CS124 — Lecture 8

Greedy algorithms

SATisfiability, NP complete

Formula – conjunction of disjunctions
(and-of-ors)

Special subclass: Horn formulae – [redacted] at most one pos literal per clause

$(x \vee \bar{y} \vee \bar{z} \vee \bar{w}) \wedge (\bar{x} \vee \bar{y} \vee \bar{w}) \wedge (\bar{x} \vee \bar{z} \vee w)$
$\wedge (\bar{x} \vee y) \wedge (x) \wedge (\bar{z}) \wedge (\bar{x} \vee \bar{y} \vee w)$

split into ① pure negative clauses – no negative literal
② one positive literal – implications

$(\bar{x} \vee y) : x \to y$

x y

x	y	exp
T	T	V
T	F	X
F	F	V
F	T	V

$(\bar{x} \vee \bar{y} \vee w)$
$(x \wedge y \to w)$

X	Y	W	exp
T	T	T	V
~~T~~	~~T~~	~~F~~	~~X~~
T	F	T	V
T	F	F	V
F	T	T	V
F	T	F	V
F	F	T	V
F	F	F	V

$(\bar{x} \vee \bar{y} \vee w) : x \wedge y \to w$
$(x) : \to x$

Alg Greedy: start with all false
while ∃ an implication that is not satisfied
set right hand side to true
[redacted]

$(x \vee \bar{y} \vee \bar{z} \vee \bar{w})$	$(y \wedge z \wedge w \to x)$	4) unsatisfiable
$\wedge (\bar{x} \vee \bar{y} \vee \bar{w})$	$\wedge$ [redacted] (pure negative)	
$\wedge (\bar{x} \vee \bar{z} \vee w)$	$\wedge (x \wedge z \to w)$	2) $y = T$
$\wedge (\bar{x} \vee y)$	$\wedge (x \to y)$	1) $x = T$
$\wedge (x)$	$\wedge (\to x)$	
$\wedge (\bar{z})$	$\wedge$ [redacted] (pure negative)	
$\wedge (\bar{x} \vee \bar{y} \vee w)$	$\wedge (x \wedge y \to w)$	3) $w = T$

greedy: only make changes when have to

check all pure negative clauses
if all satisfied – done
if not – unsatisfiable

Correctness

can satisfiable: each step only makes a clause [redacted] satisfied [redacted] never unsatisfied, it stays satisfied

can unsatisfiable: only set literals to T, because there was no other [redacted] way [redacted]

[redacted]

e.g.

$(\bar{x} \vee \bar{y} \vee w)$

$(x \wedge y \rightarrow w)$

$\underbrace{T \quad T}$ $\quad F$

only assignment of an unsatisfied Horn clause

All started with F previously had to be set to T as the only way to satisfy some other [redacted] clauses

must be set to T to satisfy this clause

if algorithm return unsatisfiable $\to \nexists$ no assignment that can satisfy all clauses

$\boxed{\text{induction}}$ $\boxed{\text{on variables set to } T}$

set ith variable to T because previous i-1 variables had to be set to T, and setting ith variable to T is the only way to satisfy the clause

---

Huffman coding

$\to$ compression scheme

DNA example

A, C, G, T

ex1 2 bits A = 00, C = 01, G = 11, T = 10

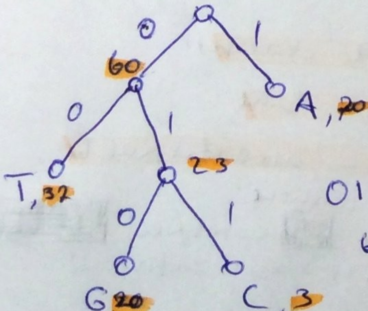
ex2			
	A - 70 million	1	70
	C - 3 million	011	9
	G - 20 million	010	60
	T - 37 million	00	74
			213 million (20% saved)

baseline 2 bits $\to$ 260 million bits

+ self description of dictionary

does not take context into account!

