen**

Linjär avbildning  $F : \mathbb{P}_3 \rightarrow \mathbb{P}_3$ ,  $F = p + p' + p'' + p'''$  Ange  $F$ :s matris i standarbasen.

Avbildar varje ellement i standardbasen  $(1 \ x \ x^2 \ x^3)$

$$F(1) = 1 + 1' + 1'' + 1''' = 1$$

$$F(x) = x + x' + x'' + x''' = x + 1$$

$$F(x^2) = x^2 + x^{2'} + x^{2''} + x^{2'''} = x^2 + 2x + 2 \cdot 1$$

$$F(x^3) = x^3 + x^{3'} + x^{3''} + x^{3'''} = x^3 + 3x^2 + 6x + 6 \cdot 1$$

$$\begin{pmatrix} 1 & 1 & 2 & 6 \\ 0 & 1 & 2 & 6 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## 8.9 Basbyte av linjära avbildningar

**Definition: Basbyte av linjära avbildningar**

låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara linjär. Låt  $\underline{e}$  och  $\underline{v}$  vara baser i  $\mathbb{V}$  och låt  $\underline{f}$

och  $\underline{w}$  vara baser av  $\mathbb{W}$ . Då gäller

$$(F)_{\underline{f}}^{\underline{e}} = T_{\underline{f}}^{\underline{w}}(F)_{\underline{w}}^{\underline{v}}T_{\underline{v}}^{\underline{e}}$$

Där vektorn kommer från höger (viktigt vid vilken ordning transformations matriserna står)

Ide: Vi omvandlar från en bas (ex standard bas) till en bas som är mer anpassat för uträkningen  
Sedan omvandlar vi igen för att få svaret i den bas vi vill ha den i (ex standard bas)

**Exempel: Basbyte av linjär avbildning vektor**

**Exempel: Basbyte av linjär avbildning polynom**

Vad är matrisen av derivatan  $\mathbb{P}_3 \rightarrow \mathbb{P}_2$  med avseende på  
basen  $\underline{u} = (x^3 \ x^2 \ x \ 1)$  av  $\mathbb{P}_3$  och  $\underline{v} = (x^2 \ x \ 1)$  av  $\mathbb{P}_2$ ?

$$(H)_{\underline{f}}^{\underline{e}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

$$T_{\underline{v}}^{\underline{f}} :$$

$$\vec{f}_1 = 1 = 0\vec{v}_1 + 0\vec{v}_2 + 1\vec{v}_3$$

$$\vec{f}_2 = x = 0\vec{v}_1 + 1\vec{v}_2 + 0\vec{v}_3$$

$$\vec{f}_3 = x^2 = 1\vec{v}_1 + 0\vec{v}_2 + 0\vec{v}_3$$

$$T_{\underline{e}}^{\underline{u}} :$$

$$\vec{u}_1 = x^3 = 0\vec{e}_1 + 0\vec{e}_2 + 0\vec{e}_3 + 1\vec{e}_4$$

$$\vec{u}_2 = x^2 = 0\vec{e}_1 + 0\vec{e}_2 + 1\vec{e}_3 + 0\vec{e}_4$$

$$\vec{u}_3 = x = 0\vec{e}_1 + 1\vec{e}_2 + 0\vec{e}_3 + 0\vec{e}_4$$

$$\vec{u}_4 = 1 = 1\vec{e}_1 + 0\vec{e}_2 + 0\vec{e}_3 + 0\vec{e}_4$$

$$(H)_{\underline{v}}^{\underline{u}} = T_{\underline{v}}^{\underline{f}}(H)_{\underline{f}}^{\underline{e}}T_{\underline{e}}^{\underline{u}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

## 8.10 Kärna och bild av en linjär avbildning

**Definition:** Kärna och bild

Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara en linjär avbildning

**Kärnan** eller **nollrummet** av den linjära avbildningen F

$$\ker(F) = N(F) = \{\vec{v} \in \mathbb{V} | F(\vec{v}) = \vec{0}\}$$

**Bilden** eller värdерummet av den linjära avbildningen F

$$\text{Im}(F) = V(F) = \{\vec{w} \in \mathbb{W} | \exists \vec{v} : F(\vec{v}) = \vec{w}\}$$

**Definition:** Isomorfism, injektiv, surjektiv

$F : X \rightarrow y$  kallas

**injektiv** om  $F(x) = F(x') \Rightarrow x = x'$

**surjektiv** om  $\forall y \in Y \exists x \in X : f(x) = y$

**bijektiv** om det är både injektiv och surjektiv

**Sats:**

Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara linjär

$F$  är injektiv omm  $\ker F = \{\vec{0}\}$

$F$  är surjektiv omm  $\text{Im } F = \mathbb{W}$  omm  $\text{rang}((F)_{\underline{w}}^{\underline{v}}) = \dim \mathbb{W}$

**Sats:** Dimensionssatsen

För varje linjär avbildning  $F : \mathbb{V} \rightarrow \mathbb{W}$  gäller

$$\dim \mathbb{V} = \dim \ker(F) + \dim \text{Im}(F)$$

**Sats:** Isomorfism

Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara en linjär avbildning. De följande är ekvivalenta

$F$  är en isomorf

$\dim \mathbb{V} = \dim \mathbb{W}$  och  $F$  är injektiva

$\dim \mathbb{V} = \dim \mathbb{W}$  och  $F$  är surjektiva

$$\det((F)_{\underline{w}}^{\underline{v}}) \neq 0$$

$(F)_{\underline{w}}^{\underline{v}}$  är en kvadratisk matris av rang  $\dim \mathbb{V} = \dim \mathbb{W}$

## 8.11 Egenvärden och egenvektorer

**Definition:** Egenvärde och egenvektor

Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara en linjär avbildning. En vektor  $\vec{v} \neq \vec{0}$  kallas en **egenvektor** till  $F$  om det finns  $\lambda \in \mathbb{R}$  så att  $F(\vec{v}) = \lambda\vec{v}$ . I detta fall kallas  $\lambda$  en **egenvärde** till  $F$ .  $\mathbb{V}_{F,\lambda} = \{\vec{v} \in \mathbb{V} | F(\vec{v}) = \lambda\vec{v}\}$  kallas **egenrummet** av  $F$  till egenvärdet  $\lambda$ .

**Definition:** Sekularpolynomet

För en matris  $A$  kallas polynomet  $\chi_A(\lambda) = \det(A - \lambda I_n)$

**sekularpolynomet** till  $A$

Om  $A = (F)_{\underline{e}}^e$ , så kallas  $\chi_A$  också sekularpolynomet till  $F$

**Exempel:**

$$F : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \text{ matrisen i standardbasen är } \begin{pmatrix} 1 & 2 & 2 \\ 0 & 2 & 1 \\ -1 & 2 & 2 \end{pmatrix}$$

1. Hittar egenvärderna

$$F(\vec{x}) = \lambda \vec{x} \Leftrightarrow A\vec{x} = \lambda \vec{x}$$

$$A\vec{x} - \lambda \vec{x} = \vec{0} \Leftrightarrow (A - \lambda I)\vec{x} = 0$$

Där  $\lambda$  är egenvärdet, och  $\vec{x}$  är egenvektorn

$$A - \lambda I = \begin{pmatrix} 1 - \lambda & 2 & 2 \\ 0 & 2 - \lambda & 1 \\ -1 & 2 & 2 - \lambda \end{pmatrix}$$

$$\det(A - \lambda I) = 4 - 2\lambda - (2 - \lambda)(4 - 3\lambda + \lambda^2) = (2 - \lambda)(-2 + 3\lambda - \lambda^2)$$

$$\det(A - \lambda I) = 0 \Leftrightarrow \lambda = 2 \vee \lambda = 1$$

2. Hittar egenvektorerna

$$\lambda = 2 :$$

$$(A - 2I)\vec{x} \Leftrightarrow \left( \begin{array}{ccc|c} 1 - 2 & 2 & 2 & 0 \\ 0 & 2 - 2 & 1 & 0 \\ -1 & 2 & 2 - 2 & 0 \end{array} \right) \sim \left( \begin{array}{ccc|c} 1 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right)$$

$$\Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2t \\ t \\ 0 \end{pmatrix} = t \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, t \in \mathbb{R}$$

Så egenrummet till  $\lambda = 2$  är linjära hörjet:  $\left[ \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \right]$

$$\lambda = 1 :$$

$$(A - 1I)\vec{x} \Leftrightarrow \left( \begin{array}{ccc|c} 1 - 1 & 2 & 2 & 0 \\ 0 & 2 - 1 & 1 & 0 \\ -1 & 2 & 2 - 1 & 0 \end{array} \right) \sim \left( \begin{array}{ccc|c} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right)$$

$$\Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -t \\ -t \\ t \end{pmatrix} = t \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, t \in \mathbb{R}$$

Så egenrummet till  $\lambda = 1$  är linjära hörjet:  $\left[ \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \right]$

## 8.12 Diagonalisering

**Definition:** Diagonalisering

Låt  $F : \mathbb{V} \rightarrow \mathbb{V}$  vara en linjär avbildning. Då kallas  $F$  **diagonaliserbar** om det finns en bas  $\underline{v}$  som består av egenvektorer till  $F$ . (Det betyder att  $(F)_{\underline{v}}$  är en diagonalmatris)

**Definition:** Algebraisk och Geometrisk multiplicitet

**Algebraisk multiplicitetet** av en egenvärde  $\lambda_0$  till en matris  $A$  är det maximala talet  $m$ , så att  $\chi_A(\lambda) = (\lambda - \lambda_0)^m p(\lambda)$  för någon polynom  $p$ .

**Algebraisk multiplicitetet** av en egenvärde  $\lambda_0$  till en matris  $A$  är  $\dim \mathbb{V}_{A, \lambda_0}$

**Exempel:**

Avför om matrisen är diagonaliserbar och isäfäl vad är den matrisen

$$M = \begin{pmatrix} -2 & 4 & 4 \\ 0 & 2 & 4 \\ 0 & 0 & -2 \end{pmatrix}$$

1. Hittar egenvärden: (se i kaptlet om enhetsvektorer)

$$(-2 - \lambda)(2 - \lambda)(-2 - \lambda) = 0$$

$$\lambda = 2 \text{ (alg mult 1)} \vee \lambda = -2 \text{ (alg mult 2)}$$

2. Hittar egenvekter: (se i kaptlet om enhetsvektorer)

$$\lambda = 2 \Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} t \\ t \\ 0 \end{pmatrix} = t \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, t \in \mathbb{R}$$

geo mult 1 därmed så är den hittills diagonaliserbar

$$\lambda = -2 \Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} s \\ -t \\ t \end{pmatrix} = s \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + t \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, s, t \in \mathbb{R}$$

geo mult 2 därmed diagonaliserbar

3. Sätter upp diagonala matrisen

$$F_{\underline{e}}^e = T_{\underline{e}}^{\underline{v}} F_{\underline{v}}^{\underline{v}} T_{\underline{v}}^e \Leftrightarrow$$

$$\begin{pmatrix} -2 & 4 & 4 \\ 0 & 2 & 4 \\ 0 & 0 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{pmatrix}^{-1}$$

## 8.13 Inre produkrum

### Definition: Skalärprodukt

Låt  $\mathbb{V}$  vara ett vektorrum. En **skalärprodukt** på  $\mathbb{V}$  är en operation som tillordnar varje par av vektorer  $\vec{u}, \vec{v} \in \mathbb{V}$  en skalär  $(\vec{u}|\vec{v}) \in \mathbb{R}$  så att följande villkor gäller:

bilinjär

1.  $(\vec{u} + \vec{v}|\vec{w}) = (\vec{u}|\vec{w}) + (\vec{v}|\vec{w})$
2.  $(\lambda \vec{u}|\vec{v}) = \lambda(\vec{u}|\vec{v}) = (\vec{u}|\lambda \vec{v})$

Symmetrisk

3.  $(\vec{u}|\vec{v}) = (\vec{v}|\vec{u})$

Positiv definit

4. Om  $\vec{u} \neq \vec{0}$  så  $(\vec{u}|\vec{u}) > 0$

### Definition: Avstånd och vinklar

Låt  $\mathbb{V}$  vara ett vektorrum med en skalärprodukt och  $\vec{u}, \vec{v} \in \mathbb{V}$

1. Längden (normen) av  $\vec{u}$  beteckas  $|\vec{u}|$  och defineras som

$$|\vec{u}| = \sqrt{\vec{u}|\vec{u}}$$

2. Avståndet mellan  $\vec{u}$  och  $\vec{v}$  defineras som  $\vec{u} - \vec{v}$

3. Om  $\vec{u} \neq \vec{0}$  och  $\vec{v} \neq \vec{0}$  defineras vinkeln mellan  $\vec{u}$  och  $\vec{v}$  som

$$\cos^{-1} \frac{(\vec{u}|\vec{v})}{|\vec{u}||\vec{v}|}$$

4. Vi säger att  $\vec{u}$  och  $\vec{v}$  är ortogonala om

$$(\vec{u}|\vec{v}) = 0$$

### Definition: Trianglar

Låt  $\mathbb{V}$  vara ett vektorrum med en skalärprodukt och  $\vec{u}, \vec{v} \in \mathbb{V}$ .

