

14ام از جلسه نهم : 25.9.93

برنامه نویسی Prolog

محمد ابراهیم شیری

دانشگاه صنعتی امیرکبیر - دانشکده ریاضی و علوم کامپیوتر

91-92 بهار

موفق باشید

لیست چیلی کاربری است و بعده ساده و مفید چیلی از برآمدهای لیست حل می شود.

List programing \Rightarrow Lisp زبان برنامه نویسی

پیمانه های برو لوگ خوب است و کاربردی است (اسا دلخواه) می باشد این از هم حاکم نند و برای اسناد بفرستند. به از خود به ترم معبر ابله : (

و هم زبانهای ماله Functional و تابعی. شباهت های زیادی دارند اما خوب قوای تایی هم دارند.

لیست ها، عملگرها و حساب

فصل سوم

لیست یکی از ساده ترین و مفیدترین ساختار است و در برنامه های برو لوگ برای عملیات مناسب از لیست استفاده می شود.

لیست یک ساختمان داده ساده عمومی است که در برنامه نویسی

غیر عددی استفاده می شود. (ترتیبی از تعدادی اجسام است)

یک لیست ترتیبی است از تعدادی ایتم مانند :

ann , tennis, tom, skiing

یکی از روش های نمایش این لیست در پرو لوگ به شکل زیر است:

[ann , tennis , tom, skiing]

که در این حالت لیست خالی را به شکل زیر نمایش میدهند : []

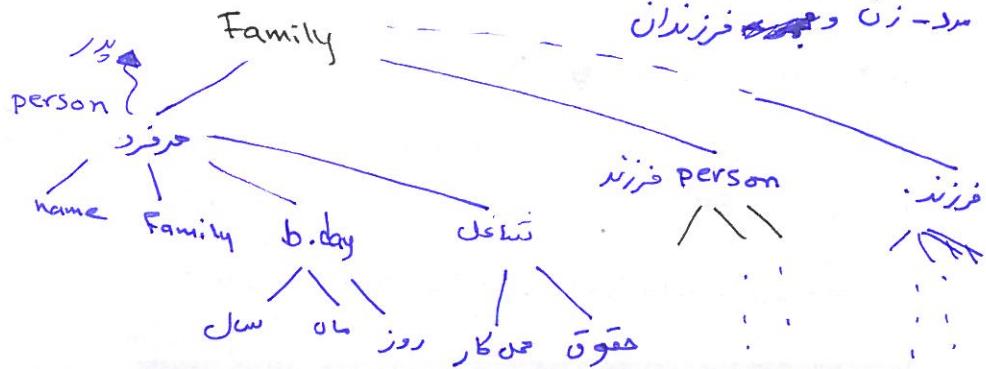
[]:

عنصر لیست حاکم برای همان مسیر کار از [] لیست خالی می تواند فرضی باشد این است که آنرا به مردمی از اعماق این لیست می کاریم این ایتم از اعداد یا کارا سر صحاب است

اما لیست می تواند هر چیز باشد. آنرا به لیست از اعداد یا کارا سر صحاب است

اطلاعات اخراجی خانواده و استنتاج حایی دهنگان از آن برسی کورس

جلسه ۴



اگر این متصفات را در این وسیله نگران آن را در پایه داره همراه را دارد.

اوین نقد برای این نتیجه که این شفاهم شناختی است.

که اینها هموزن ساخته شده اند / اینها هموزن ساخته شده اند

Family ([person | shahram, shayeghi, date, (1, 9, 60)],
 work (say pa, 3000)), [person | maryam, shayeghi,
 date (10.8.68), work]), [person | roya, shayeghi,
 date (11, 6, 86), unemployed]), [person | amir, shayeghi,
 date (12, 7, 85), unemployed]).

Family ([person | sharifi, ---

Family ([person | shekarchi, ---

اگر این های را در پایه داره من لذت میشم و تا زمانی که سوالات مایا سخن بدهم
 سوال: underline

? - Family (person(Name, Fg, ---) , person(---, ---, ---))

که کاره بکار بینه تو در و معوقی ندارد.

آنچه که برای این

date (---, ---, 4), (---, ---, 7) > 79

نمیشود این میگذرد همچنانچه برویم همچنانچه این نیست

brown(X) چه رند و نرا اجلد بیوں مهند

دنبال های brown باشد گردد. به جای \leftarrow \rightarrow می نزدیم
brown bear یا brown bear می شود.

حال با مرتبه \rightarrow big(X) یا big bear می شود؟

/ به درخط یا وجود دارد و عملی می شود برآورده می شود.

لیست هدف : black(X), big(X)

بویش را در زیر خط 5 برنامه ادame میدهیم عبارت 8 پیدا میشود
dark(Z):-brow(Z).

اولین هدف را با brown(X)

جایگزین میکنیم عبارت جدید دیگری به صورت زیر بدست می اید.

brown(X), big(X) برنامه را برای یافتن تطبیقی با brown(X)

بویش میکنیم ، این عبارت در خط 4 پیدا میشود
حال با قرار گذاشتن bear به جای X

لیست هدف تهی میشود و برنامه حاتمه میابد یعنی با جایگذاری

X=bear

اهداف برآورده میشوند.

برای حل سئه زیر لوگ باید به دنبال یوچ می‌باید در هم کار باش و مطابق باشد
 زیر لوگ یعنی مهندس اول (dark(X)) را برآورده نماید و بعد به علی (big(X)) و مطابق باشد.
 از آنکه زیر هدف حاصل برآورده شود، با استفاده از چیزی مثلاً از زیر باشید
 و من نه به خط تحریر می‌باشد. می‌باشد به دنبال (big(X)) مطابق باشد
 \Rightarrow هدف دوم \Leftarrow هدف دوم (cat) و مطابق شود. عبارت سرآمده سور
 دیگر مترادست. یعنی با قاتر 8 خط نمی‌تواند بجوابی برسد.

سوال زیر چگونه ردیابی می‌شود:

?- dark(X), big(X).

ردیابی اجرا:

لیست هدف اولیه: dark(X), big(X)

پویش برنامه از بالا به پایین جهت پیدا کردن عبارتی که با هدف اول مطابقت داشته باشد. عبارت خط 7 یافت می‌شود. dark(Z):-black(Z).

اولین هدف را با بدنه عبارت خط 7 جایگزین می‌کنیم عبارت جدید دیگری به صورت زیر بدست می‌آید.

black(X), big(X)
 برنامه را برای یافتن تطبیقی با black(X)

پویش می‌کنیم، این عبارت در خط 5 پیدا می‌شود black(cat)

حال با قرار گذاشتن cat بجای X

باید هدف دوم به شکل زیر برآورده شود big(cat)

برنامه را برای هدف big(cat)

پویش می‌کنیم هیچ عبارتی پیدا نمی‌شود با شکست مواجه می‌شویم

حال نمونه سازی X=cat

را لغو می‌کنیم و برنامه را دنبال می‌کنیم لیست هدف بصورت زیر است:

حال به خط 8 می‌رسد با استاره از طرف 8 سمع مهندس حواب برسد

← قوییز و حیند هر قبیر که حقیقتی و مشکل است (سرمه) است

Fact ۵۶
rule ۵۲

اجرای یک مثال

فصل دوم

فرض کنید برنامه زیر را داشته باشیم:

big(bear).	1
big(elephant).	2
small(cat).	3
brown(bear).	4
black(cat).	5
gray(elephant).	6
dark(Z):-black(Z).	7
dark(Z):-brown(Z).	8

نکته: روش روالي (یا رویه ای) تعین میکند که پرولوگ چگونه به سوالات جواب میدهد.

باسخ به یک سوال به این معنی است که لیستی از اهداف برآورده شوند. آنها در صورتی برآورده میشوند اگر متغیرهایی که در اهداف وجود دارند بتوانند به روشی نمونه سازی شوند که اهداف به طور منطقی از برنامه پیروی کنند.

بنابراین مفهوم روالي پرولوگ، رویه ای برای اجرای اجرای لیستی از اهداف با توجه به برنامه است.

* هایلدری لسی از اهداف برای اصرای برنام

وچایلدری

حالات هستند. $P: -Q, R$ $\rightarrow P \leftarrow Q, R$

\hookrightarrow (حاجز \leftarrow اس)

* ۹ و ۱۰ زیر عدف هستند

معنای اعلانی و رویه ای برنامه های
برولوگ

فصل دوم

برنامه های برولوگ به دو راه فهمیده میشوند اعلانی و رویه ای . برنامه زیر را در نظر بگیرید:

$P:-Q, R.$

خواندن به روشن اعلانی:
 P درست است اگر Q و R درست باشند.
 از Q و R ، P نتیجه میشود.

خواندن به روشن رویه ای:
 برای حل مسئله P ابتدا زیر مسئله Q و سپس زیر مسئله R حل میشود.
 برای برآورده شدن P ابتدا Q و سپس R برآورده میشود.

نکته: روش روایی (یا رویه ای) تعین میکند که برولوگ چگونه به سوالات جواب میدهد.

لذت روایی: چیزی که حل مسئله در برولوگ را به ماسن می خود.

ارس اعلانی برای خصم مابه طرف رود

اگر سوال زیر را بپرسیم: آیا هیچ پاره خط عمودی از نقطه (2,3)

شروع میشود

?- vertical($\text{seg}(\text{point}(2,3), P)$).

$P=\text{point}(2,Y)$

جواب پرولوگ در بالا امده.

سوال جالب دیگر: آیا پاره خطی وجود دارد که هم افقی و هم
عمودی باشد؟

?- vertical(S), horizontal(S)).

که جواب به شکل زیر است:

$S=\text{seg}(\text{point}(X,Y), \text{point}(X,Y))$

يعني هر پاره خطی که با يك نقطه مجدداً توليد شود داراي خاصيت
عمودي و افقی است

يعني هر تواند خط که نقطه باشد.

خط اول نویسید از خط دو برابر باشد سیزدهم

خط دوم نویسید که اگر برابر باشد سیزدهم

فرض کنید در یک برنامه پرولوگ خطوط عمودی و افقی به صورت دو
واقعیت به شکل زیر تعریف شده باشند:

`vertical(seg(point(X,Y), point(X,Y1))).`

`horizontal(seg(point(X,Y), point(X1,Y))).`

آنگاه جواب سوالات زیر از پرولوگ در زیر آن آمده:

?- vertical(seg(point(1,1), point(1,2))).

