

Ph.D. Comprehensive Examination

Computer Science Department
University of Miami

August 19, 2022

Student Name: Alexander
Clemens

Student Number: C12041265

Problem number	Points (10 max)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total:	

1. Data organization; Algorithms and complexity

DONE

Each item x in a set S has a unique key $key[x]$. We need to implement the following operations.

- (a) Search (S, key)
- (b) Insert (S, x)
- (c) Successor (S, x)
- (d) Predecessor (S, x)

Give the 4 running times as an $O()$ for the following implementations:

- (a) Ordered (sorted) array,
- (b) Ordered doubly linked list,
- (c) Min-Heap, and
- (d) Hash table

Ordering is on x , not key ?
these search estimates might go to
 $O(\log n)$ (binary search)

a) Ordered array

a) Search should be $O(n) \Rightarrow$ worst case, it's at the end

b) Insert should be $O(n) \Rightarrow$ worst case, needs to go at "end" (is biggest)

c) $O(1)$ (go to next/previous element in ordered list)

d) $O(1)$

b) Ordered doubly-linked list:

a) Search is $O(n) \Rightarrow$ linear scan

b) insert is $O(n) \Rightarrow$ same as search for order maintenance

c) successor

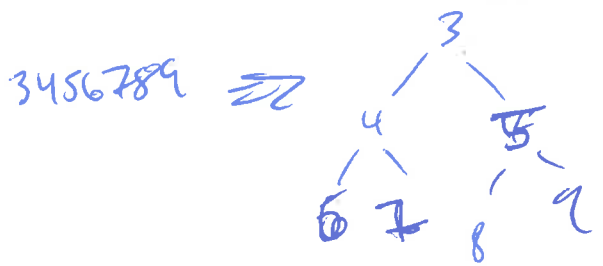
d) predecessor \Rightarrow easy: doubly linked \Rightarrow next, previous $\Rightarrow O(1)$

d) Hash table

a/b) Search & insert should be $O(1)$ (hash fn)

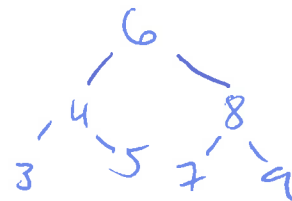
c/d) I think $O(n)$? Need to check all;
hash fn might map next biggest/smallest
somewhere else?

c) min-heap: I really don't know what this is.
 Maybe just a "higher = smaller" tree? or something?



\Rightarrow is it not ordered? like BST?

vs.



Search: if tree, maybe $O(n \log n)$?

addition: $O(n)$? needs to make sure is smallest \Rightarrow what it needs to add at the top? Bad \Rightarrow lots of swaps. Maybe only $O(\log n)$ though \Rightarrow based on depth of tree.

predecessor } $O(1)$? just go up/down (linked list implementation?)
 successor }

2. Program control and structure; Programming language and notations

Suppose that procedure *swap* is declared as follows:

```

procedure swap( x, y: integer);
  procedure f(): integer;
    var z: integer;
    begin // f
      z = x; x = y; return z;
    end // f
  begin // swap
    y = f();
  end // swap

```

DONE

Please see reverse for full answers

Describe the effect of the procedure call *swap*(i, A[i]) under each of the following parameter passing methods:

- Call-by-value
- Call-by-reference
- Call-by-value-result

↳ I'm not sure what these are (never seen these formal defs before). However, I think the ideas are:

reference value
 \downarrow \downarrow
 $x = 3 \Rightarrow$ ~~pointer~~ pointers?

so 3 is 3

$i = 3 \Rightarrow$ value = 3
 reference = (address)

$A[i] \Rightarrow$ value = address of thing @ $A[i]$
 reference = address of $A[i]$
 value-result = value of thing @ address pointed to by $A[i]$

~~AL:3~~

example

A-addr $\rightarrow A = [i_1, i_2, \dots, i_n]$

$A[i_k] \rightarrow \underset{\uparrow}{i_k} \rightarrow \underset{\uparrow}{V_k}$
 ref value result

→ flip for my best explanation in this paradigm

thought access

Okay, let's more formally define this.

