Admin

Today's mentor hours moved to 6-8pm

Assignment 4 graded

Assignment 5
- how's it going?
- part A due tonight at 11:59pm
- part B due Friday at 6pm

Course feedback

Thanks!

Course feedback

Overall, how is the class going?

1 2 3 4 5
Course feedback

Favorite thing about the course

Learning Python. It’s pretty dope.

Things to be improved

Sometimes a lot of the lecture material is covered too quickly. I would appreciate a slightly more comprehensive pace, and easier first examples.

Maybe focusing more time on reviewing the concepts that we have already learned just to make sure that the idea is really solidified so that we can use it in a program.

Honestly, it’s a little slow. But I don’t really know how to improve that given that I know literally nothing about coding and need to be walked through things.
Things to be improved

Also I would really enjoy more mentor sessions, on Monday, for example. Or even on Friday.

It would be helpful to have a tutor/mentor session on Sunday nights so that if we run into questions/issues on homework over the weekend, there is somewhere to get help before Tuesday, aka halfway through the week.

Things to be improved

I was talking to a friend about how it’d be really nice to have a buddy to share how I code a function. I thought by watching another person code in a way that is different/similar to yours, you can understand the logic/style better and faster.

Things to be improved

We haven't done that many assignments, but one of them had some grading I didn't understand. I feel like that was an anomaly though.

Things to be improved

More high-five breaks!
Other thoughts

I would really like to do Turing Machines in this class I think they’re fun & have interesting connections/applications to computation/the human mind

Comments in the future…

Grammars

Language view:
A grammar is a set of structural rules that govern the composition of sentences, phrases and words.

Computational view:
A grammar (often called a “formal grammar”) is a set of rules that describe what strings are valid in a formal language.

CFG production rules

S → NP VP

left hand side    right hand side
(single symbol)   (one or more symbols)
Grammars "generate" or "derive" strings:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really} \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } cs
\end{align*}
\]

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.
Grammars “generate” or “derive” strings:

S → A B C  
A → I  
B → really  
B → really, B  
C → like cs  

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.
Grammars "generate" or "derive" strings:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → A B C</td>
<td></td>
</tr>
<tr>
<td>A → I</td>
<td>I really like cs</td>
</tr>
<tr>
<td>B → really</td>
<td></td>
</tr>
<tr>
<td>B → really, B</td>
<td></td>
</tr>
<tr>
<td>C → like cs</td>
<td></td>
</tr>
</tbody>
</table>

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.

No more rules apply, so we're done!
CFG example

Grammars “generate” or “derive” strings:

$$S \rightarrow A \, B \, C$$
$$A \rightarrow I$$
$$B \rightarrow \text{really}$$
$$B \rightarrow \text{really, } B$$
$$C \rightarrow \text{like } cs$$

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right:

$$A \rightarrow \text{really, really, } B \, C$$

CFG example

Grammars describe a language, i.e. the strings (aka sentences) that are part of that language:

$$S \rightarrow A \, B \, C$$
$$A \rightarrow I$$
$$B \rightarrow \text{really}$$
$$B \rightarrow \text{really, } B$$
$$C \rightarrow \text{like } cs$$

$$I \text{ really, really, } \ldots \text{ like } cs$$

What language does this represent?

$$S \rightarrow aS$$
$$S \rightarrow E$$
$$E \rightarrow bE$$
$$E \rightarrow b$$

What language does this represent?

$$S \rightarrow aS$$
$$S \rightarrow E$$
$$E \rightarrow bE$$
$$E \rightarrow b$$

Two options
What language does this represent?

```
S  →  aS
S  →  E
E  →  bE
E  →  b
```

- Can do this as many times as we want
- Keeps adding more a's to the front

What language does this represent?

```
S  →  aS
S  →  E
E  →  bE
E  →  E
```

Eventually, apply second rule
What language does this represent?

S → aS
S → E
E → bE
E → b

Two options

What language does this represent?

S → aS
S → E
E → bE
E → b

aaaE

What language does this represent?

S → aS
S → E
E → bE
E → b

aaabE

What language does this represent?

S → aS
S → E
E → bE
E → b

aaaabE

What language does this represent?

S → aS
S → E
E → bE
E → b

aaaabbbE
What language does this represent?

S → aS
S → E
E → bE
E → b

aaabbE

- Can do this as many times as we want
- Keeps adding more b’s to the end

Eventually, apply second rule

What language does this represent?