Yes

?- vertical(seg(point(1,1), point(2,Y))).

no

?- horizontal(seg(point(1,1), point(2,Y))).

Y=1

اگر سوال زیر از برولوگ بررسیده شود:

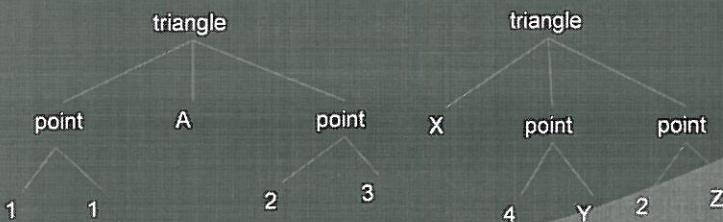
?- triangle(point(1,1), A, point(2,3))=triangle(X, point(4,Y), point(2,Z)).

کل روند نطایق موفقیت آمیز است و جواب برولوگ بصورت زیر است:

X=point(1,1)

A=point(4,Y)

Z=3



در صد اول Functor ها با هم تابع دارند
حالا باید از تو سانه ای بررسی کنیم و جایی لذاری انجام بدهیم

در جالت \mathcal{C} نمی‌شود که علیه باشد می‌رند، اما در هر رسمیات باشد.

(حالات⁽²⁾) برای تغییر دهنده‌گر در صورتی تغییر متناسب با جایگزین S نیز هر رسمی استد سطابق هم شوند.

قوانين عمومی برای تصمیم گیری اینکه آیا دو عبارت S و T باهم تطابق

دارند:

اگر S و T ثابت باشند، در صورتی باهم تطابق دارند که هر دو یک ✓

شیء باشند

اگر S متغیر و T هر چیزی باشد، باهم تطابق دارند و T جایگزین S ✓
میشود

اگر S و T ساختار باشند، تنها در صورتی تطابق خواهند داشت که: ✓
۱- S و T اجرا کننده (functor) یکسانی داشته باشند

۲- تمام اجزاء آنها نظیر به نظیر با هم تطابق داشته باشند،
جایگزینهای نتیجه شده با تطابق اجزاء معین میشوند.

روند تطابق از ریشه آغاز میشود. هنگامی که اجرا کننده ها باهم
تطابق یافتند، این فرایند به آرگومانها میرسد تا جایی که تمام
آرگومانهای نظیر به نظیر تطابق بیابند.

مثال: سوال ریر را در نظر بگیرید:

```
?- date(D,M,2001)=date(D1,may,Y1),  
date(D,M,2001)=date(15,M,Y).
```

برای برآورده ساختن هدف اول، برو لوگ متغیرها را بدین ترتیب جایگزین می کنند:

D=D1

M=may

Y1=2001

بسی از برآورده شدن دومین هدف جایگزینی به صورت ریر خاصتر می شود:

D=15

D=D1

M=may

Y=2001

Y1=2001

شال خط اول برابریت چون کسی است و دندر ۲۰۱۲ است

و در خط دهم هم Functor برابریت

اگر مرحله اول تطابق برقرار نبود پس عبارت $\text{date}(D1, \text{may}, 2001)$ سرد بررسی مارضی نباید و یک جمله نتیجه اول نام و پس برشاپ با این فرآیند را می‌نماید.

ولی دو عبارت $\text{date}(D1, \text{may}, 2012)$ و $\text{date}(D, M, 2001)$ و یا دو عبارت

$\text{point}(X, Y, Z)$ و $\text{date}(X, Y, Z)$ تطابق ندارند.

تطابق روندی است که در آن دو عبارت ورودی باهم بررسی می‌شوند که آیا باهم تطابق دارند یا نه.

اگر عبارات باهم تطابق نداشته باشند، می‌گوئیم این فرآیند قابل قبول نیست

اگر دو عبارت با هم تطابق داشته باشند، پس روند روند موفقیت امیزی است و همچنین متغیرهای در هر عبارت با مقادیری جایگزین می‌گردند تا یکسان شوند.

۵) تئمود حسی هر دو ساختار را تطابق نظیر برابر باشند

مثال: $\text{?} = \text{بارات تماه باهم تطابق دارند} (\text{از زیر و لور بسته})$

exp: $? - \text{date}(X, \text{may}, Y) = \text{date}(D, M, 2001)$

$X = D$

$M = \text{may}$

$Y = 2001$

بروکور می‌اید انتظاری می‌دهد

تطابق

فصل دوم

دو ساختار باهم تطابق دارند اگر:

آنها باهم برابر باشند.

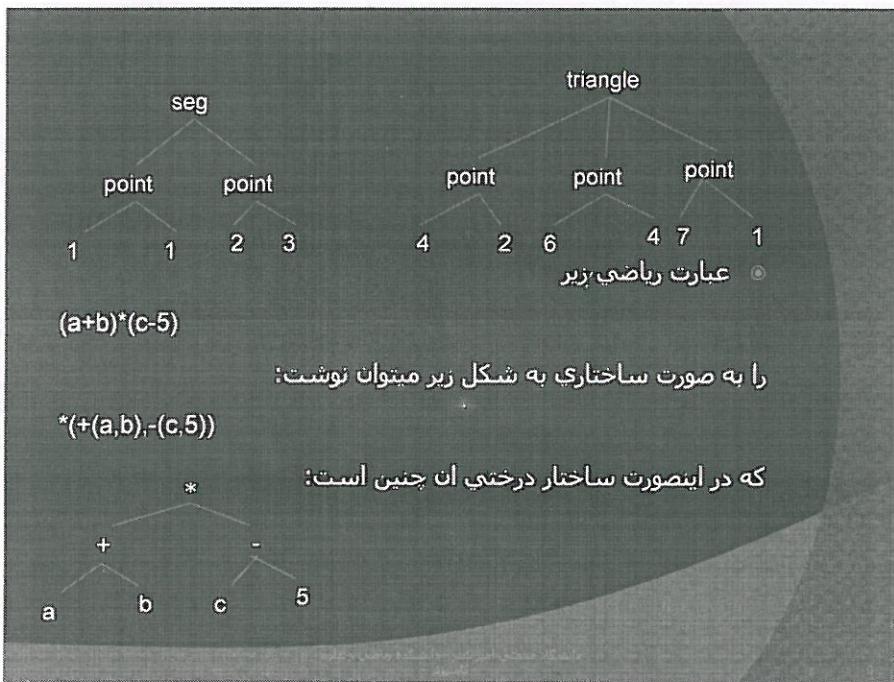
متغیرهای هر دو ساختار را بتوان به گونه‌ای به اشیاء معرفی کرد که پس از جایگزینی متغیرها توسط این اشیاء عبارات باهم مساوی باشند.

مثال: دو ساختار $\text{date}(D1, \text{may}, Y)$ و $\text{date}(D, M, 2001)$ با جایگذاری زیر برابر می‌شوند:

$D = D1$

$M = \text{may}$

$Y = 2001$

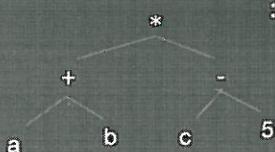


$(a+b)^*(c-5)$

را به صورت ساختاری به شکل زیر میتوان نوشت:

$*(+ (a, b), -(c, 5))$

که در اینصورت ساختار درختی آن چنین است:



هر دالد کی علامت هم باشد Factor

هر عبارت تر را میتوان به صورت گل سه تارسان داد

کی Factor است *

اعضای ساختار میتوانندشی ثابت، متغیر یا خود ساختار باشند.

مثال زیر نقاطی را در فضای دو بعدی نشان می دهند:

$P1=point(1,1)$.

$P2=point(2,3)$.

پاره خط

یک پاره خط را میتوانیم بصورت زیر تعریف کنیم:

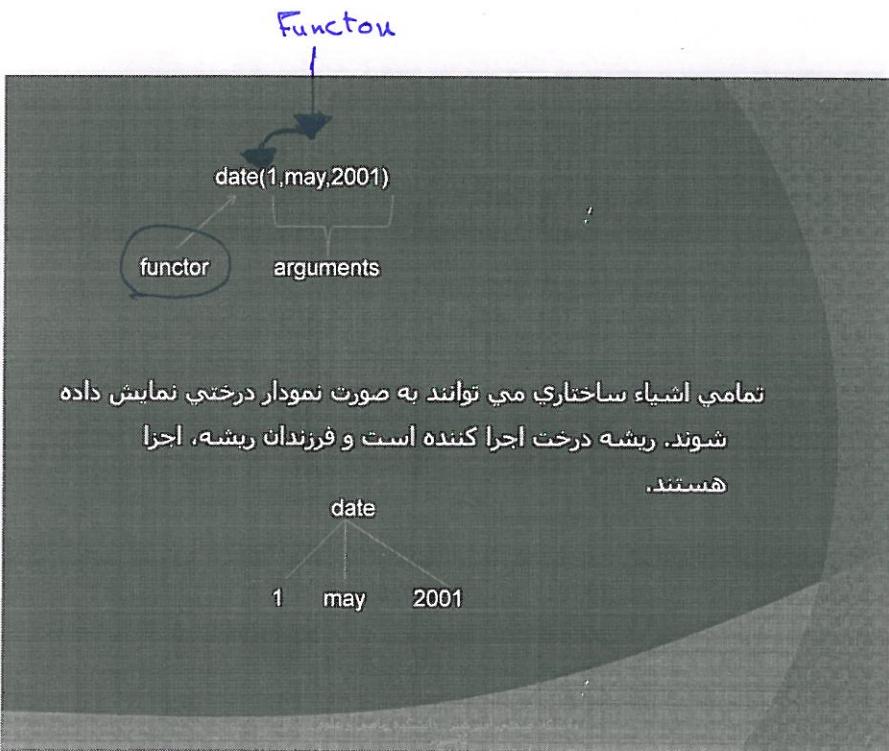
$S=seg(P1,P2)=seg(point(1,1),point(2,3))$

همچنین یک مثلث به شکل زیر تعریف میشود:

$T=triangle(point(4,2),point(6,4),point(7,1))$

نمایش درختی متناظر با این اشیاء در شکل زیر آمده است :





تمامی اشیاء ساختاری می توانند به صورت نمودار درختی نمایش داده شوند. ریشه درخت اجرا کننده است و فرزندان ریشه، اجزا هستند.

ساختارها حاصل هستند، علاوه بر این
ساختارها تردد ندارند. بهای بده می‌گزینند

ساختارها

ساختارها اشیایی هستند که چندین جزء دارند مثل تاریخ که میتواند یک ساختار با سه جزء (سال، ماه و روز) تلقی شود.