$\text{Swap}(i, A[i]) \Rightarrow \begin{cases} i \text{ (address)} = 3456 \\ i \text{ (value)} = 4 \end{cases}$

I'm constructing this to make my life easier trying to write this all out. Could also use print statements in actual program.

$A[i] \text{ (address)} = 789$

$A[i] \text{ (value)} = \text{address at } i\text{th position of } A = 101112$

$A[i] \text{ (value-result)} = \text{value @ address @ } i\text{th position of } A = 6$

b) in address mode: $\text{Swap}(3456, 789)$
(add) (add)

$\text{Swap} \left\{ \begin{array}{l} \{ \begin{array}{l} Z = 3456 \\ 3456 = 789 \\ \text{return } 3456 \\ 789 = 3456 \end{array} \} \end{array} \right. \begin{array}{l} \text{pointer swap? so} \\ 789 \text{ now points to} \\ 4 \text{ \& } 3456 \text{ now points} \\ \text{to } 101112. \end{array}$

a) in value mode: $\text{Swap}(4, 101112)$
(val) (add) seg fault (access address 4)

$\text{Swap} \left\{ \begin{array}{l} \{ \begin{array}{l} Z = 4 \\ 101112 = 4 \\ \text{return } 101112 \\ 4 = 101112 \\ \text{101112} \end{array} \} \end{array} \right. \begin{array}{l} \text{this is the Bad Thing}^{\text{TM}}: \\ \text{you usually want to swap} \\ \text{what's in the array, not the} \\ \text{address to the value in position} \\ \text{3 of the array.} \end{array}$

c) in value-result mode: $\text{Swap}(4, 6)$

$\text{Swap} \left\{ \begin{array}{l} \{ \begin{array}{l} Z = 6 \\ 6 = 4 \\ \text{return } 6 \\ 4 = 6 \end{array} \} \end{array} \right.$

actual value swapping.

I think the goal of this question (regardless of if I got the details correct) was to assess/explain the difference between references (pointers & values).

3. Software engineering

DONE

From the software engineering point of view, any software development process can be divided into several sub-disciplines:

- (a) Requirement Analysis
- (b) Functional Specification
- (c) Architectural Design
- (d) Implementation
- (e) Testing and Evaluation
- (f) Maintenance

Choose three sub-disciplines or tasks within these sub-disciplines that involve a mathematical approach, and illustrative them with examples.

↓
not entirely sure what this entails

(e) Testing & Evaluation

- an example of this might be unit testing / test-driven-development: create a (quantitative) test a given unit of code should pass, then write the code such that it passes the test
- ex. unit test for an adding function (in Python)

```
def test_add (add_fn):
```

```
    x = 2
```

```
    y = 3
```

```
    assert add_fn(x, y) == 5
```

```
    # Maybe also requires commutativity?
```

```
    assert add_fn(y, x) == 5
```

↑ this is the callable adding fn to be designed

(b) Functional specification (or maybe (a) Requirement Analysis?)

- whichever is covered by the SRS document (software requirement specifications, so I guess (a)?)

I remember prioritizing requirements (both functional & nonfunctional) numerically on a scale when I wrote mine. i.e. 0 is absolute highest priority, 5 is probably not that important

flip for last



(f) Maintenance

- one important aspect of maintenance is version control (i.e. tracking what version of the code is currently deployed).
- classically, this has been done w/ version numbers (i.e. v1.0), etc., with major releases perhaps incrementing the version # (v2.0, v3.0) & minor patches/alterations incrementing subsequent values (v1.1, 1.2, 1.3).
- also relevant in this discussion are version control systems (such as git and subversion). These allow for (numerically \geq logically) organized collaboration ~~across~~ across machines, users, and versions.

