

94, 4, ٢٤ *

زبان = Semantics + Syntax

- مفهوم زبان های فرمال هستند سیال رایج ندارند.

- اولین بوری که ایده ای مانند میانگین را مطرح کرد؟ دیلبرت

- مفهوم سیمیار بوری (Dybvig) است: آن توصیف - چیز (زبان اکسل) است.

- که این سیمیارات علاوه بر مسائل اند. (چیز - توصیف تر)

- پس در اینجا زبان برای توصیف مایه ای برآمده است.

- مفهوم بوری (2002, Mitchell) : concepts in programming languages

- Formal syntax and semantics of programming languages (Stonneger)

- Types and programming languages (Pierce)

- essentials of programming languages (Fidman)

عیایت های زبان را بررسی می کنیم و تئی نویس pair دارم ماعنیاً سه عیایت تبیین

pair، متفاہل، گلریز بینال اضافه من شود.

94, 4, ٢٨ *

$$3.14 \times 5$$

π × 5 where π = 3.14 → bind, bound

π × diameter where π = 3.14 diameter → free

circumference = π × diameter where π = 3.14 → definition

(circumference (diameter)) = π × diameter where π = 3.14

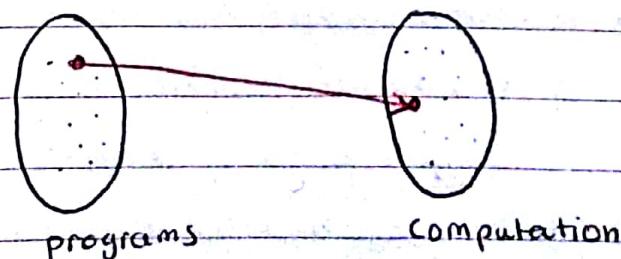
ar نوع می باشد.

ن توانیم در این مسیر این را bind می نویسیم.

ن اول program می خواهیم خود را خود می نویسیم.

Subject:
Year. Month. Date.

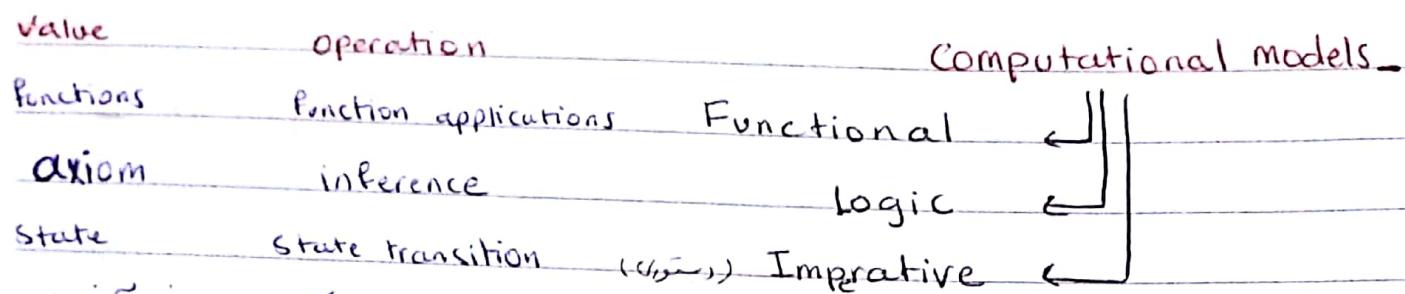
هس declaration چار واتاً مُحَاضِر احوالِ یک زانِ امن سازنے
دعا رفت:



OMID -

Subject:

Year. Month. Date.



گامی دوستی concurrent و object oriented می‌باشد که در اینجا معرفی شده است.

functional programming is based on pure functions. A function is said to be functional if it always returns the same value for the same input. Functional programming is also known as lambda calculus.

تابع f را در این operation که اینگونه عمل می‌کند.

دریم سرای همین تعقیبی است و اثر حاصل بر می‌کندار. assignment, imperative, ←
بر عقبن اثربانی و اثربن بردن در مقایسه با imperatival از این Functional reasoning statement ای از زبان imperative expression است اما زبان Imperative قوی‌تر است. این زبان ترکیبی است زبانها Imperative می‌باشد. دریم است. زبانها حالی و خود دارند و از طرفی زبانها ترسی داریم

دل مطابقی شامل inference و axiom است. برای دین در واقع logic دل اعمال inference ها و صحیح نیز است.

man (Socrates)

mortal(x) if man(x)

Mortality

OMHO

PL② many)

Subject:

Year. Month. Date.

۹۹/۵/۱*

Syntax -

کی ادھم بری امراضی کی زبان است و جزو قسمت های مشترک بازاران طبیعی است.
برای ساختن کی زبان برای این سیستم کاٹھا از روی (الگو ها) زبان طبیعی کی تحریف نهند.
"زبان زبان" نام نهاده است. عین مفهوم زبان هی آن اطول رفعه در حوزه زبان ایجاد کرد. هدف
توصیف این زبان ایجاد شد و بنابراین کی دستگاه توصیفی تابعی است.

- A grammar (I, N, P, S) consist of Four parts:

- I : A finite set of Terminal Symbols ~ alphabet

- N : A finite set of non-terminal symbols or
syntactic categories

- P : A finite set of productions or rules

(حرکت اصل تبارت بین زبان هایی که با همین چنین فرآیندهای متفاوتند است.)

- S : A start symbol

برای توصیف زبان کا صورتی ها از Metalanguage می باشد.

نوع از این محتواست:

$\langle \text{declaration} \rangle ::= \text{Var} \langle \text{variable list} \rangle : \langle \text{type} \rangle ;$

برای نشان دادن برای توصیف کدام زبان شاید بـ BNF فرضیم داشت. این محدودیت های
نمودار را درست چنین که I non-terminal فریش دهنده بـ BNF نوشت.

Vocabulary := $I + N$

چنانچه برای این سیستم کی می باشد ۳ نوع زبان برای تعریف کرد:

Type 0: (unrestricted grammars)

$\alpha ::= \beta$

- α, β are sequence of symbols in the vocabulary at

OMID = least one non-terminal occurs on the left side

Subject:

Year. Month. Date.

نحوه این سیستم می تواند گرامر نفع و راعی تراز باشند و بارز ندارد

Type 0 = Turing machine

Type 1 : (context sensitive grammars)

$$\alpha \# \beta$$

که در اینجا کمتر از دو کاراکتر نباشد

- the right side contains no fewer symbols than the left side

$$\rightarrow \alpha \langle B \rangle \beta ::= \alpha \beta \gamma, \beta \neq \epsilon$$

Type 1 = LBAs (Linear bounded Automatons)

Type 2 : (context-free grammars)

که در اینجا عبارت از α باید ترکیبی باشد

- the left side is a nonterminal.

$$\langle A \rangle ::= \alpha$$

Type 2 = PDA (push down Automatons)

Type 3: (regular grammars)

$$\langle A \rangle ::= a$$

$$\langle A \rangle ::= a \langle B \rangle$$

Type 3 = DFA, NFA

برای این سیستم می تواند شور داده شود و این سیستم می تواند اسفلوگیک برای این سیستم باشد. این سیستم می تواند این سیستم را در شور داده و این سیستم می تواند اسفلوگیک برای این سیستم باشد. این سیستم می تواند این سیستم را در شور داده و این سیستم می تواند اسفلوگیک برای این سیستم باشد.

Subject:

Year. Month. Date.

$\langle \text{Sentence} \rangle ::= \langle \text{noun phrase} \rangle \langle \text{verb phrase} \rangle$: درجہ اولیٰ

$\langle \text{noun phrase} \rangle ::= \langle \text{determiner} \rangle \langle \text{noun} \rangle$

| $\langle \text{determiner} \rangle \langle \text{noun} \rangle \langle \text{prepositional phrase} \rangle$

$\langle \text{verb phrase} \rangle ::= \langle \text{verb} \rangle | \langle \text{verb} \rangle \langle \text{noun phrase} \rangle$

| $\langle \text{verb} \rangle \langle \text{noun phrase} \rangle \langle \text{prepositional phrase} \rangle$

$\langle \text{prepositional phrase} \rangle ::= \langle \text{preposition} \rangle \langle \text{noun phrase} \rangle$

$\langle \text{noun} \rangle ::= \text{boy} | \text{girl} | \text{cat} | \text{telescope} | \text{songs} | \text{feather}$

$\langle \text{determiner} \rangle ::= \text{a} | \text{the}$

$\langle \text{verb} \rangle ::= \text{saw} | \text{touched} | \text{surprised} | \text{sang}$

$\langle \text{preposition} \rangle ::= \text{by} | \text{with}$

کوئی پرسی، کوئی نظر، کوئی سوال تھے جسکے لئے

Sentence:

a boy sang the song with a feather.

درجہ اولیٰ کا ایک نمونہ ایک

$\langle \text{sentence} \rangle$

$\langle \text{noun phrase} \rangle$

$\langle \text{Verb phrase} \rangle$

$\langle \text{determiner} \rangle$

$\langle \text{noun} \rangle$

$\langle \text{verb} \rangle$

$\langle \text{noun phrase} \rangle$

$\langle \text{prepositional phrase} \rangle$

$\langle \text{determiner} \rangle$

$\langle \text{noun} \rangle$

$\langle \text{prepositional} \rangle$

$\langle \text{noun phrase} \rangle$

a

boy

sang

the

song

with

a

feather

OMID

21

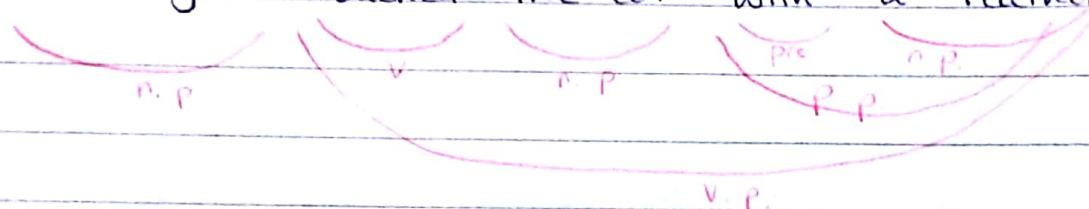
Subject:

Year. Month. Date.

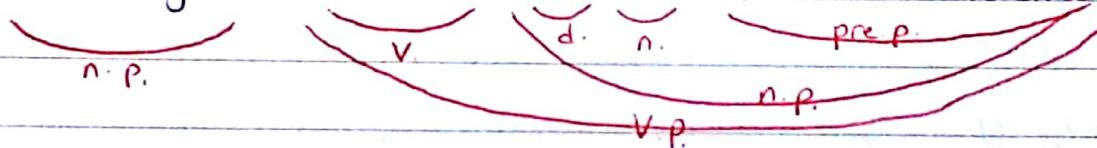
پارسی tree & derivation tree

"کلمه را بسی از رو رخت استخار داشت" جمله ای که در آن کلمه دو معنی دارد.