در زبان پرولوگ تاریخ 1 می 2012 را بصورت زیر می‌توانیم نشان دهیم:
`date(1,may,2001).`

همچنین دستور زیر یک ساختار را معرفی می‌کند:
`date(Day,may,2001).`

که در آن Day یک متغیر است و میتواند به هر شئ در حین اجرا مقدار دهی شود.

بنابراین یک ساختار در پرولوگ تشکیل شده از یک نام (که به آن اجرا کننده یا functor نیز گویند) و تعدادی آرگومانهای ورودی که اعضای ساختار را مشخص می‌کنند.

اَسْمَا در بِرولوک بَد ۱۰۳ طَرْنَد.

شَلَانَه لَو سِر لَو هِي هَسْت بَكْهَه تَهَارَه مَكَانَه در حَاقَّهه رَابِعَهه لَو لَهه

در لَكَدَهه رَسِّرَهه تَحَامَهه لَهَانَهه لَر فَرِزَنَدَهه لَهَانَهه - اَيْنَهه مَرِزَنَدَهه اَسْمَا دَارِهه مُهَمَّهه سِبَّ

پَرَهَتْهه مَعَ وَسْتَغِيرَهه بَنْهه نَامَهه

اگر متغیر بی نام در یک عبارت سوالی بکار رود، هنگامی که
برولوک به سوال پاسخ دهد، مقدار آن خروجی نیست. مثلاً اگر ما
به افرادی علاقمندیم که فرزند دارند و نه به نام فرزندان میتوان
سوال رل به سادگی به شکل زیر نوشت:

?-parent(X,_).

حوزه واژگانی نام متغیرها، یک عبارت است. یعنی اگر متغیری هم
نام در دو عبارت وجود دارد ازها دو متغیر مختلف را نشان میدهد.
البته وضعیت تابعها فرق میکند. یک اتم همیشه همان شیخ در
عبارة است که در تمام طول برنامه وجود دارد.

اگر نیاز نباشد متغیر را اعلام ننم از متغیر خط زیر میتوان استفاده نماییم
که این کد متغیر طحن است.

در داخل مجموعت ۲ چند کسی باشد و به جای آن میتوان (-) نداشت

متغیرها

متغیرها، رشته هایی از حروف، اعداد و کاراکترهای خط زیر هستند.

آنها با یک حرف بزرگ یا کاراکترهای خط زیر شروع میشوند:

X Result Object _x23 _42 Shopping_List

نکته: وقتی در عبارتی متغیری ظاهر میشود که نباید برای آن نامی در

نظر بگیریم از متغیر به اصطلاح (بی نام یا خط زیر) استفاده میکنیم

مانند مثال زیر:

haschild(X):-parent(X,Y).

را میتوان به شکل زیر نوشت

haschild(X):-parent(X,_).

نیتی حا با صرف نزد
نایتی حا با صرف کوچک
پرولوو نیت به صرف حساس است.

چند در ۱۰۰ نیت لذارم سه آنها استاد مردم نیت حرف نزد باشد نایر حکم
اعداد سوره استاده هم صحیح و طبیعی است

انتها

انتها را میتوان به سه روش ساخت:

۱- رشته ای از حروف، ارقام و کاراکتر خط زیر که با حروف کوچک شروع میشود:

ann bob x_25 x25

۲- رشته ای از کاراکترهای خاص:

... :: ==>

۳- رشته ای از کاراکترها که در نک کوتیشن قرار می گیرند:

'Tom' 'Sarah Jones' 'South_America'

اعداد

اعدادی که در برولوگ استفاده میشوند، عدد صحيٽ و طبیعی هستند (البه در فامله مجاز قابل تمايیز در کامپیوٽن)

اعداد حقیقی در برولوگ چندان کاربرد ندارند چون برولوگ در درجه اول یک زبان علanchی است و نه زبان عددی.

بعضی با علم و knowledge سرشار اند. آنها عدد.

البه که هی برخی صراحتاً اعداد نیاز ندارند. شلا بلی بیان محل

ب) بر ساختار object ها و دیالکتریک مسخن باشد.
و برای هر مسائل سمبولیک مرور استفاده ایم

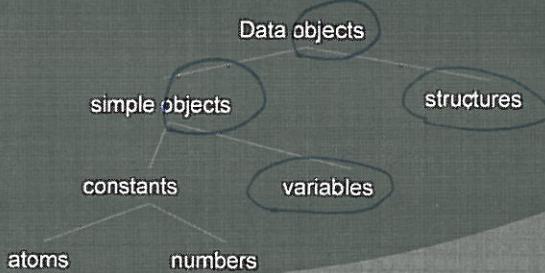
object ها از نوع دیالکتریک است. یا ساده است و یا ساختار
شایسته است

ساختار دستوری و مفهوم برنامه
های برولوق

فصل دوم

همچنان که در فصل قبل گفتیم برولوق یک زبان برنامه نویسی برای
حل مسائل غیر عددی و سمبولیک است و مناسب مسائلی است
که با اشیاء سروکار دارد.

شكل زیر طبقه بندی اشیاء در برولوق را نشان میدهد:



جلسه نهم : ۹۳.۹.۲۵

برنامه نویسی Prolog

محمد ابراهیم شیری

دانشگاه صنعتی امیرکبیر - دانشکده ریاضی و علوم کامپیوتر

بهار ۹۱-۹۲

بروزرسانی زبان طبیعی با استفاده از سروکوئی:

- ۱) یعنی جملهای را به سروکوئی بدهم مثلاً با این طبیعی و آنرا به مارسی بیان کنند.
+ با هر قوید هر دو را به آن بدهم و هم که داشتی بدهم.

طبیعی →
مارسی
لایه مواد
داستانی

مارسی
لایه مواد
داستانی

یعنی ایجاد مترجم زبان

و حفظ ایجاد تلفظ

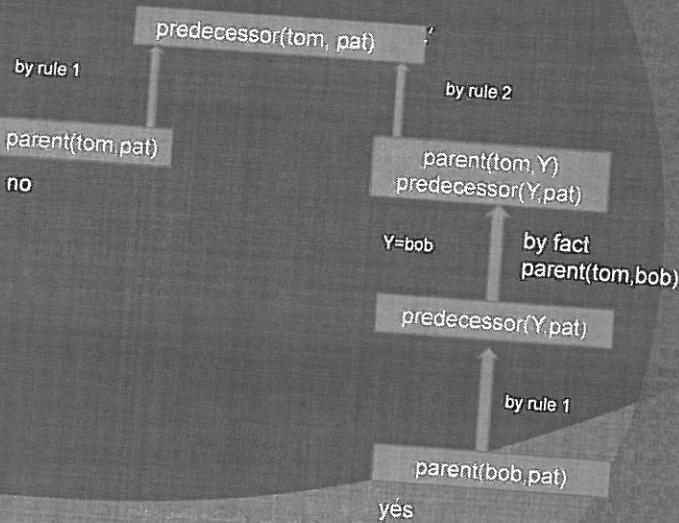
موفق باشد

۲) سبک استنایح  سبکه بر

با استفاده از سروکوئی استنایح از این دستگاهی نیست

+ لایه زیر برداشته است و استاد کتاب خود نیست (آنها در سیم و آفتاب توپخانه بودند)
از استاد همینه حسنه!

بروکل انجام قانون بعدی (خط ۸) را بررسی میکند و نتیجه شکل زیر است:



حال چگونه پرولوگ به سوال زیر پاسخ می‌دهد:

?- predecessor(tom,pat).

پرولوگ سعی می‌کند این هدف را برآورده کند.

پرولوگ این کار را با شروع از هدف (در جهت عکس) و با استفاده از قوانین اهداف جاری را با اهداف جدید جایگزین کرده و تا جایی ادامه می‌دهد که

هدف يك واقعيت ساده باشد.

اين کار با دنبال کردن خطوط برنامه از بالا به پايان و پيدا کردن اهدافي که با پرسيش مطابقت پيدا کند دنبال ميشود به شكل زير:

در جريان کار پرولوگ خط 7 برنامه را ميابد ، هدف (predecessor(tom,pat)

تبديل با جايگزاري:

X=tom, Z=pat

تبديل به هدف (parent(tom,pat) ميشود. اين هدف جزء حقائق برنامه نيست.

حالات یا آنام جدید است؟

بعضی مکالم از بالاتر ام کند اوین باید به جد predecessor خود را پخته است.

پخته است. حالات این خطا را با سوچی در زیر پرسیده هیچ کسی کند.

باید دو دفعه مقدار ممکن نظر را بگیری. بجای Z \leftarrow پست و X \leftarrow آنام

برای درسته این قانون باشد Fact آنام و پست را فقط ② درست پاسد

برای مکمل نتیجه نیزت، آنام و پست دارد که یا؟ هم بینند ندارد (لشکت می خورد)

چگونه برو لوگ به سوالات باسخ
متلاطف

فصل اول

فرض کنید برنامه زیر را داشته باشیم:

parent(pam, bob). 1

parent(tom, bob). 2

parent(tom, liz). 3

parent(bob, ann). 4

parent(bob, pat). 5

parent(pat, jim). 6

predecessor(X,Z):-parent(X,Z). 7

predecessor(X,Z):-parent(X,Y), predecessor(Y,Z). 8

حالات سلنجی قانون 8 مورد دارد. (این جاسته اهیت شرط ها و قانون ها
با صیغه هم خود داشتند) باشد قبل از قانون نویسند.
حالات شرط دوم یا بعدی همچند کسی کند.

پس باشد یا بعد از آنها کام نیزت و باشد و همچند پست باشد
که باز اول همچند و به خوبی مرد آمریکا جای و طبق بندی دارد
درست می شود می آنم صدید است.

