

$$E = \{a^i b^j \mid i \neq j \text{ and } i \neq 2j\}$$

تعداد a ها با b ها
برابر نیست

تعداد a ها با b ها
نایب برابر یا تعداد a ها
باشد.

سوال ۱) به این مجموعه E متعلق از مقوله است که تعداد آن را بصورت اجتماع چند زبان کسر
نکته دوم: ساختار متعلق از مقوله نسبت به مجموعه اجتماع بسته هستند

مثله اول: $E_1 = \{a^i b^j \mid i \neq j\}$: تعداد a ها بیشتر از b ها است (هسته)

$$S \rightarrow aAB$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow aBb \mid \epsilon$$

تولید a در ابتدا

تولید کردن a

تولید کردن a, b به تعداد مساوی

مثله دوم: $E_2 = \{a^i b^j \mid j > 2i\}$: تعداد b ها بیشتر از ۲ برابر تعداد a ها است.

$$S \rightarrow ABb$$

$$A \rightarrow aAbb$$

$$B \rightarrow Bb \mid \epsilon$$

۱. ۲ برابر a تعداد b ها
تولید b در ابتدا

۲. برابر بودن تعداد b ها

افزانه کردن b

مثله سوم: $E_3 = \{a^i b^j \mid i < j < 2i\}$: تعداد b ها از a ها بیشتر، و از ۲ برابر a ها کمتر است

$$S \rightarrow aAb$$

$$A \rightarrow aAb \mid aAbb \mid abb$$

$$E = E_1 \cup E_2 \cup E_3$$

در نهایت

چون هر کدام E_i ها $\{1, 2, 3\}$ متعلق از مقوله هستند، E هم متعلق از مقوله خواهد بود

نتیجه

سوال (۲)

in every prefix of ω the number of ϕ 's is at least the number of 1's?

$\{0,1\}^*$

این یعنی تا جایی که پیشوند را می‌بینیم، تعداد صفرها باید به تعداد یک‌ها برسد. یعنی در هر پیشوند، تعداد صفرها باید به تعداد یک‌ها برسد.

$$S \rightarrow \epsilon \mid \phi S \mid \phi S 1 S$$

بیشود. حالت هم می‌تواند باشد. — به این معنی:

پس prefix نه تعداد ϕ و 1 برابر دارد. این را می‌توان به معنی است و باید اینجا را به طرف کنیم.

در هر پیشوند، تعداد ϕ و 1 برابر است. \downarrow prefix نه در هر پیشوند، تعداد ϕ و 1 برابر است.

ادامه سوال ۲) مثلاً برای $\phi \phi S 1 S$ داریم بالا میجاء

$$S \rightarrow \phi S 1 S \rightarrow \phi \phi S 1 S$$

$$\phi \phi \epsilon 1 \epsilon = \phi \phi 1$$

$$S \rightarrow \phi S$$

$$\phi \phi S 1 S$$

$$\phi \phi \epsilon 1 \epsilon = \phi \phi 1$$

راه حل بیان اینجا :

$$S \rightarrow A | B$$

$$A \rightarrow \phi A 1 A | \epsilon$$

$$B \rightarrow \phi S | \phi A 1 B$$

دایره‌های آبی که ما داریم

$$B^* A B^*$$

تعلیمی کنیم. در جاهای ϕ و 1 با هم برابر

(A) و در B این قابل باید هم منبریم و

از یای که هم اول و هم آخر است / از ته اما نه

نه ارد

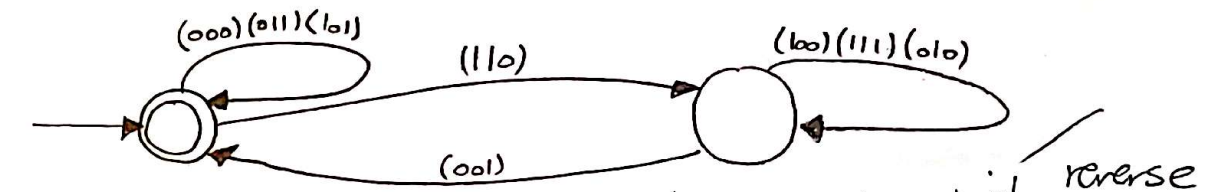
$\tilde{S} = -Y + a1 * (\varepsilon - b)$
 $S \rightarrow T+S \mid T-S \mid \tau$
 $T \rightarrow P*T \mid P/T \mid P$
 $P \rightarrow (S) \mid [S] \mid v \mid N$
 $N \rightarrow M \mid -M$
 $M \rightarrow [0-9]M \mid [0-9]$
 $v \rightarrow [a-z]G$
 $G \rightarrow [a-z]G \mid [0-9]G \mid \varepsilon$