4. Systems

DONE

- (a) _____ linked libraries can support shared library code, allowing one copy of a library routine to be used by several different processes.
absolute relative static dynamic none of these is correct
- (b) When it is not known at compile time where a process will reside in memory, _____ code must be generated.
logical physical absolute relocatable
- (c) A UNIX process calls *fork()* to create a child process as shown: $pid = fork()$;
i. What value will be assigned to *pid* in the parent process by the call to *fork()*?
the parent's process id the child's process id zero none of these
ii. What value will be assigned to *pid* in the child process by the call to *fork()*?
the parent's process id the child's process id zero none of these (I don't think a value is assigned)
- (d) The Banker's algorithm is used for deadlock prevention.
~~denial~~ prevention avoidance recovery
Deadlock = thread 2 waiting on B, resolves A, thread 2 waiting on A, resolves B.
- (e) Belady's anomaly can affect the performance of the _____ page replacement algorithm.
FIFO LRU optimal SJF
"recall" and "denial" don't make sense?
- (f) _____ access files are made of fixed length records that allow programs to read and write records in no particular order.
sequential direct logical none of these is correct
- (g) When an I/O request is being handled for a user's process, which term refers to the policy of returning control to the user process before the I/O is completed?
synchronous I/O asynchronous I/O delayed I/O none of these
- (h) Which multithreading model requires that a new kernel thread be created for each new user thread?
many-to-one one-to-one many-to-many none of these is correct
- (i) A process that does not affect, and is not affected by, another process is referred to as:
static independent cooperating dynamic unbounded

5. Software, Programming Techniques

DONE

Given that

 $B(x)$ means "x is a bear" $F(x)$ means "x is a fish", and $E(x, y)$ means "x eats y",

what is the best English translation of

 $\forall x[F(x) \rightarrow \forall y(E(y, x) \rightarrow B(y))]$?

- (a) All fish eat bears. ~~X~~
- (b) Every fish is eaten by some bear. ~~X~~
- (c) Bears only eat fish. ~~$F(x) \rightarrow$~~
- (d) Every bear eats fish. ~~X~~
- (e) Only bears eat fish. ~~X~~

for all things y

if x is a fish

for all things y

if x is eaten by y

y is a bear

all fish are eaten by bears

for everything:

if it is a fish, it is eaten

~~by a bear~~

by a bear

for all fish

if they are eaten by something

the thing is a bear

all fish are eaten by bears?

↓

closest to B?

but B has "some" $\Rightarrow \exists$ not \forall ?

→ Anything that eats a fish is a bear?