1. the girl touched the cat with a feather.



2. the girl touched the cat with a feather.



هر کلمه (کلم) دو معنی دارد (معنایی هم خواهد داشت):

(۱) دفترگری با سیمیلی یک سری می کرد.

(۲) دفترگری با رنگ پرداشت را نیز می کرد.

A grammar is ambiguous if some phrase in the language generated by the grammar has two (or more) distinct derivation trees.

این ایام هنرمندی در زبان ایرانی شود. بعضی از اعیام در روزنامه

نمایش اند و بعضی هایی اند مثل هنرمندی.

در هر صورت این ایام نیز رفع شود. در زمان میتوان ایامی را در زمان هنرمندی بررسی کرد.

من گاهی نیازی به کمیان نیست. میتوانم در مجموعه ای این ایام را برای نظری بررسی کنم.

ایام بین طبقه قابل عمل تواند در زمان توسعه چشم بینش است. راه هایی برای این ایام

layering یا if blocking مطرح می شود. ویژگی این ایام

Subject:

Year. Month. Date.

٩٤، ٧، ٤ *

نهن ماهم رصدتم بحضور باختت بریت دست استقان سازه معاشر انتخیب

گردید.

interpret کسی است که باید معاشر احتجاج کند. در زبان. عالمی تو از متأثرا با parse tree

باشد و می compilation باشد. تا این قدر این نوشتار در درون اش روشن است و منفعت اش عالی

آن برای اجرای

برای Control structure if و while ← Wren هر رسانی

باشد معاشر از control structure در زبان من با آن رسانی end باشد

این موارد می بروی. یعنی با اینکه این عکس‌های منفعت درست layered باشند.

می بینی هم روشی برای بحضور اینها است. اصطلاحاً او لوسی بندی گرده است.

پرسن لغات بروشده هم دارید که برای ساخت در پیاده سازی است و طاری حق ندارد از این

لغات به عنوان آن پارامترها کی غیر از آن syntax تعریف می شوند استفاده کنند. درین زبان میانی

ایامی و صورتی دارد.

استقان syntax را برویکس تقسیم کرد:

برآمده در واقع رشته ای از کارائی را داشت. این بسیار خواهد

کوچکترین واحدها را متحفل کند. به این کسی کسی کسی باشد. (words)

وی گویند. شنیدگرانشان این زبان: token

<token> := <identifier> | <numeral> | <reserved word> | <relation> |
<weak op> | <strong op> | := | () | ;

پس این بسیار در واقع آن فایل مایل رشته را می خواهد مقایسه کند.

آن بسیار token ها را جوانی کنند lexical analyzer نام دارد.

Var a : boolean ;
R.W I.D R.W ...

این lexer ، tokenizer ، scanner های ریگزیست

OMID

پرس

Phrase-Structure Syntax (P)

آن نیست رضیه دارد که نسبت کنند token هادر صور ساختاری از تراکنش در کارهای راهنمایی را بگیرند.

جتنا "روتا" `<var>` دلکل تعریف زبان ماشینی است، این نیست اسکالنگر

سین کس پرسر (sin parser)

در این زبان مطالب ایهام و صورت دار را، این صفاتی بـ دریل (فع ایهام هایی) بوده است که در تعاریف آن آمده است. متابع ایهام : if-else ، در تعاریف recursive ، ... این هیچ رستورانی نداشته است.

نامبر از این کو نوع لازم معلم است رشته‌ی λ Valid بنا شود. دلیل:

Program illegal is

Var a : boolean;

begin

$$a := 5$$

end

برای نوع اضافاتی static semantics مطروح نموده شد. آن محدودیت‌های syntax و type error را در این اضافات محدود نمود. این تغییر کی سرمهی دیگر اضافاتی می‌شود که در رامی فعال است. در این طبقه عین کنترل از محتواها برقرار است. Static از نوعی قوی متن برنامه مابین برسی است. همان‌طور آنکه context constraints معرفه‌های از این قوی متن برنامه قیود بر زبان است.

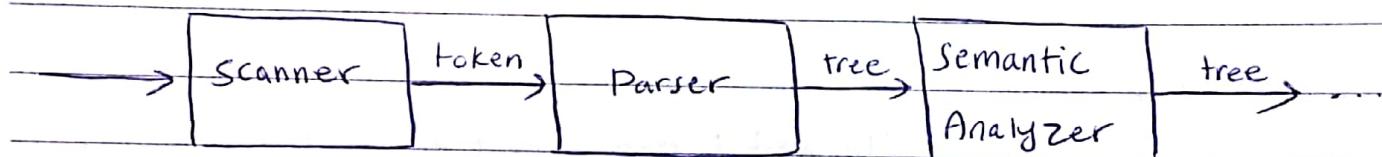
OMID — (درینیا میں علیحدہ اس جنگ از جنگوں میں دینے تبدیل ہوئے است) PL(5)

Subject:

Year. Month. Date.

- NO identifier may be declared more than once in a block.

حالت مخصوص (context sensitive) است که نتیجه از گرفتن روابط شباهت بنهایت عبارت لازم را حل کردن می‌تواند باشد. (۱) روابطی سیمی مابین non-terminal ها را با استفاده از گرسنگان غیر مخصوص نمایند. (۲) روابطی سیمی مابین terminals و non-terminals را با استفاده از گرسنگان فرعی از دست می‌آید. از روی ریخت parse tree به داشتن علی‌تلخ تفسیر دارد. این داده‌ها را می‌توان برای خودشان نیز (و) خوب است. (۳) ساخت گرسنگان فرعی خوب است.



: Semantics Error

پی ار آن سیم دلیل باز هم معلم است. اور رهایی را در رشته داشد. معرفتی اخلاقی
است که در برخایر بر مصداں برسیم که نتایج اعماق رهم و اعماق عالم get stocked شوم. مثل تفکر که
در ترسیم به متغیر علی از مقدار بیعنی ، read (از فایل هنر) ، while (true) (استحصال متفاوت دارد)
و ... بینهایی که زندگی ایجاد کنند ناچشم توان از روی Syntax و به مرور استفاده
کرد اما سائل میگویی میتوان این توانی غیرقابل هنری لارن و سیار سخت است . برای جن
این امدادات را وارد زبان انجام کنید . آن غیر از خطا هایی که با تابع داده دار قابل رفع نیست
در Static است و دیگر زمانی ریاضی غیر محدود است ما داریم غالباً کارهای کمتری . (کل عنوان
که خط کامل مسخره Syntax و Semantics کنید .

~~94, v, 11 *~~

: Semantic analyzer (سیمانتیک آنالایزر) که از روش

Attribute Grammars:

$$L = \{a^n b^n c^n \mid n \in \mathbb{Z}^+\}$$

مئون زبان رویه های دیگر:

سی تولانی است من توانم سه ماه استقل از صنعت نفیط شدم دری هشتاد هزار مقداری فراغت داشتم

OMID

Subject:

Year. Month. Date.

وزیران ملی اسلامیہ اور ایک تجزیہ کرد

<letter sequence> ::= <a sequence> <b sequence> <c sequence>

$\langle \text{a sequence} \rangle ::= a \mid \langle \text{a sequence} \rangle a$

$\langle b \text{ sequence} \rangle ::= b \mid \langle b \text{ sequence} \rangle b$

< C sequence > ::= C | < C sequence > C

اپنے تحریف رہ چکے ہیں۔ میرزا احمد شاہ کے دوسرے تحریف میں لکھا گیا ہے:

Synthesized attribute size:

گاهی این نوع از attr بی خواهی داشت و در ماقبل این λ است. نزدیکی هایش در ماقبل λ λ attr نویسید. این نوع از attr ممکن است λ attr میگشود. اگر کسی این اشیا را از دامنه ارث نماید. λ attr ممکن است λ synthesized باشد. این نوع از production rule domain attr دارای λ attr میگشند. همچنان که در ماقبل این λ است. تا آنرا در ماقبل این λ ممکن است λ این را شناخته باشیم در سرنوشتها من توانم شرط های λ است. λ predicate هم راست

$\langle \text{letter Seq.} \rangle ::= \langle \text{a seq.} \rangle \langle \text{b seq.} \rangle \langle \text{c seq.} \rangle$

condition:

$$\text{size}(\langle \text{a seq.} \rangle) = \text{size}(\langle \text{b seq.} \rangle) = \text{size}(\langle \text{c seq.} \rangle)$$

(a seq.) ::= a

Size(< a seq. >) \leftarrow 1

| < a seq >₂ a

$$\text{Size}(\langle \text{aseq} \rangle) \leftarrow \text{size}(\langle \text{aseq} \rangle_2) + 1$$

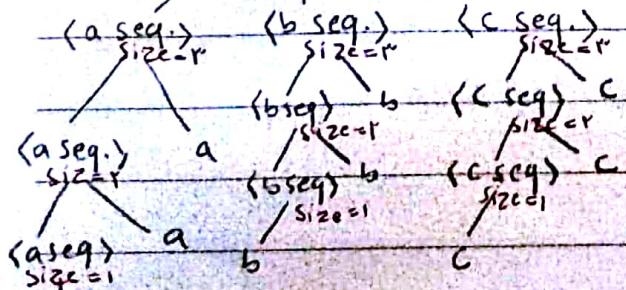
<letter sequence>

condition: true

$$\text{size}(\langle \text{baseq} \rangle) = \text{size}(\langle \text{bseq} \rangle) = \text{size}(\langle \text{cseq} \rangle)$$

طوبى هم مانند ای تغیر و کنید

الخط السادس: ما تدري بمرس auabbccccc'uuu'rrr



OMID

PLG

Subject:

Year. Month. Date.

Synthesize سازه کردن، a "the size of" operator inherited سازه از در یک دستور

$\langle b\ seq \rangle ::= b$ سازه کردن دستور $\langle a\ seq \rangle$ در inherited سازه از دستور $\langle b\ seq \rangle$

$\langle c\ seq \rangle ::= c$ سازه کردن دستور $\langle a\ seq \rangle$ در inherited سازه از دستور $\langle c\ seq \rangle$ ← پس از a class

$\langle letter\ seq \rangle ::= \langle a\ seq \rangle \langle b\ seq \rangle \langle c\ seq \rangle$

InSize($\langle b\ seq \rangle$) ← SynSize($\langle a\ seq \rangle$)

InSize($\langle c\ seq \rangle$) ← SynSize($\langle a\ seq \rangle$)

$\langle a\ seq \rangle ::= a$

SynSize($\langle a\ seq \rangle$) ← 1

| $\langle a\ seq \rangle_2 a$

SynSize($\langle a\ seq \rangle$) ← Size($\langle a\ seq \rangle_2$) + 1

$\langle b\ seq \rangle ::= b$

condition: InSize($\langle b\ seq \rangle$) = 1

| $\langle b\ seq \rangle_2 b$

InSize($\langle b\ seq \rangle_2$) ← InSize($\langle b\ seq \rangle$) - 1

پس از هر b حذف می شود

- An Attribute Grammer is a context-free grammar augmented with attributes, semantic rules, conditions.

$$G = (N, \Sigma, P, S)$$

$$p \in P$$

$$P: X_0 ::= X_1 X_2 \dots X_{np}$$

$$np > 1, X_0 \in N, X_k \in N \cup \Sigma; 1 \leq k \leq np$$

for $X \in N$

Two Finite set, $I(X), S(X)$

OMID

Subject:

Year.