سريع $S \rightarrow T+S$
 $T \rightarrow P*T \rightarrow P$
 ايضا $T = T + P * P$
 $(S) \rightarrow (T-S)$
 $T \rightarrow P \rightarrow v \rightarrow [a-z]G$
 $bG \rightarrow b\varepsilon = \underline{\underline{b}}$
 $N \rightarrow M \rightarrow -M \rightarrow -[0-9] = -Y$
 $a[0-9]G \rightarrow a1G \rightarrow a1\varepsilon = \underline{\underline{a1}}$

سوال ٣

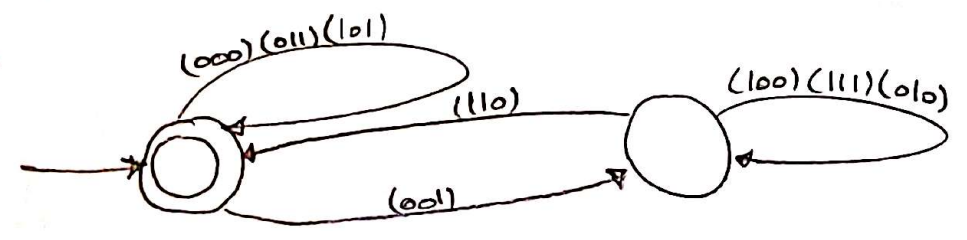
در نهایت $-Y + a1 * (\varepsilon - b)$

سوال ٤



باید برعکس

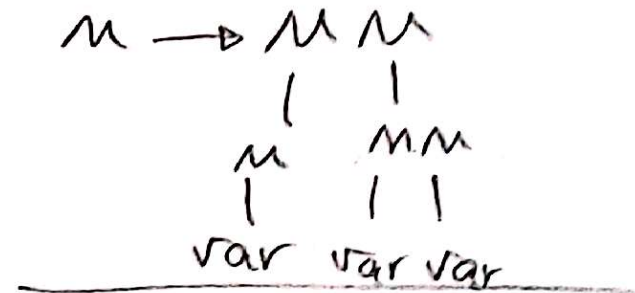
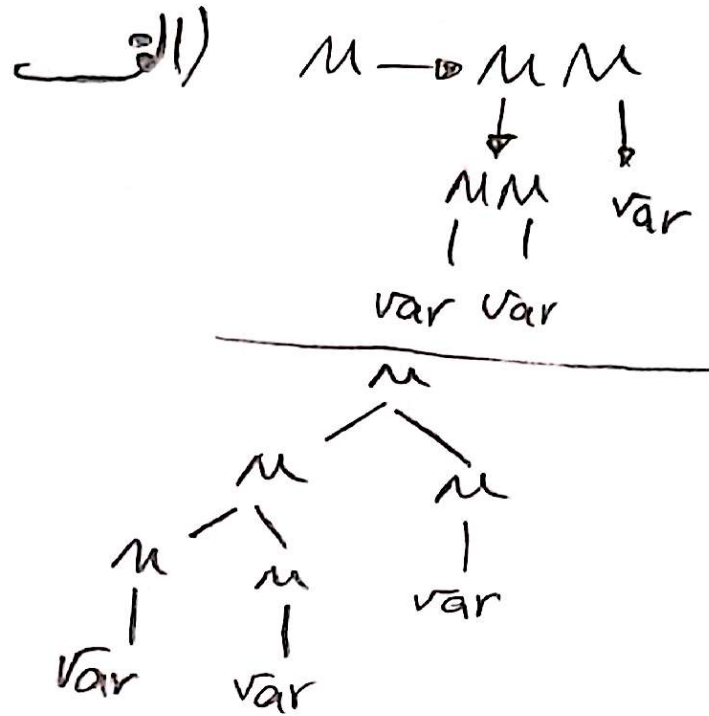
نیم (محیط طبق با همفانی) و الله جمع از است به چه این می شود ، عمل کنه ایم



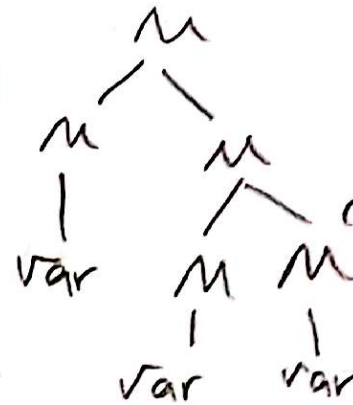
Lambda Calculus

$$M \rightarrow \text{var} \mid MM \mid \lambda \text{ var. } M \mid (M)$$

سوال ۱)



ws var var var



var غیر قابل استفاده

غیر قابل استفاده که مورد نیاز نیست

identifiers قسمتی است

که در متن می بینیم و باید در کد هم باشد

اینکه بدی هم می تواند lambda باشد که دوباره در مدل اشتقاق داریم.

