



Introduction & Language Guessing

Data Structures and Algorithms for CL III, WS 2019-2020

Corina Dima

`corina.dima@uni-tuebingen.de`

DSA-CL III course overview

What is Data Structures and Algorithms for Computational Linguistics III?

- An intermediate-level survey course
- Programming and problem solving, with applications
 - **Data structure**: method for storing information
 - **Algorithm**: method for solving a problem
- Second part focused on Computational Linguistics

Prerequisites

- Data Structures and Algorithms for CL I
- Data Structures and Algorithms for CL II
- Module: ISCL-BA-07, Advanced Programming

DSA-CL III course overview

- **Lecturers**

- Corina Dima
- Çağrı Çöltekin

- **Tutors**

- Kevin Glocker
- Teslin Roys

- **Slots**

- Monday 10:15 – 11:45 (lecture)
- Wednesday 10:15 – 11:45 (lecture)
- Friday 8:15 – 12 (lab)

- **Course website:** <https://dsacl3-2019.github.io>

Coursework and grading

- Reading material for most lectures
- **Programming assignments: 60%**
 - 2 ungraded introductory assignments
 - 6 graded assignments, one every 2 weeks
 - **60% of the grade: the best 5 assignments**
 - Graded assignments due every other Monday, 11pm, only via electronic submission (GitHub Classroom)
 - Collaboration/lateness policy: see web
- **Written exam: 40%**
 - Midterm practice exam **0%**
 - Final exam **40%**

Honesty Statement

- Feel free to **cooperate** on assignments that are **not graded**
- **Graded** assignments **must be your own work. Do not:**
 - Copy a program (in whole or in part)
 - Give your solution to a classmate (in whole or part)
 - Get so much help that you cannot honestly call it your own work
 - Receive or use outside help
- Sign your work with the **honesty statement** (provided on the website)
- Above all: **You are here for yourself, practice makes perfect**

Organizational issues

- Presence

- A presence sheet is circulated **purely** for statistics
- Experience: those who do not attend the lectures or do not make the assignments end up failing the course
- Do not expect us to answer your questions if you were not at the lectures

- Office hours

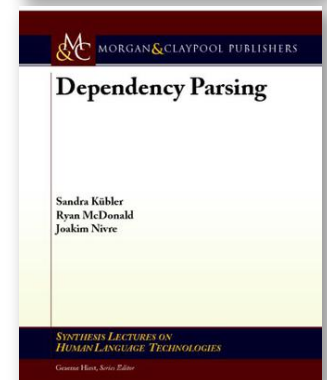
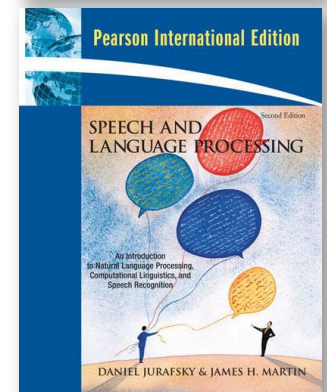
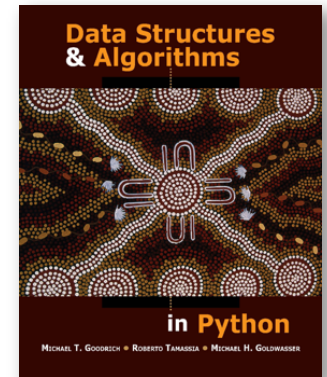
- Office hour: **Wednesday: 14:00-15:00**, please make an **appointment!**
- Please ask your questions about the material presented in the lectures during the lectures – everyone benefits
- Solutions to the assignments will be discussed after the lab deadline has passed

Registration

- Do the first assignment, A_0 (see website), until October 23rd
- Walk-through: work on an assignment with GitHub Classroom

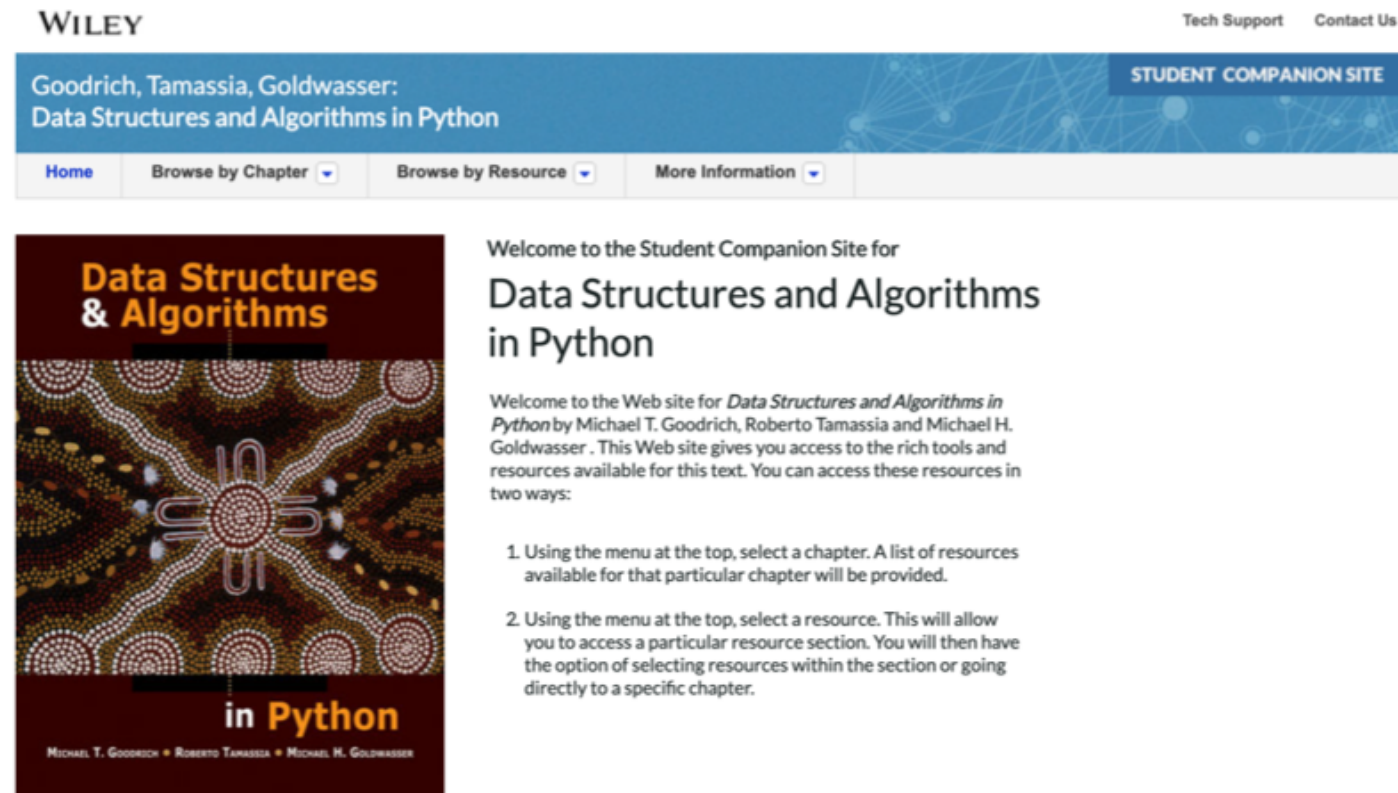
Resources (textbooks) – required reading