Month.

Date.

Class inheritance attr. pl. check

$X \sim \dots$

new Class synthesized attr

$X \sim \dots$

$$A(X) = I(X) \cup S(X)$$

$$I(X) \cap S(X) = \emptyset$$

$$I(S) = \emptyset$$

→ semantics Functions (rules)

$$Atb \in A(X)$$

$$X_0 ::= X_1, X_2, \dots, X_{np}$$

- each synthesized attribute $Atb \in S(X_0)$ has its value defined in terms of the attributes in $A(X_1) \cup A(X_2) \cup \dots \cup A(X_{np}) \cup I(X_0)$
- each inherited attribute $Atb \in I(X_k)$ for $1 \leq k \leq np$ has its value defined in terms of the attributes in $A(X_0) \cup S(X_1) \cup S(X_2) \cup \dots \cup S(X_{np})$
- each production may also have a set of conditions on the values of the attributes in

$$A(X_0) \cup A(X_1) \cup A(X_2) \dots \cup A(X_{np})$$

رسالة ملخص از هسته دهنده برای داده های پردازشی، اتفاقی کند که بسیار ساده
باشد از دریافت داده های پردازشی، راهنمایی برای آن است. این کاملاً ممکن باشد

94, V, 14 *

<assign> ::= <var> = <expr>

: دستور

<expr> ::= <var> | <var> + <var>

: int, real, float, int, float, var

<var> ::= A | B | C

: اعمال شتابی از نظر قدرتمندی

نوع سمت چه باشد باستثنی راست گیران باشد:

int, float, var

OMID

PL

Subject: _____
Year. Month. Date. _____

→ actual-type (A synthesized attribute)

<var>, <expr>

↑
intrinsic

دروخته ایستاد

→ expected-type (An inherited attribute)

<expr>

ویرایشی var ای داریم زیرا ای دیگر

استسیم شود که نباشد

: مثلاً این قیاسی طوراً را اینجا باید نمی‌شود

<assign> ::= <var> = <expr>

expected-type(<expr>) ← actual-type(<var>)

<expr> ::= <var>

↑
actual-type(<expr>) ← actual-type(<var>)

| <var>₂ + <var>₃

↑
actual-type(<expr>) ← if (actual-type(<var>₂) = int

Λ actual-type(<var>₃) = int) then int

else real end if

condition:

actual-type(<expr>) == expected-type(<expr>)

<var> ::= A | B | C

actual-type(<var>) ← look-up(<var>, string)

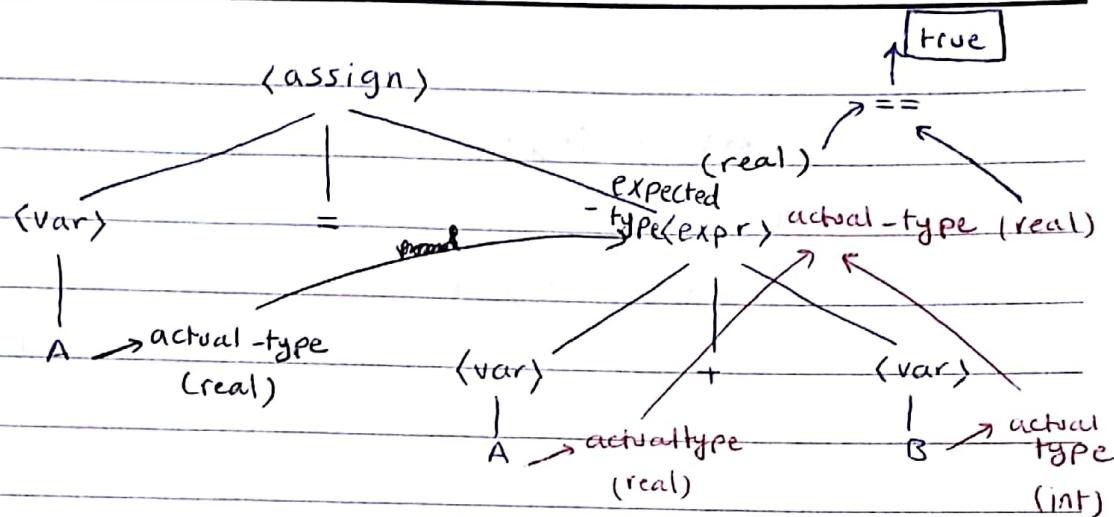
کلیزیتی را در

بررسیم parser-یوس، B → int, A → real باشیم A = A + B ای دیگر.

OMID

Subject:

Year. Month. Date.



• static semantics مفهوم اعتمادی بر داده ها در این صورتی از این اصطلاح را اصطلاح امتیازی (attribute grammar) نیز می‌نامند.

: Semantics Via Attribute Grammar

می‌دانیم که هر عدد دو رقمی را به دو رقمی دو رقمی می‌توانیم. binary numerals می‌دانیم که هر عدد دو رقمی را به دو رقمی دو رقمی می‌توانیم. چنانچه عدد 4.125 را ب ۰۱۰۰۰۱ می‌توانیم در نظام مکانیکی را داشت. همان‌طور که می‌دانیم

: int

non-terminals : <binary numeral>, <binary digit>, <bit>

terminals : ۰, ۱, .

Synthesized attribute Val (Value) از این اصطلاح است.

Local attribute Len (Length) از این اصطلاح است.

<binary numeral> ::= <binary digits>₁ . <binary digits>₂

Val(<binary numeral>) ← Val(<binary digits>₁) + Val(<binary digits>₂)
Len(<binary numeral>) ← Len(<binary digits>₁) + Len(<binary digits>₂)

<binary digits> ::= <binary digits>₂ <bit>

Val(<binary digits>₂) ← 2 · Val(<binary digits>₂) + Val(<bit>)

Len(<binary digits>₂) ← Len(<binary digits>₂) + 1

| <bit>

Val(<binary digits>) ← Val(<bit>)

Len(<binary digits>) ← 1

OMID

PL(8)

Subject:

Year. Month. Date.

$\langle \text{bit} \rangle ::= 0$

$\text{val } (\langle \text{bit} \rangle) \leftarrow 0$

| 1

$\text{val } (\langle \text{bit} \rangle) \leftarrow 1$

، pub 1101.01 ۱۳.۲۵

(binary digits)

$$\text{Val} = 13 + \frac{1}{2^2} = 13.25$$

(binary digits)

$$\text{Val} = 13$$

$$\text{len} = 4$$

(binary digits)

$$\text{Val} = 6$$

$$\text{len} = 3$$

(binary digits)

$$\text{Val} = 3$$

$$\text{len} = 2$$

(bit)

$$\text{Val} = 1$$

1

(binary digits)

$$\text{Val} = 0$$

$$\text{len} = 1$$

(bit)

$$\text{Val} = 0$$

1

$$\text{Val} = 1$$

(bit)

$$\text{Val} = 0$$

0

(bit)

$$\text{Val} = 0$$

0

(bit)

$$\text{Val} = 1$$

1

(bit)

$$\text{Val} = 1$$

1

1

Abstract Syntax

و مقابل concrete Syntax است که در "مت فیزیکی" برای این ساخته می‌گذرد. با کمی اعماق

می‌توان گفت تاکنون حسنه بود است. (زیرا در درستگاه‌ها هم می‌توان صحت نظریم.)

abstact می‌باشد که در معنای این است توجه کنید. چنان درست است این

Abstract هر دو زدنی به یک درست است شود. برای رسیدن؟ concrete

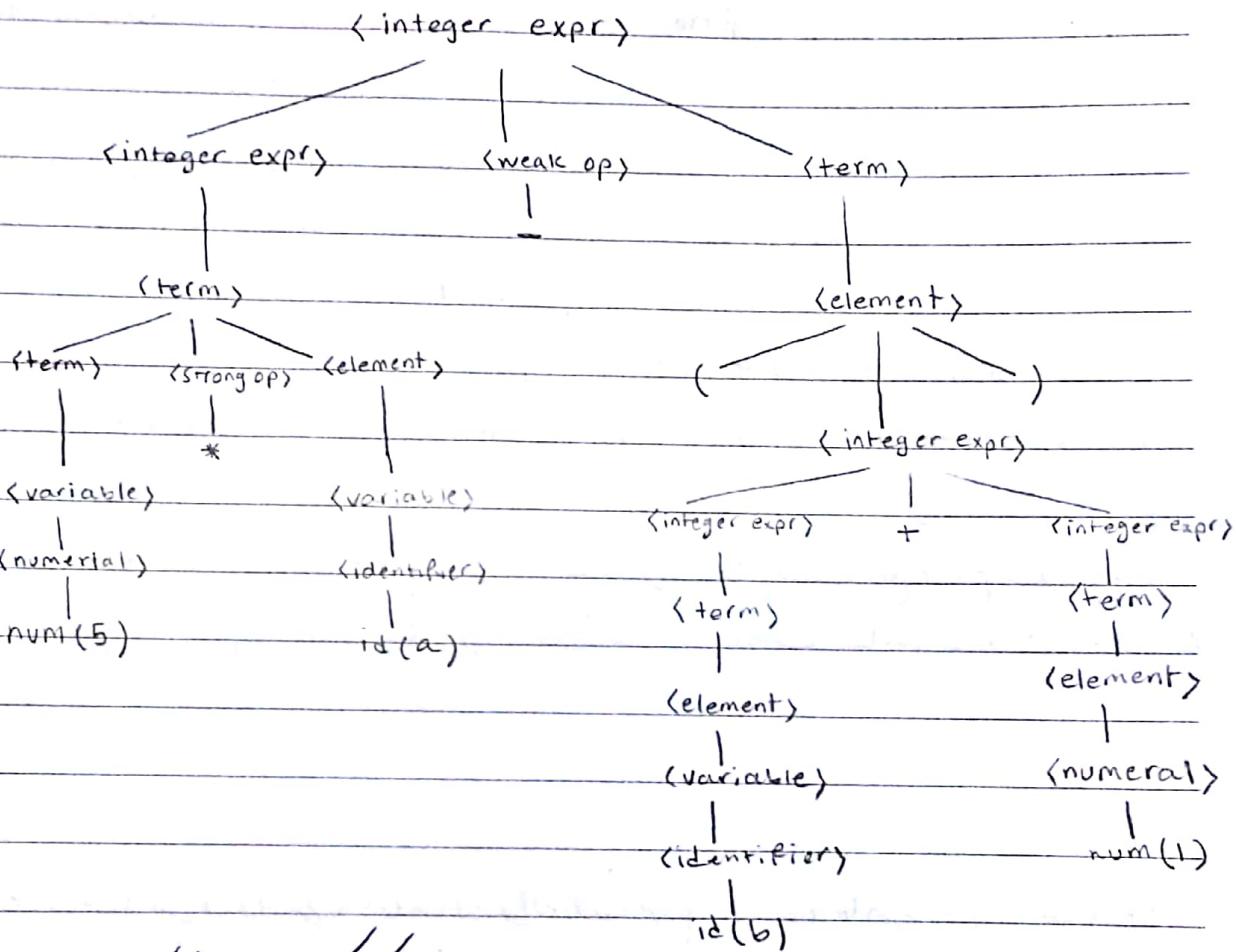
برای رسیدن؟ concrete

Subject:

Year. Month. Date.