Då bildar  $\vec{u}, \vec{v}$  och  $\vec{u} + \vec{v}$  en triangel

$$|\vec{u} + \vec{v}|^2 = |\vec{u}|^2 + |\vec{v}|^2 + 2(\vec{u}|\vec{v})$$

### Definition: Ortogonal projektion och komponent

1.  $\vec{u}_{\parallel \vec{v}} := \frac{(\vec{v}|\vec{u})}{(\vec{u}|\vec{u})} \vec{u}$  kallas den ortogonala projektionen av  $\vec{v}$  på  $\vec{u}$
2.  $\vec{u}_{\perp \vec{u}} := \vec{v} - \vec{u}_{\parallel \vec{u}}$  kallas den ortogonala komponenten av  $\vec{v}$  med avseende på  $\vec{u}$

## 8.14 Symmetriska och positiva definita matriser

**Formel:** skalärprodukt uttrykt med matriser

$$(\vec{u}|\vec{y}) = \vec{u}^t A \vec{y}$$

där  $A$  är  $(n \times n)$ -matrisen som uppfyller följande krav

1. Symetrisk om  $A^t = A$
2. Positivt definit om  $\vec{x}^t A \vec{x} > 0, \forall \vec{x} \in \mathbb{R}^n, \vec{x} \neq \vec{0}$

**Exempel:** skalärprodukt uttrykt med matriser

På  $\mathbb{R}^2$  definierar vi  $(\vec{x}|\vec{y}) = x_1y_1 + 3x_1y_2 + ax_2y_1 + bx_2y_2$

För vilka värden på  $a$  och  $b$  är detta en skalärprodukt?  $a, b \in \mathbb{R}$

Lösn:

1. Symetrisk:  $(\vec{x}|\vec{y}) = (\vec{y}|\vec{x}) = y_1x_1 + 3y_1x_2 + ay_2x_1 + by_2x_2$

$$(\vec{x}|\vec{y}) = \vec{x}^t \begin{pmatrix} 1 & 3 \\ a & b \end{pmatrix} \vec{y} \Rightarrow a = 3 \text{ pga summetri}$$

2. Pos def:  $(\vec{x}|\vec{x}) > 0, \forall \vec{x} \neq \vec{0}$

$$\begin{aligned} (\vec{x}|\vec{x}) &= (x_1x_2) \begin{pmatrix} 1 & 3 \\ a & b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2 + 3x_1x_2 + 3x_2 + x_1 + bx_2^2 \\ &= x_1^2 + 6x_1x_2 + bx_2^2 = (x_1 + 3x_2)^2 + (b - 9)x_2^2 \end{aligned}$$

vilket endast är positivt då  $b > 9$

## 8.15 On-baser och Gram-Schmidt-ortonormalisering

### Definition: ON-bas

Låt  $\mathbb{V}$  vara ett vektorrum med en skalärprodukt.

En mängd  $\{\vec{u}_1 \dots \vec{u}_n\} \subseteq \mathbb{V}$  kallas ortonormal (ON) om

$$(\vec{u}_i | \vec{u}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$|\vec{u}_i| = 1$$

Dvs alla vektorer är ortogonala mot varandra och att längden ska vara 1

### Definition: Koordinater i en ON-bas

Låt  $\underline{u} = (\vec{u}_1 \dots \vec{u}_n)$  vara en ON-bas i  $\mathbb{V}$  och  $\vec{v}, \vec{w} \in \mathbb{V}$ . Då är

$$1. \quad \vec{v}_{\underline{u}} = \begin{pmatrix} (\vec{v} | \vec{u}_1) \\ \vdots \\ (\vec{v} | \vec{u}_n) \end{pmatrix}$$

$$2. \quad (\vec{v} | \vec{w}) = (\vec{v} | \vec{u}_1)(\vec{w} | \vec{u}_1) + \dots + (\vec{v} | \vec{u}_n)(\vec{w} | \vec{u}_n) = \vec{v}_n \bullet \vec{w}_n$$

### Definition: Ortogonal komplement

Låt  $\mathbb{V}$  vara ett vektorrum med en skalärprodukt och

$\mathbb{U} \subseteq \mathbb{V}$  ett underrum. Då är

$$\mathbb{U}^\perp = \{\vec{v} \in \mathbb{V} | (\vec{u} | \vec{v}) = 0, \forall \vec{u} \in \mathbb{U}\}$$

ett underrum som kallas det ortogonal komplementet till  $\mathbb{U}$

**Exempel: Hitta en ON-bas med Gram-Schmidt ortonormalisering**

$$U = [\vec{u}_1, \vec{u}_2, \vec{u}_3] \subseteq \mathbb{R}^4, \vec{u}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \vec{u}_2 = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}, \vec{u}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Hitta ON-bas

$$[\vec{u}_1] \subseteq [\vec{u}_1, \vec{u}_2] \subseteq [\vec{u}_1, \vec{u}_2, \vec{u}_3]$$

där  $[\vec{u}_1] = U_1$ ,  $[\vec{u}_1, \vec{u}_2] = U_2$ ,  $[\vec{u}_1, \vec{u}_2, \vec{u}_3] = U_3$

$$\text{ON-bas } U_1 : f_1 = \frac{1}{|\vec{u}_1|} \vec{u}_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$\text{ON-bas } U_2 : f_1, f_2 \vec{u}_{2 \perp \vec{u}_1} = \vec{u}_2 - (\vec{u}_2 | f_1) f_1$$

$$= \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \end{pmatrix} - \frac{1}{\sqrt{3}}(1+0+2+0) \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

$$f_2 = \frac{1}{\vec{u}_{2 \perp \vec{u}_1}} \vec{u}_{2 \perp \vec{u}_1} = \frac{1}{\sqrt{3}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

$$\text{ON-bas } U_3 : f_1, f_2, f_3 \vec{u}_{3 \perp \vec{u}_2} = \vec{u}_3 - (\vec{u}_3 | f_1) f_1 - (\vec{u}_3 | f_2) f_2$$

$$= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} - \frac{1}{\sqrt{3}}(0+0+1+0) \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{\sqrt{3}}(0+0+1+1) \frac{1}{\sqrt{3}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} - \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - \frac{2}{3} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow f_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Kontrollera att  $f_1, f_2, f_3$  är ortogonalala mot varandra

## 8.16 Isometriska avbildningar och spektralsatsen

**Definition:** isometrier

Låt  $\mathbb{V}$  och  $\mathbb{W}$  vara vektorrum med skalärprodukter.

En linjär avbildning

$$F : \mathbb{V} \rightarrow \mathbb{W}$$

kallas en isomerik om  $|F(\vec{v})| = |\vec{v}|$

En isometrisk linjär avbildning kallas också en isometri.

dvs. Så är längden oförendrad.  $F$  är en isometri omm

$$(F(\vec{u})|F(\vec{v})) = (\vec{u}|\vec{v})$$

**Egenskaper:** isometrier

**Sats:** Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara en isometri. Då är  $F$  injektiv

**Sats:** Låt  $F : \mathbb{V} \rightarrow \mathbb{W}$  vara en iventerbar isometri.

Då är  $F^{-1} : \mathbb{W} \rightarrow \mathbb{V}$  en isometri

**Sats:** Låt  $F : \mathbb{U} \rightarrow \mathbb{V}$  och  $G : \mathbb{V} \rightarrow \mathbb{W}$  vara en isometrier.

Då är  $G \circ F : \mathbb{U} \rightarrow \mathbb{W}$  en isometri

**Sats:** spektralsatsen

Låt  $\mathbb{V}$  vara ett ändligt dimensionellt vektorrum med en skalär produkt och  $F : \mathbb{V} \rightarrow \mathbb{V}$  en linjär avbildning.

Då är följande vilkor ekvivalenta

1.  $F$  är symmetrisk
2.  $\mathbb{V}$  har en ON-bas av bestående av egenvektorer till  $F$

Låt  $A$  vara en  $(n \times n)$ -matris. Då är följande vilkor ekvivalenta

1.  $A$  är symmetrisk
2. Det finns en ortonormal matris  $T$  så att  $D = T^{-1}AT$  är en diagonalmatris

## 8.17 Andragradskurvor och andragradsytor

$$Q(x) = 1 \Leftrightarrow 1x^t Ax = 1 \Leftrightarrow y^t Dy = 1 \Leftrightarrow \lambda_1 y_1^2 + \lambda_2 y_2^2 = 1$$

Om  $\lambda_1 = \lambda_2$  så berkriver  $Q(x) = 1$  en cirkel

Om  $\lambda_1 > 0$  och  $\lambda_2 > 0$  så berkriver  $Q(x) = 1$  en ellips

Om  $\lambda_1 > 0$  och  $\lambda_2 < 0$  så berkriver  $Q(x) = 1$  en hyperbel

**Exempel:** Bestäm fomren

$$Q(\vec{x}) = \frac{14}{5}x_1^2 + \frac{11}{5}x_2^2 + \frac{14}{5}x_1x_2$$

$$Q(\vec{x}) = (x_1 \ x_2) \begin{pmatrix} 14/5 & 2/5 \\ 2/5 & 11/5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$1. \text{ Egenvärden } \det A = \left(\frac{14}{5} - \lambda\right)\left(\frac{11}{5} - \lambda\right) - \frac{4}{25} = 0$$

$$\lambda_1 = 2 > 0$$

$$\lambda_2 = 3 > 0$$

Därmed så beskriver  $Q(x) = 1$  en ellips

2. Egenvektor får vi en ON-bas

3. Ställer up ekvationen och ritar ut den

$$Q(\vec{y}) = y^{-1} \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \vec{y} = 2y_1^2 + 3y_2^2 = 1$$

## 8.18 System av linjära differentialekvationer

$$y' = Ay$$

$$T^{-1}AT = D$$

$$y = Tz \Rightarrow y' = Tz' \wedge y' = Ay \Leftrightarrow z' = dz$$

$$z = \begin{pmatrix} c_1 e^{\lambda_1 t} \\ c_2 e^{\lambda_2 t} \\ \vdots \\ \vdots \\ c_n e^{\lambda_n t} \end{pmatrix} \text{ där } c_i \in \mathbb{R}^n \text{ och } y = Tz$$

$$c = T^{-1}y_0 \text{ där } z(0) = c = \begin{pmatrix} c_1 \\ \vdots \\ \vdots \\ c_n \end{pmatrix}$$

**Exempel: Bestäm fomren**

Lös följande system av differentialekvationer

$$\begin{cases} \frac{dx(t)}{dt} = 2x(t) + y(t) + z(t) \\ \frac{dy(t)}{dt} = x(t) + 2y(t) + z(t) \quad x(0) = 3, y(0) = 2, z(0) = 1 \\ \frac{dz(t)}{dt} = x(t) + y(t) + 2z(t) \end{cases}$$

1. Skriver up systemet

$$\begin{pmatrix} \frac{dx(t)}{dt} \\ \frac{dy(t)}{dt} \\ \frac{dz(t)}{dt} \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \text{ Där matrisen är } A$$