- *Data Structures & Algorithms in Python* by Michael Goodrich, Roberto Tamassia and Michael Goldwasser, 2013, Wiley
 - available in the university network:
<https://ebookcentral.proquest.com/lib/unitueb/detail.action?docID=4946360>
- *Speech and Language Processing*, Dan Jurafsky and James Martin, 2nd Edition, 2008, Prentice Hall
 - Draft chapters of the 3rd edition available
 - See <https://web.stanford.edu/~jurafsky/slp3/>
- *Dependency Parsing*, Sandra Kübler, Ryan McDonald and Joakim Nivre, 2009, Morgan and Claypool



Resources (web)

- Book site for the first part of the class: <http://bcs.wiley.com/he-bcs/Books?action=index&bcsId=8029&itemId=1118290275>
- Source code
- Hints for solving exercises



The screenshot shows the Wiley Student Companion Site for the book "Data Structures and Algorithms in Python" by Goodrich, Tamassia, and Goldwasser. The page features a blue header with the Wiley logo and navigation links for "Tech Support" and "Contact Us". Below the header is a navigation bar with "Home", "Browse by Chapter", "Browse by Resource", and "More Information" options. The main content area displays the book cover on the left and a welcome message on the right. The book cover has a dark red background with a complex, symmetrical pattern of white and gold dots and lines, and the title "Data Structures & Algorithms in Python" in white and gold text. The authors' names are listed at the bottom of the cover. The welcome message on the right reads: "Welcome to the Student Companion Site for Data Structures and Algorithms in Python. Welcome to the Web site for *Data Structures and Algorithms in Python* by Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser. This Web site gives you access to the rich tools and resources available for this text. You can access these resources in two ways: 1. Using the menu at the top, select a chapter. A list of resources available for that particular chapter will be provided. 2. Using the menu at the top, select a resource. This will allow you to access a particular resource section. You will then have the option of selecting resources within the section or going directly to a specific chapter."



Why Study Algorithms?

Their impact is broad and far-reaching.

Internet. Web search, packet routing, distributed file sharing, ...

Biology. Human genome project, protein folding, diagnosis, ...

Computers. Circuit layout, file system, compilers, ...

Computer graphics. Movies, video games, virtual reality, ...

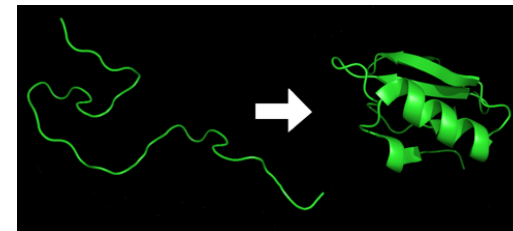
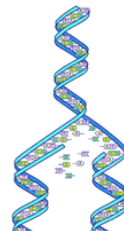
Security. Cell phones, e-commerce, voting machines, ...

Multimedia. MP3, JPG, DivX, HDTV, face recognition, speech recognition, ...

Social networks. Recommendations, news feeds, advertisements, ...

Physics. N-body simulations, particle collision simulation, ...

...



Write text? (soon)

- **OpenAI GPT-2**, a transformer-based language model, generates text samples in response to a sample input that is human-written
- It is able to adapt to the style and the content of the provided sample
- Trained on **40GB of Internet text**
- Objective – predict the next word given all the previous words in some text
- More on:
<https://openai.com/blog/better-language-models/>

```
SYSTEM PROMPT (HUMAN-WRITTEN)  In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."
```



Why Study Algorithms?

The Nobel Prize in Chemistry 2013



© Nobel Media AB. Photo: A. Mahmoud

Martin Karplus

Prize share: 1/3



© Nobel Media AB. Photo: A. Mahmoud

Michael Levitt

Prize share: 1/3



© Nobel Media AB. Photo: A. Mahmoud

Arieh Warshel

Prize share: 1/3

The Nobel Prize in Chemistry 2013 was awarded jointly to Martin Karplus, Michael Levitt and Arieh Warshel "for the development of multiscale models for complex chemical systems."

- They are instruments for developing new research

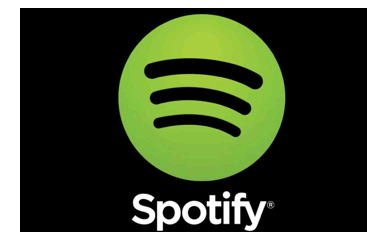
The computer – your Virgil in the world of atoms

Chemists used to create models of molecules using plastic balls and sticks. Today, the modelling is carried out in computers. In the 1970s, **Martin Karplus**, **Michael Levitt** and **Arieh Warshel** laid the foundation for the powerful programs that are used to understand and predict chemical processes. Computer models mirroring real life have become crucial for most advances made in chemistry today.



Why Study Algorithms?

- It is a profitable endeavor



What is Ahead?

Lecture schedule

The course plan is subject to change.

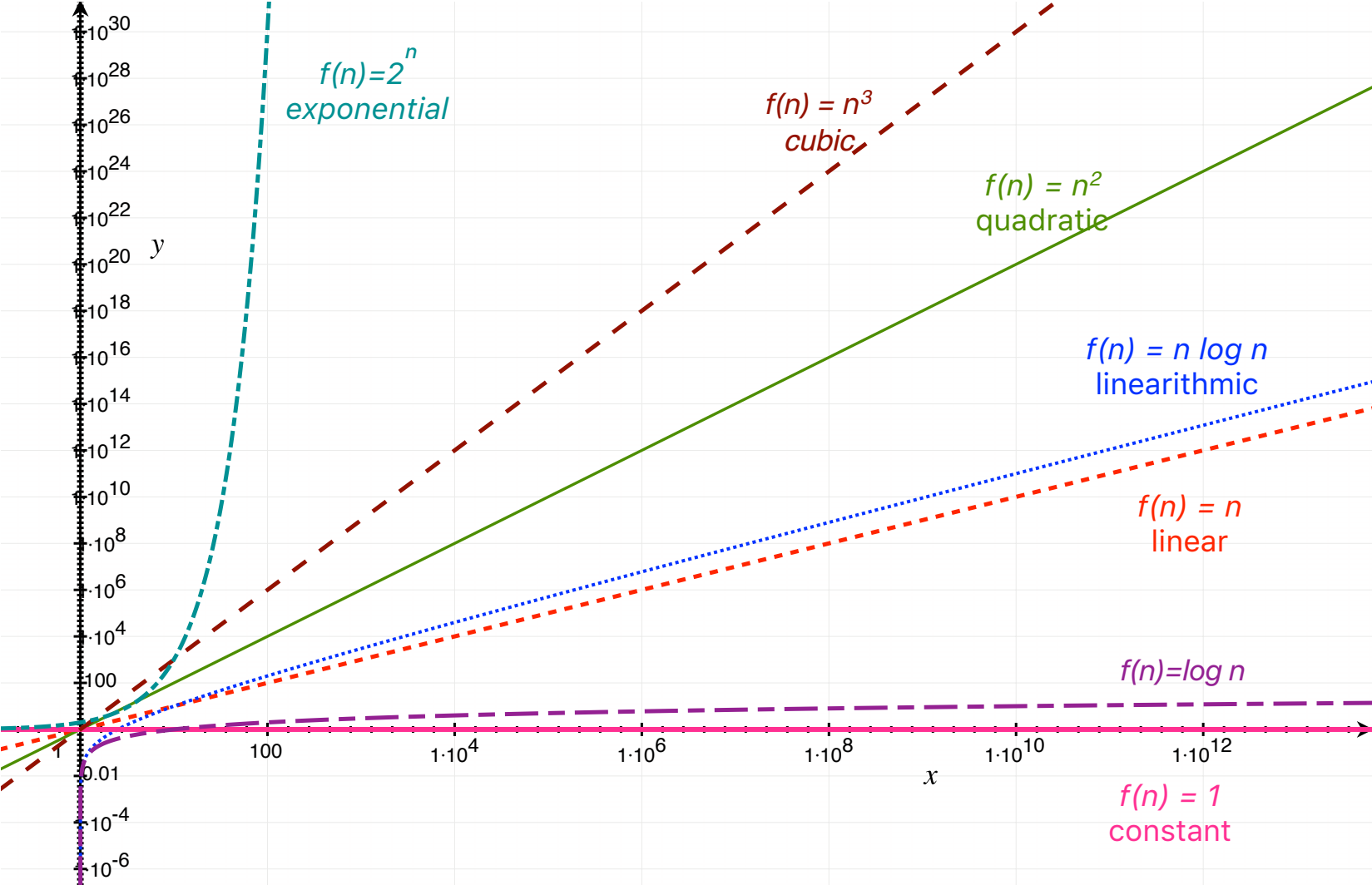
Week	Monday (lecture)	Wednesday (lecture)	Friday (lab)
01	Oct 21 <i>Introduction & Language Guessing</i>	Oct 23 <i>Analysis of Algorithms</i>	Oct 25 <i>Introduction to Python, Assignment 0.1 (ungraded)</i>
02	Oct 28 <i>Sorting: Insertion Sort & Quicksort</i>	Oct 30 <i>Priority Queues, Binary Heaps & Heapsort</i>	Nov 1 <i>No class</i>
03	Nov 04 <i>String Distance Measures</i>	Nov 06 <i>Tries</i>	Nov 08 <i>Graded Assignment 1</i>
04	Nov 11 <i>Undirected Graphs</i>	Nov 13 <i>Undirected Graphs (cont'd)</i>	Nov 15 <i>Graded Assignment 1 (cont'd)</i>
05	Nov 18 <i>Directed graphs</i>	Nov 20 <i>Directed graphs (cont'd)</i>	Nov 22 <i>Graded Assignment 2</i>
06	Nov 25 <i>Minimum Spanning Trees</i>	Nov 27 <i>Shortest Paths</i>	Nov 29 <i>Graded Assignment 2 (cont'd)</i>



