# Discrete Maths

Given a grammar, generate sentences in the language as follows:
(derivation steps)

1. Begin with the starting symbol as current status of string
2. If any portion of the current status of string matches the left hand side of production, replace the portion of the string by right hand side of the production.

3. Any string of terminals obtained by repeating step 2 is a sentence in the language.

P.S.
If during step 2 there exists more than one production, in that case any one of the productions can be chosen.
On the other hand, if we reach where no productions can be applied then its dead end then restart from S with different combinations.

Sentence → noun_phrase
intransitive_verb_phrase

→ noun_phrase
intransitive_verb adverb

→ noun_phrase
intransitive_verb rapidly

→ noun_phrase runs ~~rapidly~~

→ article noun ~~runs~~ rapidly

→ ~~article~~ dog runs rapidly

→ a dog runs ~~rapidl~~y.

Example

Construct a grammar for

$$L = \{ aaaa, aabb, bbaa, bbbb \}$$

Also, derive example strings.

$S \rightarrow aaaa \mid aabb \mid bbaa \mid bbbb$

---

$S \rightarrow AA$

$A \rightarrow aa$

$A \rightarrow bb$ ..

Example $\overset{i.e.}{b}$ bbab abbab    0, 3, 6, ... number of a's

$L = \{x \mid x \in \{a, b\}^*$, the number of a's in $x$ multiple of 3 $\}$

$T = \{a, b\}$    $N = \{S, A, B\}$

$S \rightarrow bS$    # 0 times a's
$S \rightarrow b$    # just b

$S \rightarrow aA$    1st occurance of a
$A \rightarrow bA$    Any inbetween b's
$A \rightarrow aB$    2nd time a's
$B \rightarrow bB$    Any inbetween b's
$B \rightarrow aS$    3rd time a and repeat
$B \rightarrow a$    3rd time a itself.

$S \Rightarrow bS$    # ~~R1~~    $S \to bS$

$\Rightarrow bbS$    # ~~R1~~    $S \to bS$

$\Rightarrow bbaA$    #    $S \to aA$

$\Rightarrow bbabA$    #    $A \to bA$

$\Rightarrow bbabaB$    #    $A \to aB$

$\Rightarrow bbababB$    #    $B \to bB$

$\Rightarrow bbababbB$    #    $B \to bB$

$\Rightarrow bbababbaS$    #    $B \to aS$

$\Rightarrow bbababbab$    #D    $S \to b$

Example:

show your observation about below grammar

$S \rightarrow bS$

$S \rightarrow b$

$S \rightarrow aA$

$A \rightarrow aB$

$B \rightarrow aS$

what is more or less compare to previous ?

Example

show your observation about language for below grammar

$S \to aB$

$S \to bA$

$A \to a$

$A \to aS$

$A \to bAA$

$B \to b$

$B \to bS$

$B \to aBB$

* sentence may start with 'a' or 'b'

* NT A has one more a than b's

* NT B has one more b's than a's

* S adds on either a for B
  or b for A

  that is
  total number of a's and b's
  are always exactly same.

* positions may vary)?

Example

FSM

$n_1 = n_2$ : E

$n_1 < n_2$ : Small

$n_1 > n_2$ : Big

Information Processing machine

Assume ; Lower order digits are fed first.

| $n_2$ | $n_1$ | |
|---|---|---|
| 00 | 00 | E |
| 01 | | S |
| 10 | | S |
| 11 | | S |
| 00 | 10 | B |
| 01 | | B |
| 10 | | E |
| 11 | | S |
| 00 | 01 | B |
| 01 | | E |
| 10 | | S |
| 11 | | S |
| 00 | | |
| 01 | | |
| 10 | | |
| 11 | | |

L, L, L, E

| state | Input n1 n2 00 | n1 n2 01 | n1 n2 10 | n1 n2 11 | output |
|---|---|---|---|---|---|
| A equal | A | C | B | A | Equal |
| Big | B | C | B | B | Big |
| C small | C | C | B | C | Small |