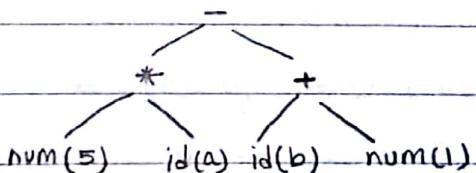
$5 * a - (b + 1)$

: Wren دستگاری عبارت را در درخت نمایش دهد



ابویز ریخت قسمت های اضافی زیرا در می توان ریخت کوچکتر را به شکل زیر در راهات

بکار برد:



آنچه بترفیم رید، این است که مقادیر آغازین مکان است تا ریشم باز بروم. تاکنین این ریخت بسیار خواهد بود.

Subject: _____
Year. Month. Date.

44, V, 11 *

بر. هنر ریتمیک انتقال اینها ممکن است قبل از نتیجه خروجی به شکل زیر است:

minus (times(5 , id(a)) , plus(id(b) , 1))

درین رخدت Abstract syntax tree می باشد که عبارت های رشته و عبارت های داده را در گزینه های انتخابی می نماید. برای اینکه یک داده را در گزینه هایی که در درخت استخراج شده باشد، باید این رخدت را بازگردانید. این رخدت داشته باشد که در مورد این داده ها باشند: ۱) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۲) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۳) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۴) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۵) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۶) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۷) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۸) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۹) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند. ۱۰) ترتیبی (order) که در آن داده هایی که در درخت قرار دارند، مطابق با ترتیبی است که در متن اصلی آنها داشتند.

Semantics

Operational

Denotational

Axiomatic

ترصیح و تعریف این سه کار رخته است. در کام هم بخش های زیادی شامل می شوند.

درین (برن) درین انتخاب و Abstract machine کی Operational است. این نوٹ کی دلیل

تغییر حالت، State transition، تغییر حالت انتقالی

• Transform terms \rightarrow transform terms

OMID

Subject:

Year. Month. Date.

سازه های این روش که در کنکور مانند abstract syntax tree ها می باشند را در اینجا معرفی خواهیم کرد. این روش دو نوع است: denotational (دلالتی) و axiomatic (اصطلاحی). در دلالتی معمولی مفهوم معرفت مبتنی بر تعریف است. در اصطلاحی مفهوم معرفت مبتنی بر اصول است. این دو روش می توانند برابر باشند. مثلاً در اینجا در دلالتی معرفت اصولی در آورده ایم (Axiomatic) و در اصطلاحی معرفت اصولی در آورده ایم (Denotational).

$t ::= \text{true}$

false

$\text{if } t \text{ then } t \text{ else } t$

0

$\text{succ } t$

$\text{pred } t$

$\text{iszero } t$

مثال: (گزینه های معرفتی در اینجا معرفت دلالتی است.)

در اینجا از نظر تئوری آن معرفت دلالتی است.

$\text{value} ::= \text{true}$

false

$n \vee$

$n \vee ::= a$

$\text{succ } n \vee$

برای معرفت دلالتی (One-step evaluations relation) "on terms" می باشد. این روابط را می توان از این شکل درست نمود: $t \rightarrow t' \vdash (t, t') \in \rightarrow$. این روابط را می توان از این شکل درست نمود: $\frac{\text{axiom}}{\text{succ}(a) \in \text{N}}$. این روابط را می توان از این شکل درست نمود: $\frac{\text{succ } a \in \text{N}}{\text{succ(succ}(a)) \in \text{N}}$. این روابط را می توان از این شکل درست نمود: $\frac{\text{succ } a \in \text{N}}{\text{succ(succ}(a)) \in \text{N}}$.

برای معرفت دلالتی در اینجا معرفت دلالتی می باشد: $\frac{a_1 \in \text{N}, a_2 \in \text{N}}{\text{node}(a_1, a_2) \in \text{N}}$. این روابط را می توان از این شکل درست نمود: $\frac{}{\text{empty} \in \text{N}}$.

OMID

PL

Subject:

Year. Month. Date.

در اینجا با دیگر طرزهای داشت صنعتی خود می‌کارم و توان برای مفاهیمی که مانند عالی و پایین را در آن معرفی کنم که شال و مفهومی را باید نزدیک کرد.

برای این امر فناوری را باید تعریف کنیم:

Axiom:

if true then t_2 else $t_3 \rightarrow t_2 \Rightarrow$ if true then t_2 else $t_3 \rightarrow t_2$ مطابق با زیر کل طبق مطابق با زیر کل

Axiom:

if False then t_2 else $t_3 \rightarrow t_3$

proper rule:

$t_1 \rightarrow t'_1$

if t_1 , then t_2 else t_3

\rightarrow if t'_1 , then t_2 else t_3

$t_1 \rightarrow t'_1$

$\text{succ } t_1 \rightarrow \text{succ } t'_1$

$\text{pred } 0 \rightarrow 0$

$\text{pred}(\text{succ } n_{V_i}) \rightarrow n_{V_i}$

$t_1 \rightarrow t'_1$

$\text{pred } t_1 \rightarrow \text{pred } t'_1$

$\text{iszero } \emptyset \rightarrow \text{true}$

$\text{iszero } (\text{succ } n_{V_i}) \rightarrow \text{false}$

$t_1 \rightarrow t'_1$

$\text{iszero } t_1 \rightarrow \text{iszero } t'_1$

پس مدرسین از این اخوندی های بات را به رست آور.

if true then $\text{succ}(\text{pred}(\text{succ}(\emptyset)))$ else \emptyset

OMID $\rightarrow \text{succ}(\text{pred}(\text{succ}(\emptyset)))$
 $\rightarrow \text{succ}(\emptyset)$

Subject:
Year. Month. Date.

ترمھایی کو درست صب قرار دادن کرنا \rightarrow Normal Form می گویند مثل $Succ(\emptyset)$ نام ترکیب عام Value ها formal form است. اگر فقط سه rule اول $true$ و $false$ و $if-then-else$ کو از پرداخته باشیم. و گزینه شال نمی این دیرگی را بذارد. مثل $Pred(x,y)$. اگر زان - خوبی مارس شد، شا - انتظار می رود که این دیرگی برقرار باشد. اشطال هم درستیف Rule هایی باشد. این زان safe نیست. یعنی در زمان اصرار مغلوب است از حرکت مانعه در $get\ stock$ کند. یعنی از نظر syntax اشطالی نادرستی را اجرا به اشطال می خورد.

94, v, 114 *

- * بخش عمده ای از تابع قابل از افراحت است: interpret
- (لینک از روی) حاصل AST مقدار را به دست بیاریم: interprete
- سبل حاصل AST بینیک را به راهی سلحنج پاسیں ران اجرای تائین: compile
- * زیر نویسنده style را نسبل به معنی semantics داریم
- عقیقیت درون دلیل تأثیری inductive است (نمایی از آنست) و گریدر را axiom هاد و rule ها از این
- اصطلاحات باقی می‌مانند rule scheme، rule و inference، term و free variable، unification و substitution.
- شرط درستی این عبارات می‌تران: رشته ها را evalute کرد
- شرط درستی این عبارات می‌تران: Step by Step evaluation
- شرط درستی این عبارات می‌تران: Term های ران کامپیوٹ می‌باشند

گاهی مقداری را با $t \rightarrow t' \rightarrow t'' \rightarrow \dots \rightarrow V$ نمایش داده که V مقداری است که مقداری t را در t' و t'' و ... تغییر نماید. این مقداری را **Value** می‌نامند. این مقداری را می‌توان $t \rightarrow t' \rightarrow \dots \rightarrow t^{(n)}$ نوشت که $t^{(n)}$ مقداری است که مقداری t را در t' و \dots تغییر نماید.

ابست ترجمی (درین) است در جایگزینی static semantics، $b \leftarrow$ یعنی stick الگوریتم است.

OMID -

PL II

Subject: _____
Year. Month. Date. _____

زیادتی Safe یعنی است رهیم برای این برایش زد و بیل از اینها بیش نیستند
ضادها را داشته اند exception handler * .
برای Safe شود، باید غیر از
* غیر تواند یک برایش نداشته باشد که اینها را در پرسش

Theorems - If $t \rightarrow t'$ and $t \rightarrow t''$, then $t' = t''$.

(Determinacy of one-step evaluation)

ناتایج متعارف از دو روش ارزیابی کردن این اثبات را دارند
(ایجاد این اثبات را داشت) (ایجاد این اثبات را داشت) (ایجاد این اثبات را داشت)
ایجاد این اثبات را داشت (ایجاد این اثبات را داشت) (ایجاد این اثبات را داشت) (ایجاد این اثبات را داشت)

Definition - The multistep evaluation relation " $\xrightarrow{*}$ " is the reflexive and transitive closure of one-step evaluation.

ایجاد $\xrightarrow{*}$ کوچکترین رابطه ای از \rightarrow

Smallest $\xrightarrow{*}$ that

1) if $t \rightarrow t'$, then $t \xrightarrow{*} t'$
2) $t \xrightarrow{*} t$
3) if $t \xrightarrow{*} t'$ and $t' \xrightarrow{*} t''$, then $t \xrightarrow{*} t''$

(Corollary)

If $t \xrightarrow{*} t'$ and $t \xrightarrow{*} t''$ where t' and t'' are both normal forms then $t' = t''$

Theorem - For every term t , there is some normal form t' such that

$t \xrightarrow{*} t'$

OMID

Subject: _____
Year. Month. Date.