What is Ahead? (cont'd)

07	Dec 02	Dec 04	Dec 06 <i>Graded Assignment 3</i>
08	Dec 09	Dec 11	Dec 13 <i>Graded Assignment 3 (cont'd)</i>
09	Dec 16	Dec 18	Dec 20 <i>Graded Assignment 4</i>
Sem. break	<i>No class</i>	<i>No class</i>	<i>No class</i>
10	Jan 06 <i>No class</i>	Jan 08	Jan 10 <i>Graded Assignment 4 (cont'd)</i>
11	Jan 13	Jan 15	Jan 17 <i>Graded Assignment 5</i>
12	Jan 20	Jan 22	Jan 24 <i>Graded Assignment 5 (cont'd)</i>
13	Jan 27	Jan 29	Jan 31 <i>Graded Assignment 6</i>
14	Feb 03 <i>General Summary/Q&A</i>	Feb 05 <i>Discuss Practice Exam</i>	Feb 07 <i>Exam</i>

Complexity



Sorting



Bundesarchiv, Bild 183-22350-0001
Foto: Junge, Peter Heinz | 20. November 1953

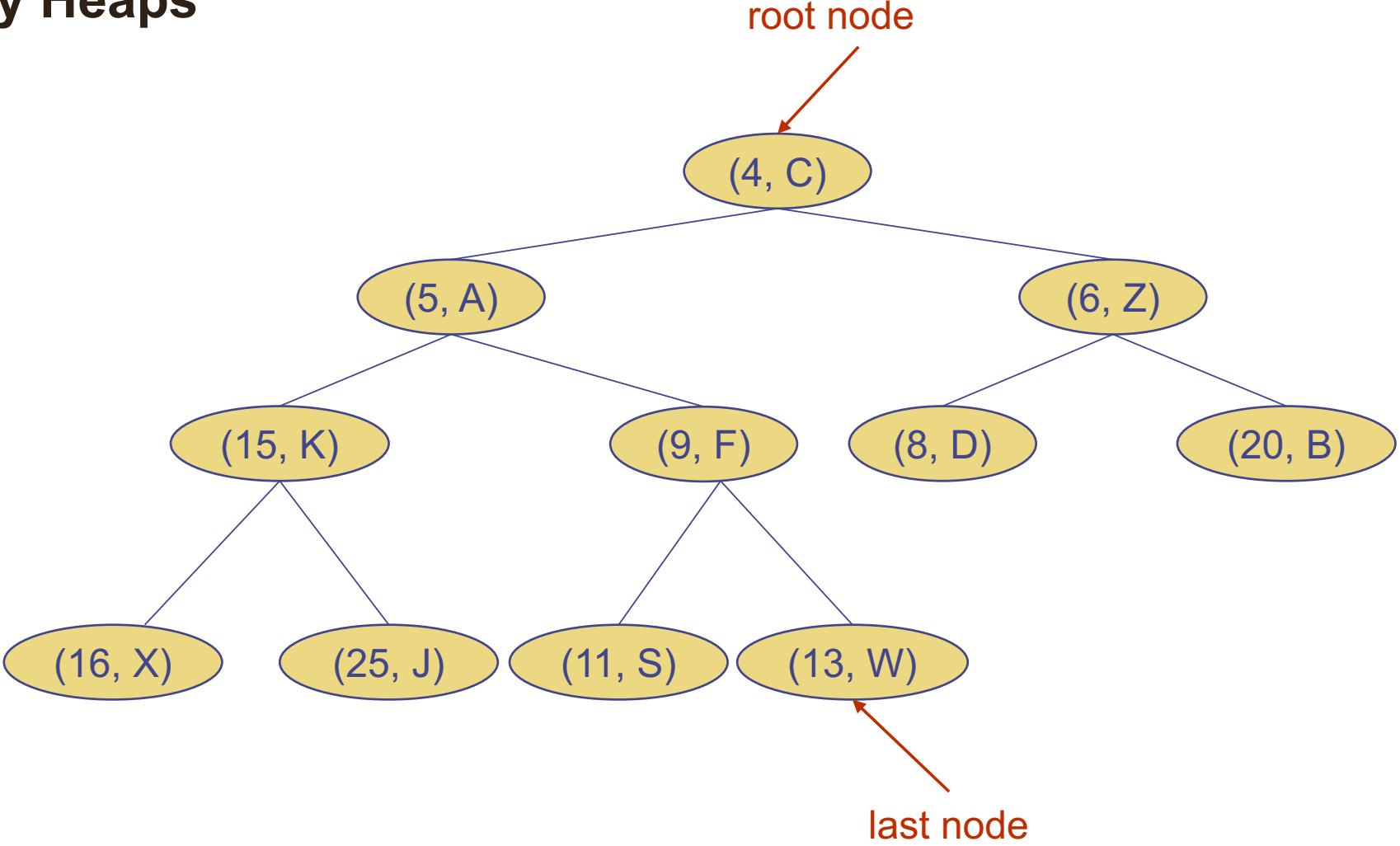
https://en.wikipedia.org/wiki/File:Bundesarchiv_Bild_183-22350-0001,_Berlin,_Postamt_O_17,_P%C3%A4ckchenverteilung.jpg