2. Diagonalisering

2.1 egenvärden

$\lambda_1 = 1$  (multiplicitet 2),  $\lambda_2 = 4$  (multiplicitet 1)

2.1 egenvektor

$$\lambda_1 = 1 \Rightarrow A - I = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \Rightarrow \vec{v}_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \wedge \vec{v}_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

$$\lambda_2 = 4 \Rightarrow A - 4I = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix} \Rightarrow \vec{v}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\text{Diagonal ekv blir } A = TDT^{-1} \Rightarrow T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix} \wedge D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

3. Finner den allmäna lösningen

$$u' = Du \Rightarrow \begin{cases} u_1 = c_1 e^t \\ u_2 = c_2 e^t \\ u_3 = c_3 e^{4t} \end{cases} \text{ Där } c_1, c_2, c_3 \in \mathbb{R}$$

$$v' = Av \text{ Där } A^{-1}v' = Tu \Rightarrow v = Tu$$

$$v = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} c_1 e^t \\ c_2 e^t \\ c_3 e^{4t} \end{pmatrix} = \begin{pmatrix} c_1 e^t + c_2 e^t \\ c_2 e^t + c_3 e^{4t} \\ -c_1 e^t - c_2 e^t + c_3 e^{4t} \end{pmatrix}$$

4. Finner lösningen till begynelsevärdet

$$\left( \begin{array}{ccc|c} 1 & 0 & 1 & 3 \\ 0 & 1 & 1 & 2 \\ -1 & -1 & 1 & 1 \end{array} \right) \sim \left( \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{array} \right) \Rightarrow \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

$$\begin{cases} x(t) = e^t + 2e^{4t} \\ y(t) = 2e^{4t} \\ z(t) = -et + 2e^{4t} \end{cases}$$



## Chapter 9

# Computer System with Project Work

## 9.1 M1

### 9.1.1 CPU

The central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

### 9.1.2 Register

A processor register is a quickly accessible location available to a computer's central processing unit (CPU). Registers usually consist of a small amount of fast storage. A CPU only has a small number of registers.

### 9.1.3 Memory

Memory refers to the computer hardware integrated circuits that store information for immediate use in a computer; it is synonymous with the term "primary storage". The memory is much slower than the CPU register but much larger in size. Sources

### 9.1.4 CPU context

At any point in time, the values of all the registers in the CPU defines the CPU context. Sometimes CPU state

### 9.1.5 Memory allocation

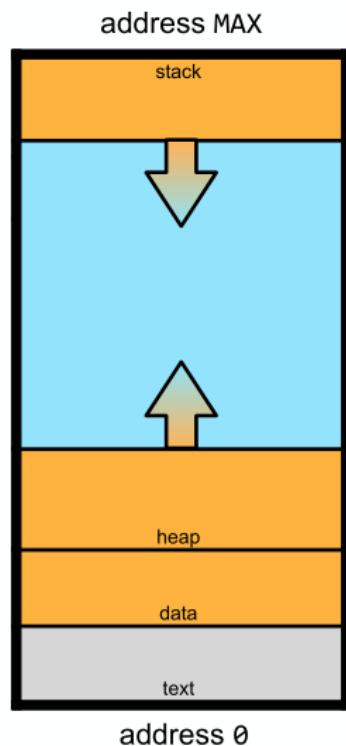
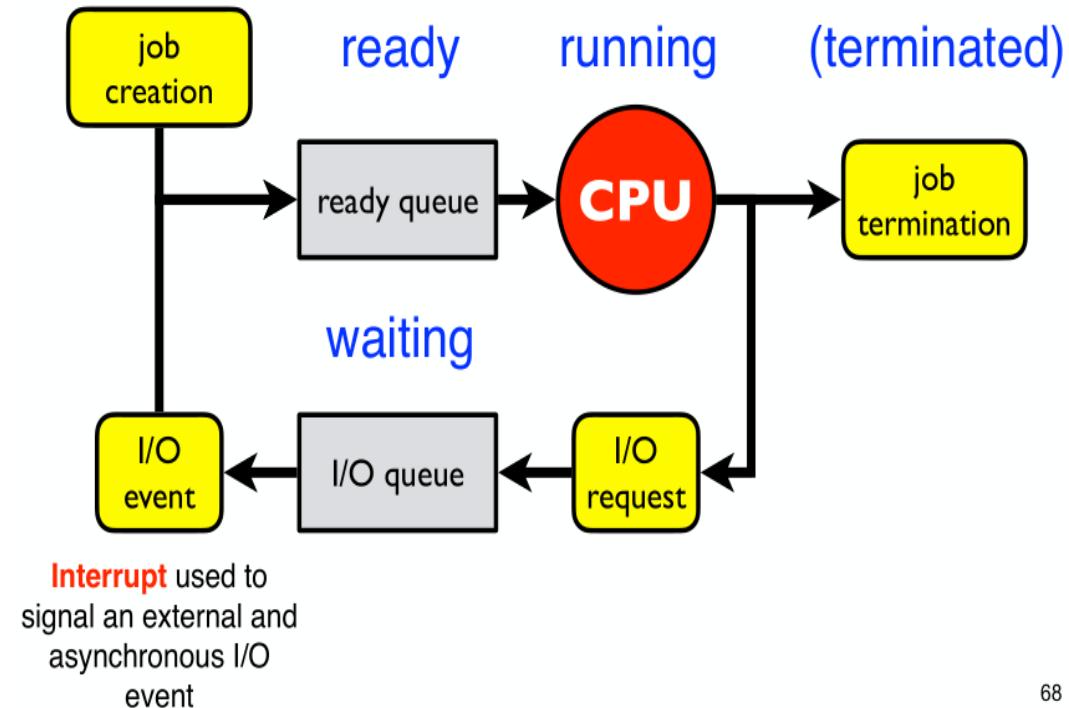


Figure 9.1: memory-allocation

### 9.1.6 Kernal

The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system.

### 9.1.7 Multiprogramming



68

Figure 9.2: multiprogramming

## 9.2 M2

### 9.2.1 Network communication

- TCP for a reliable byte stream service
- UDP for an unreliable message forwarding service
- Four different delivery models: uni/broad/multi/any-cast
- The client/server model is common in application design
- Sockets API can be used for network programming
  - A socket is a communication handle abstraction
  - Properties of a socket can be set through socket options

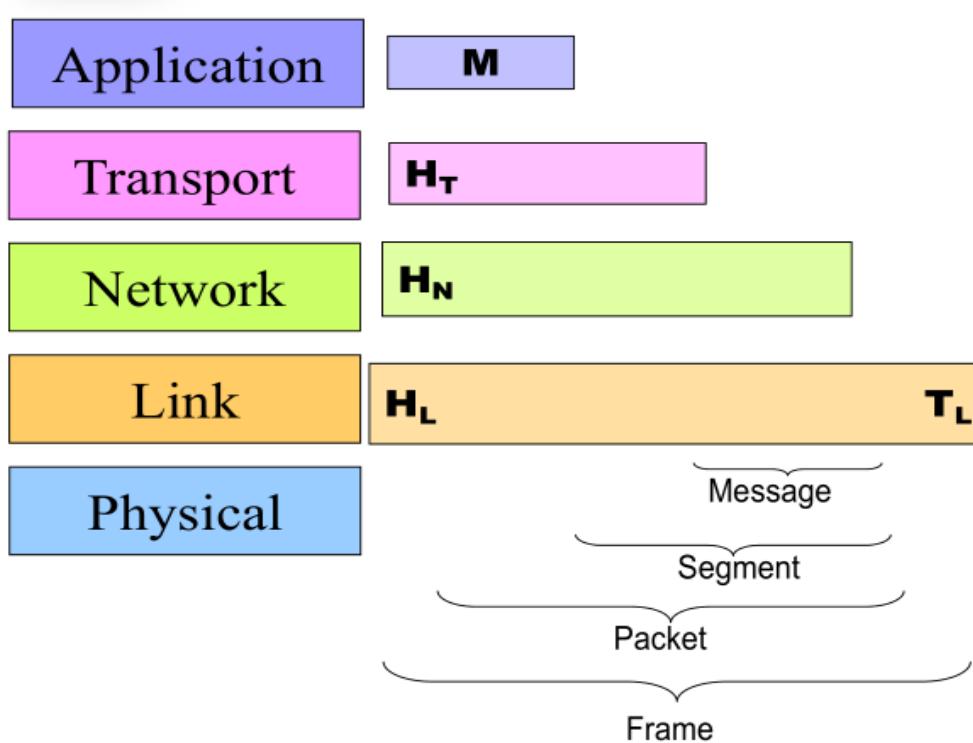


Figure 9.3: layering

### 9.2.2 Processes

- **Zombie**: A terminated process is said to be a zombie or defunct until the operating system removes it.
- **Orphan**: An orphan process is a process whose parent process has terminated, though it remains running itself.
- **Signals**: are a limited form of inter-process communication used in the kernel.
- **File descriptor**: The kernel keeps a table with information about a process's open file descriptors.
- **Pipes**: An (anonymous) pipe is a simplex FIFO communication channel that may be used for inter-process communication.

## 9.3 M3

- DNS är ett distribuerat system av servrar
- ARP mappar IP-address till länklageraddress
- En IP-address har två delar för att identifiera nätverk och dator

### 9.3.1 Dispatcher

Another component that is involved in the CPU-scheduling function is the dispatcher, which is the module that gives control of the CPU to the process selected by the short-term scheduler. It receives control in

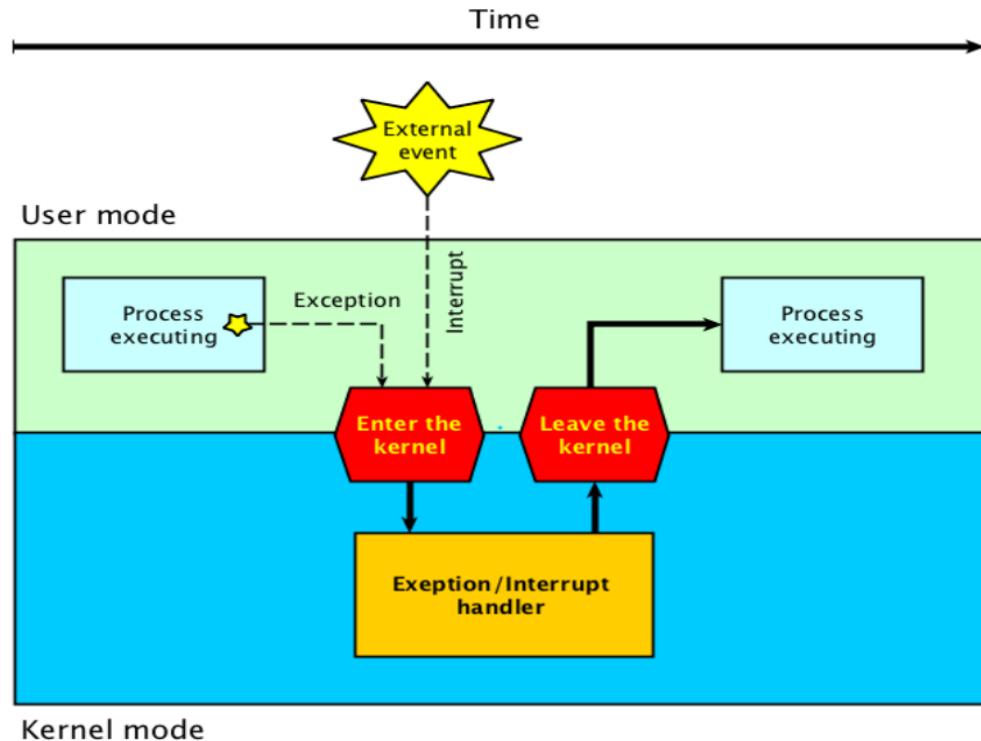


Figure 9.4: user-kernal-mode

kernel mode as the result of an interrupt or system call.

The CPU scheduler selects one process from among the processes in memory that are READY to execute. The scheduler dispatcher then gives the selected process control

### 9.3.2 PSB

The process control block (PCB) is a data structure in the operating system kernel containing the information needed to manage a particular process.