prefix free property: no string is prefix of [redacted] another



C C T A A A G

011 | 011 | 00 | 1 | 1 | 1 | 010

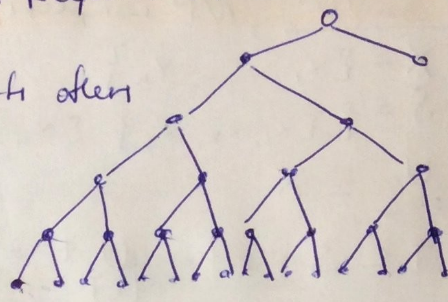
60 + 70 + 37 + 23 + 20 + 3 = 213 million

e.g. AAA ... AA CCC ... CC GGG ... TTT ... could be compressed in < 213 by taking context information into account

encoding: build tree
decoding: read from tree
} thinking not implementing

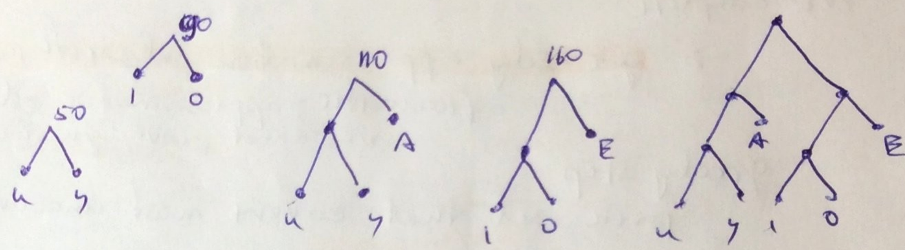
Greedy [redacted] step [redacted], put most frequent at the top

NO ↳ if 16 characters, keeping 1 at [redacted] depth, hurts others



- put lowest frequencies to the bottom

A	60
E	70
I	40
O	50
U	20
Y	30



correctness proof

- WLOG, the two least frequent nodes are siblings at the deepest level in opt. Huffman tree.

Base Case

Proof by contradiction: if they are not at deepest level, swap with two at the deepest level and get a better tree

[redacted] [redacted] contradiction to optimality

induction

1) combine preceding into [redacted] single node
2) two least frequent are siblings at the deepest level in opt Huffman tree
   same pf. by contradiction
3) expand previously combined nodes

Set Cover, NP complete

$X = \{x_1 \ldots x_n\}$

$S = \{\{x_1, x_2, x_3\}\ \{x_4, x_8, x_4, x_{10}\}\ \{x_{12}, x_{12}, x_{26} \ldots\}\}$

$\bigcup_{s \in S} = X$

$T \subseteq S$ s.t. $\bigcup_{t \in T} t = X$, $|T|$ minimum

NP complete

→ greedy approximation algorithm

(Heuristic = approximation alg. where we cannot prove guarantees [redacted])

greedy step:

pick set that covers most uncovered elts

		S_5		S_4		
S_1	1	2	3	4	5	6
S_2	7	8	9	10	11	12
S_3	13	14	15	16	17	18

optimal set cover: S_1, S_2, S_3

greedy set cover: S_4, S_5, S_1, S_2, S_3

$|X| = n$

claim: let k be the size of the smallest set cover, then the greedy alg. finds a cover of size

$$\le k \ln n$$

<u>Proof.</u> let $y_i \le |X|$ be uncovered elts after i sets chosen

$y_0 = |X| = n$

at least 1 set must cover $\ge \frac{n}{k}$ elts ← pigeonhole princ.

$$y_1 \le n - \frac{n}{k} = n\left(1 - \frac{1}{k}\right) = y_0\left(1 - \frac{1}{k}\right)$$

$$y_2 \le y_1\left(1 - \frac{1}{k}\right) \le y_0\left(1 - \frac{1}{k}\right)^2$$ [redacted]

$j = \lceil k \ln n \rceil$

$$y_j \le n\left(1 - \frac{1}{k}\right)^{k \ln n}$$

$$\le n\left(e^{-1/k}\right)^{k \ln n}$$

$$< 1 \quad \checkmark$$

$\left(1 - \frac{1}{k}\right) \le e^{-1/k}$ Taylor exp.

[redacted]

$\left(1 - \frac{1}{k}\right) < e^{-1/k}$

y_j is integer, must be 0, all covered