Priority Queues

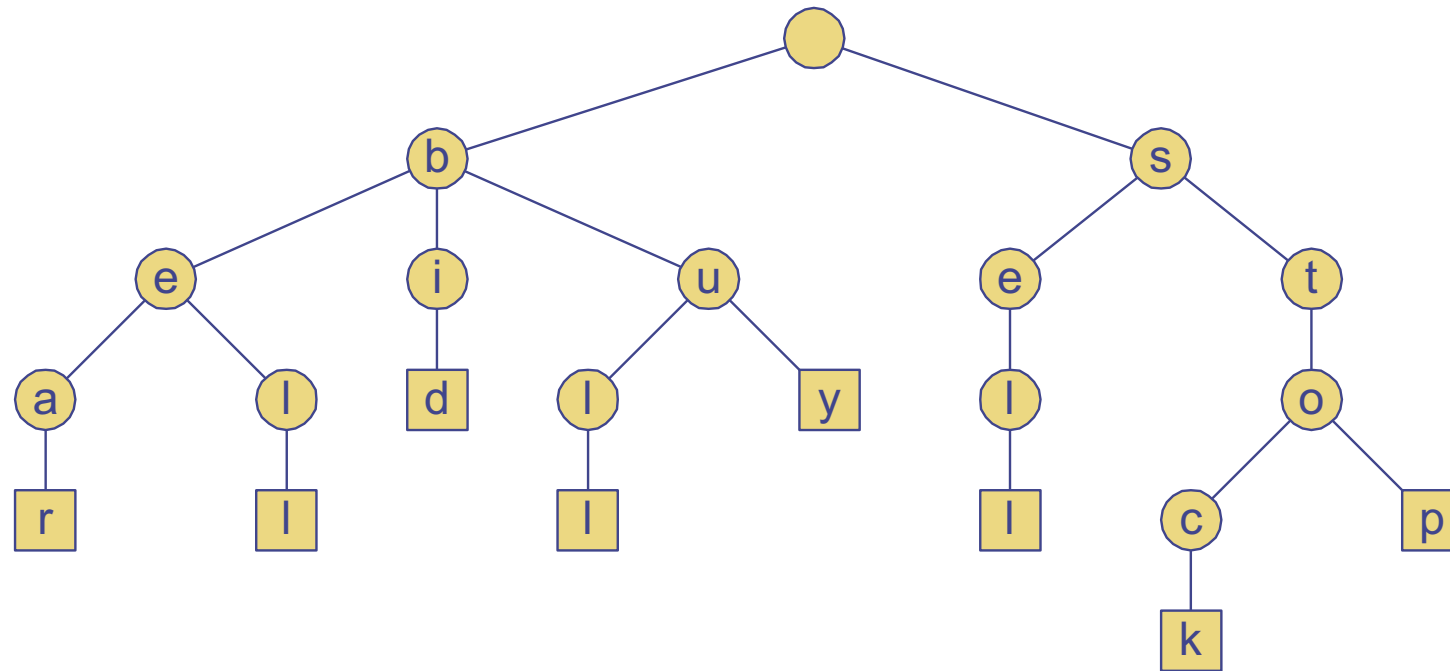
Operation	Return Value	Priority Queue
P.add(5,A)		{(5,A)}
P.add(9,C)		{(5,A), (9,C)}
P.add(3,B)		{(3,B), (5,A), (9,C)}
P.add(7,D)		{(3,B), (5,A), (7,D), (9,C)}
P.min()	(3,B)	{(3,B), (5,A), (7,D), (9,C)}
P.remove_min()	(3,B)	{(5,A), (7,D), (9,C)}
P.remove_min()	(5,A)	{(7,D), (9,C)}
len(P)	2	{(7,D), (9,C)}
P.remove_min()	(7,D)	{(9,C)}
P.remove_min()	(9,C)	{}
P.is_empty()	True	{}
P.remove_min()	"error"	{}

Binary Heaps



Tries

- Example: standard trie for the set of strings $S = \{ \text{bear, bell, bid, bull, buy, sell, stock, stop} \}$



Undirected Graphs

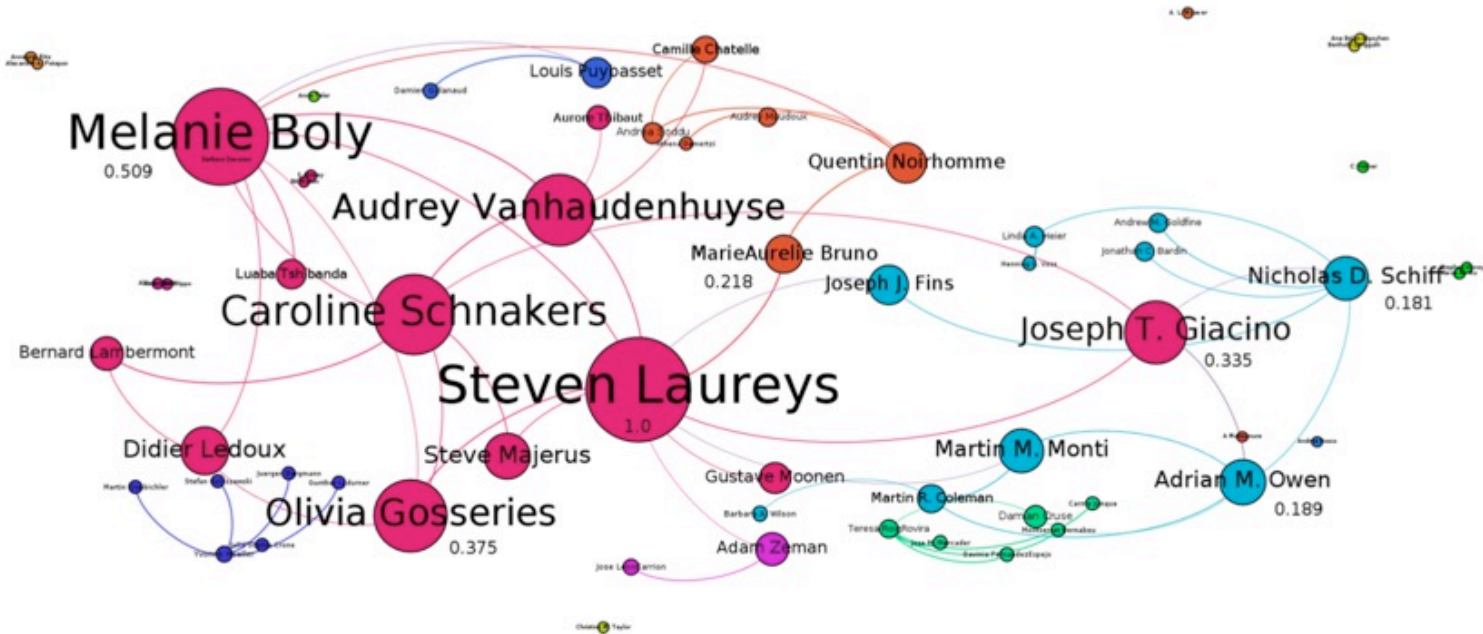
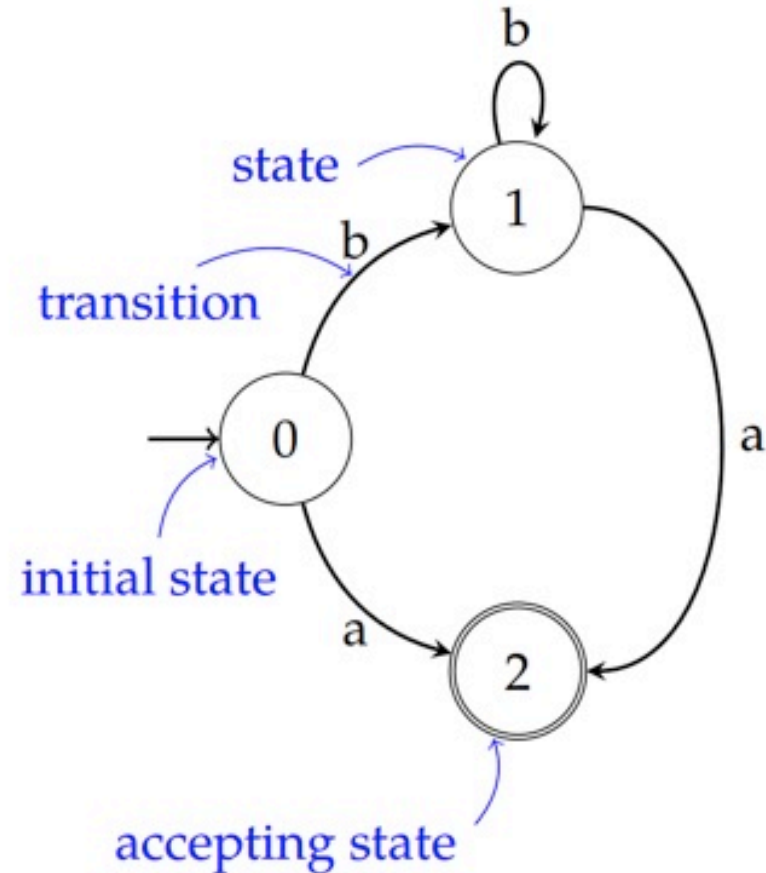


