

20 February 2025

Thursday

Announcement:

Assignment Submission week 3 Quiz ২০১,

regEx identifier \wedge
অ্যালিটেনেটিভ, constant \wedge
অ্যালিটেনেটিভ lab ৩

Common RegEx Symbols:

• Kleene Closure:

a^* $\rightarrow \epsilon, a, aa, aaa, \dots$

Concatenation:

$a \cdot b = ab$

• Positive Closure:

a^+ $\rightarrow a, aa, aaa, \dots$

↪ minimum একাগার স্থলত রয়েছে।

Union:

$a \cup b$ or alb অথবা

অথবা অথবা, (a or b)

$(a \cup b)^*$ $\rightarrow \epsilon, a, b, abbab, \dots$

Bracket এর precedence ক্ষমতা বেশি রয়েছে,

এখনও সবচেয়ে তরফে higher precedence অথবা Kleene Closure এর।

Positive closure

$a^* b + c$

Kleene Closure, Positive Closure

precedence রয়েছে

Concatenation

Union

• $[]$ \rightarrow exactly one character at a time রয়েছে,

↪ example,

$[b, r, n]ed \rightarrow bed, red, ned,$

$[b, r, n]^*$ \rightarrow Kleene Closure অথবা \wedge choose portion
multiple time রয়েছে,

• $-$ \rightarrow range রূপায়,

$[a-z]^*$ $\rightarrow zacb\dots\dots$

a থেকে z range এর মধ্যে তারা একাই character choose
হচ্ছে তার ফলস্থূলু ক্লিনে ক্লিনে closure এর হো �choose multiple times

$a^* \rightarrow \epsilon, a, aa, aaa, aaaa, \dots$

$a^+ \rightarrow a, aa, aaa, aaaa, \dots$

Now suppose our string is aa exactly into two parts a and -

now we have to match string is "aa". so our regEx.

For this,

- $a^m \rightarrow a$ will be repeated m times.
(fixed)

$\therefore a^2 = aa \rightarrow$ fixed 2 times.

at least m times repeat upto n times at max n times.

$a^{m,n} \rightarrow$ at least m times.

example: $a^2 \rightarrow aa, aaaa, \dots$

at least m times and at most n times repeat upto n times.

$a^{m,n} \rightarrow$ at least m times at max n times.

example: $a^2, 4 \rightarrow aa, aaa, aaaa$

"a" → exactly a character.

• normally - sign ∞ range शब्द,

■ Suppose वर्तमान में कोई string "a-z" regEx फॉर्मूला विकल्प बनाए जाए, a or z तो easily express कर सकते हैं; but क्या अन्यतर - (hyphen) express कर सकते हैं?

→ "—" को literally hyphen शब्द

→ - को also फॉर्मूला विकल्प range शब्द,

→ \-

literal character { "—" }

→ यह इसके लिए एक विकल्प को बनाता है।
express कर सकता है।

∴ "a-z" can be expressed as,

a. "-".z or a.(\-).z

compiler एवं defined फॉर्मूला रूप \rightarrow , " " string

○ यहाँ literal character विकल्प express कर सकता है।

• \$ द्वाये end of line शब्द,

ab \$ cd \rightarrow ab
 cd

regEx
↓
NFA
↓
DFA

} then
match
होगा।

• ? द्वाये शब्द zero or one occurrence.

colour? \rightarrow colon, colour.

bracket वर्तमान \rightarrow For
NTS की Question गती,
325 तक 323 विं 213
205 विं,

Theory \Rightarrow regEx generate
विकल्प बनाते हैं तो तो बनाते हैं
DFA

- Create 5 strings using the given regEx,

$[a-z A-Z -] [a-z A-Z -]^*$

- a
- z a A Z b
- Back -
- b E - n
- apple
- - Ant

Regular Definitions:

- regEx को एक ग्रामार्थ ना कहते हैं लात्यर इसे उनकी जांच करने की वजह से इसका अर्थ बताया जाता है।
- kind of like a variable ; it stores expression stored in memory.

letter → $[a-z A-Z -]$

letter definition is corresponding action.