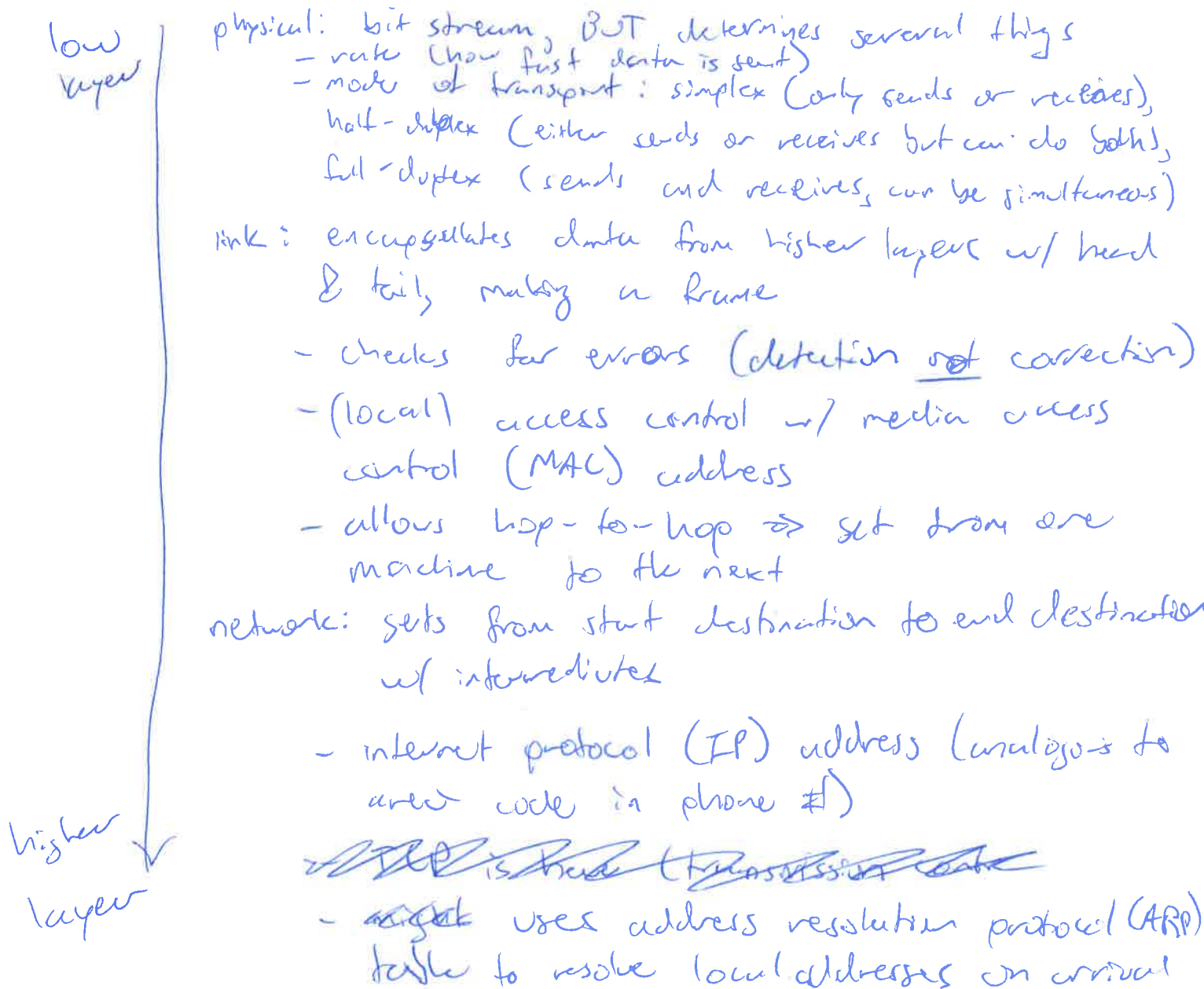
This is A. It's E.

6. Networking and Communications

DONE

- (a) Draw a diagram showing layers of the Internet Protocol Stack and briefly discuss role of each layer.
- (b) Describe functions of each layer when a file is transferred from a source to destination using (file transfer protocol (FTP)).

a) 5 layers are: physical, link, network, transport, application



low
layer

transport: packaging of larger files into segments (fragmentation)