Figure 2 Co-authorship graph of NIMCS and related research. Nodes represent authors; edges represent co-authorship. Graph layout uses the ForceAtlas2 algorithm. Clusters are calculated via Louvain modularity and delineated by color. Frequency of co-authorship is calculated via Eigenvector centrality and represented by size.

Image from Alex Garnett, Grace Lee and Judy Illes. 2013. *Publication trends in neuroimaging of minimally conscious states*. PeerJ.

Finite State Automata

- States are represented as nodes
- Transitions are shown by the edges, labeled with symbols from an alphabet
- One of the states is marked as the *initial state*
- Some states are accepting states



credit: introduction to finite state automata by C. Çöltekin



Parsing

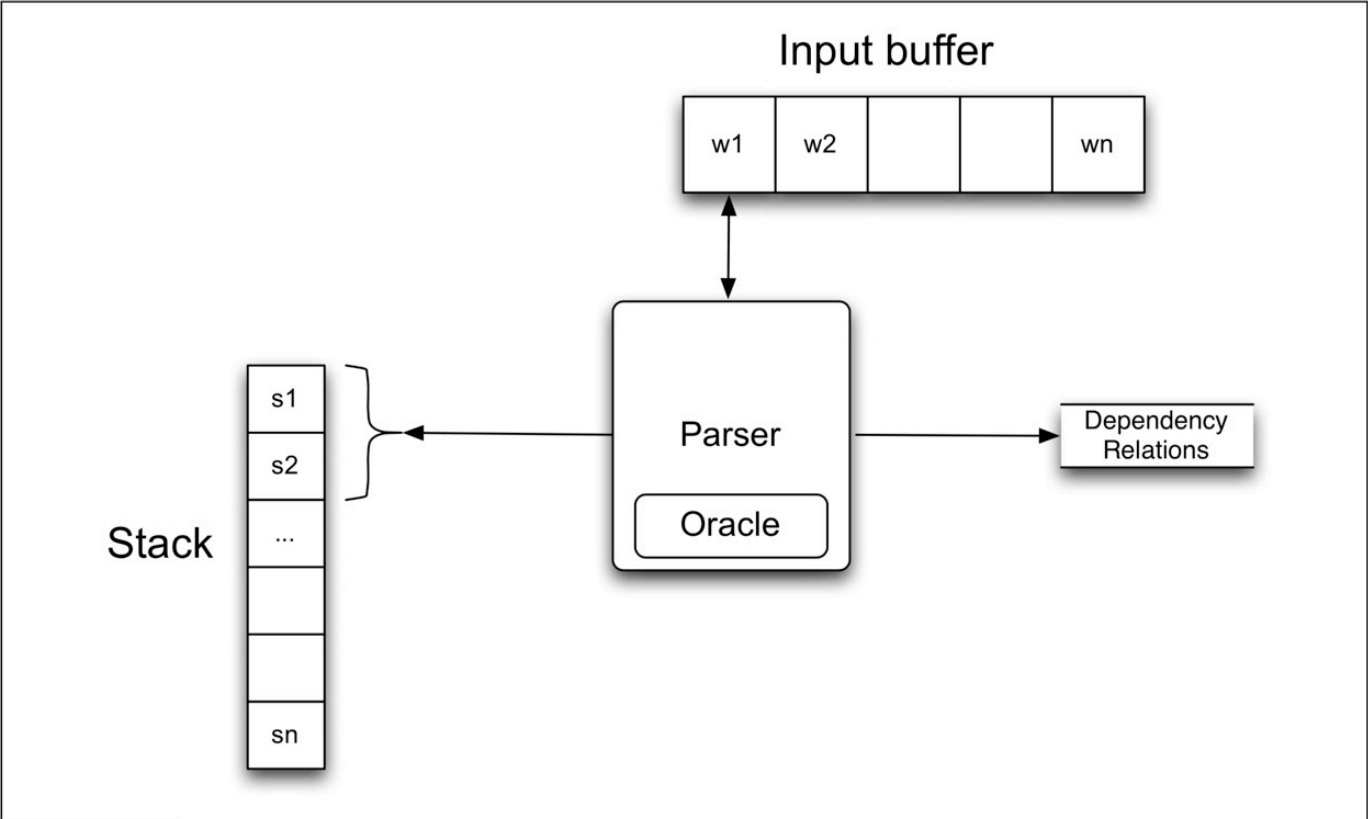
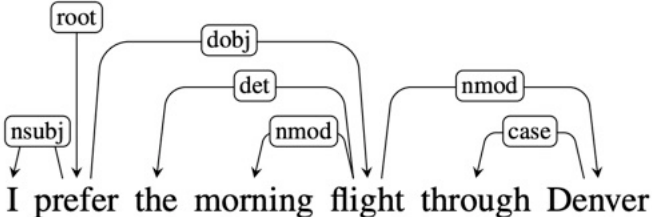


Figure 15.5 Basic transition-based parser. The parser examines the top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

credit: Jurafsky & Martin, SLP 3, chapter 15, Dependency Parsing

Language Guessing / Language Identification

Language Guessing Applications

- Web browsers use language identification and offer to translate the page when it is not in the computer's default language
- Google Translate uses language identification to determine the source language of the text to be translated
- In computational linguistics, it is important to know what language the text is in, in order to determine what linguistic tools are appropriate for processing it

Language 1

Landau-Symbole

Landau-Symbole werden in der **Mathematik** und in der **Informatik** verwendet, um das **asymptotische Verhalten** von **Funktionen** und **Folgen** zu beschreiben. In der Informatik werden sie bei der Analyse von Algorithmen verwendet und geben ein Maß für die Anzahl der Elementarschritte oder der Speichereinheiten in Abhängigkeit von der Größe der Eingangsvariablen an. Die **Komplexitätstheorie** verwendet sie, um verschiedene **Probleme** danach zu vergleichen, wie „schwierig“ oder aufwendig sie zu lösen sind. Man sagt, „schwere Probleme“ wachsen **exponentiell** mit der Instanz oder schneller und für „leichte Probleme“ existiert ein Algorithmus, dessen Laufzeitzuwächse sich durch das Wachstum eines **Polynoms** beschränken lassen. Man nennt sie *(nicht) polynomiell lösbar*.