(λ-calculus) Lambda Calculus

هیدریت پسیون بینی یک ماشین Abstract بینی سیمی را پاسخ را مطرح کرد. درین سیمایلات ترنسیگ و ترنسیگ یک ماشین صالت طراحی کرد. برداشته هم بودند. این را محاسبه رامعنای دارند. ترنسیگ یک ماشین صالت طراحی کرد. برداشته هم بودند. این را محاسبه رامعنای دارند. ترنسیگ یک زبان استفاده دارند محاسبات را با آن احکام نوار و استیت ها کاریک کرد. جرج در اصل یک زبان استفاده دارند محاسبات را با آن احکام دارد. اثبات منشود محاسبات جرج با عالیین ترنسیگ مقابل است. درسته هم تز جرج ترنسیگ مطرح شد که محاسبات در مقابل این درس تجسس و همیشه ای ای ای توان کاریک کرد. ترنسیگ calculus آن است که زبانی است که سایر اراده خواشان فی لجیاند و می توان تمام سایر زبانها را با آن تبدیل کرد. نیز در معنا شناسی زبانها برین زبان را تاریخی شود. به عنی دلیل برای زبان اصلی calculus (core calculus) گویند. یک عرضی دیگر آن بین زبان Functional گویند. یک عرضی دیگر آن بین زبان concurrent گویند. به عنی این زیرسی concurrent است. به عنی object calculus بین زبان های متفاوت است.

$t ::= x \mid \lambda x. t \mid tt$ abstraction. application.

94, V, Pa *

این زبان سیارقوی است و می‌توان همین‌جهان را با آن ترسیم کرد اما اعملاً برای برآمدها
مشتمله بکار گرفته نمی‌شود.

* قانون : ترم های این زبان از ص بیشتر باشند: $st \rightarrow (st)^n$

$$S \vdash U \vee V \rightarrow ((S \vdash U) \wedge (S \vdash V))$$

*قازنی: امور production ها نیز از راست بُلست بُزیر باشند.

$\lambda x. \lambda y. x y x$

$$\lambda x. (\lambda y. (xy)x)$$

Definition: An occurrence of the variable x is said to be bound when it occurs in the body t of an abstraction.

~~λn.t~~

binder contains
OMID PL II

Subject:

Year. Month. Date.

Combinator

bound ≠ Free

Open Combinator Free $\xrightarrow{\sim}$, Closed $\xleftarrow{\sim}$ Combinator Free $\xrightarrow{\sim}$ مترجلاً

Ex: $F(x) = x^2 + 5 \rightarrow x$ is bounded no free var.

$F(x) = x^2 + y \rightarrow x$ is bounded. y is free.
 $(\lambda x. x)y$

Ex) $(\lambda x. x)x$

↓
bounded

↓
Free

$\lambda x. x x \xrightarrow{R.D.} \lambda x. (x x)$

↓
bounded

$\lambda x. x \rightarrow$ identity function (untyped calculus)

النهاية مترجلاً مجهولة

Ex) if $R = \{x \mid x \notin x\}$ then $\lambda R R \in R \rightarrow R \notin R$!

(if $R \notin R \rightarrow R \in R$!)

• $\lambda x. (\lambda y. (\lambda x. x y)) y x$ → opened term
↓ b ↓ f ↓ bound

مقدمة بحثية مترجلاً مجهولة

: Operational Semantics

Operational Semantics

$(\lambda x. t) t' \rightarrow [x \mapsto t'] t \sim [t/x] t$

Beta Reduction

The term obtained by replacing all free occurrences of x in t by t'

Ex) $(\lambda x. (\lambda x. x)x)y \rightarrow (\lambda x. x)y$

OMID

Subject:

Year. Month. Date.

$$(x)(\lambda x. x(\lambda x. x)) \rightarrow (\lambda x. x)(\lambda x. x)$$

$$((\lambda x. x(\lambda x. x)) u) r \rightarrow u(\lambda x. x) r$$

این از abstraction (که تعریف را در خارج از داشت) : (reducible expression) redex
 abstraction که قابل تغییر داشت است. یعنی فقط $(\lambda x. t) t'$
 در $\lambda x. t$ application بجزی t اعمال کرد.

: Evaluation Strategy

که در اینجا redex که تغییر داشت درست

تک β -Reduction

که redex مخصوص جمله تغییر داشت

$$(x)(\underline{\text{id}}(\lambda x. x)(\underline{\text{id}}(\lambda x. x)(\lambda z. (\underline{\text{id}}(\lambda x. x))z))) = \text{id}(\underline{\text{id}}(\lambda z. \underline{\text{id}}z))$$

$$\rightarrow \text{id}(\lambda z. \underline{\text{id}}z) \xrightarrow{2} \text{id}(\lambda z. z) \xrightarrow{3} \lambda z. z \not\rightarrow$$

ادلهای معتبرین درین ترتیب : normal-order ①

$$\text{id}(\underline{\text{id}}(\lambda z. \underline{\text{id}}z)) \rightarrow \text{id}(\lambda z. \underline{\text{id}}z) \rightarrow \lambda z. \underline{\text{id}}z$$

$$\rightarrow \lambda z. z \not\rightarrow$$

② + که abstraction دارد : call-by-name ②

$$\text{id}(\underline{\text{id}}(\lambda z. \underline{\text{id}}z)) \rightarrow \text{id}(\lambda z. \underline{\text{id}}z) \rightarrow \lambda z. \underline{\text{id}}z \not\rightarrow$$

OMID

PL ۱۴

Subject:

Year. Month. Date.

abstraction

رسانید redex : سیب ترین و بیرنگ ترین اول ماته به شرطی است : call-by-value (E)

value از ترکیب سایع (غير) صورتی :

$$id(id(\lambda z.idz)) \rightarrow id(\lambda z.idz) \rightarrow \lambda z.idz \rightarrow$$

حکم صورتی دستیاری :

است که در آنها مفهوم رامحابب از نوع call-by-name است

از lazy است . این گونه است call-by-value (I). اینها

جی بین (ترمیمی) می‌باشد . تفاوت اینها با نیازمندی دنبال شود

لیکن lazy صورتی و بخوبی می‌باشد .

94, v, ۳ *

• Civil E. strategy + Semantic + Syntax CS607

• Programming in the λ -calculus

حکم داری λ -calculus است . محدودیت‌هایی برای اینها نداشت از این

است که اینها turing-complete هستند . اینها می‌توانند هر مسئله ای را حل کرد . عام

لیکن تابعی است که نیمه از مجموعه مایندری (non-reducible) است . اینها می‌توانند محدودیتی داشته باشند که ممکن است می‌توانند β -reduction

لیکن توابعی است که نیمه از مجموعه مایندری (non-reducible) است . اینها می‌توانند $\lambda(x,y).t \rightarrow \lambda x.\lambda y.t$ بین تغییراتی اعمال کنند .

Currying

حکم داری $f(x,y)$ است که نیمه از مجموعه تابعی از لامبردا را دارد .

$(\lambda x.\lambda y.t) u v$

$\hookrightarrow ([x \mapsto u](\lambda y.t)) v \rightarrow [y \mapsto v][x \mapsto u](\dots, t)$

• church boolean

اول ساختن عبارات boolean دارد . (آپنی در واقعیت ذهنی شوربی) صفحه

در کتابخانه غایش بخوبی دارد .) انتظار کار از boolean بعنوان

OMID

$\text{neg} \vee \rightarrow \text{neg} = \lambda b. b \text{ false true}$

Subject
Date

غایلی ریکارڈ رائے مفہوم مربوطہ ضریش دستہ ہے۔ بیداری سے، ایک ایسا
implement عبارت ایجاد کرے کہ خودی ساخت مربوطہ بے بیان، اسی مراحل کا
 $\text{true} \rightarrow \lambda t. \lambda f. t$
 $\text{false} \rightarrow \lambda t. \lambda f. f$

مثال) $\text{true } u \vdash \rightarrow (\lambda t. \lambda f. t) u \vdash v \rightarrow (\lambda f. u) v \rightarrow v$

$\text{false } u \vdash v \rightarrow (\lambda t. \lambda f. f) u \vdash v \rightarrow (\lambda t. u) v \rightarrow (\lambda t. v) = v$

$\text{test} = \lambda L. \lambda m. \lambda n. L m n . \lambda l. l n m$

(مثال) $\text{or test true } v \vdash w$

$\rightarrow (\lambda L. \lambda m. \lambda n. L m n) \text{ true } v \vdash w$

$\rightarrow (\lambda m. \lambda n. \text{ true } m n) v \vdash w$

$\rightarrow (\lambda n. \text{ true } v n) w \rightarrow \dots \text{ true } v w \xrightarrow{*} v$

(مثال) $\text{or and } v w \rightarrow \text{and} = \lambda b. \lambda c. b c \text{ false}$

$\rightarrow (\lambda b. \lambda c. b c \text{ false}) \text{ true false}$

$\rightarrow (\lambda c. \text{ true } c \text{ false}) \text{ false}$

$\rightarrow \text{ true false false} \xrightarrow{*} \text{ false}$

(مثال) $\text{or OR } v w \rightarrow \text{OR} = \lambda b. \lambda c. b \text{ true } c$

: Church numeral 0₂

$(\lambda s. \lambda z. z)$ یعنی 0₂ کا church number

$c_0 = (\lambda s. \lambda z. z) \rightarrow 0$

$c_1 = (\lambda s. \lambda z. sz) \rightarrow 1$

$c_2 = (\lambda s. \lambda z. s(sz)) \rightarrow 2$

:

: 3₂ کا church number Successor نوں

$\text{scc} = \lambda n. \lambda s. \lambda z. s(nsz)$

$\rightarrow \text{scc } m = \lambda s. \lambda z. s(msz) = (\lambda s. \lambda z. s(s(\dots)s)) sz$

PAPCO

PL(15) $\Rightarrow \lambda s. \lambda z. s(s(\dots(sz)\dots)) = c_{m+1}$

$m+n$
diagram) add m n

$$\text{add} = \lambda m. \lambda n. m \text{ succ } n \rightarrow \text{inapply } n \rightsquigarrow \text{succ } m$$

$$\Leftarrow \text{add} = \lambda m. \lambda n. \lambda s. \lambda z. m s (n s z)$$

$m \times n$ (مثال) times = $\lambda m. \lambda n. m (\text{add } n) \circ_0 \rightarrow \text{combinator } \text{add } n \rightsquigarrow m$
 $(\text{add } n \dots (\text{add } n (\text{add } n \circ_0)) \dots) \leftarrow \text{inapply}$

m^n (مثال) power = $\lambda m. \lambda n. n (\text{times } m) \circ_1 \rightarrow \text{combinator } \text{times } m$
 $\cdot \text{inapply } \circ_1 \circ_0 \dots \text{times } m$

pair (مثال) pair = $\lambda F. \lambda S. \lambda b. b F S$

constructor First = $\lambda p. p \text{ true} \rightarrow$ کارکردن برگردانن

observer Second = $\lambda p. p \text{ false} \rightarrow$ کارکردن برگردانن

$$\text{thus } \text{First}(\text{pair } v w) = (\lambda b. b v w) \text{ true} = \text{true} v w = v$$

94 / A/P *

→ داده از این توان به اشغال مختلف به عاشیل گذاشت. چنین مخفدم و درگیری

حایی است که رولین عاشیل داده ها بیهان است. مفهوم interface را توصیف من کنم و کاری به پایه سازی

آن نمایم، اینسته پیاده سازی باشد آنکه interface را با ساختگو باشد.