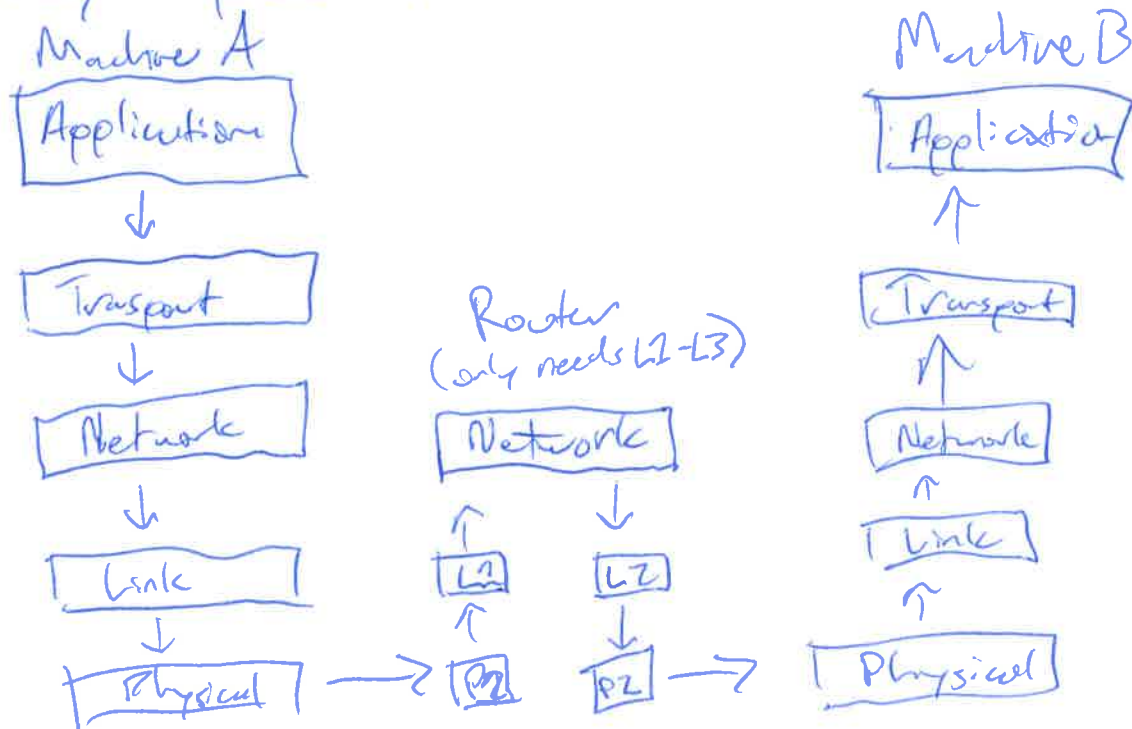
- max frame size in link layer means multiple transport layer packets might be required for one large file
- TCP (transmission control protocol) is a transport layer protocol

application: highest layer, controls, dictates what to send

- FTP (file transfer protocol) is an application layer protocol

highest
layer

Diagram:



5) FTP establishes 2 TCP connections on ports 20 & 21. One is a control, one is for data transmission. (This is somewhat analogous to buses in ~~open~~ comp. architecture). TCP uses 3-way handshake to open stream (e.g. 1) "Hi" 2) "Cool, ready?" 3) "ready!"). Breaks application layer data (i.e. file in FTP) into packets, sends these over physical layer in link layer frames by network layer IP address.

7. Algorithms and complexity

DONE

Describe an algorithm that takes two input lists of integers $A = a_1, \dots, a_n$ and $B = b_1, \dots, b_m$ and delivers the list of all the elements that belong to A but not to B . A and B do not contain redundant elements, however, the elements of A and B might have a large range.

The algorithm should run in $O(n \log m + m \log m)$ time.

Okay: general approach:

- sort A & sort B , can sort B in $O(n \log n)$ _{each}
- now can just go through each list simultaneously \Rightarrow linear search $\Rightarrow O(n)$ or $O(m)$
- $O(n \log n)$ dominates runtime; will be ^{whichever is larger} $O(n \log n)$ for A + $O(m \log m)$ for B

I ensured differences, not only A :

Only diff. is not saving out B -list values

$$O(n \log n + m \log m)$$

\rightarrow not $O(n \log m + m \log m)$

① Something like merge sort of A & B to get ordered $A \cup B$
 - merge sort is $O(n \log n)$ with input size n then and scales well with large arrays

② Linear comparison: approach \Rightarrow take the smaller of the two start values (if different). If same \Rightarrow move to next element in list. If we run out of one list: the rest of the other is unique, add it we're done.

Or (see reverse for implementation)

Recursive approach:

THIS IS CORRECT

scan-return-difference (A (ordered), B (ordered), C):

end conditions {
if A & B are empty: return C
if A is empty & B is not:
return ~~many C's~~
if B is empty & A is not:
return C + A (many concat)

↓
this can start
as an empty
array
(C = [] in
python)

Case #1:
first element
is the same →
it's in both

if $A[0] == B[0]$:

return scan-return-dif ($A[1:]$, $B[1:]$, C)

↳ "the rest of A", etc.