(من و عزیز) دالا - : (من و عزیز) جد

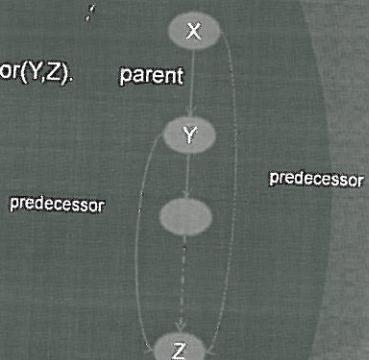
(من و مان) جد و (مان و عزیز) دالا - : (من و عزیز) جد

به تدریم خط اول به های حروف باید لجه لذات است (از خود نویس این را است دلخواه)

این مسئله توسط قوانین بازگشتی به سادگی قابل حل است. با استفاده از قوانین بازگشتی "جد" را به صورت زیر تعریف میکنیم:

$\text{predecessor}(X, Z) :- \text{parent}(X, Z).$

$\text{predecessor}(X, Z) :- \text{parent}(X, Y), \text{predecessor}(Y, Z).$



نکته: بدون استفاده از قوانین بازگشتی نمیتوان مسائل بیجیده پرولوگ را حل

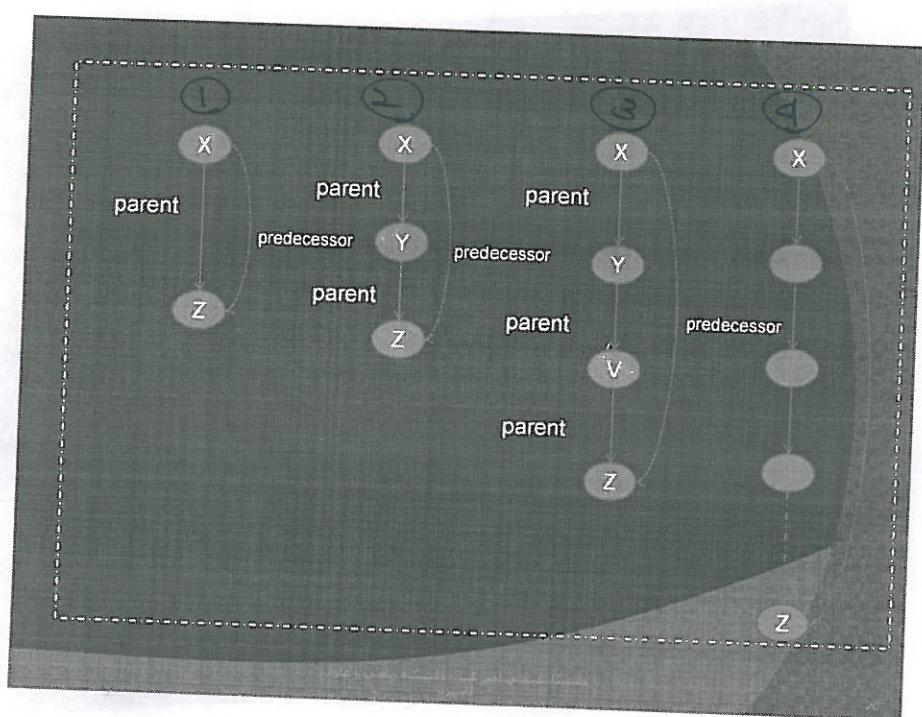
نمود.

(نقشه روکشی توکل
همی ممکن باز نیست است)

شرط اول شرط توقف است)

۲) اسرمه و والدی و لا والد ۲ باشد، سه ۶ در ۱۲ است. و بهین ترتیب

متوالی نظر داریم که قبل نویسه شده است. آنرا کمتر ریشه نست و بین جمله متن
قىلا باشد از خانم نازنین هادی شروع کنید و همین ادوارتان را تعریف کنید تا به حضرت «دم
بر سینه آرامی شود خانم فنا دلیل هر باب بله!» موصیم به (ها) تاون! (د)؛) البته دهان بپردازید.
می توان از عائون برگشته استفاده کرد.



مُوافِق بَارِلِسَيْ بِسِارِ رَصْم حَسَنَد دَانْغُطَافَنِيرِسِرِيْ پِرَولُوك هَدْرَمَا بَه خَاصِر هَمِينْ قَوَاعِد
اَسْت . مَثَلًا $loop$ اَنْمِي خَواهِدِ جُونْ خَورَسَنْ با دَسَّوَرَات بَارِلِسَيْ لَوِيْ اِيجَا دَمِيْ كَنْد
الْبِهَ دَقَتْ دَائِسَهْ بَاشِيدَهْ مَوَاعِد بَارِلِسَيْ بَايدَهْ صَهَهْ سُرَطْ تَعَمَضْ رَادَائِسَهْ بَاشِد
هَنَالْ زِيرِ بَاهِئْ هَهْ سَهَوَسِيمَا حَوايدِ بَارِلِسَيْ رَالِهِسِيْ لِينَد .

قَعْدَ تَعْرِيفِ جَهْرِيْ بِيرِزِيرِكْ اِسَكْ يَا predecessor اَسْت

قوانين بازگشتی

فصل اول

- قانونی بازگشتی، قانونی است که خود را فراخوانی میکند.
 - قوانين بازگشتی ، پرولوگ را قادرمند میکند و امکان این را می دهد شرطهایی که بیان آنها در حالت عادی راحت نیست را ساده کند. از این قوانین در پرولوگ شدیداً استفاده می شوند.
 - مثلا برای تعریف حد در حالت عادی نیاز به زنجیره ای از قوانین هستیم که عملاً بصورت نامحدود ادامه پیدا می کنند و مشکل ساز میشود مانند زیر:
- ```

predecessor(X,Z):-parent(X,Z).
predecessor(X,Z):-parent(X,Y),parent(Y,Z).
predecessor(X,Z):-parent(X,Y),parent(Y,V),parent(V,Z).
.....

```

مانند شکل زیر، این برنامه خیلی طولانی میشه و تنها تا فاصله خاصی عمل میکند.

واقعیتها عباراتی هستند که فقط سر دارند و بدن آنها حالي است.  
سوالات تنها بدن دارند. قوانین هم سر دارند و هم بدن دارند.  
در هنگام محلسیات یک متغیر می تواند با یک شئ جایگزین شود و  
می گوییم که متغیر اعلان شده است.

متغیرها کمیت عمومی هستند و به صورت "برای همه" خوانده می  
شوند، برای مثال قانون زیر

$\text{haschild}(X) :- \text{parent}(X, Y)$ .

به دو صورت می تواند خوانده شود:

(الف) برای تمام  $X$  ها و  $Y$  ها اگر  $X$  والد  $Y$  است انگاه  $X$  یک فرزند  
دارد.

(ب) برای تمام  $X$  ها  $X$  فرزند دارد اگر یک  $Y$  وجود داشته باشد که  $X$   
والد  $Y$  باشد.

ـ قوانین زیراگاه حکم کردن ه نویسید  
ـ میراث مانعون در تصریف قدرتمنه می شوند

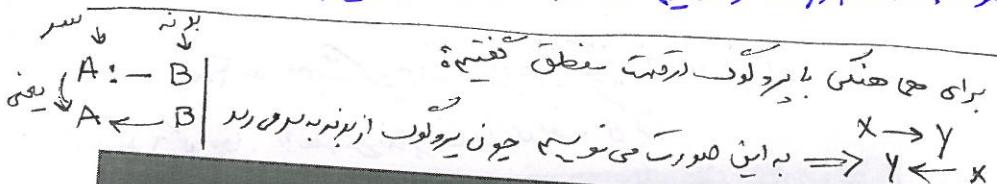
### نکات مهم:

- برنامه های پرولوگ با اضافه کردن عبارتهاي جديده گسترش مي يابند.
- شرطهاي پرولوگ سه نوع مي باشند: واقعيتها، قوانين و سوالات.
- واقعيتها عباراتي هستند که همواره صحيح مي باشند.
- قوانين عباراتي هستند که بسته به شروط خاصي مي توانند صحيح باشند.
- با استفاده از سوالات کاريکر مي تواند بفهمد که جه حيزهایي صحيح مي باشند.
- شرطهاي پرولوگ از دو قسمت سرويدنه تشکيل شده است.  
بدنه از لیست هدفها تشکيل شده که با کاما از هم جدا شده اند.  
کاما به معنی اتصال است.

سررايی حدا هم سرائی

برنهام می تواند پنده صفت داشته باشد:  
براساس حسین کانون هایی می توان بزرگ را تعریف نمود. (البته والدین را که  
کلاه X والدین را که اسک آسر بود اند ولا باشد و لا والر 2 باشد.

حالا (مندوه و مادان و عزیز) parent+ :- (من و عزیز) grandparent



از قوانین در بروولوگ می توانیم برای تعریف روابط پیچیده استفاده کنیم مثل:  
برای تعریف پدر بزرگ بصورت زیر تعریف می کنیم:

$\text{grandparent}(X, Z) :- \text{parent}(X, Y), \text{parent}(Y, Z).$



تمثیل: از قوانین در بروولوگ برای تعریف روابط پدر، مادر، خواهر، برادر، جد  
بدری، جد مادری و ... استفاده کنید.

حالاً Fact هاو relation هارا نویسیم تا بدهیم  
 چه نویسیم. پرلوو می تواند  $\exists$  چنین داشته باشد  
 predicate و relation ها و Fact ها و موالات.

- ① قوانین
- ② موالات
- ③ مطالعات

در اینجا Fact ها صحبت کردم آنکه رابطه باعوای صرسی نزدیم.  
 اولین خط از اینها: رابطه فرزند بورن را تعریف می کنیم

تعریف روابط توسط قوانین

فصل اول

فرض کنید بخواهیم رابطه offspring را بعنوان عکس رابطه parent تعیین کنیم اینکار بصورت زیر انجام میگیرد:

offspring(Y,X) :- parent(X,Y).

به این عبارت که يك قانون گفته میشود بدین معنی است:  
 برای تمام X ها و Y ها، Y offspring X است اگر X يك parent Y باشد

هر قانون از دو قسمت تشکیل شده سر (عبارت سمت راست) و بدنه (عبارت سمت چپ) است.

offspring(Y,X) :- parent(X,Y).

سر

بدنه

راست

ما اینجا در در در مفهوم تواند چه کسی والدی کنی ایست که داده ای داشته باشد  
 حاصل نشان دادم ماه فرزند mom است. البته اینها مفهوم والدین را نویسیم و می نوشود جزوی تربیت مخصوص نوون هاری یا بدری هم برداشت  
 اینها کوئی برای هم این مفهوم مخصوص نوون را بغير فرزندی Fact وارد نمی کنیم زیاده ساده می شود.  
 من تو را نه کنی خانم با ۱۴۲ بتوییم در دیاسین خودیان

اُس پاسئی باشد که سرچ دخدر زیرین لیست  $DL$  می دارد.  
ا هدف از  $\tilde{P}$  به راست برسد و نشود.