The Long-term scheduler (LTS) (aka job scheduler) decides whether a new process should be brought into the ready queue in main memory or delayed.

The medium-term scheduler (MTS) temporarily remove processes from main memory and places them in secondary storage and vice versa, which is commonly referred to as "swapping in" and "swapping out".

### 9.3.3 IO bound/CPU bound

An I/O-bound process spends more time doing I/O than computations and is characterised

An CPU-bound process spends more time doing computations and is characterised by few very long. Interactive processes interact constantly with their human users.

Batch processes do not interact with human users.

Real-time processes have very strong scheduling requirements.

### 9.3.4 Schedular algorithms

The first come, first served (commonly called FIFO – first in, first out) process scheduling algorithm is the simplest process scheduling algorithm. Processes are executed on the CPU in the same order they arrive to the ready queue.

Shortest Job First (SJF) scheduling assigns the process estimated to complete fastest, i.e, the process with shortest CPU burst, to the CPU as soon as CPU time is available.

An extension of SJF where the currently running process is P preempted if the CPU burst of a process A arriving to the ready queue is shorter than the remaining CPU burst of the currently running process.

A priority number (integer) is associated with each process. The CPU is allocated to the process with the highest priority (smallest integer = highest priority).

Round Robin (RR) is a scheduling algorithm where time slices are assigned to each process in equal portions and in circular order.

A multi-level queue scheduling algorithm is used in scenarios where the processes can be classified into groups based on properties like process type, CPU time, IO access, memory size, etc.

In computer science, starvation is a problem encountered in multitasking where a process is perpetually denied necessary resources. Without those resources, the process can never finish its task.

Ageing is used to ensure that jobs with lower priority will eventually complete their execution.

## 9.4 M4

### 9.4.1 Concurrency

The ability of different parts or units of a program, algorithm, or problem to be executed out-of-order or in partial order, without affecting

### 9.4.2 Parallelism

In parallel systems, two tasks are actually performed simultaneously. Parallelism is when tasks

### 9.4.3 Client-Server mode

A distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

### 9.4.4 Threads

A thread of execution is the smallest sequence of instructions that can be managed independently by a scheduler, which is typically a part of the operating system.

#### Atomic

In concurrent programming, an operation (or set of operations) is atomic if it appears to the rest of the system to occur at once without being interrupted.

### Race condition

A race condition or race hazard is the behaviour of an electronic, software or other system where the output is dependent on the sequence or timing of other uncontrollable events.

### Data race

A data race occurs when two instructions from different threads access the same memory location

### 9.4.5 Locks

In general, any solution to the critical section problem requires a tool/abstraction - a lock.

### 9.4.6 Spinlock

A spinlock is a lock where a task simply waits in a loop ("spins") repeatedly checking until the lock becomes available.

### 9.4.7 TestAndSet

The TestAndSet instruction atomically first sets the value at the target address to True and return the old value stored at the target address.

### 9.4.8 Swap

The Swap instruction atomically swaps the content of two memory

### 9.4.9 Bounded waiting

A bound must exist on how many times a task has to wait in order to enter a critical section.

### 9.4.10 Semaphores

The hardware solutions to the critical section problem using Swap and TestAndSet are complicated for programmers to use directly.

### 9.4.11 Mutex lock

Many locking libraries provides a special mutex lock which can only be used to provide mutual exclusion to critical sections.

### 9.4.12 Deadlock

In concurrent computing, a deadlock is a state in which each member of a group is waiting for some other member to take action, such as sending a message or more commonly releasing a lock.

### 9.4.13 Mutual exclusion

A chopstick can only be held by one philosopher at the time. Updating the amount of rice must be atomic.

### 9.4.14 Deadlock prevention

Preventing deadlocks by constraining how requests for resources can be made in the system and how they are handled (system design).

#### 9.4.15 Deadlock avoidance

The system dynamically considers every request and decides whether it is safe to grant it at this point,

#### 9.4.16 Clock Synchronization

- External synchronization
- Internal synchronization
- Hard in an asynchronous system

#### 9.4.17 Mutual exclusion

- Critical section
- Need for coordination:
- Evaluation criteria for different solutions

#### 9.4.18 Desired properties of Transactions

- Atomicity
- Consistency
- Isolation
- Durability

#### 9.4.19 WiFi

- Several standards, from 2Mbit/s to 100's of Mbit/s
  - Using multiple channels for higher speeds
- Range: in general 20-100m, depending on the environment
- Can be used in ad hoc or infrastructure mode
  - Ad Hoc: like a wireless Ethernet
  - Infrastructure: a coordinating base station
- Networks identified with SSID:s
  - Used to scramble signal somewhat
- Encryption with WEP (bad), WPA(better), WPA2 (best)

#### 9.4.20 Bluetooth

- Several different versions
  - Different transmission power (and hence ranges)
- Different profiles for different applications
- Frequency hopping 1600 times/s in ISM band
- Peer to peer or piconet networks
- Max 7 simultaneous connections

## Channels in ISM band

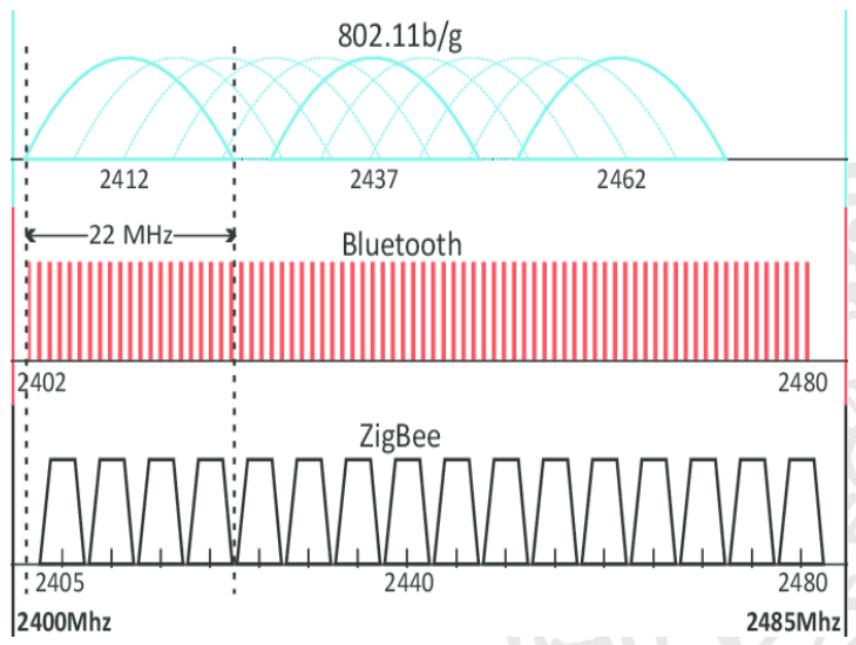


Figure 9.5: ISM-band

### 9.4.21 ISM-band

### 9.4.22 Properties of a medium

- Bandwidth B
  - Frequency range for signals transmitted in medium
- Signal/noise ratio S/N, SNR
  - Frequency range for signals transmitted in
  - Signal level / noise level
  - Measured at the receiver
  - Can be approximated with Maxwell's equations
- Capacity  $C=B \log_2 (1+S/N)$

### 9.4.23 Sampling

Bandwidth B Hz is sampled at  $2^*B$  Hz

### 9.4.24 Multithreading models

### 9.4.25 Bounded buffer

A bounded buffer lets multiple producers and multiple consumers share a single buffer. Producers write data to

Many-to-One	One-to-One	Many-to-Many	Two-level-model
<b>Examples:</b> <ul style="list-style-type: none"> <li>★ Solaris Green Threads</li> <li>★ GNU Portable Threads</li> </ul>	<b>Examples:</b> <ul style="list-style-type: none"> <li>★ Windows NT/XP/2000</li> <li>★ Linux</li> <li>★ Solaris 9 and later</li> </ul>	<b>Examples:</b> <ul style="list-style-type: none"> <li>★ Solaris prior to version 9</li> <li>★ Windows NT/2000 with the ThreadFiber package</li> </ul>	<b>Examples:</b> <ul style="list-style-type: none"> <li>★ IRIX</li> <li>★ HP-UX</li> <li>★ Tru64 UNIX</li> <li>★ Solaris 8 and earlier</li> </ul>

Figure 9.6: Multithreading-models

#### 9.4.26 Priority inversion

A higher priority task is “preempted” by a lower priority one.

### 9.5 M5

#### 9.5.1 Single contiguous

Single contiguous allocation is the simplest memory management technique. All the computer’s memory, usually with the exception of a small portion reserved for the operating system, is available to the single application.

#### 9.5.2 Swapping

A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution.

#### 9.5.3 Memory resident

Certain programs, can be marked as being memory resident, which means that the operating system is not permitted to swap them out to a storage device; they will always remain in memory.

#### 9.5.4 Partitioned allocation

Partitioned allocation divides primary memory into multiple memory partitions, usually contiguous areas of memory.

#### 9.5.5 Logical address space

A logical address is the address at which a memory cell appears to reside from the perspective of an executing application program.

#### 9.5.6 Address binding

Address binding is the process of mapping the program's logical (or virtual addresses) to corresponding physical memory addresses.

#### 9.5.7 Memory management unit (MMU)

The MMU is a computer hardware unit having all memory references passed through itself, primarily performing the translation of logical memory addresses to physical addresses.

#### 9.5.8 Fragmentation

Fragmentation is a phenomenon in which storage space is used inefficiently, reducing capacity and often performance.

#### 9.5.9 Compaction

Move all processes to one end of the address space producing one large hole at the other end of the address space.

#### 9.5.10 Memory protection

Memory protection between processes implemented by associating a valid bit with each entry in the per process page table.

#### 9.5.11 Shared pages

Paging makes it possible for processes to share parts of their memory space with each other.

#### 9.5.12 Translation lookaside buffer

A translation lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location. The TLB stores the recent translations of virtual memory to physical memory.

#### 9.5.13 Operating System

Controls the hardware and coordinates its use among the various application programs for various users.

#### 9.5.14 Access methods

Data can be accessed in different patterns. These patterns can be divided into two main categories: Sequential access, Random access (aka direct access)

#### 9.5.15 RAM

The main memory aka primary memory is a form of Random Access Memory (RAM).

### 9.5.16 Volatile memory

Volatile memory, in contrast to non-volatile memory, is computer memory that requires power to maintain the stored information; it retains its contents while powered on but when the power is interrupted, the stored data is quickly lost.

### 9.5.17 Persistent data storage

In computer science, persistence refers to the characteristic of state that outlives the process that created it.

### 9.5.18 The file

A file is a named collection of related information that is recorded on secondary storage.

### 9.5.19 Block data storage

A file is made up of fixed length logical records (blocks)

### 9.5.20 File control block

A File Control Block (FCB) is a file system structure in which the state of an open file is maintained.

### 9.5.21 Directory

A directory is a file system cataloging structure which contains references to other computer files, and possibly other directories.

### 9.5.22 Contiguous allocation

Each file occupies a set of contiguous blocks on the disk.

### 9.5.23 Linked allocation

Each file is a linked list of disk blocks: blocks may be scattered anywhere on the disk.

### 9.5.24 Indexed allocation

Indexed allocation brings all pointers together into an index

### 9.5.25 FAT

File Allocation Table (FAT) is a computer file system architecture and a family of industry-standard file systems utilizing it. The FAT file system is a legacy file system which is simple and robust.

### 9.5.26 iNode

In Unix-style file systems the file control block is called iNode. The inode is a data structure that describes a filesystem object such as a file or a directory.

### 9.5.27 Directory

A directory is a file system cataloging structure which contains references to other computer files, and possibly other directories.

### 9.5.28 M6

#### 9.5.29 Classic historical ciphers

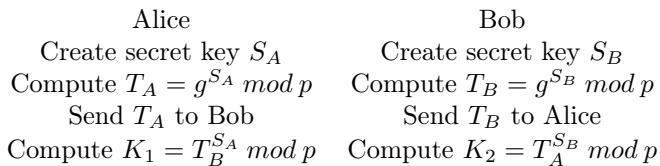
Caesar cipher (monoalphabetic), Vigenere cipher (polyalphabetic).