Case #2:
first element in
A is smaller
→ unique
→ save it,
keep going

if $A[0] < B[0]$:

add $A[0]$ to C ($C.append(A[0])$)

return scan-return-dif ($A[1:]$, B, C)

Case #3:
Same as #2
but B is
smaller

if $A[0] > B[0]$:

~~add B[0] to C~~

return scan-return-dif (A, $B[1:]$, C)

You could do a wrapper like:

wrap-dif (A (unsorted), B (unsorted)):

A-sort = merge-sort (A)

B-sort = merge-sort (B)

return scan-return-dif (A-sort, B-sort, [])

Yes not in language?

abacabb or



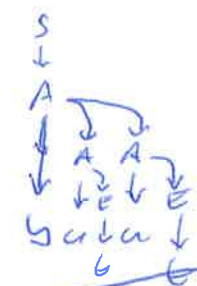
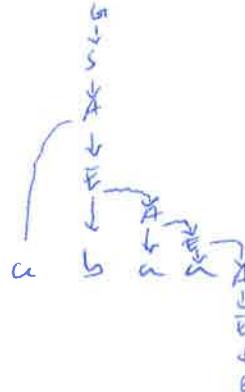
8. Automata and language theory

Consider the following grammar:

- $G \rightarrow S \$ \$$
- $S \rightarrow A M$
- $M \rightarrow S | \epsilon$
- $A \rightarrow a E | b A A$
- $E \rightarrow a B | b A | \epsilon$
- $B \rightarrow b E | a B B$

abacabb

abacabb



Please see reverse for final answers

- Describe the language that the grammar generates in English.
- Show a parse tree for the string a b a a.
- Is the grammar LL(1)? If so, show the parse table; if not, identify a prediction conflict.

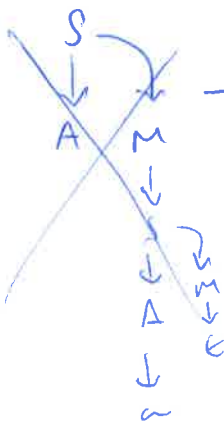
☆☆☆

a) Okay, let me start by just making some strings

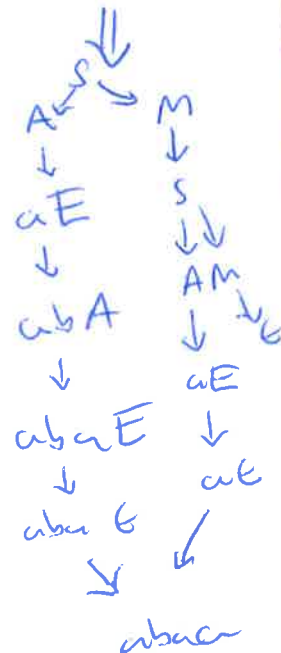
$L = [\epsilon, a, \text{~~ab~~, ~~aba~~, ~~abab~~, ~~abacabb~~, abacabb, abacabb, ...}$

$b a, b a b, b a b b, a b a b a \Rightarrow$ is it "any string from alphabet $\{a, b\}$ s.t. there can be the

b) don't formally remember how to do parse trees, but



abac + a

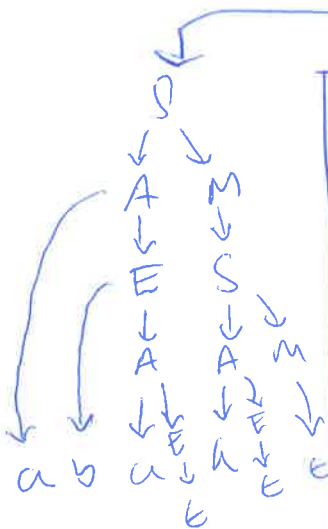


some letter in a row at most 2x? (i.e. yes: as above) no: any letter

NO: If can have AT most 1 triple repeat.

no... maybe no more than 2 bs?

maybe any ab string?