If input $n_1$ digits value so far > $n_2$ digits value so far

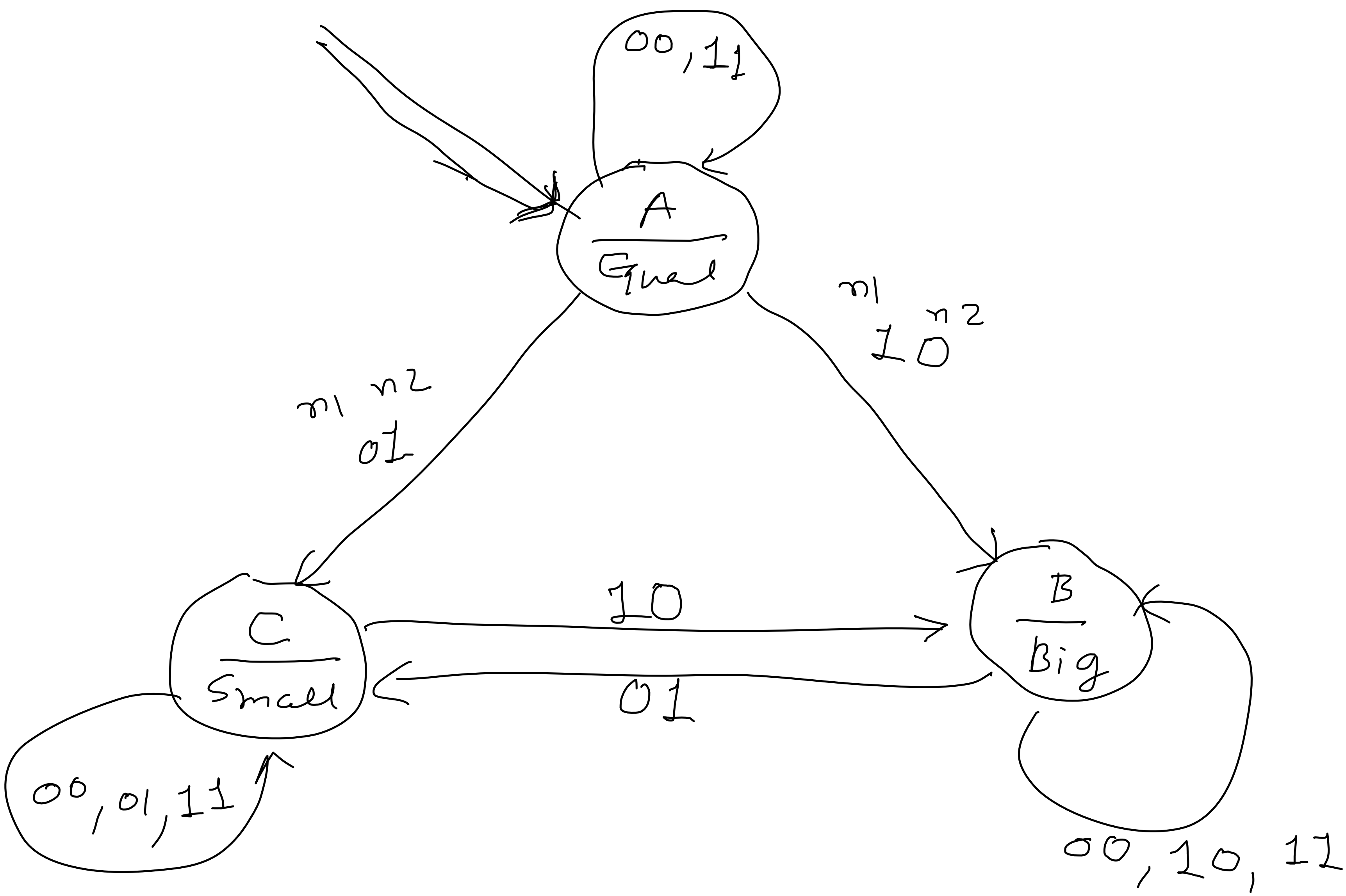
present BIG

| n1 | n2 |
|---|---|
| 0 | 1 |

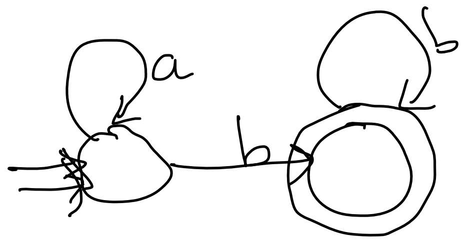
when higher most significant side '1' is present in n2 then n2 becomes > n1.