→ زبان گوئی از باشد که نتیجه به عاشیل داده ها را ترسیس نماید.

→ زبان گوئی از باشد که بر زبان نویس جوابش را داده دوگری عاشیل باشد.

مثالی به numeral abstraction برای n با توابع چرخ پیاده سازی شده اند و پیری

تابع روی آنها تعریف می شود.

در اک برآمده های ریاضیات imperative هستند ولی سایر اصطلاحها را قیاس می سازند

زعنون حل کننده مسائل دینامیکی های که برآشی را فتن و سادگی را می آورد فنده از

اتقای از کنند. برای reasoning بیان صفت است.

PAPCO

Subject:

Year. Month. Date.

Recursion

نیابت برگشت از هم زین قابل تهیی نیز زبان است. اگر دیک زبان این افعال وجود نداشته باشد آن زبان نویس تواند turing complete باشد و مام سالم را حل کند. برگشت بزرگ برگشت بزرگ است. حکایت بازی های این برگشت بزرگ بروجت ترکیبی است.

را در λ-calculus بحث کنیم؟

اگر $F: A \rightarrow A$ اگر تابع \leftarrow Fixed-point.

x_0 is fixed point of F if $F(x_0) = x_0$.

نمایه شوند $f(x) = x^2$ می شود.

تابع فاکتوریل (Factorial):

$$f: N \rightarrow N$$

$$f(0) = 1$$

$$f(n+1) = (n+1) \times f(n)$$

$$f(n) = \begin{cases} 1 & \text{if } n=0 \\ n \times f(n') & \text{if } n=n'+1 \end{cases}$$

$$f: n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n \times f(n') & \text{if } n=n'+1 \end{cases}$$

Functional F \leftarrow

Functional F : functions \rightarrow functions

$$N \rightarrow N$$

$$N \rightarrow N$$

برخوبی خروجی از این فرمت است.

$$F(f) = f'$$

where

$$f': n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n \times f(n') & \text{if } n=n'+1 \end{cases}$$

" f' " : $F(f)$, where $F(n) = n^2$

$$\hookrightarrow F(f) = f' : n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n \times n^2 & \text{if } n=n'+1 \end{cases}$$

Subject:

Year. Month. Date.

دراسته داریم $f'(0) = 1, f'(1) = 0, f'(k) = k$, ...

" \tilde{f} : $F(g)$, where $g(n) = n - 1$

$$F(g) = g': n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n(n-2) & \text{if } n \neq 0 \end{cases}$$

دراسته \tilde{f} $\rightarrow g'(0) = 1, g'(1) = -1, \dots$

" \tilde{f} : خواهد شد $F(F) = F$ \leftarrow درباره $f' \downarrow F$ اگر

$$F(F) = F'$$

$$f: n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n \times F(n') & \text{if } n=n'+1 \end{cases} \rightarrow \text{دلتانی}$$

$$F(F) = f': n \rightarrow \begin{cases} 1 & \text{if } n=0 \\ n \times F(n') & \text{if } n=n'+1 \end{cases} \rightarrow \boxed{F(F) = F}$$

The factorial function is Fixed point of F .

" \tilde{f} : $N \rightarrow N$

$$f(0) = 0$$

$$f(1) = 1$$

$$f(n) = f(n-1) + f(n-2)$$

the Fibonacci function is the fixed point of the functional:

$$F(g): n \rightarrow \begin{cases} 0 & \text{if } n=0 \\ 1 & \text{if } n=1 \\ g(n-1) + g(n-2) & \text{ow} \end{cases}$$

دلتانی داریم \tilde{f} \rightarrow functional اگر دلخواه باشد

اگر سیران نهایت دلتانی تابع داشته باشد دلتانی است (recursion λ -calculus)

OMID

Subject:

Year. Month. Date.

fix : Fixpoint combinator ← معنى - معنى

fix F = the Fixed-point of F (معنی فیکس)

→ fix $\lambda F. \lambda n. \text{if } n=0 \text{ then } I \text{ else } n * (F^{n-1})$ apply

→ fix $\lambda g. \lambda n. \text{if } n=0 \text{ then } 0 \text{ else if } n=1 \text{ then } 1$
else add (g^{n-1}) (g^{n-2})

Fix \rightarrow مفهوم ثابت

* * $Y = \lambda F. (\lambda x. F(x x))(\lambda x. F(x x))$

"this → Omega = $(\lambda x. x x)(\lambda x. x x)$ } Diverge
 $\rightarrow (\lambda x. x x)(\lambda x. x x) \rightarrow \dots$ obvious
↳ not normal form

$YF = (\lambda x. F(x x))(\lambda x. F(x x))$

= $F((\lambda x. F(x x))(\lambda x. F(x x)))$ } Y is
call-by-name
fixed-point
combinator

↳ YF is the fixed point of F

↳ YF is the fixed point of F

$F = \lambda F. \lambda n. \text{if } n=0 \text{ then } I \text{ else } n * (F^{n-1})$. دلالة (دلالة)

$(YF)^2 = F(YF)^2$

= $\underbrace{(\lambda F. \lambda n. \text{if } n=0 \text{ then } I \text{ else } n * (F^{n-1}))}_{+} (YF)^2$

OMID PL(14) = $(\lambda n. \text{if } n=0 \text{ then } I \text{ else } n * ((YF)^{n-1}))^2$

stucked محاشرت است و میتوانیم λ -calculus را ترجمه کنیم *

Subject:

Year. Month. Date.

شوندی *

$$= \text{if } 2 = 0 \text{ then } 1 \text{ else } 2 * ((YF) 1)$$

$$= 2 * ((YF) 1)$$

= ...

94, A, V *

Denotational Semantics *

to mean S. th. is denote . من برای Operational semantics تأثیر میگیریم

"this program P denotes the function $f(n) = 1 + 2 + \dots + n$

Object language \rightarrow metalanguage "عنوانی" میگیریم

با این معنی مطابق زبان را در مطالعه

زنگولی مطالعه زبان را در مطالعه

مثل زبان طبیعی :

یک دو تری لم به این معنی دارد. بین های هر عبارت compositionality

برگزایی برای ساختن عبارات کوچک تر است که در آن متضمن شود

$$\text{thus } B = B'$$

$$P = P'$$

$$Q = Q'$$

$$\text{if } B \text{ then } P \text{ else } Q = \text{if } B' \text{ then } P' \text{ else } Q'$$

→ "this: Binary numbers :

$$e ::= n \mid e + e \mid e - e$$

$$n ::= b \mid nb$$

$$b ::= 0 \mid 1$$

: شوری pars tree (or semantics)

$E(\text{parse tree}) = \text{Arithmatic expression}$

$[e]$

OMID

$\rightarrow (\text{parse tree for } e)$

Subject:
Year. Month. Date.

$$[e_1] \xrightarrow{+} [e_2] \leftarrow [e_1 + e_2] \quad \text{U. "in"}$$

$$E\{[0]\} = 0$$

سے دارم

$$E[1] = 1$$

$$E[nb] = E[n] * 2 + E[b]$$

$$E[e_1 + e_2] = E[e_1] + E[e_2]$$

\rightarrow **Def:** $e ::= v \mid n \mid e + e \mid e - e$ (Arithmetic Expressions with Variables)

Page 12 of 12

d ::= 0 | 1 | 2 | ... | 9

$V := x | y | z | \dots$

米

States ($s : \text{variables} \rightarrow \text{values}$)

جزء مفهومی $\rightarrow E$ (parse tree , state)

مقدار x را state معرفی کن $\rightarrow E[x] = s(x)$ دس میتوانیم داشت .

$$E[0](s) = 0$$

$$E[\bar{e}_1](s) = 1$$

$$E[9](s) = 9$$

$$E[\lceil n_d \rceil](s) =$$

Engineering

$$E[\text{nd}](s) = E[n](s) * 10 + E[d](s)$$

$$E[e_1 \pm e_2](s) = E[e_1](s) \pm E[e_2](s)$$

اسے پہلے کام مانند اسی مدل تعریف اسے partial function کہا جاتا ہے

Subject:

Year. Month. Date.

→ This: while programs:

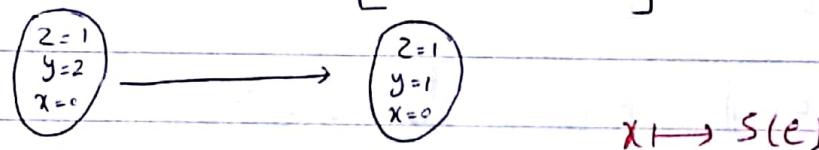
$P ::= x := e \mid P; P \mid \text{if } e \text{ then } p \text{ else } P \mid$

while e do P

$C[[P]] : \text{states} \rightarrow \text{states}$

هر بارہ ایک تابع از محالت ہے
اسے سوچو

وہی $C[[x := 0; y := 1]](S) = S[x \mapsto 0, y \mapsto 1]$



Semantics $\rightarrow C[[x := e]](S) = S[x \mapsto E[e](S)]$

$C[[x := y + 2]](S) = S[x \mapsto E[y](S) + 2]$

$$= S[x \mapsto E[y](S) + 2]$$

$C[[P_1 ; P_2]](S) = C[P_2](C[P_1](S))$

$$= \underbrace{(C[P_2] \circ C[P_1])}_{\text{fog} = f(g(x))}(S)$$

$C[[\text{if } e \text{ then } p_1 \text{ else } p_2]](S) =$

if $E[e](S)$ then $C[P_1](S)$ else $C[P_2](S)$

$C[[\text{while } e \text{ do } p]](S) =$

if not $E[e](S)$ then S

else $C[[\text{while } e \text{ do } p]](C[P](S))$

Chained

P Chained

OMID

Subject: _____
Year. Month. Date. _____

99, 1, 9 *

Q) $P = x := 2 ; \text{while } x > 0 \text{ do } x := x - 1 ?$

$C [[P]](S)$

$= C [[x := 2 ; \text{while } x > 0 \text{ do } x := x - 1]](S)$

$= C [[\text{while } x > 0 \text{ do } x := x - 1]](C [[x := 2]](S))$

$= C [[\text{while } x > 0 \text{ do } x := x - 1]](S[x \mapsto 2])$

$= \text{if } \neg S[x \mapsto 2](x_0) \text{ then } S[x \mapsto 2] \text{ else}$

$C [[\text{while } x > 0 \text{ do } x := x - 1]](C [[x := x - 1]](S[x \mapsto 2]))$

$= C [[\text{while } x > 0 \text{ do } x := x - 1]](\underbrace{(S[x \mapsto 2])[x \mapsto 1]}_{S[x \mapsto 1]})$

$= C [[\text{while } x > 0 \text{ do } x := x - 1]](S[x \mapsto 0]) = S[x \mapsto 0]$

بعض الـ semantics غير الـ standard

و (Abstract Interpretation ...)