آرگومونهای روابط می توانند اشیاء ثابت مثل  $tom$  و  $bob$  (با حروف کوچک) و یا متغیر مثل  $X$  و  $Y$  (با حروف بزرگ) باشند.

سوالاتی که از پرولوگ پرسیده میشود می توان از يك هدف و یا چند هدف تشکیل شده باشد

چند هدف مانند سوال زیر که در اینجا "،" بمعنی اتصال هدفهای  $parent(X, ann), parent(X, pat).$

یک پاسخ به سوال بسته به اینکه هدف مورد نظر قابل دستیابی باشد یا نه میتوان مثبت یا منفی باشد

اگر پاسخ مثبت باشد می گوییم هدف مورد نظر قابل دستیابی است و در غیر اینصورت قابل دستیابی نیست

اگر چندین جواب وجود داشته باشد پرولوگ با درخواست کاربر (را وارد کردن ) آنها را پیدا خواهد کرد.

## ( لزرو نمی تواند بخوبی خواهد )

سوال زیر بدین معنی است که والد مشترک  $ann$  و  $pat$  چه کسی است؟  
?-  $\text{parent}(X, ann)$ ,  $\text{parent}(X, pat)$ .

جواب پرولوگ چنین است:

$X=bob$

خلاصه:  
در پرولوگ تعریف یک رابطه مانند از قوانین در پرولوگ می توانیم برای  
تعریف روابط پیچیده استفاده کنیم مثلا برای تعریف بذر بزرگ بصورت زیر  
تعریف می کنیم:

parent می تواند از پرولوگ در باره رابطه های تعریف شده در برنامه  
به سادگی و با مشخص کردن اشیاء رابطه انجام میگیرد.  
کاربر می تواند از پرولوگ در باره رابطه های تعریف شده در برنامه

سوال نماید  
یک برنامه پرولوگ از جند عبارت تشکیل شده و هر عبارت با نقطه پایان  
میاند.

سوال پیچیده تر نیز که برای زبان باشد و لئن که داده و الدعیم باشد یعنی برای زیر جمله و  $\rightarrow$  و  $\leftarrow$  and (یا ادارنده باید برگردان) برای این اطلاعات (الایندر ۶) هم می‌تواند برگردان را ببرهد و هم خارج برگردان را برای این اس KB دنبال می‌کند و یا همیشه در دو انتها باشد و هم و هم و الداعی باشد که هر چیز بالاتر از  $\text{mij}$

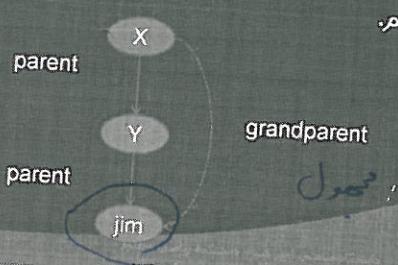
می‌توان سوالات پیچیده تری نیز بررسید مثلاً:  
?- parent(Y, jim), parent(X, Y).

يعني X و Y را طوری پیدا کن که دورابطه بالا همزمات برقرار باشد

X=bob

Y=pat

در واقع ما بدون تعریف مستقیم پدر بزرگ ما از پرولوگ پدر و پدر بزرگ را بررسیدیم.



در اینجا برگردان یادگیری معرفی می‌شود. من کوایم به هر دو نیز هم مطرح نمود  
هم کسی را معرفی نهاده هم داده Ann باشد و هم داده pat باشد  
(مفعول بعد) یعنی داده است که pat و Ann داده است

? - parent(X, ann), parent(X, pat).

X=bob

لعن مهندسی داده‌سنجی و داده‌گردانی دهد و همین طور خرمندان را،  
(اسناد حیاتیه منشأ درست از امیر طرد)

?- parent(X, Y).

اگر سوال زیر را بپرسیم:

جوابهای برو لوگ به ترتیب به صورت زیر است:

1 } X=pam

Y=bob;

2 } X=tom

Y=bob;

3 } X=tom

Y=liz;

...

نکته: ما می‌توانیم خروجی جوابها را با وارد کردن Enter بجای متوقف کنیم.

آئر پرولوں چند تا جواب داشتے ہیں میں سوال کیا تھا والد (parent) کیوں چہ کسی والد نہیں اس تھا۔ یعنی ہم والدھارا کی خواہیم، و چند تا جواب کی خواہیم یعنی ہم لا مجموع اس تھا و ہم لا محظوظ اس تھا۔ آئا پرولوں نے تو صہ بہ اولین fact در B کا نہایت مدد و مدد برائی میں صوابی رکھ رہا تھا؟ و اس ترا صوابی بعدی رامی دھر دھر،

؟ حنای ۲۵ رابہ طائف دھر.

سوال زیر بہ این معنی است کہ فرنڈان bob چہ کسانی ہستند:  
?- parent(bob, X).

X=ann

جواب پرولوگ بصورت زیر است:

حال اگر فرنڈن بعدی را بخواہیم کافیست؛ وارد کنیم

X=pat

جواب بعدی پرولوگ

no

با وارد کردن یک دیگر پرولوگ جواب میدهد

مسموم کردی باب چہ لسانی رابہ عفران خریز ندارد  
در KB باب دو فرزند رارد آنما و Pat۔ هر دو برابر بود.  
پرولوں براساس ترتیب B کا جواب را نہیں سمجھدی، و داردی نہیں  
و حرباب مسموں عباری رابہ صادق دھر و دسر حربابی نہیں دھر دھر.

کیمی دلخواه در حسابات معمول می‌گذرد. معمول می‌گذرد که آن باشد پاسخ می‌ردد در غیر این صورت  
نمی‌تواند ۵۰٪

سوال جالب دلخواه نه حسنه بروز است از بروز این دست در حسنه والدیت است. این دست در حسنه والدیت است.

خط اول . معنی آن نه رامی خواهیم در بینیم چه کسی هست؟  
پاسخ طهط است.

\* بروز نست به حروف حساس است.

سوالات جالب دیگری که از پرولوگ میتوان بررسید چه کسی والد pat است؟ سوال به شکل زیر بررسیده میشود:

?- parent(X, pat).

X=bob

جواب پرولوگ بصورت زیر است:

و با اینکه چه کسی فرزند pat است به صورت زیر:  
?- parent(pat, X).

X=jim

جواب پرولوگ

نکته: پرولوگ نسبت به حروف حساس است. ثابتها با حروف کوچک و متغیرها با حروف بزرگ شروع میشوند.

برآمده است

pat والد هم است (برآمده) اینها relation است.  
برآمده بروز صواب داده به سوال از بالا به پاسنی داشته و حکم ممنونه parent بوده است.

تابع برآمده یک object وابسته object هم بود. (یک کلمه است)

relation از قدری object هست true ؟ اینجا نه است (یک جمله است)

sim دلخواه دلخواهی بسیار دلخواه بسیار دلخواه

- آنده جواب دادن به سوالات (عن) برو لوگ آنده است.  
 آن خط اول یعنی آیا باب والد هست ؟ نه اسن عصر فنوری  
 و ازایی است. ترتیب جملات ترتیب حتم است.

و قی ای برنامه با برو لوگ ارتباط برقرار کند از برو لوگ میتوان سوالات زیر  
 بررسید مثلا ایا bob والد pat است؟ سوال به شکل زیر بررسیده

?- parent(bob, pat). میشود:

Yes

جواب برو لوگ بصورت زیر است:

?- parent(liz, pat). برای سوال زیر:

No.

جواب برو لوگ

زیرا برو لوگ چیزی درباره این که liz والد pat است نگفته است.  
نکته: برو لوگ در جهان بسیه عمل میکند یعنی به هر آنچه که در باگاه  
 دانش خود ندارد جواب no میدهد.

Liz و pat اربابی باهم نمایند. (راهنمای pat است) آن در B است  
 استاد ففت از فلتون باش محمد احمد باشند !!!

آن برسن نازینه همارتی برابر pat است مهندس (صحرای طما) !!

خلاصه نویسیده هر کسی را محترم است. چون شویم محمد است.

Fact ها و مفاسد رایا ساخت ارسطو هریه اول می نویسیم

کام پیر باب است . (رابطه والد بودن بین تاکادهاب) والد آخر  
برت مذکور هم کابل (نهانگردی است) .

در نویت دیده من نویسیم و با سیورندا می کنیم به سورت آمریک مضر پرولوگ  
آنرا اجرا کند در ماره اجرا می شود . و مس آماره اجرا شد من سوالاتی از آن مطرح کرد.

این حقیقت که tom والد bob است را میتوان در پرولوگ به صورت زیر

parent(tom, bob).

نوشت:

parent را در اینجا بعنوان یک رابطه بین tom و bob (رابطه والدی)  
تعريف میکنیم.

کل روابط والدی درخت بالا بصورت زیر تعریف می شوند:

parent(pam, bob).

parent(tom, bob).

parent(tom, liz).

parent(bob, ann).

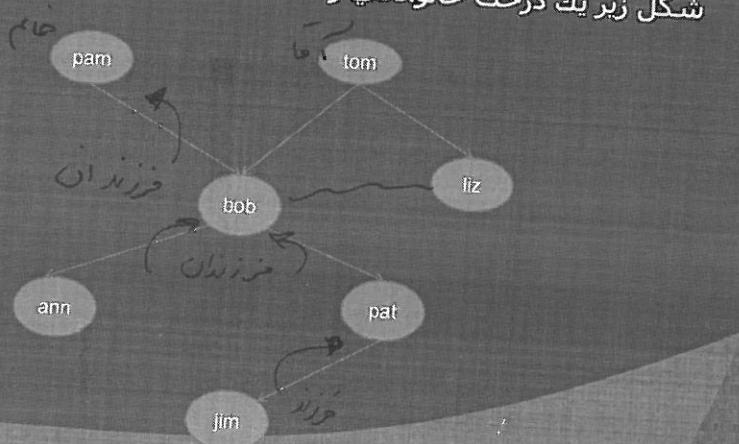
parent(bob, pat).

parent(pat, jim).

فصل اول

معرفی روابط توسط واقعیتها

شکل زیر یک درخت خانوادگی را نشان میدهد



پریلوگ زبان حوس مفسری باست برای حل مسائل حوس مفسری دنیا است  
و در زبان مدلول مفسر ریزن زبان است.

وظیفه مفسر چنان که ممکن است  
هر زبانی طایبا نیز دارد. آن سیم پریلوگ خوب است از جمی رعد.