Language 2

Cota superior asintótica

En [análisis de algoritmos](#) una **cota superior asintótica** es una [función](#) que sirve de cota superior de otra función cuando el [argumento](#) tiende a infinito. Usualmente se utiliza la [notación de Landau](#): $O(g(x))$, Orden de $g(x)$, coloquialmente llamada Notación **O Grande**, para referirse a las funciones acotadas superiormente por la función $g(x)$.

Formalmente se define:

$$O(g(x)) = \left\{ \begin{array}{l} f(x) : \text{existen } x_0, c > 0 \text{ tales que} \\ \forall x \geq x_0 > 0 : 0 \leq |f(x)| \leq c|g(x)| \end{array} \right\}$$

Una función $f(x)$ pertenece a $O(g(x))$ cuando existe una [constante](#) positiva c tal que a partir de un valor x_0 , $f(x)$ no sobrepasa a $cg(x)$. Quiere decir que la función f es inferior a g a partir de un valor dado salvo por un factor constante.

La cota superior [asintótica](#) tiene gran importancia en la [Teoría de la complejidad computacional](#) cuando se definen las [clases de complejidad](#).

Language 3

Notăția Big O

De la Wikipedia, enciclopedia liberă

Notăția Big O este o notație matematică care descrie **comportamentul la limită**^(d) al unei **funcții** atunci când **argumentul**^(d) tinde la o anumită valoare sau la infinit. Este una din notațiile inventate de **Paul Bachmann**,^[1] **Edmund Landau**^{(d), [2]} și alții, numite colectiv **notațiile Bachmann-Landau** sau **notațiile asimptotice**.

În **informatică**, notația Big O este folosită pentru a **clasifica algoritmi** în funcție de felul în care timpul lor de rulare sau cerințele lor de spațiu cresc pe măsură ce crește dimensiunea datelor de intrare.^[3] În **teoria analitică a numerelor**^(d), notația Big O este adesea folosită pentru a exprima o legătură între diferența dintre o **funcție aritmetică**^(d) și o aproximare mai bine înțeleasă; un exemplu celebru de astfel de diferență este termenul rest din **teorema numerelor prime**.

Notatia Big O caracterizează funcțiile după de vitezele lor de creștere: funcții diferite cu aceeași viteză de creștere pot fi reprezentate folosind aceeași notație O.

Litera O este folosită deoarece viteza de creștere a unei funcții este numită și **ordin al funcției**. O descriere a unei funcții în ceea ce privește notația Big O, de obicei, oferă doar o **limită superioară**^(d) a vitezei de creștere a funcției. Cu notația Big O mai sunt asociate mai multe alte notații, folosind simbolurile o , Ω , ω , și Θ , pentru a descrie alte tipuri de limite ale vitezelor de creștere asimptotică.

Language 4

Büyük O gösterimi

Vikipedi, özgür ansiklopedi

Büyük O (Big-Oh) gösterimi matematiksel bir gösterim olup **işlevlerin** (fonksiyonların) **asimptotik** davranışlarını tarif etmek için kullanılır. Daha açık şekilde anlatmak gerekirse, bir işlevin büyümesinin **asimptotik üst sınırını** daha basit başka bir işlev cinsinden tanımlanması demektir. İki temel uygulama alanı vardır: **matematik** alanında genellikle kırılmış bir **sonsuz serinin** kalan terimini karakterize etmek için kullanılır; **bilgisayar bilimlerinde** ise **algoritmaların** bilgi işlemsel **karmaşıklığının** **çözümlemesi** için kullanılır.

Bu gösterim ilk olarak **Alman sayılar kuramcısı Paul Bachmann** tarafından **1892** yılında yazdığı *Analytische Zahlentheorie* kitabında kullanılmıştır. Gösterim bir başka Alman matematikçi olan **Edmund Landau** tarafından yaygın kullanıma sokulmuştur, bundan ötürü bazen **Landau sembolü** olarak da anılır. Büyük O, İngiliz dilindeki "order of" yani bir şeyin derecesi anlamına gelen söz öbeğini hatırlatmak amacı ile kullanılıyordu ve ilk olarak büyük **omicron** harfi idi; günümüzde büyük O kullanılmakta ve **0 sayısı** hiç kullanılmamaktadır.

Language 5

ランダウの記号

出典: フリー百科事典『ウィキペディア (Wikipedia) 』

ランダウの記号 (ランダウのきごう、**英**: Landau symbol) は、**関数の極限**における値の変化度合いに、おおよその評価を与えるための記法である。

ランダウの漸近記法 (asymptotic notation)、**ランダウ記法** (Landau notation) あるいは主要な記号として O (オーもしくは**オミクロン** O 。数字の0ではない) を用いることから (バッハマン-ランダウの) **O -記法** (Bachmann-Landau O -notation^[1])、**ランダウのオミクロン**などともいう。

記号 O は「程度」の意味のオーダー (Order) から。

なおここでいうランダウは**エドムント・ランダウ**の事であり、『**理論物理学教程**』の著者である**レフ・ランダウ**とは別人である。

ランダウの記号は**数学**や**計算機科学**をはじめとした様々な分野で用いられる。

Language 6

大O符号 [编辑]

维基百科，自由的百科全书

大O符号（英語：Big O notation），又稱為漸進符號，是用于描述函数渐近行为的数学符号。更确切地说，它是用另一个（通常更简单的）函数来描述一个函数数量级的渐近上界。在数学中，它一般用来刻画被截断的无穷级数尤其是渐近级数的剩余项；在计算机科学中，它在分析算法复杂性的方面非常有用。

大O符号是由德国数论学家保罗·巴赫曼在其1892年的著作《解析数论》（*Analytische Zahlentheorie*）首先引入的。而这个记号则是在另一位德国数论学家艾德蒙·朗道的著作中才推广的，因此它有时又称为朗道符号（Landau symbols）。代表“order of ...”（……阶）的大O，最初是一个大写希腊字母“O”（omicron），现今用的是大写拉丁字母“O”。

Any Ideas?