$[a-z A-Z -] [a-z A-Z -]^* \rightarrow [letter] [letter]^*$

digit → $[0-9]$

integer → $[digit] [digit]^*$

$[a,b,c]$ → $0-9$

(a,b,c) → $[0-9]$

• $(0-9)$

cd $(a^* b c) e^*$

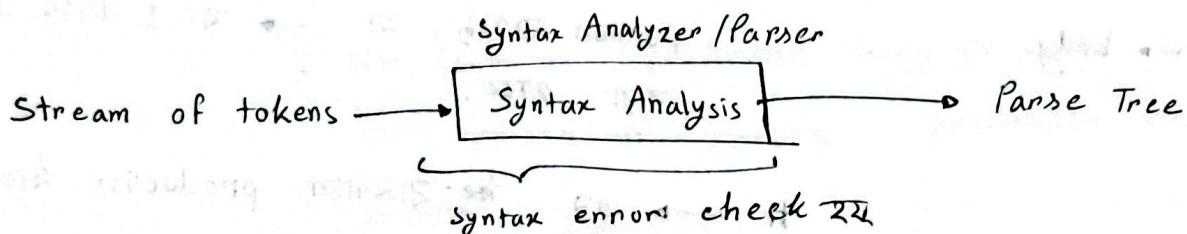
$(a b^*) \rightarrow$ $b \Rightarrow$ यह क्लिने closure.

(alb)*

- Quotation match string এ পাইলে "" বা \"
- lexeme match করে \Rightarrow regEx এ মিল হলে then token generation.

Syntax Analysis Phase

- Syntax analysis phase এ parse tree generate করা,
- হমের parse tree এ স্ট্যাকে stack ফর্মে push করা,
- Syntax error checking এর CFG ফর্মে,



- CFG এর মিল করে, automate করে, generate parse tree.
- CFG থেকে এর parse tree (table) generate করি, এতে তথ্য ক্ষেত্র চেক করি
- Parser, parsing করে generate parse table এর মাধ্যমে code করি যেখানে সেটি কো-কেশ করে।
- Parser always left to right parse করে। এবং আসের token generation ও left to right এর।
- CFG \rightarrow Context Free Grammar
 - CFG এর 4 BT term কি?
 - i Terminal *
 - iii Start *
 - ii Non Terminal *
 - iv Production Rules *

Parse tree গুরুতে সঠিক means
my stream of tokens are correct.

$$\begin{array}{l} S \rightarrow E \\ E \rightarrow T + E \\ T \rightarrow F \\ F \rightarrow id \end{array} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{grammar}$$

ଶବ୍ଦ ଏବଂ ଅନେକ ଶବ୍ଦରେ = ପରିଚାର

- Start \rightarrow expression
- expression \rightarrow term + factor
- term \rightarrow factor
- factor \rightarrow id

grammar
CFG

ଏହାରେ, ଏହି ପ୍ରକାର ପ୍ରକାର ଏହି ଏହି

head \rightarrow body

portion ପରିଚାର. ଯା \rightarrow ଯାହା ନିଯମ ଆବଶ୍ୟକ

\rightarrow ଏହି କେବଳତାର production head

\rightarrow ଏହି କେବଳତାର production body

Production Rules: Production Head: Production Body

Start:

- Start ନିଯମ starting point denote କରୁଣ୍ଟାର, ଡାକ୍ତର ଗ୍ରାମ୍‌ପାଠୀ
- Starting symbol ଏବଂ ସମ୍ବନ୍ଧୀୟ ପ୍ରକାର ନକାରାତ୍ମକ ନାହିଁ, not necessary.

Start: start

- Start ଏହି ରେ କବିତାର starting point ଏବଂ ଅନେକ ଶବ୍ଦରେ
- ଶବ୍ଦରେ କବିତାର ପରିଚାର ନିଯମ start କରିବାର, start ଏହି
- ଏହି କବିତାର production rules ଏବଂ ଶବ୍ଦରେ

Terminal : Direct token എന്നൊരു lexer മുൻപ് pars ചെയ്യും.

Non Terminal: • Grammar portion

- യാതൊരു production head അല്ലെങ്കിൽ, production head ഒരു ടക്സിംഗ് നാം ടക്സിംഗ് ഓഫ്, non terminal production body എന്നും ടക്സിംഗ് production head എന്നും പറയാം,
- Token നാം
- ടക്സിംഗ് നാം ടക്സിംഗ് production head അല്ലെങ്കിൽ
- Non terminal production body എന്നും കാണുന്നത് ടക്സിംഗ് നാം ടക്സിംഗ് production head അല്ലെങ്കിൽ പ്രൈസ് എല്ലാം
- Non terminal hence:
expression, term, factor

id, +, * എന്നൊരു direct tokens നോ എന്നൊരു terminal .

Terminal: id, +, *

↳ Production body എന്നും, അല്ലെങ്കിൽ production head അല്ലെങ്കിൽ

Start one kind of non terminal, special kind of
non terminal as it denotes the starting point #