Nonstandard
semantics *

برمجة بستدر لـ program analysis

Subject: _____
Year. Month. Date. _____

بازم تر تفسیر دستورات هر تعبیر قبل از استفاده، مقداردهن شد باشد. جمله این بحث است که کلیل و ناممکن (undecidable) است. برای حل باشتر اطلاع را ایجاد کرد. الگوریتم مقاطعه کارانه برای این سیستم تأثیرگذشت درست است، درست است اند اما اگر تغییر نموده باشد (False negative) اعتماد کرد.

$$\text{states} = \{\text{error}\} \cup \{s: \text{variable} \rightarrow \{\text{init}, \text{uninit}\}\}$$

$$E[e](s) = \begin{cases} \text{err} & \text{if } e \text{ contains any variable } y \text{ with } s(y) = \text{uninit} \\ \text{OK} & \text{otherwise} \end{cases}$$

الجبری: $E[e](s)$ \leftarrow err اگر e شامل مت�یری باشد که در s نداشته باشد.

$$C[n := e](s) = \text{if } E[e](s) = \text{OK} \text{ then } s[x \mapsto \text{init}] \text{ else error}$$

$$C[P_1 ; P_2](s) = \text{if } C[P_1](s) = \text{error} \text{ then error else } C[P_2](C[P_1](s))$$

$$(S_1 \oplus S_2)(v) = \begin{cases} \text{init} & \text{if } S_1(v) = S_2(v) = \text{init} \\ \text{uninit} & \text{o.w.} \end{cases}$$

$$C[\text{if } e \text{ then } P_1 \text{ else } P_2](s)$$

$$= \text{if } E[e](s) = \text{err} \text{ or } C[P_1](s) = \text{error} \text{ or } C[P_2](s) = \text{error} \\ \text{then error else } C[P_1](s) \oplus C[P_2](s)$$

OMID

(list processor)



نیست و در دینامیک صدای دارد. بیان این سیاست برای پردازش زبان پر از است impractive ازین بار برای هشتاد و عنین در پردازش استفاده نمایند کارگزینه شده است. پردازش را فکل ساخته شده به حساب می آید.

99, ۸, ۱۵ *

در مفهوم معمولی محاسبات symbolic طبقه بندی می شود. محاسبات عالی با صورت symbolic نیست. اما این زبان های مخصوص مورد استفاده نیست و عمومی است. در این مابین های کم به lisp (اصناف شده، زبانی برای پردازش آن) به وجود آورده اند. می گویند functional-lisp functional-lisp نیست. اما در واقع من توانم یا کامپلاین side effect را هم اثرا ندارم. عادت مکمل pure (pure, lisp) نیست دلیل این توانم به صورت pure functional نیست اما این pure نیست. پس نه کوتاه این pure نیست.

ویرگی می خوبد motivation application

بررسی abstract machine طراحی شده. در واقع می بینیم این آن است. از تواند abstract concrete باشد. افزایش در قدر و جایگزین این امثله دارد. اگر زیادی رفتی باشد، compatibility، portability و درجه داشت. دیگر زیادی نادستی باشد، efficiency، را لازم است در دهن نیست. در IBM 704 lisp می بینیم این کرید این را بگزینیم زیادی سهی نیز نداشت turing complete theoretical foundation

و همان در lisp این قابلیت را اثبات کرد.

parse این استفاده می شود. علی را با استرمیکن prefix از lisp (۱۲).

$$(+ 1 2 3) = 1 + 2 + 3$$

$$(* (+ 2 3) (+ 1 2)) = (2 + 3) * (1 + 2)$$

Subject:

Year. Month. Date.

لisp یعنی atoms and cells هست. اتم خاص تراشیده است، سیبلیک باشد، ... رکل کوچکترین واحدی سیار موردعیب من تراشیده است.

$\langle \text{atom} \rangle ::= \langle \text{smb} \rangle \mid \langle \text{num} \rangle$

$\langle \text{smb} \rangle ::= \langle \text{char} \rangle \mid \langle \text{smb} \rangle \langle \text{char} \rangle \mid \langle \text{smb} \rangle \langle \text{digit} \rangle$

$\langle \text{num} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{num} \rangle \langle \text{digit} \rangle$

نیز اتم خاص به عبارت nil بود. در دو زیر حالتی null مانند است.

(dotted pairs) S-Expression \leftarrow lisp هست

$\langle \text{sexp} \rangle ::= \langle \text{atom} \rangle \mid (\langle \text{sexp} \rangle . \langle \text{sexp} \rangle)$

مثال : book از این میان IBM این گرفته است.

(a . b)

(a . (b . (c . d)))

Functions and Special Forms

cons (ا) = $\overset{\text{cons}}{\text{cons}}(a, b)$ دو یادویست مارکیپ می کند. \leftarrow cons

atom ، eq ، cdr ، car \leftarrow این را به تعریف می کند

content of address \leftarrow content of address

هست یانه decrement register reg

eval ، quote ، define ، lambda ، cond شرطی \leftarrow می بیند

ارس می کند این تعریف تابع

(eval e) (quote a) \leftarrow جمله eval

(lambda (x)(+x 2)) (cond (p₁, e₁) ... (p_n, e_n))

set \leftarrow imperative ، "set" ، eval

side effect \leftarrow این توابع دستورات دارد.

rplaca = $\overset{\text{rplaca}}{\text{rplaca}}$ با آنچه دو یادویست مانند

rplacd =

set =

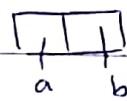
OMID setq =

Subject:

Year. Month. Date.

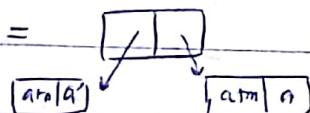
(quote cons) = 'cons

(cons a b) =



، الـ

(cons 'a 'b) =



(+ 1 2) = 3

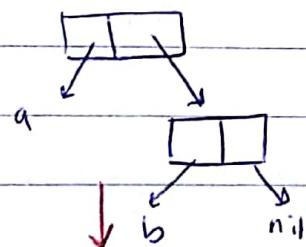
'(+ 1 2) = اخْرَجْتُ (لِسْتَ) (عَرْفَتُ)

car ('(+ 1 2)) = +

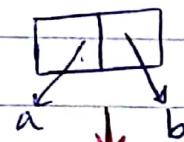
cdr ('(+ 1 2)) = 12

→ '(+ 1 2) = (+ . (1 . (2 . nil)))

'(a b)



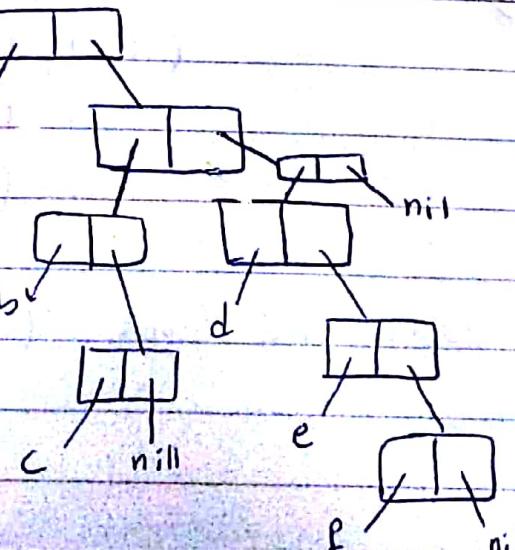
'(a . b)



(cons 'a 'b)

(cons 'a '(b))

'(a (b c) (def)) →



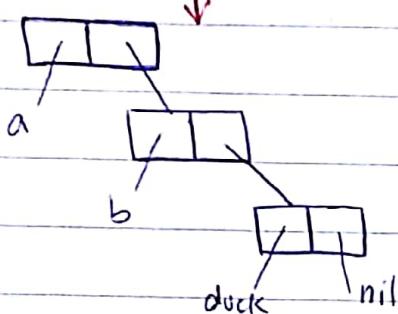
OMID

PL 10

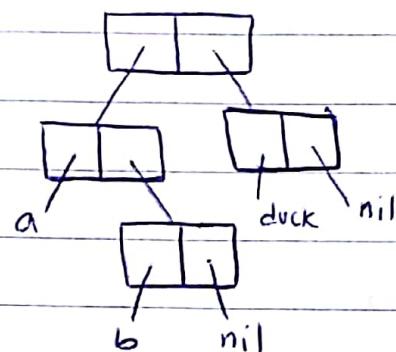
Subject:

Year. Month. Date.

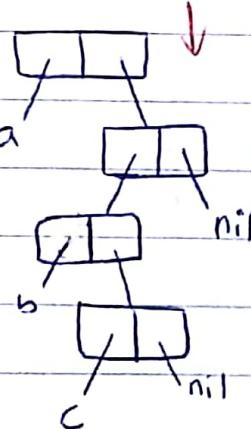
'(a b duck)



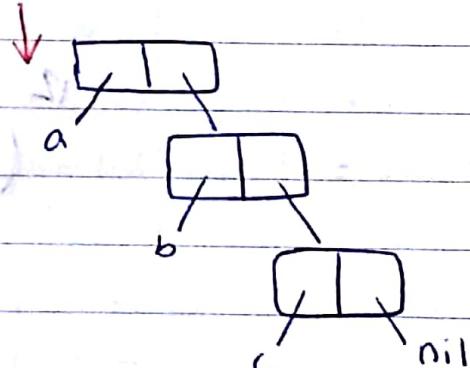
'((a b) duck)'



'(a (b c))'



'(a b c)'



بيانی زبان برنامه نویسی *

'(a buf duck PL)

→ (defin find (Lambda(x y)

مثال: بحث در لیست

```
( cond ((equal y nil) nil)
        ((equal x (car y)) x)
        (true (find x (cdr y))))))
```

Q: (find 'duck '(a buf duck PL))

OMID

Subject:

Year. Month. Date.

۹۴/۸/۱۴ *

قبل از lisp را بخواهیم imprative بودند. ارسن زبانی برای P و تغییبات شرکت می‌باشد

$(\text{cond } (P_1, e_1) \dots (P_n, e_n))$ expression

و ترکیب دیگر کمتر آن است که شان را در مدل مدلینگ می‌سازد است

در عمل کسی کوچک دیدار شود جزئیات بیشتری در اینجا lisp abstraction می‌باشد از این سایت برترین خودش را لایه کرد.