#### 9.5.30 Modern cryptography

If a lot of smart people for a long time have failed to solve a specific problem, it is unlikely that a solution will appear soon.

- Encryption method is well-known
- Secret guarded by a n-bit key
  - Encryption and decryption in  $O(n)$  time
  - Key guessing in  $O(2^n)$  time
- Key management is crucial
- CIA triad represent desirable properties
  - Confidentiality
  - Integrity
  - Availability

#### 9.5.31 Public key exchange



#### 9.5.32 Certification Authority (CA)

Issues digital certificates: Digitally signed with the private key of the CA, Authorize a public key.

#### 9.5.33 Chain of trust

#### 9.5.34 Self-issued certificates

Some sites present self-issued certificates: A little like designing your own drivers license.

#### 9.5.35 Usage of modern cryptography

- Symmetric cryptography in encrypted sessions
  - public-key cryptography not fast enough
- Asymmetric cryptography in certain situations
  - To establish a symmetric key
  - Digital signatures and verification
- Cryptographic hashing for verification of authentic data

## PUBLIK KEY KRYPTO

idea-instructions.com/public-key/  
v1.1, CC by-nc-sa 4.0

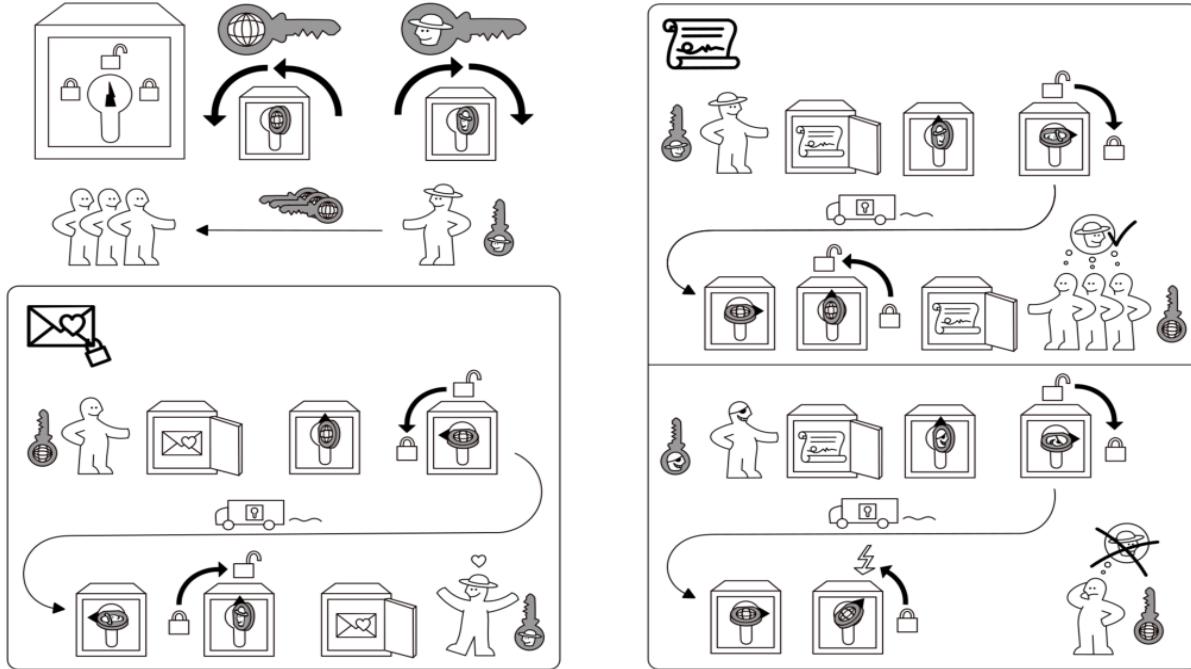


Figure 9.7: publik-key-krypto

### 9.5.36 Man-in-middle attack

the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other

### 9.5.37 Chosen-plaintext attack

presumes that the attacker can obtain the ciphertexts for arbitrary plaintexts. The goal of the attack is to gain information that reduces the security of the encryption scheme

### 9.5.38 Side-channel attack

any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself. Timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited.

### 9.5.39 Replay attack

valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and re-transmits it, possibly as part of a spoofing attack by IP packet substitution. This is one of the lower-tier versions of a man-in-the-middle attack. Replay attacks are usually passive in nature.

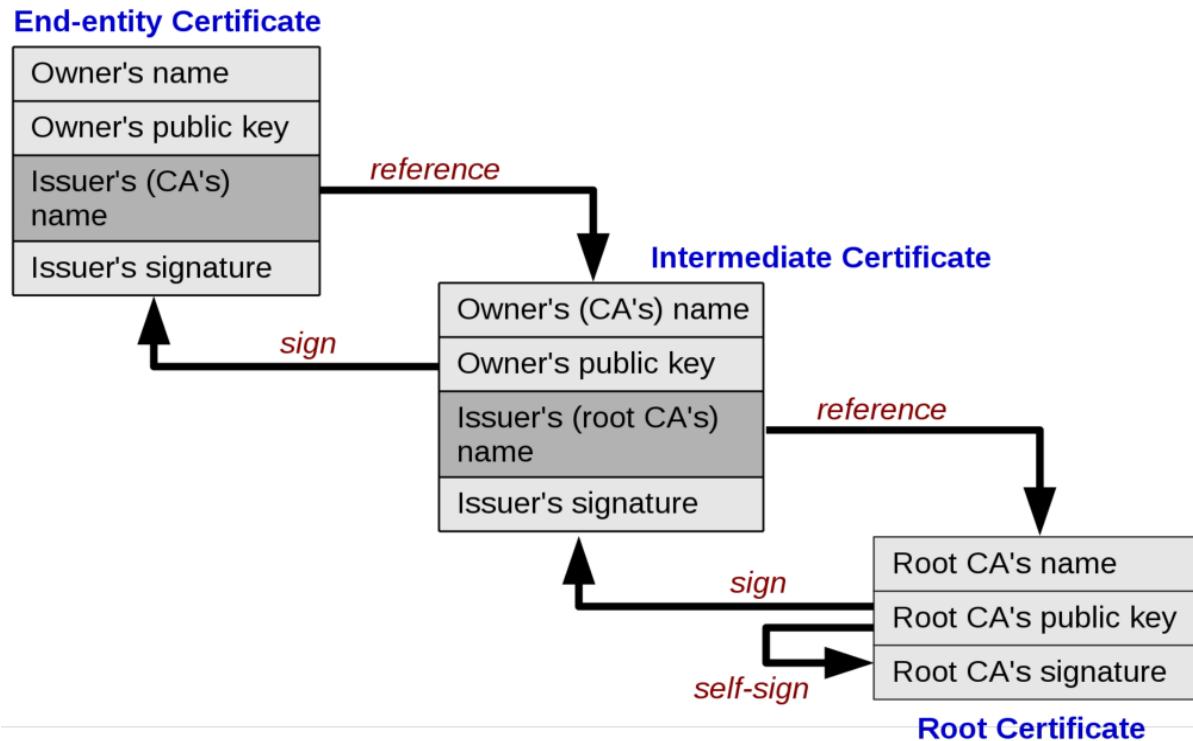


Figure 9.8: chain-of-trust

#### 9.5.40 Security in the Internet stack

- Application layer: DNSSEC
- Transport layer: SSL/TLS
- Network layer: IPsec
- Link layer: WEP/WPA, MAC filtering, ...

#### 9.5.41 Firewall

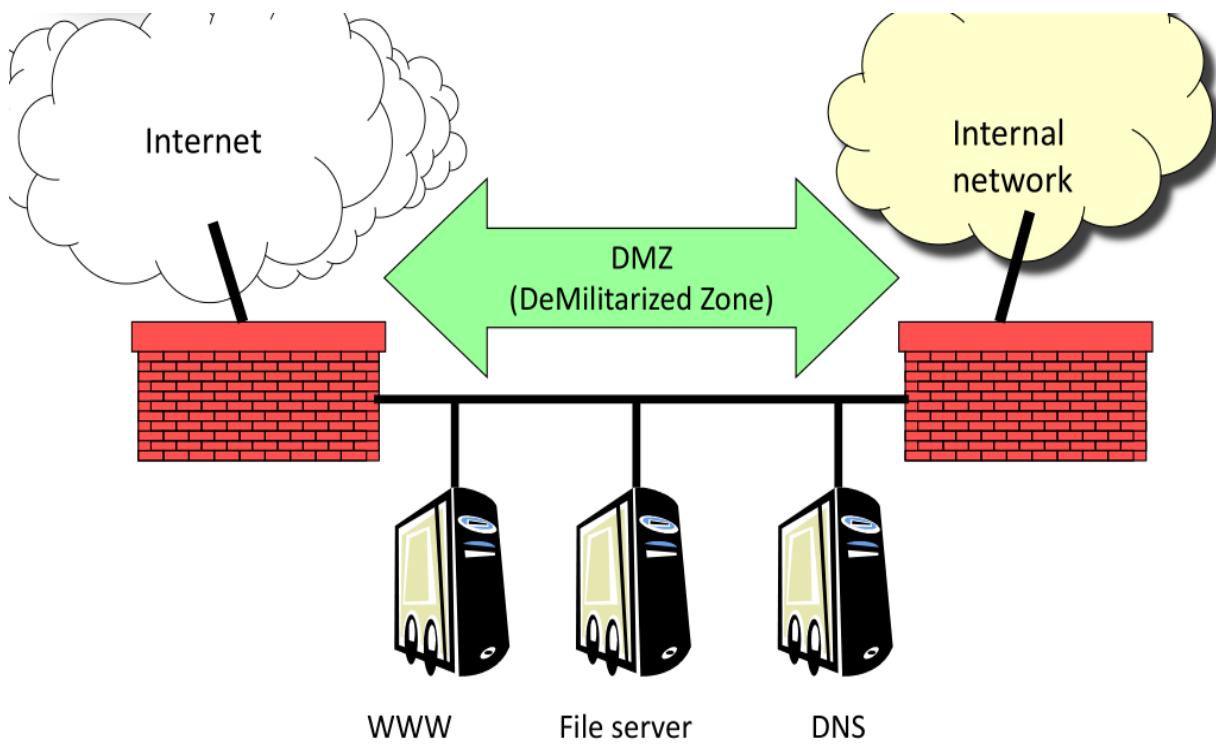


Figure 9.9: firewall

## Chapter 10

# System Design with User Perspective

## 10.1 Documentation

- One of the most important things in life is to understand...
- I want you to understand WHY you need to document a program!
- Then the HOW becomes much easier!
- The documentation is the user interface to the program code!
- It is possible to work without it, but who want to code in assembler?

### 10.1.1 Implicit documentation

- Names can tell a story (meaningful names)
- Names can also be classifying the variable (x,y positions. i,j,k index)
- Choose the right algorithms and data structures

### 10.1.2 Explicit documentation

- Several different versions of documentation
  - Meta documentation (Each file: Description, author, protocols, file dependencies)
  - External document (Manuals/Installation instruction, list of tree structures)
  - Inline documentation (Difficult code, TODO)

## 10.2 Human factors

### 10.2.1 Color combinations

There are some color combinations with humans have difficulty processing on of with is Red text on Blue background and Blue text on red background.

### 10.2.2 Personas

Who are the real people who will use this program. With that information make sure it is well suited for them and there needs as well as for their difficulties.

### 10.2.3 Memory

Our memory is like a net wire each connection is from one piece of information to another there for memory needs to be associated with the necessary areas. For instance when we design a web shop we should have the checkout on the right top since for most web shops have that and therefore people make that connection that this is a web shop therefore the checkout is on the top right.

### 10.2.4 What is needed to design a system?

- Requirement specification
- Specify in detail the requirements on the software
  - What should it be able manage?
    - \* Number of customers, number of items in store, etc.
  - Which functions are needed?
- Order a beverage, check stored items, etc.

### 10.3 Model-View-Control

Initially a Object-Oriented (interface) Design model. Purpose: Making the View separated from the Mode.