- How can the language of a text be guessed?
- (Brainstorming)

Method

- We can write an **algorithm for guessing the language of a text**
 - Using simple **n-gram statistics**
 - Using a small amount of training data
 - With high accuracy
- Method of Canvar and Trenkle, 1994. *N-Gram-Based Text Categorization*.
 - Based on computing and comparing **profiles of n-gram frequencies**
 - First, **compute profiles on a training set** of data containing different language samples
 - For a **new document** whose language has to be guessed: **construct a profile and compare it to each of the training profiles**; select the **language with the smallest distance** to the new profile as the “winner”

First Step: Computing the language profile

- As in Canvar and Trenkle, 1994. *N-Gram-Based Text Categorization*.
 - Identify and count each 1-, 2-, 3-, 4- and 5- gram of the text
 - Sort the n-grams by frequency (most frequent first)
 - Retain the most frequent 300 n-grams

N-grams

- An n -gram is a n -character-long continuous slice of a string
- Each n defines a separate set of n -grams
- E.g. different n -grams for the word *bananas*

n-gram type	resulting n-grams
1-grams	b a n a n a s
2-grams	ba an na an na as
3-grams	ban ana nan ana nas
4-grams	bana anan nana anas
5-grams	banan anana nanas

Example: *bananas* - n-grams & their frequencies

n-gram type	resulting n-grams
1-grams	b a n a n a s
2-grams	ba an na an na as
3-grams	ban ana nan ana nas
4-grams	bana anan nana anas
5-grams	banan anana nanas



n-gram	freq
a	3
b	1
n	2
s	1
ba	1
an	2
na	2
as	1
ban	1
ana	2
nan	1
nas	1

n-gram	freq
bana	1
anan	1
nana	1
anas	1
banan	1
anana	1
nanas	1

Example: *bananas* - frequencies, reverse-sorted (+ lexically)

n-gram	freq
a	3
an	2
ana	2
n	2
na	2
anan	1
anana	1
anas	1
as	1
b	1
ba	1
ban	1

n-gram	freq
bana	1
banan	1
nan	1
nana	1
nanas	1
nas	1
s	1

Example: *bananas* - from frequencies to ranks

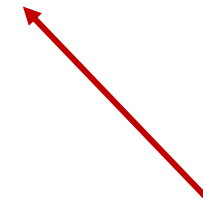
n-gram	freq
a	3
an	2
ana	2
n	2
na	2
anan	1
anana	1
anas	1
as	1
b	1
ba	1
ban	1

n-gram	freq
bana	1
banan	1
nan	1
nana	1
nanas	1
nas	1
s	1



n-gram	freq
a	1
an	2
ana	3
n	4
na	5
anan	6
anana	7
anas	8
as	9
b	10
ba	11
ban	12

n-gram	freq
bana	13
banan	14
nan	15
nana	16
nanas	17
nas	18
s	19

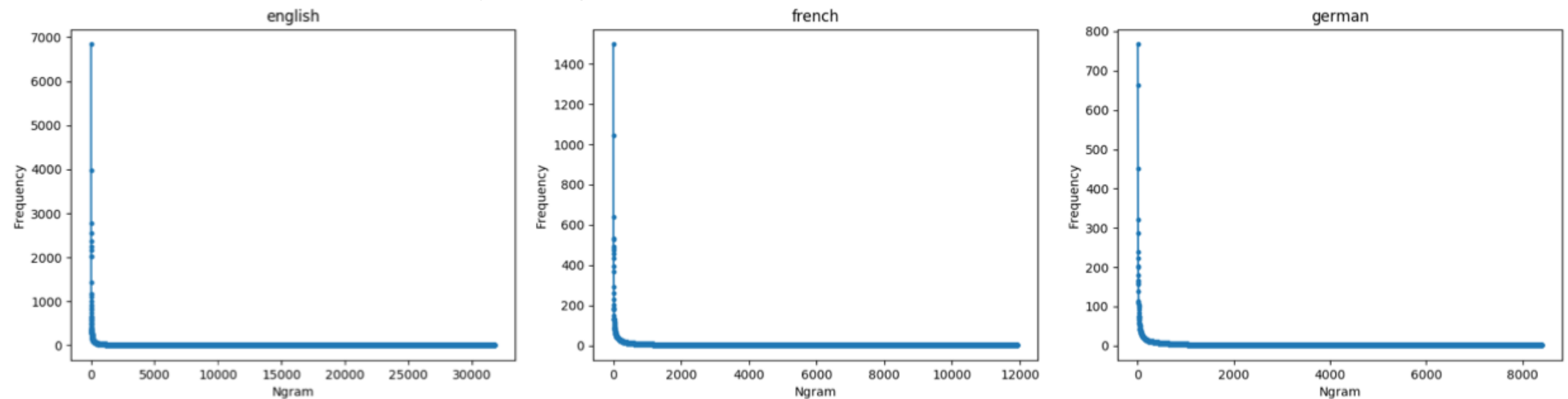


Profile for the word *bananas*



Zipfian Distribution

- When plotting the n -gram frequencies in rank order, the language profiles display a Zipfian distribution
- Zipf's law: the n -th most common word in a human language text occurs with a frequency inversely proportional to n .
- This means that the most frequent word will occur twice as many times as the second most frequent word, three times more than the third most frequent, etc.
- Also holds for the frequency of n -grams



Two English Samples – top n-grams

```
'_', 6839  ' ', 4185
'e', 3970  'e', 3027
't', 2783  'n', 1964
'o', 2553  'o', 1790
'a', 2374  't', 1748
'n', 2248  'i', 1655
'r', 2175  'a', 1620
'i', 2033  's', 1476
's', 2033  'r', 1199
'h', 1440  'c', 925
'l', 1167  'h', 879
'e_', 1164 'l', 870
'd', 1117 'd', 780
'_t', 1007 'e_', 765
'c', 931  'u', 695
'u', 893  'm', 683
's_', 827 'g', 586
'm', 815  's_', 555
'th', 748 'f', 510
'g', 658  'en', 507
'w', 654  '_t', 490
'f', 648  'in', 458
```

```
'p', 631  'th', 445
'_a', 621 'p', 439
'he', 592 '_a', 421
't_', 590 'n_', 397
'_th', 577 'he', 369
'an', 571 'on', 364
'y', 559  'an', 348
'd_', 539 '_th', 340
'er', 527 'y', 327
'in', 505 'es', 325
're', 500 'ti', 325
'n_', 491 '_o', 321
'_w', 471 'd_', 305
'_o', 451 'er', 303
'r_', 433 're', 298
'or', 416 'at', 294
',', 401  'the', 280
',_', 393 'b', 267
'b', 390  'ge', 258
'the', 390 't_', 249
'on', 376 '_the', 248
'es', 365 '_i', 246
```

```
'o_', 363 'om', 244
'to', 360 'se', 244
'_s', 358 'v', 241
'nd', 356 'he_', 234
'.', 350  ',', 231
'_the', 345 '_s', 231
'at', 343 'of', 230
'._', 341 '_of', 226
'v', 339  'gen', 222
'y_', 319 'f_', 221
'_c', 318 'te', 221
'_to', 313 '_of_', 220
'ha', 312 'of_', 220
'he_', 306 'al', 219
'en', 297 'no', 219
'_i', 296  ',_', 215
'ou', 295 '_g', 215
've', 290 'y_', 215
'nd_', 282 'the_', 212
'ng', 282 '_c', 211
'k', 281  '_the_', 211
'_f', 275 'di', 210
```

- The top n-grams from a text containing President's Obama State of the Union Address from 2014 (left), and a Wikipedia entry on the human genome (right) will tend to have many n-grams in common, despite the difference in topic



Two English samples – lower n-grams

```
's_', 93 'seq', 75
',_a', 91 'sequ', 75
'pr', 91 'seque', 75
'ric', 91 'uman', 75
'rt', 91 'NA_', 74
'mer', 90 'hu', 74
'wor', 90 'rs', 74
'e_a', 89 '_seq', 73
'ad', 87 '_sequ', 73
'ry', 87 'un', 73
'wh', 87 '_a_', 72
'Am', 86 '_l', 72
'_Am', 86 'ul', 72
'_wh', 86 'DN', 71
'ess', 86 'DNA', 71
'her', 86 'tha', 71
'ie', 86 'uence', 71
'ill', 86 '_D', 70
'lo', 86 '_hu', 70
'_ne', 85 'ac', 70
'po', 85 'id', 70
's_to', 85 's,', 70
```

```
'meri', 84 's_', 70
'rica', 84 'ut', 70
'we_', 84 's_t', 69
'Ame', 83 'din', 68
'Amer', 83 'e_a', 68
'Ameri', 83 '_hum', 67
'_Ame', 83 '_huma', 67
'_Amer', 83 'hum', 67
'eric', 83 'huma', 67
'ERICA', 83 'human', 67
'meric', 83 'is_', 67
'pl', 83 '_tha', 65
'_for', 82 'ding', 65
'_we_', 82 'ding_', 65
'gh', 82 'iv', 65
'ir', 82 'lo', 64
't_t', 82 'pro', 64
'ts', 82 'na', 63
'_wor', 81 'pe', 63
'd_t', 81 '_DN', 62
'e_w', 81 '_DNA', 62
```

- Around rank 300 the n-grams become more specific to the **topic** of the particular article: on the left, about *America*, on the right about *human, DNA, sequence* etc.