8. lisp abstract machine .

عبدالجعفری که باعث می‌شوند . expression

پس از این اعمام exp فناوری . عدی کلام است .

: صفتی است که stack را باشد . معنیت صفتی هم است .

heap stack local : در حال اینجا global و همچو هم در

Heap : دردهای global و عویض کرده باید بیشتر نگفته ایم نیست . راهنمایی برای هم

heap . ممکن است اثراورکی را Heap بر A-list هم داشته باشیم .

مجموعه ای از cons هاست .
($\boxed{\square}$)

در زبانی تمام برنامه های اینها به شکلی می‌شود . میان زبانهای مختلف

تفاوت مختلفی داشت خود را دارند . تفاوت آنها در مدلینگ تسلیم میان آنها به که مائین است .

و گرینه خروجی هم است . تفاوت در این اینها در تفاوت این است که اراضی این برنامه زبان قرار می‌گیرد .

وقت (b) کسیم یک اش اور به رفت
a b a b

Programs as Data .

نهایی program را متعاطار \rightarrow entity که با این امثله رسانی از خود را می‌دهد .

که یک مجموعه ای از رشته های درست نارضی .

روی program معلم است . مشوار باشد وی را lisp این افعال به راسخ دعوه را در

OMID

PL ۲۱

Subject:
Year. Month. Date.

Ques) A lisp function to substitute expression x for all occurrences of y in expression z and then evaluate the resulting expression.

resulting expression.

(true (cons (Subtitude expl var (car exp2))

(Substitute expl var (cdr exp2)))))))

فرضیہ \rightarrow یہ atom است یا یہ رشہ \rightarrow اگر atom است فریئن جو شودگی
ہے تو جب رشہ یہ بارصکٹور میں راست رشہ چک شود۔ تابعیں فریڈر کے درمیان
میں var کے اندر رابطہ expl معنی منکرنے۔

(define substitute-and-eval (lambda (x y z)

(eval (substitution x y z))))

صدهم جگه زیرینی فرستند و هم احرا من کنند. که به رشته مانند بزم از این امور نگاه مانند

Higher-order functions

تابع مرتبه اول \rightarrow مدل اشناید Valve است و فرم اصلی تابع.

تابع سرتیپی لعم \leftarrow درودی ائڑے کی تابع است کہ آن تابع یک value مہ گیری۔

تابع مرتبه‌ی سه \rightarrow در دری اشن یک تابع مرتبه‌ی دو است.

▷ $((\lambda x. (\text{cons } x '(A B))) 'Apple)$

'(Apple A B)

OMID

Subject:

Year. Month. Date.

نوشتن یک تابع برای تابع f و g را که $f \circ g$ نویل دهد.

(define compose (lambda (f g) (lambda (x) (f (g x)))))

تابع
درین
بررسی = تابع است

درین هم تابع است که باهم ترکیب می شود.

94, ۸, ۲۱ *

فیلم اگر لست بگیر، آن تابعیست که برای اجرای آن racket را در طبقه بین eval

کاری دارد که من توان آن تابعیست را رفع کرد.

(define (atom? x)

(and (not (null? x))

(not (pair? x))))

(define substitute (lambda (exp1 var exp2)

(cond ((atom? exp2) (cond ((eq? exp2 var) exp1)

(true exp2)))

(true (cond ((equal? null exp2) null)

(true (cons (substitute exp1 var

(car exp2)

(substitute exp1 var (cdr exp2))))))

▷ (substitute_and_eval '(+ 1 2) 'x '(+ x x x))

۹

سی بازابلیت eval، پس از تغییرهای لول (بهصرورت زنگنه عبارت را حذف کرد. این طرا

درین مکان یکسان می بینیم.

▷ (substitute_and_eval 'b 'x '(cons 'a 'x))

'(a b)

(دو) لست نشست

OMID

PL

Subject:
Year. Month. Date.

D (substitute-and-eval '(cons 'a 'b) 'x '(cons 'b x))
'(b a.b)

▷ (substitute-and-eval '(cons 'a (cons 'c 'd)) 'x '(car (cons x '(2 3))))
'(a c d)

هیچنیں مدرسہ کی قبل درود رکس تو ایک ہم بیٹ کر دیں۔

D (P 2)

8

هذا، إن `highorder` هي `func` تأخذ `func` كـ `param`، ثم ترجع `func` مع `param` مضاف.

D (define square (lambda (x) (* x x)))

D (maplist square '(1 2 3 4))

'(14916)

8 Garbage Collection

در طول اجرای سیال حافظه ای برگیرید و راستاد نیست را خواهی کرد. یا خود را این مانع است
مالار مثل D_{S1} و D_{S2} و یا باعده در طول برآوردها است این عمل را این رهند شن C .
اعمایی که از تو پیشست انجام می رهند محسن: برآوردهای سازی مدارهای موصلی این موضوع باشد.
که بعد مس درجه overheat.

ارجمندی سے ایک سب سے بہترین حالت اقتصادی performance و سرعت۔

OMID siloso, error global, see e

(cons 'a '(b)) \rightarrow '(a b)

Subject: (cons 'a 'b) \rightarrow '(a b)

Year. Month. Date.

گاہی ایام درست میں اسی کو اپنے ایجاد کرنے کا سلسلہ تھا۔

At a given point in the execution of a program P, a memory location m is garbage if no completed execution of P from this point can access location m.

جیسا کہ اسی بنا پر دیگر کوئی کام نہیں کر سکتا۔

وارد گرد تا سورداش کوئی کام نہیں کر سکتا۔

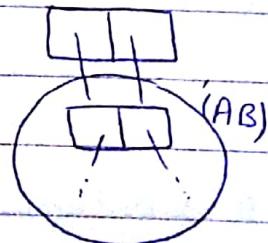
Mark-and-Sweep (1)

reference_count (2)

94, R, ۲۴ *

' e₁ e₂ کو اسی کام کے بعد e₁ کو '(car (cons e₁ e₂)) کی طرح مارک کر دیا جائے۔

(lambda (x) (car (cons x x))) '(AB)



: mark-and-sweep (1)

tag کو 1 کر دیں۔ اس کو لیست میں اضافہ کرو۔

: mark-and-sweep (2)

Set all tag bits to 0

start from each location used directly in the program.

Follow all links, changing the tag bit of each cell visited to 1.

Place all cells with tags still equal to 0 on the free list.

OMID

PL ۲۴

Subject:
Year. Month. Date.

روشن reference count ہم بیں مبادلت بہرین روشن سیت۔
 جیسے تبلیغ رکورڈ مزایا دھانن رہنگی کہ بھروسہ تھوڑے حامل
 ہستے ہیں اسی پر جو اپنے error prompt نہیں ، سریار ہی سائی دارم۔ اگر دستے
 برناصر نہیں ملے تو زبان برائی سخت تر خواهد بود و از طرف آئان رکون برائی نہیں ہم جزو دلایا
 جان کاسٹ :

عملیاتی انتخاب (Select) که دارای یک لیست را می‌پرورد و آن را با ایجاد از نظر بود را برگرداند:

```
(define select (lambda (x list)
  (cond ((equal list nil) nil)
        ((equal x (car list)) (cdr list))
        (true (select x (cdr list))))))

```

در صورت که دیگر سیازی بے list باشد، اگر ~~نخواهد~~ باید ایناًت، سررا
متداول باشد.

```
typedef struct cell cell;
```

Struct cell {

Cell * car, * cdr;

1

```

cell* Select (cell* x, cell* list) {
    cell* ptr;
    for( ptr = list; ptr != Ø; )
        { if (ptr->car == x) return (ptr->cdr);
          else ptr = ptr->cdr; }

```

لرستان تاریخ میل . و آنده دیدا به خان کردن فعال اعماق فکر نشده است .
قرنوار عل خان کردن این صورت سق اعماق شد .

Subject: _____
Year. Month. Date.

```

→ → Cell * select ( Cell * x, Cell * list ) {
    Cell * ptr, * previous;
    for ( ptr = list, ptr != 0, )
        if ( ptr->car == x ) return ( ptr->cdr );
        else previous = ptr;
        ptr = ptr->cdr;
    → → free ( previous );
}

```

پس چون) تابیت خواهی بود و خود را نماید، ماید و نمایند این صورت
درست در هنر نیاز حافظه را پاک نماید که تا بهار از درون خود بگیرد.

لار referential transparency خواصی دارد. این مقدار عبارت را بازگشتن چنین نام عرقی ندارد.

He was called William Rufus because of his red beard.

ویلیام IV \rightarrow قدر \downarrow
اگر دلیل خارجی بر از این مکانیزم نباشد، باید این دشمنی کاملاً خود را در خود داشته باشد.

لisp از زبانی ماتم می‌باشد که اصولاً دارای گیره بصریت pure Functional است. به عبارت دیگر همه کارها را با استفاده از تابعیت function-oriented نمایش می‌دهد. در اینجا lisp با استفاده از functional monad چنین کارها را می‌نماییم: car , set ، replacecar ، let و map که مانند assignment در پیش از اینها معرفی شدند.

(rplaca x y) replace the address field of cons cell x
with y

(replace x y) ~ ~ ~ decrement ~ ~ ~ ~ ~

OMID

PL YE

Subject: _____
Year. Month. Date. _____

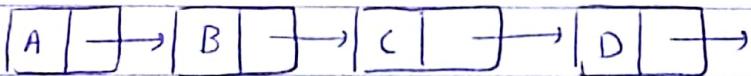
این رستورانت بعثت می شود Side effect
$$(\lambda(x)(\text{cons}(\underline{\text{car } x})(\text{cons}(\text{replaca } x \ c) (\underline{\text{car } x}))))$$

$$(\text{cons } a \ b) \ a \ c$$

نیز نیز بعثت میگذرد که بسیار خوب می شود درین توان اصل درخت را بخواهیم
کرد و از این هم نمایند. در مثال میباشد $(\text{car } x)$ که تأثیر قرارگیرنده است و
بعثت پیغامش نمایند.

دستورانت دارای side effect است اگرینه که برای سریع هدایا کمی میگیرد
سیار کند. پس فقط دلایل دو دستورانت دارید که درین توان بازدهی سریع هدایا ایجاد

x:



فرزن کنید برای این بگوییم که عیوب سعیم را باید چه بجاییں :

PURE
 $\xrightarrow{*} (\text{cons}(\text{car } x)(\text{cons}(\underline{\text{cadr } x}) (\text{cons} 'y (\underline{\text{cddr } x}))))$

car of cdr

cadr of cdr of cdr

impure
 $\xrightarrow{*} (\text{replaca } (\text{cddr } x) 'y)$

OMID —————