### مقدمه ای بر برولوگ

#### فصل اول

برولوگ یک زبان برنامه نویسی برای حل مسائل غیر عددی و سمبولیک است و مناسب مسائلی است که با اشیاء سر و کار دارد.  
برولوگ از سه قسمت پایگاه دانش، مفسر، و رابط کاربر تشکیل شده است

پایگاه دانش شامل حقیقت ها و قوانین است که بر اساس قوانین منطق مرتبه اول بیان می شوند، می باشد.  
وظیفه مفسر تهیه جواب به پرسش کاربر بر اساس این دانش است.  
رابط کاربر محل ارتباط کاربر با سیستم است. کاربر پرسش خود را مطرح و سیستم برولوگ به آن جواب میدهد.

## مقدمه

- ❖ فصل 7 : گزاره های تعبیه شده بیشتر
- ❖ فصل 8: شیوه و روش برنامه نویسی
- ❖ فصل 9: اعمال روی ساختمان داده ها

## مقدمه

حداقل فصلهای زیر را میخوانیم:

❖ فصل 1: مقدمه ای بر پرولوگ

❖ فصل 2: ساختار دستوری

❖ فصل 3: لیستها، عملگرها، حساب

❖ فصل 4: استفاده از ساختارها

❖ فصل 5: کنترل ردیابی شاخه ای

❖ فصل 6: ورودی و خروجی

## مقدمه

كتب مرجع :

- Prolog programming for artificial intelligent, 3<sup>rd</sup> ed.

هوش مصنوعی و برنامه نویسی پرولوگ  
نویسنده: ایوان برتوکو  
مترجم: رامین مولاناپور

# برنامه نویسی Prolog

محمد ابراهیم شیری

دانشگاه صنعتی امیرکبیر - دانشکده ریاضی و علوم کامپیوتر

بهار ۹۱-۹۲

هر ساختار که اجرای شده باشد Functor دارد و هدو tail به عنوان اعماقی لیست هستند در در داخل پریسٹ هر کدامیکی نیز head رأس لیست است و tail دنباله آن است

همچنان که گفتیم لیست خود یک ساختار است ولذا باید بتوانیم آنرا نیز در قالب یک ساختار بنویسیم.

لیست میتواند به عنوان ترکیبی از دو چیز در نظر گرفته شود:

1- بخش اول که شر لیست نامیده میشود (Head)

2- بخش باقیمانده که دنباله لیست نامیده میشود (Tail)

در این صورت لیست را در قالب یک ساختار به شکل زیر نمایش میدهیم:  
(Head, Tail)

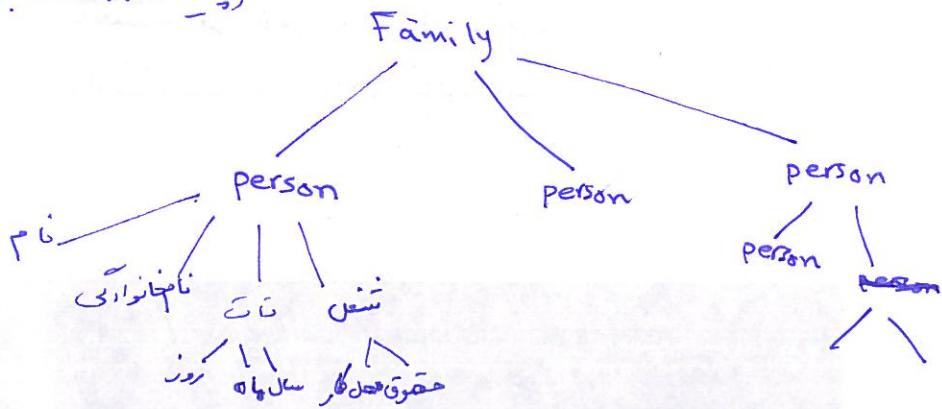
که ". در اینجا نقش اجرا کننده را بازی میکند.

عموما Head میتواند هر شئ پرولوگ برای مثال یک اتم یا یک متغیر و یا هر شئ دیگر باشد.

دنباله یا Tail خود باید یک است باشد (میتواند خالی و یا فقط یک عنصر داشته باشد)

دیا صند محضر

(پاہنا دارہ استاجر)



family (person ( ) , person ( ) , [ ] )

بے لین صورت میں شدہ اطلاعات سر بوط بے ظنوارہ رائسر ری  
سوال: مطلوب ہے نام و نام خانوادی ظمہ ہیچ خانہ رکارہ حداہم می فریز رازند  
دریں جا چون شوہر ہم نے (—) مہ لائلوم۔

? - family ( -> person ( N, F, — , unemployed ( ) , [ ] ) , [ ] )

سوال: سانکر بالائی ۳۰ سال ہستے (سرویز پسے بر سرال) (انجاماتمہ)

? - Family ( -> person ( N, F, date ( — , — , ۶۰ ) , [ ] ) , [ ] )

[ — , — , [ — ] ) y < 1984

سوال: کھایاں کے دریاں ٹھارات طرفی لئے جی

? - family ( person ( N, F, — , work ( — , عمل کار ) , [ ] ) , [ ] )  
لے یاںک ٹھارات ملا

نحوه نیش ابرائیتده را در این

لیست

• (head, tail)

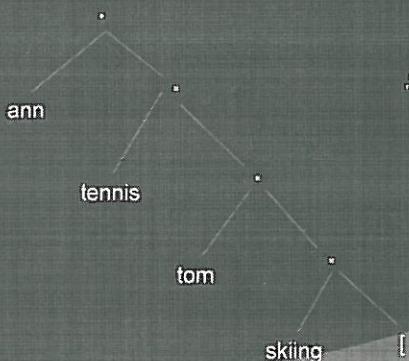
↓  
اگر این ترتیب  
• (ann, [ ])

• (ann, .(tennis, [ ]))

بنابراین طوری که در این شیوه نمایش لیست قبلی به صورت زیرهم  
نمایش داده میشود:

(ann, .(tennis, .(tom, .(skiing, [ ]))))

که در این صورت نمایش درختی این لیست به شکل زیر است:



سوان ۳: حقیقی بالا در مدلینو حسنه طبقه و ...

? - family (person (N, F, -, works (-, S)), person (Fateme  
-, -, -, -, 2000))

اعضاده از utility ها:

(برای ساده تر نمودن کار) این utility ها در پایه دسن انسان می شوند

husband (x): - family (x, -, -) بیکاری مرد

wife (x): - family (-, x, -) طلاق از همسر

child (x): - family (-, -, children), member (x, child)  
این هدف آن است که از زیربنای children

date of birth day (person (-, -, D, -), D). تاریخ تولد

salary (person (-, -, -, unemployed), 0).

Salary (person (-, -, -, work (-, S)), S).

حقوق پردازندگی برای کارکرده و خود را در می بیند و می خواهد. (کارکرده شرکت را در فریز نماید)

و خیریگ شمعه هر سه درین بازن و سوابست یا بحیث است. (کارکرده شرکت را در فریز نماید)

and or و یا مدرده شود)

exist (person): - husband (person); wife (person);

child (person).

ظاهره ای که درین عرصه از راه رفته و بین

? - wife (person (N, F, date (-, -, 1994, unemployed)))

- توان مردی در بازن یا فرزند

\* javan (person): - exist (person), date birth (person), date

(-, -, 1994))

تعریف شفته جوان

برولوگ از یک روش دیگر نیز برای نمایش یک لیست استفاده میکند و آن با استفاده از نوار عمودی است که سر را از دنباله در لیست را جدا میکند مانند شکل زیر:

$L=[\text{Head} \mid \text{Tail}]$

با این نمایش لیست مثال ما به شکل زیر در میاد:  
head tail  
[ann | [tennis, tom, skiing]]

با این نمایش یک لیست با راههای مختلف میتوان نوشت برای مثال:

$[a,b,c] = [ a \mid [b,c] ] = [ a, b \mid [c] ] = [a,b,c \mid []]$

نکته: در زبان برنامه نویسی برولوگ از هر سه روش بالا برای نمایش یک لیست استفاده میشود.

جلسہ ۱۲ :

سوال آئندہ می خواہم کہ سب جس سال پہلے (رامکم زیر لائے)

? - husband (person), date of birth day (person, date (-, -, - و (۱۹۵۴، salary (person, -، -)، (S, F, -

مسائل بازستہ نئی میقوعہ کم میں زندگی درہم می خرد نہ استیشون دمر

درین رہارس سری ہاعہ توان بسوالات مختلف پاسخ داد (ابنہ بازستہ ایام تعریف کر

سوال : خانہ میں کہ سوچاں حوالہ دارند؟ (سوچنے کا سچھوہ باغنا انتہا لئے)

سوال : شوہر بی کاربین عار - خانہ مشتعل

? - family (person (N, F, -)، unemployed), person (-, -, -، work (-, -، -، -)

سوال مجموع درآمد خانوارہ ۲۰ ملیون باشد صلا

پیدا در KB کے total برابر مجموع درآمد تعریف می گئی

total ([ ], 0).

total ([ person | tail ], sum ) :- salary (person, S),

total (list, rest), sum is S + rest.

مجموع درآمد

کے خانوارہ

مجموع درآمد

- خط اول سفرت بتوقف . لیست خالی بنشود

برای مجموع درآمد salary فردو بقیہ افراد

? - family (husband, wife, children) [ بقیہ بھی هم بعنایت لیست ]

(77) total ([ husband, wife | children]), income),  
income > 20, -,-

مثال برنامه نویسی:

Hobbies1=.(tennis,.(music,[ ])),

Hobbies2=[skiing, food],

L=[ann,Hobbies1, tom, Hobbies2].

L=[ann, [ tennis, music],tom,[ skiing,food] ]

جلسہ 12 :

دھنیفم جبل children زبرہ سرورت لست در تظر دریتم .

چون total باللیت کارچن لندنگ حقوقی ه را با چم جمع و لز

پڑھنا ہیں children لیت با شد و نہ ععنو و نک تک آنا رامروڈیا بہ  
حراریم = ( میڈاریم tail )

جلسہ 13 باریخ 16.10.93.

اساد در مرور مبایسٹ متعلق درس ھی یا 5 سرور  
و رفع اسٹکل ردنڈوں نے تو ستم .

### فصل سوم

#### برخی عملیات روی لیستها

لیست ها را میتوان مانند مجموعه ها در نظر گرفت با این تفاوت که در مجموعه ها ترتیب عناصر مهم نیست و یک عنصر نمیتواند تکرار شود ولی ترتیب عناصر در لیست مهم است و یک شی در لیست میتواند تکرار شود.