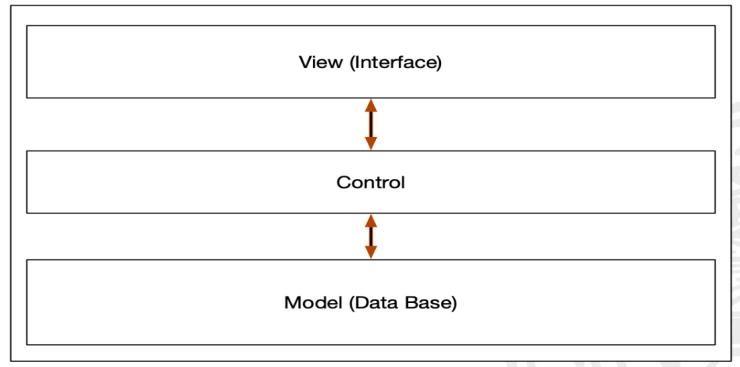


Figure 10.1: MVC

- View is only presentation of a Designed Interaction!
- Start by modelling the Model and Control (as a command Interface)
- Then implement the View (both control and presentation).
  - Create a good metaphor to the things that happen in



## Chapter 11

# Probability and Statistics DV

## 11.1 Statistisk mått och begreppet sannolikhet

### 11.1.1 Begrepp

- Deterministiska modeller: enkla modeller som inte tar hänsyn till fel
- Sannolikhetsteori: modelera slumpmässiga fel
- s..k: a..u slumpmässig data
- Beskrivande statistik:
  - Population: Alla bilar i uppsala
  - Stickprov: 100 utvalda bilar i uppsala
  - Enhet: En av det 100 utvalda
  - Variabler: Motorstyrka, dragkrok, automat etc
    - \* Kvalitativa: (dragkrok, automati/manual)
    - \* Kvantitativa: (Motorstyrka, vikt)
  - Tvärsnitts data: Befolknings i svenska städer 1 jan 2018
  - Longitudinella data: Befolking i uppsala under 1 jan 1960 till 2020
- Statistisk mått:
  - Lägesmått:
    - \* Aretmetisk modelera:  $\bar{x} = \frac{\sum_{k=1}^n x_k}{n}$
    - \* median: (1, 1, 2, 4, 4, 4, 7), (1, 1, 2, 4, 4, 4)
  - Kvartiler: Tre punkter, fyra i hjälmarna, som delar upp tal serie
    - \* 1:a kvartilen (Nedre kvartil) mittpunkten av den nedre halvan
    - \* 3:e kvartilen (Övre kvartil) mittpunkten av den övre halvan
  - Spridningsmått:
    - \* Kvartal bred: 3:e kvartalen – 1:a kvartalen
    - \* Stickprovs standard devianse (standard deviation):  $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$
    - \* Savaraten (skatt plots): varge enkel två ???
    - \* Spridnings diagram: Positiv korelation, negativ korelation, ingen korelation, perfect korelation
- Diskret variable: räknerligt många värden
- Kontinuerlig variabel: tar oräknerligt många värden, värden i ett interval
- Diskreta visualiseras: med stolpdoagram
- Kontinuerlig variabel visualisering: med histogram
- Lådiagram:
- Sannolikhetsteori: Vi antar att ett förseks genomförs  $n$  gånger obunden av varandra. En händelse  $A$  inträfar  $f$  gånger
  - Frekvenskvoteten:  $\frac{f}{n}$
  - Experimentellt bestört närmevärde: frekvenskvoten på  $p(A)$ 
    - \* Def: (Frekvensbaserad sannolikhet):  $p(A) \lim_{n \rightarrow \infty}$
    - \* Def: (Klassisk sannolikhet) Anta att fersek kan utföras på  $m$  olika olika sätt varav  $g$  är gynnsama (innebär  $A$ ). Då är  $p(A) = g/m$

- Utfallsrum:  $\Omega$  alla möjliga värden som slumpvariabeln kan ta
- Händelser: är delmängden av utfalsrummet
- Slumpvariabler: Stokastisk variabel

**Kotmogorovs axiom**

*I* För varje händelse  $A$  gäller att  $P(A) \geq 0$

*II* Sannolikhet för utfalsrummet är 1  $P(\Omega) = 1$

*III* Om  $A$  och  $B$  är händelser och  $A \cap B = \emptyset$

(oförsemliga händelser) gäller  $P(A \cup B) = P(A) + P(B)$

$$\Rightarrow P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

## 11.2 Sannolikheter och slumpvariabler

### Användbara räkneregler

- 1  $P(A^*) = 1 - P(A)$
- 2  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- 3  $P(A|B) = \frac{P(A \cap B)}{P(B)}$

### 11.2.1 Betingade sannolikheter

Type	Dygtig	Defekt	Tot
Äldre	170	10	180
Ny	115	5	120

$A$  = "Slumpmässigt vald produkt är dyglig"

$B$  = "Slumpmässigt vald produkt är tillverkad vid äldre maskin"

$$P(A) = \frac{285}{300} = 0.95$$

$C$  = Slumpmässigt vald produkt är dyglig givet att den är tillverkad vid äldre maskiner

$$P(C) = P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{170}{180} \approx 0.94$$

### 11.2.2 Kedjor av händelse

$$P(A \cap B \cap C) = P(C|A \cap B)P(A \cap B) = P(C|A \cap B)P(B|A)P(A)$$

### Bayes sats

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

### Lagen om total sannolikhet

$$P(B) = P(B|A)P(A) + P(B|A^*)P(A^*)$$

### 11.2.3 Oberonde händelser

$$P(A|B) = P(A)$$

Betingad sannolikhet:  $P(A \cap B) = P(A|B)P(B)$  om A och B är oberonder  $\Rightarrow$   
 $\Rightarrow P(A \cap b) = P(A)P(B)$

### Födelsedagsparadoksen

$$1 - \left( \prod_{k=1}^n \frac{365-k}{365} \right)$$

### Slumpvariablel

Är en function från utfalsrumet  $\Omega$  till någon mängd  $E$   $X : \Omega \rightarrow E$

$$X \in \{0, 1, 2, 3\}$$

$$P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) = 1 \text{ Kolmogorovs axiom}$$

$$\text{Sannolikhetsfaktor? är } P_X(x) \quad P_X(x) = P(X = x)$$

## 11.3 Fördelningar

Diskreta fördelningar	Kontinuerliga fördelningar
$\Omega = \{1, 2, 3\}$	$\Omega = [0, 1]$
$\Omega = \{0, 2, \dots\}$	$\Omega = \mathbb{R}$
Sannolikhetsfunktion $p_X(x) = P(X = x)$	Täthetsfunktion $f_X(x) = \int_a^b f_X(x)dx = P(a \leq x \leq b)$ $P(X = x) = 0$
$\sum_{x \in \Omega} p_X(x) = 1$	$\int_{\Omega} f_X(x)dx = 1$
$E[X] = \sum_{x \in \Omega} x p_X(x)$	$E[X] = \int_{\Omega} x f_X(x)dx$
$V[X] = E[X^2] - E[X]^2$	$V[X] = E[X^2] - E[X]^2$
$F_X(x) = \sum_{i \leq x} p_X(i)$ $= p(X \leq x)$	$F_X(x) = \int_{-\infty}^x f_X(x)dx = 1$ $= P(X \leq x)$ $= P(X < x)$

### 11.3.1 Binomial-fördelningar

Tillämpning: man utför något n antal gånger med sannolheten p att det lyckas.

Om  $X$  är binomialfördelad med paramter  $n$  och  $p$ . Då gäller

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, \dots, n$$

$$X \sim Bin(n, p)$$

$$E[X] = np$$

$$V[X] = np(1-p)$$

$$dbinom(x, n, p)$$

$$pbinom(x, n, p, lower.tail = FALSE)$$

### 11.3.2 Possion-fördelningar

Tillämpning: För att modellera sällsynta händelser.

Om  $X$  är possionsfördelad med paramter  $m$ . Då gäller

$$P(X = x) = \frac{m^x}{x!} e^{-m}, \quad x = 0, 1, 2, \dots, n$$

$$X \sim Po(\mu)$$

$$E[X] = \mu$$

$$V[X] = \mu$$

$$dpois(x, \mu)$$

$$ppois(x, \mu, lower.tail = FALSE)$$

### 11.3.3 Likformig/rektangulär-fördelningar

Tillämpning: Lika fördelade inom ett interval.

Om  $X$  är likformig på intervallet  $[a, b]$ . Då gäller

$$f_X(x) = \frac{1}{b - a}, \quad a \leq x \leq b$$

$$X \sim Re(a, b)$$

$$E[X] = (a + b)/2$$

$$V[X] = (b - a)^2/12$$

$$dunif(x, a, b)$$

$$punif(x, a, b)$$

### 11.3.4 Exponential-fördelningar

Tillämpning: Livslängd/väntetid.

Om  $X$  är exponentialfördelad med paramter  $a > 0$ . Då gäller

$$f_X(x) = \frac{1}{\lambda} e^{-x/\lambda}, \quad x \leq \text{lambda}$$

$$X \sim \text{Exp}(\lambda)$$

$$E[X] = 1/\lambda$$

$$V[X] = 1/\lambda$$

$$\text{dexp}(x, \lambda)$$

$$\text{pexp}(x, \lambda)$$

### 11.3.5 Normalfördelning-fördelningar

Tillämpning: Alltmöjligt.

Om  $X$  är normalfördelad med paramter  $\mu$  och  $\sigma^2$ . Då gäller

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad -\infty < x < \infty$$

$$X \sim N(\mu, \sigma^2)$$

$$E[X] = \mu$$

$$V[X] = \sigma^2$$

$dnormal(x, \mu, \sigma)$  täthetsfunktionen (behövs inte för kontinuerliga)

$pnorm(x, \mu, \sigma)$  fördelningsfunktion (behövs inte för kontinuerliga)

### 11.3.6 Läges och spridningsmått

#### Väntevärde

$$(\text{Diskret}) \quad E[X] = \sum_{x \in \Omega} x P_X(x)$$

$$(\text{Kontinuerlig}) \quad E[X] = \int_{\Omega} x f_X(x)$$

**varians**

Om  $X$  är slumpvariabler med väntevärde  $\mu$ . Då gäller

$$\text{(Diskret)} \quad V[x] = \sum_{x \in \Omega} (x - \mu)^2$$

$$\text{(Kontinuerlig)} \quad V[X] = \int_{\Omega} (x - \mu)^2 f_X(x) dx$$

**Alternativ form**

$$\text{(Diskret)} \quad E[X^i] = \sum_{x \in \Omega} x^i P_X(x)$$

$$\text{(Kontinuerlig)} \quad E[X^i] = \int_{\Omega} x^i f_X(x) dx$$

$$V[x] = E[x^2] - (E[x])^2$$

## 11.4 Olikheter

### 11.4.1 Markovs olikhet

Om  $X$  är ickenegativ ( $x \leq 0$ ) och  $a > 0$ . Då gäller

$$p(x \geq a) \leq \frac{E[x]}{a}$$

### 11.4.2 Thebysjovs olikhet (chebyshev)

Om  $X$  är en slumpvariabel med  $E[X] = \mu$  och  $V[X] = \sigma^2$ . Då gäller

$$p(|x - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

### 11.4.3 Fördelnings funktioner

$$\text{(Diskret)} \quad p(X \leq x) = F_X(x) = \sum_{i \leq x} P(X = i)$$

$$\text{(Kontinuerlig)} \quad p(X \leq x) = F_X(x) = \int_{-\infty}^x f_X(t) dt$$

### Normalfördelning

Vi kan inte integrera täthetsfunctionen -ingen stängd form för fördelings funktionen  
special fall:  $X \sim N(0, 1)$  (standard normalfördelning)

Vi betecknar fördelnings funktionen  $\Phi(x) = P(X \leq x)$

**kvarter**

För  $0 < x < 1$  defineras  $\alpha$ -kvantilen  $x_\alpha$  till  
slupvariabel  $x$  som en lösning till  
 $F_X(x_\alpha) = 1 - \alpha$