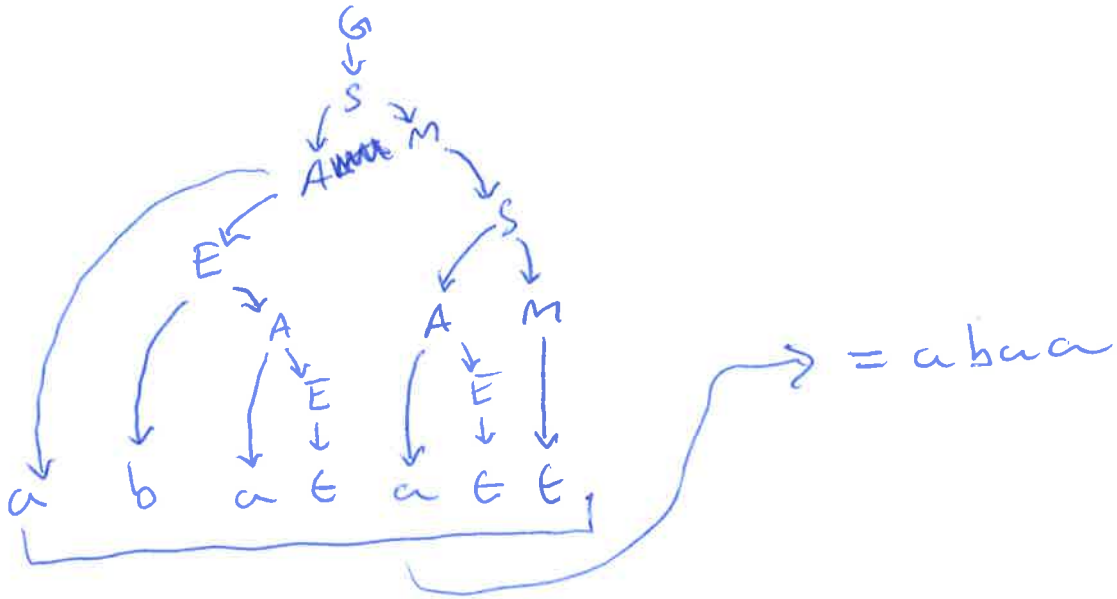


a) I think this grammar describes any string such that the alphabet is $\{a, b\}$. NOPE: "b" is not in the language. Hmm. Well, no string in the language starts with b. Maybe $L = \{ w \text{ s.t. } w[0] \neq b \mid \{a, b\}^* \}$?

I think this is it. Could get opposite by replacing

$S = AM$ with $S = BM$. Nope (bAA) \Rightarrow any string in language must have an a in it.

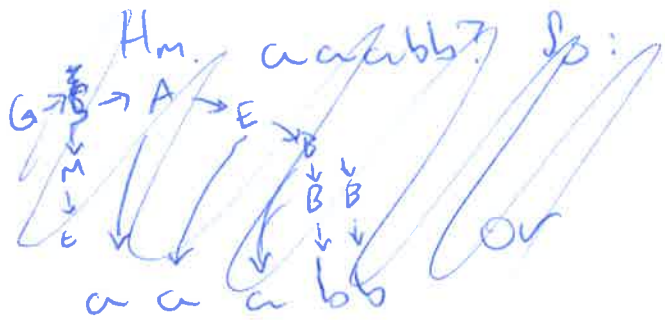
b) I don't formally remember rules for parse trees, but here is my best effort. The key is abc + a ($\approx A + M \Rightarrow A$)



c) Okay, again, don't remember exactly what $LL(1)$ is.

BUT \Rightarrow "predictor conflict" \Rightarrow I'm guessing this means

"Does there exist a string in the language that can be parsed multiple ways?" If yes \Rightarrow not LL(1), if no \Rightarrow LL(1).