همانند مجموعه ها یک سری عملیات روی لیستها قابل تعریف است مانند بررسی این که عنصری در لیست است یا نه، حذف عنصری و یا اضافه کردن عنصر جدید به لیست، الحاق دو لیست و یا عملیات دیگر.

در اینجا بعضی از این عملیات را بررسی میکنیم.

جلسه ۱۴

جلسه چهاردهم . ۹۳. ۱۰. ۲۳

کنسل ریاضی شاخص ای خودکار

## برنامه نویسی Prolog

محمد ابراهیم شیری

دانشگاه صنعتی امیرکبیر - دانشکده ریاضی و علوم کامپیوتر

بهار ۹۱-۹۲

۱

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

اگر لیست داشته باشیم که هر دو عضوی از باشد و سریم آنها معنی آن است  
 یعنی همین حقیقتی true است.

یعنی اگر لیست داشته که هر دو عضوی از باشد و سریم آنها معنی آن است. نزدیک  
 میتوان یک مختصت tree را نیز در میم.

اگر هر یادگیری بنا شود (اسعاده در تطابق) یعنی هر حد بنا شود (حد راهبردی کریم)  
 آن وقت برای اینکه عضوی از لیست باشد باید عضوی از tail باشد

## ۵ عضویت

برنامه رابطه عضویت می تواند طبق الگوی زیر باشد:

X عضوی از لیست L است اگر

-1 X سر باشد و یا

-2 X عضوی از دنباله L باشد.

اینها را میتوان بصورت زیر نوشت:

`member(X, [X | Tail]).`

`member(X, [Head | Tail]) :- member(X, Tail).`

در اینصورت هدف (L , member(X , L)) وقتی درست است که X در L نخ

داده باشد برای مثال:

`member(b, [a, b, c]).`

`member([b, c], [a, [b,c]]).`

درین صورت وقتی همچندی از tail بودن هم عضو از list است

پس باید هر کدام عضویت یک عضو در لیست همین روش طبق کافی است

(به اینصورت باز نمی شود) اما اخوه به جاین برسد که هر جزءی از  
 حکم دوم هدفها را صادر کند به اینصورت باز نمی شود.

حالا اگر سوال کنیم کیم آیا ط عضویت L و D است یا نمی خواهد که لا مجبوب هر دو

اگر سریم C ط عضویت L و D است باز هم تولید yes است

## کنترل ریدیابی شاخصه‌ای

## عمل نیجم

پرولوگ در صورت لزوم برای برآورده کردن هدف، عمل ریدیابی شاخصه‌ای را انجام می‌دهد.

ریدیابی شاخصه‌ای خودکار یک مفهوم مهم در برنامه نویسی است جراحت:

۱. برنامه نویس از سختی برنامه نویسی ریدیابی شاخصه‌ای معاف می‌شود.
۲. ریدیابی شاخصه‌ای کنترل نشده ممکن است باعث عدم کارایی شود.

با استفاده از امکان "برش" در پرولوگ می‌توانیم ریدیابی شاخصه‌ای را کنترل نماییم.

2

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

اما ابرهیم<sup>آ</sup> با عنوانیت [L و L] هست جواب من در زنده (بود) هست اما  
که نیست:  
۴- مکالمه الحال:

خط اول: یعنی آر لیست خانی را با این پایه تغییری به جزء L به مابهود (قطعه پایه هد) آر لیست اول آنها باشد حالا آر دولیست داشته باشیم لیست اول حدداست داشت اما است ولیست درم . حلقه تعریف باید لیست بر هدنه هدآن هاست چون و لایت اول آنها من شود تن لیست هر بر ما حدآن است هاست را؛ ta؛ دخواست و ملایم طوری دست میگردید از آنها ناد لایت دلایلی بروت من آنها.

درست می باشند ولی هدف زیر درست نیست

`Member(b, [a,(b ,c )]).`

### الحال

عملگر الحال باید طوری تعریف شود که لیستی را به انتهای لیست دیگر  
وصل کند بنابراین این عملگر سه پارامتر ورودی میگیرد L1 و L2  
بعنوان دو لیست ورودی و L3 بعنوان الحال شده دو لیست (در صورتی  
که بصورت متغیر وارد شود پرولوگ بعنوان جواب اثرا محاسبه میکند)  
عملگر الحال را بنابراین بر اساس ۲ قانون مبنوایم تعریف کنیم حالتي که  
لیست اول تهی است در که در این حالت باید لیست دوم و سوم  
یکسان باشند.

و حالتي که لیست اول تهی نیست که در تعریف مشخص شده

عملگر الحال را بنابراین بصورت زیر تعریف میکنیم:

`conc([], L,L).`

`conc([X | L1] , L2, [X | L3]):-conc(L1, L2, L3).`

برنامه صریح باز نیست احتمال شود.

سرطا  
ترفع

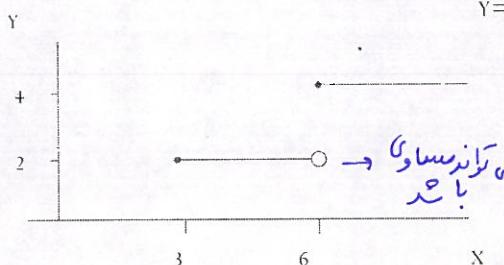
تابع دو مرحله اي زير را در نظر بگيريد که در ان رابطه بين  $X$  و  $Y$  را مي توان با سه

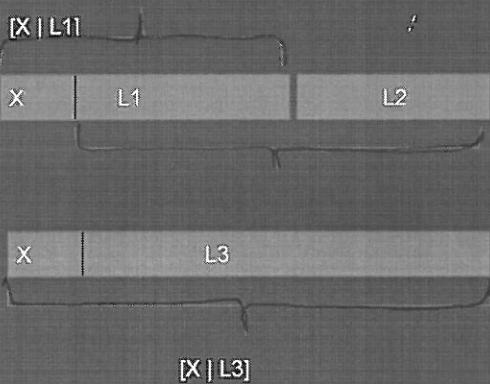
قانون زير تعبيين كرد:

1- اگر  $X < 3$  باشد آنگاه  $Y=0$

$$y=2 \quad \text{اگر } 3 \leq X \leq 6 \quad \text{و } X < 6 \quad \text{آنگاه } 3 \leq Y < 4$$

2- اگر  $6 \leq X$  باشد آنگاه  $Y=4$





حال این برنامه میتواند لیستها را بهم بیوند دهد. مثال:

?- conc([a,b,c], [1,2,3],L).

L=[a,b,c,1,2,3]

برای انجام اجرایی بزرگتر دستورات را از اول بررسی کنید

در خط اول بهجای  $\neq$   $\leftarrow$   $\rightarrow$  قرار می‌دهد. زیرا شود  $\neq$ .

چون نه از ۲ کو کمتر است و بسته ممکن است از طرفی لا باشد  $\neq$   $\leftarrow$   $\rightarrow$   $\neq$ . هم نه از ۳ کو کمتر است. نه مقصه‌ی نشود نتیجه می‌دهد که لا باشد برابر باشد. همان شرط  $\neq$  هم می‌خواهیم این دو شرط را یا هم برقراری کنیم می‌باشد.

برنامه را به صورت زیر مبنویسیم:

$$F(X,0) : - X < 3.$$

$$\rightarrow 3 = < 4$$

$$F(X,2) : - 3 \leq X, X < 6.$$

$$F(X,4) : - 6 \leq X.$$

دو آزمایش که عدم کارانی این برنامه را نشان میدهد ارائه میدهیم:

آزمایش ۱: با مطرح شدن سوال زیر پرولوگ چگونه عمل میکند

$$\leftarrow (1,Y), 2 < Y. \quad (۱) \text{ به شرطی که } 2 < Y \text{ باشد}$$

- اول هدف  $(1,Y)$  اجرا میشود و ۲ با صفر نمونه سازی میشود که در این صورت دو میں

هدف  $< 0$  میشود که مردود است.

- پرولوگ همچنانز طریق ردیابی شاخه‌ای، دو راه بی استفاده دیگر را امتحان میکند که

نیازی به بررسی انها نمیباشد ولی پرولوگ از این امر بی خبر است.

دو شتاب زیر در لیستهای دو طبقه داشتند. این با لیست آن و آن دو طبقه می‌شود. پس لیستی می‌شود. (لگاریتمی عضو را مقبول نمی‌کند) (تریب راهنمایی نیز برای این دو شتاب داشتند)

خط ۳ کد زیر:  $\text{conc}([L1, L2]) \rightarrow [L1 \cup L2]$  این است (زیر مجموعه مجموعه باشد)

اگر  $L1$  و  $L2$  که همچنان مجموعه هستند این است را ایجاد کنید لیست  $L$  است

باشد.

در خط ۴ کوچکترین مجموعه برای مجموعه می‌شوند و هر دوی این توان توانست

می‌توان زیرخط ۴ را اینکه  $\text{conc}(L1, L2) \rightarrow [L1 \cup L2]$  هستند مجموعه می‌شوند این باید و بود داشته باشند

?-  $\text{conc}([a, [b, c], d], [a, [ ], b], L).$

$L = [a, [b, c], d, a, [ ], b]$

نوشتند فجذب برنامه عملگر عضویت با استفاده از عملگر الحال به صورت زیر:

$\text{member}(X, L) :- \text{conc}(L1, [X | L2], L).$

این برنامه می‌گوید  $X$  یک عضوی از لیست  $L$  است اگر  $L$  بتواند به دو لیست تجزیه شود بطوریکه لیست دومی دارای سر  $X$  باشد.

برای بررسی اینکه آیا  $X$  عضوی از لیست  $L$  است یا خیر:

ابتدا بررسی میکنیم که آیا سر  $L$  مساوی  $X$  است یا نه و سپس

بررسی میکنیم که  $X$  عضوی از دنباله  $L$  است یا خیر

بطور ساده تر این برنامه را به شکل زیر هم میتوان نوشت:

$\text{member}(X, L) :- \text{conc}([ ], [X | L], L).$

الآن تو خانم هنرمند حم خواهد بود !!!

اما تو خانم هنرمند ساین حم بدارن !!!