S → aS
S → E
E → bE
E → b

aaabb...bE

Notational convenience

S → aS
S → E
E → bE
E → b

S → aS | E
E → bE | b

Grammar represents all strings with zero or more a’s followed by one or more b’s
Often many ways to write the same language

\[
\begin{align*}
S & \rightarrow aS \mid E \\
E & \rightarrow bE \mid b \\
S & \rightarrow aS \mid E \\
E & \rightarrow Eb \mid b \\
S & \rightarrow aS \mid aS \mid E \\
E & \rightarrow Eb \mid b
\end{align*}
\]

What languages do these represent?

\[
\begin{align*}
S & \rightarrow aEa \mid bEb \\
E & \rightarrow Ea \mid Eb \mid a \mid b \\
S & \rightarrow aSb \\
S & \rightarrow ab \\
S & \rightarrow aaS \mid abS \mid baS \mid bbS \mid \varepsilon
\end{align*}
\]

What languages do these represent?

all strings of a’s and b’s that start and end with the same letter

\[
\begin{align*}
S & \rightarrow aEa \mid bEb \\
E & \rightarrow Ea \mid Eb \mid a \mid b \\
S & \rightarrow aSb \\
S & \rightarrow ab \\
S & \rightarrow aaS \mid abS \mid baS \mid bbS \mid \varepsilon
\end{align*}
\]

Writing CFGs

Write a CFG to represent the language containing all strings that start with a.

\[
\begin{align*}
S & \rightarrow aT \\
T & \rightarrow Ta \mid Tb \mid \varepsilon
\end{align*}
\]
Writing CFGs

Write a CFG to represent the language containing all strings with exactly two bs.

$$S \rightarrow TbTbT$$
$$T \rightarrow Ta \mid \epsilon$$

CFG: Another example

Many possible CFGs for English, here is an example (fragment):

$$S \rightarrow NP \ VP$$
$$VP \rightarrow V \ NP$$
$$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$$
$$AdjP \rightarrow Adj \mid Adv \ AdjP$$
$$N \rightarrow \text{boy} \mid \text{girl}$$
$$V \rightarrow \text{sees} \mid \text{likes}$$
$$Adj \rightarrow \text{big} \mid \text{small}$$
$$Adv \rightarrow \text{very}$$
$$DetP \rightarrow a \mid \text{the}$$

Derivations in a CFG

What can we do?

$$S \rightarrow NP \ VP$$
$$VP \rightarrow V \ NP$$
$$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$$
$$AdjP \rightarrow Adj \mid Adv \ AdjP$$
$$N \rightarrow \text{boy} \mid \text{girl}$$
$$V \rightarrow \text{sees} \mid \text{likes}$$
$$Adj \rightarrow \text{big} \mid \text{small}$$
$$Adv \rightarrow \text{very}$$
$$DetP \rightarrow a \mid \text{the}$$
Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adv → very
DetP → a | the

NP VP

What can we do?
Derivations in a CFG

\[
S \rightarrow NP \ VP \\
NP \rightarrow DetP N | DetP AdjP N \\
VP \rightarrow V NP \\
DetP \rightarrow a | the \\
Adj \rightarrow \text{big | small} \\
Adv \rightarrow \text{very} \\
N \rightarrow \text{boy | girl} \\
V \rightarrow \text{sees | likes} \\
AdjP \rightarrow \text{Adj | Adv AdjP} \\
DetP \rightarrow \text{DetP AdjP N} \\
N \rightarrow \text{boy | girl} \\
V \rightarrow \text{sees | likes} \\
Adv \rightarrow \text{very} \\
Adv \rightarrow \text{very} \]

the boy VP

the boy likes NP

the boy likes a girl

the boy likes a girl
Derivations of CFGs

Derivation history shows a tree:

```
S
  |      |      |
NP  V   VP
  |      |      |
 DetP N  V
  |      |      |
  the  boy  likes
  |      |      |
  DetP N
  |      |      |
  a  girl
```

Another CFG example

```
S → NP VP
VP → V | V ADV
NP → ART PreNP
PreNP → N | ADJ PreNP
ADV → furiously | soothingly | intentionally
ADJ → colorless | green | smelly
ART → the | a
V → sleeps | eats | swims | sprints
N → idea | bagel | milk | cow
```

What can we generate?

One last example

```
S → N
S → ( S )
S → S + S | S - S
S → S * S | S / S
N → 0 | 1 | 2 | ... | 9
N → N N
```

What language does this CFG represent?

One last example

```
S → N
S → ( S )
S → S + S | S - S
S → S * S | S / S
N → 0 | 1 | 2 | ... | 9
N → N N
```

All arithmetic expressions!
Given a CFG and a sentence, determine the possible parse tree(s)

I eat sushi with tuna

What parse trees are possible for this sentence?

How did you do it?

What if the grammar is much larger?

What is the difference between these parses?

Midterm

Average: 26.3 (82.5%)

Median: 27.5 (86%)

Quartiles:
- Q1: 30 (97.5%)
- Q2: 27.5 (86%)
- Q3: 25 (78%)