ب)

بدون غیر معمول
(de, r)

$$M \rightarrow A \mid MA$$

$$A \rightarrow \text{var} \mid \lambda \text{ var. } A \mid (M)$$

$$M \rightarrow A \mid \lambda \text{ var. } M \mid (M)$$

$$A \rightarrow A \text{ var} \mid \text{var}$$


```

function List [a:b] (List L, int a, b) {
    if (a == b) return L[a]
    return L[a] + L[a+1:b]
}

```

که مربوط به این
کتابی که نتوانست عناصر را
تاریخ

سوال ۹) با فرض اینکه می‌توان به تعدادی از عناصر لیست دسترسی پیدا کرد (مثلاً $L[1:n]$ یعنی از عنصر دوم تا آخر لیست را فرد می‌دهد)

الف) Function List Swap-declarative-in-list (List L) {

List temp = L[L.length()-1] + L[1:L.length()-2] + L[0]

return temp
}

ب) Function List Swap-imperative-in-list (List L) {

int temp = L[0]

L[0] = L[L.length()-1]

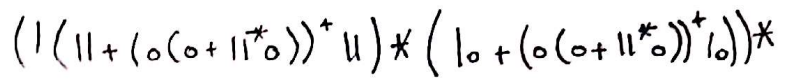
L[L.length()-1] = temp

return L

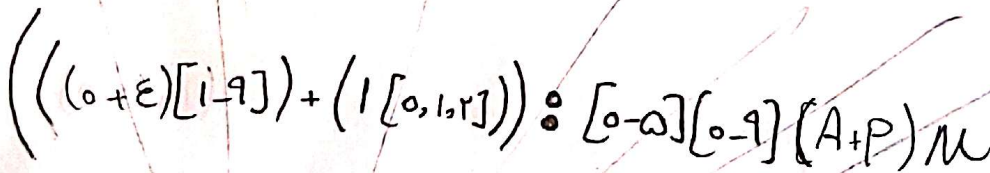
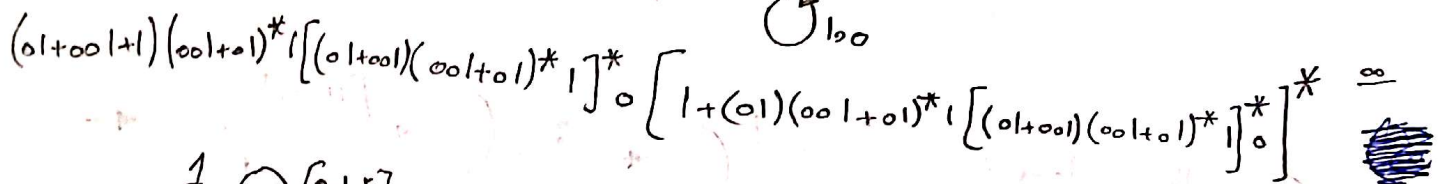
}

ج) در روش declarative ما می‌پسندیم سافت کد لیست جدید به یک خطه تعریف کنیم که $O(n)$ هزینه دارد. اما در روش imperative به صورت
inplace در خود لیست و صرفاً با تعریف یک متغیر اضافی به کار swap کردن که $O(1)$ هزینه دارد، نیازمندیم.

ماہانہ نوٹسنگ سرگرمی: GNFA دھوکہ دہہ دونوں State جانا بہائی
سرگرمی واعدہ بن سیم



راه حل: تبدیل به GNFA
دو نوع state حالتی به اسم
نام ببریم



trap

سوال ۸

$$0^*(1^*00^*)^*1^*0^*$$

الف) مجموعه اعداد دودویی که شامل 1 و 0 باشد

$$(0+1)^*00^*$$

ب) مجموعه اعداد دودویی که به 0 ختم شود

$$-(1^*0^*1^*)^*1^*(0^*1^*0^*)^*0^*$$

ج) رشته‌های دودویی که تعداد 1 در آن زوج و تعداد 0 در آن فرد باشد

$$w = 110010010101010101$$

در نظر گرفتن 1ها

الف) 110010010101010101

ب) 110010010101010101 → اینها زنده نیست → panic mode ↓
اینها د میس

ج) 110010010101010101

در نظر گرفتن 0ها

$$110010010101010101$$

شامل شدن 1ها (الف) → زوج و فرد (ج)

اولیها را از الف به ج در نظر می‌گیریم