9. Discrete Structures

DONE

Recall that the Hamiltonian Cycle Problem is the problem of deciding, on input graph G , whether G has a cycle that visits all the nodes exactly once. Show that this problem is polynomial time decidable if the input is restricted to the graphs with the property that each node has at most two neighbors (i.e., at most two adjacent nodes).

Thought
Process

Okay. UM. Maybe DFS have a start node?

When you hit the start node again, it's a cycle.

Normally, scales like $O(\text{Nodes} + \text{Edges})$.

Infinite edges \Rightarrow 2 scalars, huge. Restrict edges to

2 per node $\Rightarrow 2 \times \# \text{ nodes} \Rightarrow O(\# \text{ nodes})$

$\Rightarrow O(\text{Nodes})$ only.

DFS:

queue a start node

while the queue isn't empty:

take out a node

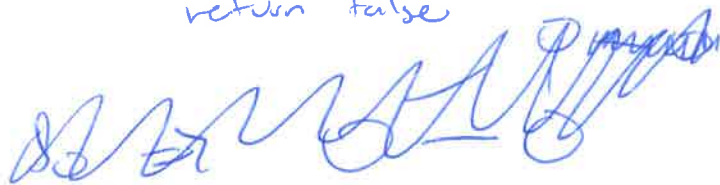
if ~~was~~ it's the start node:

we did it! return true

if we haven't visited it yet: (track of node visited or ^{something})

add to queue it's children (realistically, one other child)

return false



This works because @ most
2 edges restricts to a
set # of possible graphs:



10. Other Topics

DONE

Give a detailed explanation of any one approach to machine learning. Give a substantial example that illustrates the technical operation of the approach, and demonstrates interesting knowledge learned.

→ Okay. Broadly, the field of ML can be grouped as follows:

Unsupervised Learning

- no labels for input data
- exploratory
- examples:
 - o clustering
 - o dimensionality reduction
- concrete examples
 - o Principal Component Analysis (PCA)
 - o K-means clustering
 - o uniform manifold approximation & projection (UMAP)

Supervised Learning

- labels for input (training) data
- goal is to learn mapping of input data to either a categorical label (classification) or a value \hat{y}
- example models range from simple (linear regression) to complex (neural nets)

→ Unsupervised Learning Example: K-Means clustering

- K-means is based on the idea of grouping sets of data into a User-defined set of K clusters.
- it has a 2 step algorithmic approach:

- ① Assignment: each datum is assigned to the cluster centroid C to minimize the Euclidean distance from the datum to C .
- ② Update: the centroids C_1, C_2, \dots, C_K are moved to minimize the variance in distance to the points in their cluster?

This continues until no datum changes assignment to centroid (local) minimum

K-means has several ^{issues} restrictions, though:

- the initial placement of cluster centers (random? how to determine?)
- the # of clusters (user-defined)
- this algorithm can get stuck in local minima (i.e. AN assignment, but not THE BEST assignment)

Supervised Learning Example: Convolutional neural nets for image classification

- Training set of X images $\times 224$ wide $\times 224$ high $\times 3$ color channels
 \hookrightarrow input tensor is $X \times 224 \times 224 \times 3$

- Goes through:

- convolutional layers: inputs are convolved w/ a filter/kernel (squares, odd $\Rightarrow 3 \times 3$, etc.)

- activation layers: alters raw outputs from conv. layer, common functions include RELU \curvearrowright

o RELU function allows all positive values to remain unchanged, sets (-) vals to 0 (\nearrow)

$$\text{RELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

- fully connected (FC) layers: all neurons (individual weighting bins) in one layer connected to those in next



- Softmax layer \Rightarrow normalizes inputs s.t. they form a probability dist. (an array w/ # of classes cells, sums to 1)

- Train network by 2-pass method: propagate training data forward, get predicted outputs, use predictions to backpropagate to alter weights at each layer.

Ex. VGG16

Architecture:

Input \rightarrow 5 blocks of (convolutional layer + activation layer) \rightarrow

23 FC layers \rightarrow softmax layer

Weights of model trained on ImageNet (millions of img./1k classes) available through TensorFlow