English vs. French – top n-grams

'_ ', 6839	'_ ', 1497
'e ', 3970	'e ', 1044
't ', 2783	's ', 641
'o ', 2553	'n ', 533
'a ', 2374	'i ', 527
'n ', 2248	'a ', 491
'r ', 2175	'r ', 484
'i ', 2033	't ', 473
's ', 2033	'u ', 459
'h ', 1440	'o ', 433
'l ', 1167	'l ', 394
'e_', 1164	'e_', 367
'd ', 1117	's_', 290
'_t ', 1007	'c ', 262
'c ', 931	'd ', 232
'u ', 893	'p ', 205
's_', 827	'_l ', 185
'm ', 815	'm ', 184
'th ', 748	'é ', 182
'g ', 658	'es ', 180
'w ', 654	'_d ', 179
'f ', 648	'_t_', 149
'p ', 631	'_p ', 134
'_a ', 621	'en ', 134

'he ', 592	'_ ', 131
't_', 590	'_ ', 131
'_th ', 577	'le ', 131
'an ', 571	'ou ', 122
'y ', 559	'v ', 119
'd_', 539	'', 116
'er ', 527	'on ', 115
'in ', 505	're ', 108
're ', 500	'_c ', 106
'n_', 491	'nt ', 106
'_w ', 471	'de ', 102
'_o ', 451	'es_', 98
'r_', 433	'_s ', 95
'or ', 416	'.', 91
'', 401	'_ ', 91
'_ ', 393	'_de ', 89
'b ', 390	'_a_', 89
'the ', 390	'r_', 88
'on ', 376	'er ', 87
'es ', 365	'q ', 87
'o_', 363	'qu ', 87
'to ', 360	'us ', 87
'_s ', 358	'ur ', 85
'nd ', 356	'_e ', 83

'.', 350	'an ', 83
'_the ', 345	'f ', 79
'at ', 343	'ce ', 78
'_', 341	'n_', 77
'v ', 339	'_a ', 76
'y_', 319	'ns ', 74
'_c ', 318	'is ', 70
'_to ', 313	'de_', 69
'ha ', 312	'te ', 66
'he_', 306	'ti ', 66
'en ', 297	'tr ', 66
'_i ', 296	'i_', 65
'ou ', 295	'ra ', 64
've ', 290	'_q ', 63
'nd_', 282	'_qu ', 63
'ng ', 282	'la ', 63
'k ', 281	'_de_', 62
'_f ', 275	'_n ', 62
'_h ', 272	'ai ', 60
'st ', 272	'me ', 60
'ar ', 269	'_le ', 59
'to_', 268	'g ', 59
'_the_', 261	'it ', 59
'the_', 261	'se ', 58
'_to_', 258	'_la ', 57
'it ', 258	'la_', 57
'ea ', 246	'us_', 56
'_b ', 245	'_la_', 55
'_an ', 241	'_m ', 54
'ne ', 240	'ent ', 54
'_m ', 237	'le_', 54

- By contrast, the top n-grams from two different languages will have very different distributions of the first 300 n-grams



Second Step: Comparing Two Profiles

- Given the document profile p and the language profile q , the distance between the two profiles is defined as:

$$d(p, q) = \sum_{\text{ngram} \in p} |\text{rank}(\text{ngram}, p) - \text{rank}(\text{ngram}, q)|$$

- The *rank* of an n-gram is the rank of the n-gram in a given profile, or the size of the profile if the n-gram is not part of the profile

Example: Comparing Two Profiles (from Canvar and Trenkle, 1994)

	p (document)	q (language)	$ \Delta $
most frequent	th	th	0
	ing	er	3
	on	on	0
	er	le	2
	and	ing	1
	ed	and	no-match=max
least frequent

Basic Algorithm

```
 $L \leftarrow \{\langle \text{language}, \text{profile} \rangle\}$   
 $p \leftarrow \text{document profile}$   
 $guess \leftarrow \epsilon$   
 $guess\_dist \leftarrow \infty$   
for all  $(l, q) \in L$  do  
     $dist \leftarrow d(p, q)$   
    if  $dist < guess\_dist$  then  
         $guess\_dist \leftarrow dist$   
         $guess \leftarrow l$   
    end if  
end for
```

Results (from Canvar and Trenkle, 1994)

Article Length	≤ 300	≤ 300	≤ 300	≤ 300	> 300	> 300	> 300	> 300
Profile Length	100	200	300	400	100	200	300	400
Newsgroup								
australia	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
brazil	70.0	80.0	90.0	90.0	91.3	91.3	95.6	95.7
britain	96.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
canada	100.0	100.0	100.0	100.0	100.0	*99.6	100.0	100.0
celtic	100.0	100.0	100.0	100.0	99.7	100.0	100.0	100.0
france	90.0	95.0	100.0	*95.0	99.6	99.6	*99.2	99.6
germany	100.0	100.0	100.0	100.0	98.9	100.0	100.0	100.0
italy	88.2	100.0	100.0	100.0	91.6	99.3	99.6	100.0
latinamerica	91.3	95.7	*91.3	95.7	97.5	100.0	*99.5	*99.0
mexico	90.6	100.0	100.0	100.0	94.8	99.1	100.0	*99.5
netherlands	92.3	96.2	96.2	96.2	96.2	99.0	100.0	100.0
poland	93.3	93.3	100.0	100.0	100.0	100.0	100.0	100.0
portugual	100.0	100.0	100.0	100.0	86.8	97.6	100.0	100.0
span	81.5	96.3	100.0	100.0	90.7	98.9	98.9	99.45
Overall	92.9	97.6	98.6	98.3	97.2	99.5	99.8	99.8



Results (from Canvar and Trenkle, 1994)

- Categorized results over several dimensions
- Article length over or under 300 bytes
 - Hypothesis: shorter articles should be more difficult to classify, because there is less text to construct the n-grams from
 - Result: system only slightly sensitive to length
- Varied the profile length: 100, 200, 300 and 400 n-grams in the profile
 - Profile length did have an impact on performance
 - Almost perfect classification with 400 n-grams

Task difficulty

- The task can be made more difficult by
 - Adding more languages, especially similar languages or language dialects
 - Trying to identify very short fragments
- There are newer methods that use more sophisticated statistical modelling and/or machine learning to identify languages
 - Radim Řehůřek and Milan Kolkus. 2009. *Language Identification on the Web: Extending the Dictionary Method*. CICLing 2009

Thank you.

Acknowledgements

- The introduction to algorithms section is based on the introductory slides for Algorithms, 4th edition, by Sedgewick & Wayne
- The language guessing section uses materials from the DSA-CL III Introductory slides from 2017 by Daniël de Kok