**11.4.4 Oberonede slupvariabel**

Två slumpvariabler  $x_1 \wedge x_2$  kallas oberonade om  
 $P(x_1 \in A \cap x_2 \in B) = P(x_1 \in A)P(x_2 \in B)$   
 för alla mängder  $A \subseteq \Omega_1$ ,  $B \subseteq \Omega_2$

$$\begin{aligned} E[Y] &= a + bE[x] \\ V[Y] &= b^2V[x] \end{aligned}$$

**räkneregler**

$$\begin{aligned} Y &= a_1x_1 + a_2x_2 + \dots + a_nx_n \text{ slupvariabler } x_i \text{ konstanter } a_i \\ E[Y] &= a_1E[x_1] + a_2E[x_2] + \dots + a_nE[x_n] \end{aligned}$$

**Exempel (505): räkneregler**

Låt  $X \sim N(10, 4)$ ,  $Y \sim N(3, 1)$  vara oberonade slumpvariabler.  
 Beräkna sannolikheten att  $X > 3Y$

$$\begin{aligned} \text{Låt } Z &= X - 3Y \Rightarrow E[Z] = E[X] - 3E[Y] = 10 - 3 * 3 = 1 \\ V[Z] &= E[X] + (-3)^2E[Y] = 10 = (-3)^2 * 3 = 13 \\ \text{Enligt normalfördelingen så får vi } Z &\sim N(1, 13) \\ \text{Därmed kan vi beräkna följande sannolikhet } P(X - 3Y > 0) &= P(Z > 0) = 1 - P(z \leq 0) \\ 1 - \Phi(-1/\sqrt{13}) &= 1 - \Phi(1 - \Phi(1/\sqrt{13})) = \Phi(0.28) = 0.61 \end{aligned}$$

**väntevärde av produkter**

Om  $x_1, \dots, X_n$  är oberonade då gäller  
 $E[x_1, \dots, X_n] = E[x_1] \dots E[x_n]$

### Special fall

$$\bar{x} = \frac{1}{n}(x_1, \dots, X_n)$$

där  $x$  är obunden och diktördelade med

$$E[x_i] = \mu, V[x_i] = \sigma^2$$

$$E[\bar{x}] = \mu, V[\bar{x}] = \frac{\sigma^2}{n}$$

### 11.4.5 Fördelning av summar

Binomialfördelning, Poissonfördelning, Linjärkombination av normalfördelade variabler.

### 11.4.6 Central gränsvärdessatsen (CGS)

Exakt fördelning för summar är svårt i allmänhet  
därmed använder man CGS

$$Y = X_1 + X_2 + \dots + X_n \\ Y \sim N(\mu_Y, \sigma_Y^2)$$

#### Example

$$X_i \sim Bin(1, 0.2)$$

$$Y = \sum_{i=1}^{30} X_i$$

$$P(Y \leq 8) = ?$$

$$CGS: Y \sim N(\mu_Y, \sigma_Y^2)$$

$$\mu_Y = E[Y] = 30 * E[X_i] = 30 * 0.2 = 6$$

$$\sigma_Y^2 = V[Y] = 30 * V[X_i] = 30(1 * 0.2 * 0.8) = 4.8$$

$$P(Y \leq 8) = pnorm(8, 6, sqrt(4.8)) \approx 0.82$$

$$P\left(\sum_{i=1}^{30} X_i \leq 8\right) \approx 0.87$$

## 11.5 Simulering av slumptal

### 11.5.1 äkta slumpprämsiga tal

Hårdvara genererade tal:

- Tärningar, roletthjul ..
- Radiaktivit sönderfall
- Atmosfäriskt brus (random.org)

### 11.5.2 Pseudoslumpmässiga tal

Dator genererade slump tal. Kommer att upredasig.

- Tar in en seed som input för att generera slumptal
- Default seed är oftast tid
- det är diterministiska
- Har en period med nya tal sedan så uprepas det sig

#### Von Neumann

1. Väljer seed:  $u_0 = 0.1111$
2. Skapar  $y_0 = 1111$
3. Beräknar  $y_0^2 = 1234321$
4. Fyller på från venster med noll för att få 8 siffror 01234321
5. Skapar genom att ta det fyra mittersta siffrorna  $y_1 = 2343$
6.  $0.2343 \Rightarrow y_1^2 = 5489649 \Rightarrow y_2 = 4896 \Rightarrow u_2 = 0.4896$

#### Kongruens

$$V_{n+1} = aV_n + b \pmod{c}$$

$$\text{Vi är ett heltal mellan } 0 \text{ och } c-1 \text{ och } u_1 = \frac{V_i}{c}$$

a,b,c måste väljas noggrant (talteori, c måste vara ett primtal)  
vanliga val är  $a = 7^7 = 16807$ ,  $b = 0$ ,  $c = 2^{31} - 1 = 2147483647$   
hat svagheter, används inte längre

#### XOR-generator

Extrem snabb, lätt att förstå

1. Väljer seed:  $m$  binära bits (heltal mellan 0 och  $2^m - 1$ )
2. Skifta alla bits  $l$  steg åt vänster och fyll i från höger med nollor
3. XOR med seed och det skiftade talet (seed update)
4. Skifta seed update  $m - l = r$  steg åt höger och fyll i nollor från vänster
5. XOR med seed update och högerskiftet
6. Konvertera till ett tal mellan 0 och 1

### Mersenne Twister

#### Exempel (505): räkneregler

Givet 5 psudoslumpmässiga tal från  $Re[0, 1]$

$$u_1 = 0.8147, u_2 = 0.9058, u_3 = 0.1270, u_4 = 0.9134, u_5 = 0.634$$

Simulera 5 dosernar? från slumpvariablen  $X$  med

$$f_X(x) = \frac{x}{2}, 0 \leq x \leq 2$$

Beräna  $E[x]$  och jämför med medvärder av de sammulerade dosetaner

Tar fram primitiva funktionen  $F_X(x) = \int_0^x \frac{t}{2} dt = [t^2/4]_0^x = \frac{x^2}{4}$

$$(Tar fram inversen) y = \frac{x^2}{4} \Rightarrow x = 2\sqrt{y}$$

$$x_1 = F_x^{-1}(u_1) = 2\sqrt{x} = 1.8052$$

$$x_2 = F_x^{-1}(u_2) = 2\sqrt{x} = 1.9035$$

$$x_3 = F_x^{-1}(u_3) = 2\sqrt{x} = 0.7127$$

$$x_4 = F_x^{-1}(u_4) = 2\sqrt{x} = 1.9114$$

$$x_5 = F_x^{-1}(u_5) = 2\sqrt{x} = 1.5905$$

$$\text{Beräknar väntevärdet } E[x] \int_0^2 x \frac{x}{2} dx = [\frac{x^3}{6}]_0^2 = \frac{8}{6} = \frac{4}{3} \approx 1.33$$

## 11.6 Statistikens grunder

### 11.6.1 Allmänt

Skattning av en okänd parameter  $\theta$  från en familj  $F_X(\theta)$  är en funktion

$$\hat{\theta} = t = g(x_1, x_2, \dots, x_n)$$

### 11.6.2 Medelfel

$\sigma^2 \wedge p$  kan vara okända

Vi kan definera medelfellet genom att använda skattningen

$$s = \hat{\sigma} \wedge \hat{p}$$

$$V[\hat{\mu}] = \frac{\sigma^2}{n}$$

$$V[\hat{p}] = \frac{\hat{p}(1 - \hat{p})}{n}$$

$$d[T_\mu] = \frac{s}{\sqrt{n}}$$

$$d[T_p] = \sqrt{\frac{\hat{p}(1 - \hat{p})}{n}}$$

Utifrån medelfelet definierar vi konsistens och effektivitet

**Konsistans**  $V[T] \rightarrow 0$  då  $n \rightarrow \infty$

**Effektivitet** Givet estimatoren  $T_1, T_2$  så är  $T_1$

effekten att  $T_2$  om  $V[T_1] < V[T_2]$  ( $D[T_1] < D[T_2]$ )

#### Exempel

Vi beräknar variansen på  $T_\mu$  och  $T_p$

$$V[T_\mu] = V[1/n(x_1, x_2, \dots, x_n)]$$

$$V[T_\mu] = 1/n^2(V[x_1] = V[x_2] + \dots + V[x_n])$$

$$V[T_\mu] = \frac{1}{n^2}n\sigma^2 = \frac{\sigma^2}{n}$$

$$V[T_p] = V[x/n]$$

$$V[T_p] = \frac{1}{n^2}V[x]$$

$$V[T_p] = \frac{1}{n^2}np(1 - p)$$

$$V[T_p] = \frac{p(1 - p)}{n}$$

### 11.6.3 Skattning av variansen

$N(\mu, \sigma^2)$  - Vill skatta  $\sigma^2$  det går att visa att

$$S^2 = \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

är en venteriktig skattning av variansen

Vi kan visa att

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2 + (n_3 - 1)S_3^2}{(n_1 - 1) + (n_2 - 1) + (n_3 - 1)}$$

är en väntevärdsriktig skattning av  $\sigma^2$

**Def**

Låt  $A$  och  $B$  vara funktioner av  $x_1, x_2, \dots, x_n$  så att  
Då kallas  $[A, B]$  ett  $100(1 - \alpha)$ -procent  
konfidensintervall för  $\theta$  (med konfidensgrad  $1 - \alpha$ )

**11.6.4 Väntevärdsriktig****Exempel**

$$X \sim N(m_1 - m_2, 4), Y \sim N(m_1 + m_2, 5)$$

a. Visa att  $\hat{m}_1 = (X + Y)/2$  är väntevärdsriktig skattning av  $m_1$

$$E(\hat{m}_1) = E[(X + Y)/2] = 1/2(E[X] + E[Y]) = 1/2(m_1 - m_2 + m_1 + m_2) = \frac{1}{2}2m_1 = m_1$$

därmed så är den väntevärds riktig

b. Beräkna standardavvikelsen för  $\hat{m}_1$

$$D(\hat{m}_1) = \sqrt{V(\hat{m}_1)} = \sqrt{V\left(\frac{x+y}{2}\right)} = \sqrt{\frac{1}{4}(V[X] + V[Y])} = \frac{\sqrt{9}}{2} = \frac{3}{2}$$

**11.6.5 Konfidensintervall för  $\mu$  från  $N(\mu, \sigma^2)$  med känt  $\sigma$** 

$$\text{Estimatorn } \hat{\mu} = \bar{x} = \frac{1}{n}(X_1 + X_2 + \dots + X_n)$$

$$\bar{X} \sim N(\mu, \sigma^2/n)$$

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1)$$

$$A = \bar{X} - \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

$$B = \bar{X} + \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

### 11.6.6 Example

vid tillverkningsprocessen av axlar av rundstål kontrollers diametern.

