

به نام خدا

محمد مهدی آقاجانی

۹۳۳۱۰۵۶

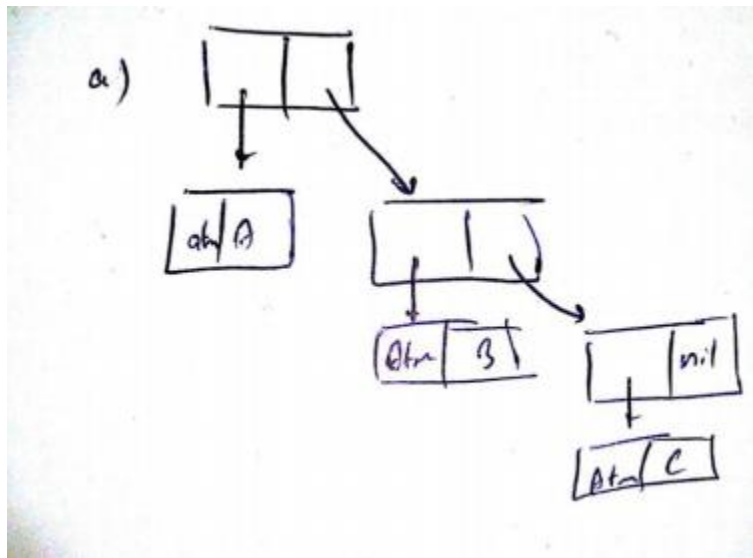
تمرین سوم

استاد : دکتر سلیمان فلاح

فصل سوم

سوال ۱

(a)



(b)

`cons(cons('A 'B.C) 'B.C)`

(c)

`(define superConcat (lambda(x)(cons(car(x) cdr(x)))))`

`(cons(superConcat(cons('A 'B.C))))`

سوال ۲

(الف)

امکان ندارد زیرا اگر π را تعریف نکنیم حالت halt پیش می آید. شرط ها به طور متوالی اجرا میشوند و اگر به شرطی برسیم که تعریف نشده باشد موجب توقف برنامه میگردد

(ب)

برای پیاده سازی چنین سیستمی به همروندی نیاز داریم تا شرط ها همه با هم محاسبه گردند تا اگر مقدار یکی true شد بتوان مقدار کل را محاسبه کرد و نیازی نباشد تا همه یا اولین شرط را فقط بررسی کنیم در این صورت اگر یکی از شروط هم تا بی نهایت رفت مشکلی پیش نمی آید

(ج)

$$\left(\text{define odd}(x) \text{ cond } ((\text{eq } x \ 0) \text{nil}) ((\text{eq } x \ 1) t) \left((> x \ 1) (\text{odd}(-x \ 2)) \right) \left((< x \ 0) (\text{odd}(+x \ 2)) \right) \right)$$

(د)

برای scor روش a مناسب می باشد زیرا به طور متوالی اجرا میگردد

برای por روش b بهتر میباشد زیرا موازی اجرا میگردد در این روش otherwise باید کامل تعریف شده باشد

سوال ۴

(a)

$F(h\ xs)$

(b)

$G: \text{maplist}$

$H: \text{car}$

(c)

$\text{Compose}(f\ h)$

سوال ۵

(a)

خیر زیرا ممکن است برنامه ما به آن دسترسی نداشته باشد ولی برنامه های دیگر به آن دسترسی داشته باشند همچنین ممکن است اشاره گر آن هنوز وجود داشته باشد ولی برنامه ما به آن دسترسی نداشته باشد در این صورت الگوریتم مک کارتی آن را زباله تشخیص نمیدهد

(b) بله زیرا ادامه هر برنامه میتواند شامل هر نوع ترتیب از دستورات car , cdr باشد اگر توسط هیچ ترتیبی از این دستورات قابل دسترسی نباشد پس با استفاده از بیس هم نمیتوان به آن دسترسی داشت

(c) این موضوع امکان پذیر نیست که به طور همزمان بفهمیم چه خانه ای از حافظه توسط هیچ یک از برنامه های در حال اجرا استفاده نمیشود بدون صرف سربار امکان پذیر نیست. برخی از متد ها البته این امر را تا حدی امکان پذیر میکند مانند کشیدن گراف استفاده از حافظه.

سوال ۶

(a) سلول c میماند و بقیه دور ریخته میشوند.

(b) شمارنده آدرس (p q) یکی صفر نمیشود و کاهش نمی یابد در نتیجه در حافظه به عنوان فضای استفاده نشده باقی میماند

سوال ۸

(a)

$$t(n) = O(1) + \max(t(n-1), t(n-2)) = O(1) + t(n-1) \rightarrow t(n) \in O(n)$$

(b)

list: (1 2 3)

cons(rplcd list 4) (cdr list) → (1 4 4)

cons (future(rplcd list 4))(cdr list) → (1 4 2 3)

(c)

مقدار *e1 OR e2* نامشخص خواهد بود در حالیکه *future(e1) OR e2* برابر *true* خواهد بود

(d)

به علت وجود همروندی ، خطا ها به صورت موازی اجرا میشوند و ترتیب خطا ها مشخص نیست و باعث میشود که در مواقع تولید چند خطا *error handler* هایی به طور تصادفی اجرا گردند. در نتیجه برنامه ها در اجرا های مختلف نتایج متفاوت خواهند داشت.