$\therefore$ n1 < n2

SMALL C.


00, 11
A
Equal
n1 n2
01
n1
n2
10
C
Small
10
01
B
Big
00, 01, 11
00, 10, 11

Write grammar for below FSM




$a^* b b^*$

T {a, b}

NT = {S, A }

S start

$S \to aS$

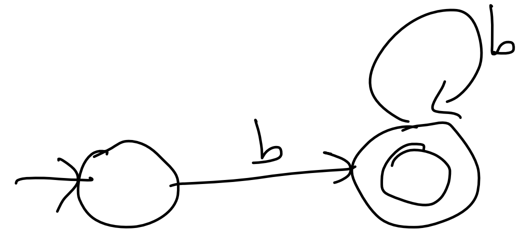
$S \to b$

$A \to bA$

$S \to bA$

$A \to b$

Accept {A}




T = {b}

NT = {S }

$S \to bA$

~~S~~ $\to b$

$A \to bA$

$A \to b$

??

## Example

Given two machines m1 & m2

Are they equivalent machines?

**m1**

| | Input 0 | Input 1 | output |
|---|---|---|---|
| A | B | C | 0 |
| B | B | D | 0 |
| C | A | E | 0 |
| D | B | E | 0 |
| E | F | E | 0 |
| F | A | D | 1 |
| G | B | C | 1 |

**m2**

| | Input 0 | Input 1 | Output |
|---|---|---|---|
| A | H | C | 0 |
| B | G | B | 0 |
| C | A | B | 0 |
| D | D | C | 0 |
| E | H | B | 0 |
| F | D | E | 1 |
| G | H | C | 1 |
| H | A | E | 0 |

$\pi_1$

$\Pi_0$ ~~based on unique output~~

$= \{ \overline{ABCDE}, \overline{FG} \}$

$\Pi_1$

| | | |
|---|---|---|
| AB | *BB | +CD |
| AC | BA | CE |
| AD | BB | CE |
| AE | BF | CE |
| BC | BA | DE |
| BD | BB | DG |
| BE | BF | DE |
| CD | AB | EG |
| CE | AF | EE |
| DE | BF | EE |

* For state A on 0 to B
For state B on 0 to B
Hence, BB is successor of AB on 0 input

+ For state A on 1 to C
For state B on 1 to D
Hence, CD is successor of AB on 1 input.

| | 0 | 1 | Will be together |
|---|---|---|---|
| AB | ABCDE ✓ | ABCDE ✓ | Yes |
| AC | ABCDE ✓ | ABCDE ✓ | Yes |
| AD | ABCDE ✓ | ABCDE ✓ | Yes |
| AE | ABCDE ✗ FG | ABCDE ✓ | NO |
| BC | ABCDE ✓ | ABCDE ✓ | Yes |
| BD | ABCDE ✓ | ABCDE ✓ | Yes |
| BE | ABCDE ✗ FG | ABCDE ✓ | NO |
| CD | ABCDE ✓ | ABCDE ✓ | Yes |
| CE | ABCDE ✗ FG | ABCDE ✓ | NO |
| DE | ABCDE ✗ FG | ABCDE ✓ | NO |