Följande diametrar (mm) uppmätttes:

30.02, 30.12, 30.07, 29.95, 30.05, 29.90, 30.01

Konstruera ett konfidensintervall, med konfidensgrad 0.95, för den förväntade diametern.

$$n = 7, \alpha = 0.05$$

$$\bar{X} = \frac{30.02 + 30.12 + 30.07 + 29.95 + 30.05 + 29.90 + 30.01}{n} = \frac{210.12}{7}$$

$$s = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{6} ((30.02 - \bar{x})^2 + (30.02 - \bar{x})^2)$$

### 11.6.7 Konfidensintervall för $p$ från binomialfördelad

Finns många alternativa metoder

$$Bin(n, p) \sim N(np, np(1-p)) \text{ om } np(1-p) \geq 10$$

$$\Rightarrow I_p = [\hat{p} \pm \lambda \sqrt{\hat{p}(1-\hat{p})/n}]$$

konfidensintervall för  $\theta$  (med konfidensgrad  $1 - \alpha$ )

### 11.6.8 Konfidensintervall för skillnad i väntevärde

Vill ofta jämföra två grupper

$$x_1, \dots, x_n \text{ från } N(\mu_1, \sigma_1^2)$$

$$y_1, \dots, y_n \text{ från } N(\mu_2, \sigma_2^2)$$

Söker  $\mu_1 - \mu_2$ . Om detta intervall inhåller 0 då kan vi med konfidensgrad  $1 - \alpha$  säga att det är skillnad på väntevärdena

### Okända varianser

$$s_p^2 = \frac{(n-1)s_1^2 + (m-1)s_2^2}{(n-1) + (m-1)}$$

$$\frac{\bar{X} - \bar{Y} - (\mu_1 - \mu_2)}{s_p \sqrt{1/n + 1/m}}$$

$$I_{\mu_1 - \mu_2} = [\bar{X} - \bar{Y} \pm t_{\alpha/2}(n+m-2)s_p \sqrt{1/n + 1/m}]$$

### 11.6.9 Ensidiga intervall

Låt  $A$  och  $B$  vara funktioner av  $x_1, \dots, x_n$

För ett nedåt begänsat konfidensintervall gäller

$$p(\theta \geq A) = 1 - \alpha$$

För ett uppåt begrensat gäller

$$p(\theta \leq B) = 1 - \alpha$$

### 11.6.10 Stickprov i par

Fotgängare	A	B
1	43	32
2	81	90
3	11	7
4	49	31
5	22	26
6	143	168
7	24	31
8	56	39
9	31	29
10	53	57

Tiden det tar för A respektive B att upptäcka forgängaren baserat står i tabellen

Konstruera ett lämpligt konfidensintervall baserat på datan. Vilken algoritm borde användas

Vi väljet ett konfidensintervall på 95%,  $Z = A - B$

$$\bar{Z} = 3/10 = 0.3, s_Z^2 = \frac{1}{10-1} \sum_{i=1}^{10} (Z_i - \bar{Z})^2 = 13.08, t_{0.025}(t) = 2.26$$

$$I_\mu = [\bar{Z} \pm t_{0.025}(t) \frac{s}{\sqrt{n}}]$$

$$= [-9.05, 9.65]$$

0 finns i intervallet, med konfidensgrad 0.95 kan ingen skillnad påvissas  
mellan algoritmerna samla in mer data, alternativt använd vilken algoritm som helst

## 11.7 Regression

### 11.7.1 Modell

Givet observationsparen  $x_1, \dots, x_n$  och  $y_1, \dots, y_n$  ansätter man följade modell

$$Y_i = m + kx_i + \epsilon_i \text{ Där } \epsilon_i \sim N(0, \sigma^2)$$

### 11.7.2 Modellens giltighet

$$\text{Konrelationskoeficent: } r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$$

#### Förklaringsgrad

$$R^2 = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$$

#### Residualer

$$e_i = y_i - (\hat{m} + \hat{k}x_i)$$

Residualer bör uppfylla visa krav

- (1) konstat varians, oberonde av x
- (2) Residualerna bör vara oberonde av varandra
- (3) Residualerna bör vara normalfördelade

Vi bedömmmer dessa visuelt

- (i) Plottar residualerna i ett histogram, där vi kan se om det kan vara normal fördelat
- (ii) Plottar residualerna i q-q plot  
På x-axeln kvatiler från en s..fördelning  
På y-axeln kvatiler från residualerna
- (iii) Ritar ett spridnings diagram över x-värderna mot residualerna  
ej ett mönster  $\Rightarrow$  från X
- (iv) Ritar ett spridnings diagram över residualerna mot det förutspodda y-värdet  
vill se ett jämt utpridning utan mönster

### 11.7.3 Användning ag modellen

Knfidensintervallet för parametern k

$$V[\hat{m}] = \frac{\sigma^2}{n} \frac{1}{S_{xx}} \sum_{i=1}^n x_i^2$$

$$V[\hat{k}] = \frac{\sigma^2}{S_{xx}}$$

## 11.8 Stokastiska processer

### 11.8.1 Bornulli processer

Kommunikationskanal, överföring av slumpmärsiga data. Tiden är uppdelad i luckor (slots)  $k = 1, 2, 3, \dots, n$  varje lucka kan hantera ett paket

### 11.8.2 Poisson processer

**Proposition:**

För en poissonprocess  $N(t), t \geq 0$  gäller att

- (a) inkreten  $N(t_1), N(t_2) - N(t_1), \dots, N(t_k) - N(t_{k-1})$  är oberoende slumpvariabel för alla  $0 \leq t_1 \leq \dots \leq t_k$  och
- (b)  $N(t) - N(s) \sim Po(\lambda(t-s))$

**Ex:**

Man att antal fel på en kommunikationskabel är 1.7. Total antal fel beskrivs av en poisson process med parameter  $\lambda = 1.7$ . Vad är sannolikheten att det finns mer än två fel på 0.5 kilometer?

$$\begin{aligned} N(0.5) &\sim Po(\lambda(0.5 - 0)) = Po(0.85) \Rightarrow P(N(0.5) > 2) = 1 - P(N(0.5) \leq 2) \\ &= 1 - ppois(2, 0.85) \approx 0.0549 \end{aligned}$$

### Förtunning

Låt  $N(t)$  vara en poisson process med parameter  $\lambda$

Låt  $J_n, n \in \mathbb{N}$  vara en följd av i.i.d.  $Be(p)$

$$P(J_i = 1) = p, P(J_i = 0) = 1 - p$$

$$M(t) = \sum_{k=1}^{\infty} \mathbf{1}_{T_k \leq t} J_k$$

Vi kan tolka att vi skippar vissa händelser

### Superposition

Låt  $N(t)$  och  $N_2(t)$  vara en poisson process med parameter  $\lambda_1$  och  $\lambda_2$

Vi vill visa att  $M(t) = N_1(t) + N_2(t)$  är en poisson process med parameter  $\lambda_1 + \lambda_2$

**Ex:**

Antal inkomade samtal till en mobil kan beskrivas som en poissonprocess med paramter 0.5 per time och antal sms som en poissonprocess medparamter 2 två fel på 0.5 kilometer?

- (a) sannoliket att ingen kommunikationskabel inkommer på en time  $N_1(t)$  med parameter  $\lambda_1 = 2$ , -antal sms
- $N_2(t)$  med parameter  $\lambda_2 = 0.5$ , -antal samtal
- $M(t) = N_1(t) + N_2(t)$  med parameter 2.5, -antal kommunicationer
- $= 1 - ppois(2, 0.85) \approx 0.0549$

### Spatial process

En samling punkter i en region  $S \subseteq \mathbb{R}^2$

Om  $N(A)$  räknar antal punkter i en mängd  $A$  (där  $A$  är mätbar)  
då gäller att  
 $*N(A) \sim Po(\lambda|A|)$   
 $*$  Om  $A \cap B = \emptyset$  är  $N(A)$  och  $N(B)$  oberonnde

### M/M/ $\infty$ -modellen

$N(t)$  -Poissonprocess, ankomster upp till tid t  
ankomsterna tar tid att behandla  
hur många ankomster pågår vid en viss tid?  
 $X_t$  -antal pågående ankomster vid tid t  
Vi antar att varje ankomst behandlas på  $\exp(\lambda)/tid$

## 11.9 Markovkedjor

En Stokastisk process  $X_n$  i diskret tid med diskret tillståndsrum  $E$  kallas en Markovkedja om den har Markovegenskapen  
 $P(X_n = x_n | X = x_0, X_1 = x_1, \dots, X_{n-1} = x_{n-1}) = P(X_n = x_n | X_{n-1} = x_{n-1}), \forall x_1 \in E \wedge n \geq 1$   
och övergångssanolikheten  $p_{xy} = P(X_n = y | X_{n-1} = x)$  är oberonnde från  $n$   
Vi fokuserar på  $E = 0, 1, \dots, r$  och  $E = \mathbb{Z}_{\geq 0}$

Övergångsmatris:

$$\begin{pmatrix} p_{00} & p_{01} & \dots & p_{0r} \\ p_{10} & p_{11} & \dots & p_{1r} \\ \vdots & \vdots & \dots & \vdots \\ p_{r0} & p_{r1} & \dots & p_{rr} \end{pmatrix}$$

### 11.9.1 Ehrenfestmodellen

Oavsett startpunkt tenderar kedjan mot ett ekvilibrium

Övergångsmatris:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1/r & 0 & 1 - 1/2 & 0 & \dots & 0 & 0 \\ 0 & 2/r & 0 & 1 - 2/r & \dots & 0 & 0 \\ \vdots & & & & & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 1 & 0 \end{pmatrix}$$

#### Def: stationär

En fördelning  $\pi$  kallas stationär för en Markovkedja med övergångsmatris  $P$  om den löser ekvationssystemet

$$\pi = \pi P$$

- $\pi$  är en egenvektor för  $P$  med egenvärde 1
- Om  $\pi$  är ursprungsfördelningen  $p^0 = \pi \Rightarrow p^n = \pi, \forall n \geq 0$

#### Def: asymptotisk

En fördelning  $\pi$  är en asymptotisk fördelning för Markovkedjan  $X_n$  om  $\lim_{n \rightarrow \infty} P(X_n = k) = \pi_k, \forall k \geq 0$   
oberende avursprungsfördelningen  $p^0$

Asymptotiska fördelningar är alltid stationära, dock inte tvärt om

#### Def: Irreducibel, Aperiodisk

En markovkedja  $X_n$  kallas

- Irreducibel om  $P(X_n = j | X_0 = i) > 0$  för något  $n$  och alla  $i, j \in E$
- Aperiodisk om största gemensamma delaren av mängden  $n : P(X_n = i | X_0 = 1) > 0$  är 1 för alla  $i$

Om en kedja är irreducibel och något  $p_{ii} > 0$  är kedjan aperiodisk

#### Def: Tillståndsrummet

Låt  $X_n$  vara aperiodisk och irreducibel

- (1) Om tillsåndsrummet är ändligt finns en unik stationär fördelning som också är asymptotisk
- (2) Om tillsåndsrummet är oändligt, då om en stationär fördelning existerar är den unik och asymptotisk

**Exempel: Markovkedjor**

$$P = \begin{pmatrix} 3/4 & 0 & 1/4 \\ * & 1/3 & 0 \\ 1/4 & 1/2 & ** \end{pmatrix}$$

- (a) Ange \* och \*\*
  - (b) Rita övergångsgrafen
  - (c) argumentera för att kedjan är aperiodisk och irreducibel
  - (d) Bestäm den stationära fördelningen
- (a)  $* = 1 - (1/3 + 0) = 2/3$ ,  $** = 1 - (1/4 + 1/2) = 1/4$
- (b) Rita övergångsgrafen
- (c) Som man ser på övergångsgrafen att den man kan ta sig till alla positioner därmed så är den irreducibel vilket också medföljer aperiodisk
- (d)  $(\pi_0 \pi_1 \pi_2) = (\pi_0 \pi_1 \pi_2) \cdot p \begin{pmatrix} 3/4 & 0 & 1/4 \\ 2/3 & 1/3 & 0 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}$
- $$\begin{cases} \pi_0 = 3/4\pi_0 + 2/3\pi_1 + 1/4\pi_2 \\ \pi_1 = 0\pi_0 + 1/3\pi_1 + 1/2\pi_2 \\ \pi_2 = 1/4\pi_0 + 0\pi_1 + 1/4\pi_2 \end{cases}$$
- $\pi_2 = \pi_0/3$ ,  $\pi_1 = \pi_0/4$
- $\pi = (12\ 3\ 4)$  Normalisering:  $\pi = (12/19\ 3/19\ 4/19)$

**11.9.2 Google-kedjan**

Google skapar en graph av alla sidor  
Tillståndet efter n steg beskrivs av en Markovkedja  
med denna övergångsmatrisen  
Google letar efter den asymptotiska fördelningen på  
kedjan och rankar sidorna i sökresultatet enligt sannolikheterna  
i den asymptotiska fördelningen

**11.9.3 Hashfunktioner**

En funktion  $h$  som tar  $n$  visare och sparar som någon av  $m$  möjliga hasvärden ( $m < n$ )

**11.9.4 Kollisionmodell**

Modeleras med Bernoulli-processen

### 11.9.5 Markov Buffer/Markovkö

$X_n$  -anatal packet som anländer under slot  $n$

$Q_n$  -anatal packet i kö slutet av slot  $n$