FG

0: AB → ABDC ✓

1: DC → ABDC

Yes

FG stays together

$\Pi_1 \quad \{ \overline{ABCD}, \overline{E}, \overline{FG} \}$

Generate $\Pi_2$

| | 0 | 1 | |
|---|---|---|---|
| AB | ABCD | ABCD | Yes |
| AC | ABCD | ABCD E X | NO |
| AD | AB CD | ABCD E X | NO |
| BC | ABCD | ABCD E X | NO |
| BD | ABCD | ABCD E X | NO |
| CD | ABCD | ABCD E | Yes |
| FG | ABCD | ABCD | Yes |

$\Pi_2 \Rightarrow \{ \overline{AB}, \overline{CD}, \overline{E}, \overline{FG} \}$

~~Generating~~ $\Pi_3$ || Given $\Pi_2$ $\{\overline{AB}, \overline{CD}, \overline{E}, \overline{FG}\}$

| | 0 | 1 | |
|---|---|---|---|
| AB | AB | CD | yes |
| CD | AB | E | yes |
| FG | AB | CD | yes |

$\Pi_3 = \{\overline{AB}, \overline{CD}, \overline{E}, \overline{FG}\}$

$\Pi_2$ and $\Pi_3$ are same exact.

Hence, stop the ~~algorithm~~.

Final, $\{\overline{AB}, \overline{CD}, \overline{E}, \overline{FG}\}$ m1 m2

$\Pi_0$ based on unique output

$\{ \overline{ABCDEH}, \overline{FG} \}$

$\Pi_1 = \{ \overline{ACDEH}, \overline{B}, \overline{FG} \}$

$\because AB \xrightarrow{0} HG \quad H \in A\boxed{B}CDE\underline{H}$
while $G \in A\underline{G}$

$\Pi_2 = \{ \overline{ADH}, \overline{CE}, \overline{B}, \overline{FG} \}$

$\because$ Seperation

$AC \xrightarrow{1} \quad C \in A\underline{C}D\,EH$
while $B \in \underline{B}$

$AE \xrightarrow{1} \quad C \in ACDEH$
while $B \in B$

$DE \xrightarrow{1} \quad C \in ACDEH$
$B \in B$

other side

$CE \xrightarrow{0} AH \in \underline{A}CDE\underline{H}$

$\xrightarrow{1} BB \in \underline{\underline{B}}$

CE will remain together.

$\Pi_3 = \{ \overline{ADH}, \overline{CE}, \overline{B}, \overline{FG} \}$

As $\Pi_2 = \Pi_3$

stop algo.

m 2

$\{ \overline{ADH}, \overline{CE}, \overline{B}, \overline{FG} \}$

$m_1$

| | | 0 | 1 | |
|---|---|---|---|---|
| A' | $\overline{AB}$ | $\overline{AB}$ | $\overline{CD}$ | 0 0 ⇒ 0 |
| B' | $\overline{CD}$ | $\overline{AB}$ | $\overline{E}$ | 0 0 ⇒ 0 |
| C' | $\overline{E}$ | $\overline{FG}$ | $\overline{E}$ | 0 ⇒ 0 |
| D' | $\overline{FG}$ | $\overline{AB}$ | $\overline{CD}$ | 1 1 ⇒ 1 |

| | 0 | 1 | |
|---|---|---|---|
| A' | A' | B' | 0 |
| B' | A' | C' | 0 |
| C' | D' | C' | 0 |
| D' | A' | B' | 1 |

$m_2$

| | | 0 | 1 | | |
|---|---|---|---|---|---|
| A' | $\overline{ADH}$ | $\overline{ADH}$ | $\overline{CE}$ | 0 0 0 | 0 |
| B' | $\overline{CE}$ | $\overline{ADH}$ | $\overline{B}$ | 0 0 | 0 |
| C' | $\overline{B}$ | $\overline{FG}$ | $\overline{B}$ | | 0 |
| D' | $\overline{FG}$ | $\overline{ADH}$ | $\overline{CE}$ | 1 1 | 1 |

| | 0 | 1 | |
|---|---|---|---|
| A' | A' | B' | 0 |
| B' | A' | C' | 0 |
| C' | D' | C' | 0 |
| D' | A' | B' | 1 |

Both $m_1$ & $m_2$ are equivalent machines.