اینها به نظر اساتید و معلم روز عطیل اورده رانشانه هم این هم مسئولیت نیز

آتایی ست یعنی هم اورده شدند !!! (لیکن بورج اساتید را تحسین نمی‌زنند)

النفع عدم گروهی نوبه و انتها

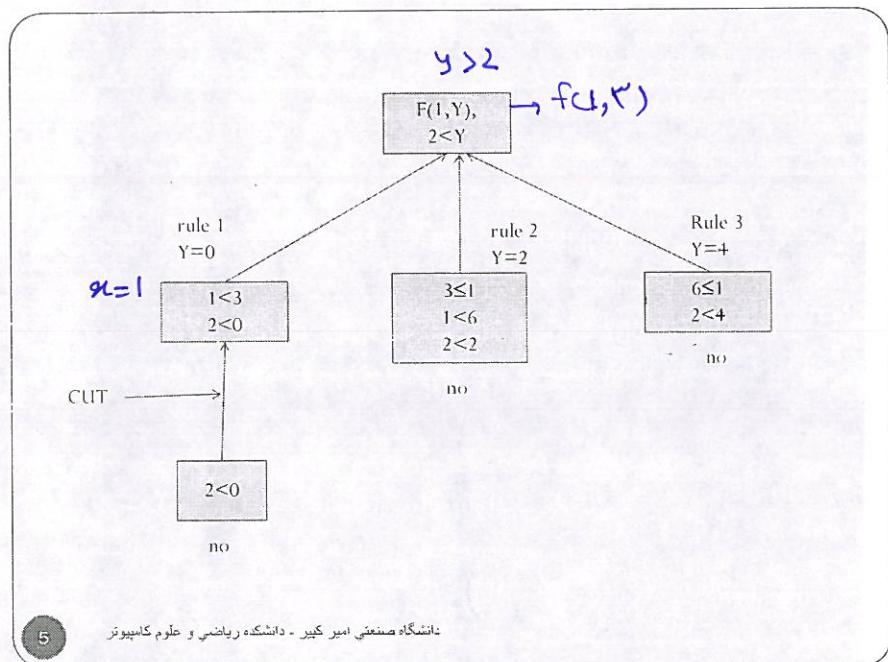
93.10.23 - 2:03 نامه (راهنما و محسن):

نمایابی مقدار دهنده می کنیم و همچنان می کنیم

دستور اول  $f(0,0)$  به جای  $\text{no}$   $\leftarrow$  ۱ مراری داشتم  $y=0$  درست

است هرمان  $y > 2$  هم باید باشد این دو نا باهم شائقرن است پس جواب  $\text{no}$  می دهد.

حالا شرط دوم را بررسی می کنیم به جای  $\text{no}$   $\leftarrow$  ۱ مراری داشتم  $y < 4$  صادر است. همچنان به سرانجام شرط سوم می برمی.



دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

به جای  $\text{no}$   $\leftarrow$  ۱ مراری داشتم. جون باید  $y > 2$  باشد از نظر برقراری است دست کنید. شرط اول را چک کردم و به جای  $\text{no}$   $\leftarrow$  ۱ لذاشتم. برقرار بودن جواب  $\text{no}$  داد. همچنان دوچوری می خواهیم به سرگذشت بلوکیم نه دو شرط دیگر را چک نماییم؟ هر آنچه در طبقه برآمده است از ۳ است  $\leftarrow$  ۱ است. ولی سهلا سرگذشت چک می نماییم خودهم همچنان می کند. و برخانم هرمان برمی شود

کاربر لیست افنا خواهد کنید تا بتواند برسد مقداری در حدود

مقدار داده و tail آن را می شود پس این عضو همیشہ به انتباری لیست افنا خواهد شد.

Add(X,L,[X|L]).

(قدرت استاد گریف هسن نمی خواهد رفعی بین، جای انتظار را راهنمایی برداشته باشد)

فعلاً بده لیست افنا کنید سایر بیرون از tail به جای این رفعی حم اگر ده مثلاً حرف سریع است.

اضافه کردن یک ایتم به لیست

برای اضافه کردن یک ایتم جدید به لیست به آسانی میتوان اثرا به ابتدای لیست اضافه نمود:

Add(X,L,[X|L]).

حذف یک ایتم از لیست

عملگر حذف یک ایتم از لیست میتواند مانند عملگر عضویت باشد، مجدداً دو حالت داریم:

اگر X سر لیست باشد، نتیجه پس از حذف X دنباله لیست خواهد شد.

اگر X در دنباله باشد، بنابراین از انجا حذف میشود  
این عملگر به شکل زیر تعریف شود:

del(X,[X|Tail],Tail).

del(X,[Y|Tail],[Y|Tail1]) :- del(X,Tail,Tail1).

هدف اینست: آن را عضوی در لیست به نام آور داریم این را حذف کنیم. (2) حالت داریم

59) اگر آن و ماسریت باشد می تواند tail را بمحفوظه داشته باشد. (خط اول  $\leftarrow \rightarrow$ )  
~~del(X,[Y|Tail],[Y|Tail1]) :- del(X,Tail,Tail1).~~

اگر همانند آن را از لیست حذف کنیم به حدود ۵ و ۶ است بنابراین می باید آن را حذف شود پس حذف آن را باز نمی خواهیم داشت. لیست بده به  $\leftarrow \rightarrow$  آن و بایسند. حذف آن همچنان علی ایست و tail پس از tail یک از حذف آن مقدمه است، درست شد.

با میدیر برنامه اس نویسیم و طریق دلخیم نه این ترتیب برقرار نبوده باشد سرانجام بقیه نزدیک در پروگرام نویسیم  $\text{Cut}$ . همان‌ها هم دلخیم بقیه را هم کنند هم نویسیم اگر مصدق کرد  $f(x, 0)$  به شرطی که قدر بود دستور سرانجام بقیه نزدیک و کات کن والسلام (کات کن رید) (الشخصیت راسته باش) اگر آنها و کات بود و این ترتیب برقرار شد سرانجام بعدی نزدیک رشکل بررس و نفعه در نهایتاً اگر  $f(a)$  در حالت ~~هست~~ بود بعدی و  $a$  برابری حالت

نتیجه: پرولوگ از طریق ردیابی شاخه ای راه های بیهوده را بررسی میکند. برای اینکه پرولوگ ردیابی شاخه ای را انجام ندهد این کارا را میتوانیم با استفاده از مکانیزم برش انجام دهیم.

'برش' که با علامت ! نشان داده به عنوان یک شبیه هدف در جای مناسب بین اهداف اضافه میشود. در زیر برنامه قبل با استفاده از برش بازنویسی شده:

بررسی دهنده  $\leftarrow$  آماری دلخیم  $\leftarrow$  دستورات  $\leftarrow$  بهای

$F(N, 0)$ : :-  $X < 3, !.$

$F(N, 2)$ : :-  $3 = < X, X < 6, !.$

$F(N, 4)$ : :-  $6 = < X.$

با توجه با ظاهر شدن زیر هدف ! در جاهانی که در برنامه امده پرولوگ از ردیابی شاخه ای جلوگیری میکند. (برنامه کارا ترجیح نشود).

حال اگر سوال زیر را پرسیم:

?-  $f(1, X), 2 < Y.$

6

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

هایی که همان بزرگتر نیاز نداشتم با این آخری برش نمی خواهد  
قانونی نیست بروز سرانجام بعدی (بعدی و عدو دنیار) من کات نمی کنم (جذب)  
نه، خوب لازم نهان ساره ترتیب  $f(a, b)$   $\leftarrow$   $a < 3, !.$   
 $f(a, b)$   $\leftarrow$   $a = < b, !.$

—  $f(a, b)$   $\leftarrow$  دستور نیازی ترتیب  
را بمنو نویسیم همان برنامه را ساره ترمی نویسیم

مباهث اسلامیه آمده : اعدامن بواره head من هم باشد در این سورت که tail را می دهد. و اگر هدمو نباشند و لا باشد مصلحه این هم از tail حذف شود پس هم بدیری من دعوه نونام دارد  $\rightarrow$  tail حامل حذف هم باز است اما حالات امثال :  $\rightarrow$  را لازمه لجه و طبقه هذف کن . سایر لجه و طبقه هم نمود. حالات امثال مذکور را مشاهده کن  $\rightarrow$  اول  $\leftarrow$  است زیرا ترین  $\rightarrow$  را هذف می کند آنرا بعده بمه آبر  $\rightarrow$  بعدی را هذف می کند و همین طورتا آخرین  $\rightarrow$  و هم آنرا آنرا قبیری  $\rightarrow$  در صورت این که دلخواه کن (باها) دو هر اجراء فقط یک را هذف می کند رعایت داشته باشید در هر برای  $\rightarrow$  را هذف می کند

### مثال

?- del(a, [a ,b , c , d], L).

L=[b, c, d]

نکته : عملگر del غیر قطعی می باشد یعنی اگر چند رویداد X در لیست باشد این عملگر قادر خواهد بود با اجرای متعدد آنها را حذف کند. در هر اجرا فقط یکی حذف میشود مانند مثال زیر:

?- del(a,[ a, b, a ,a, c] ,L).

L=[ b ,a ,a ,c];

L=[a,b, a,c];

L=[a,b,a,c];

no

حذف آنرا

این دلیل آن است که وقتی می خواهیم محتویات اول بسود بگیریم بود و باز بعدی بسود بگیریم این را هم نمی توانیم سری را هذف کن (برای حذف هم باز برنامه نویسی که سورت باز نمی شود) برای این نوشت که هم  $\rightarrow$  ها هذف نشود. جواب  $\rightarrow$  هم با این معنی است اما  $\rightarrow$  محتویات نمی توانیم باز حذف کنیم (این نویسندگان را در نظر نداشتم) (این نویسندگان را در نظر نداشتم)

(60)

(هذف آن در بعدن بود در tail + عمل حذف را نیام می داشت)

کی کمی حذف می کند

کلاس مقطع شد استاد متصرف خ زمزد زاده و دو سکان بود کابیان بعد درین بن.

استاد بان خوبی نوبره.  
جلسه ۱۴

خوب بزیرتا در این صورت هم منشد نزدست :

زیده نیازی نیست خط خدا کند چون ~~بیکار~~ و می خواست دوم بود فی کسر

چک می کند که آن را بگذر فقط برای موحای بزرگوار زد (۶۷) (۶۸)

چون مکباه رچک ترد آر نبود ~~مهربان~~ سرانجام بعدی (۱۱۱)

یه می توان بزم امرا ساره تو طارم تریت با استفاده از عمل برس cut

پرولوگ شاخه دست چپ شکل بالا را تولید خواهد کرد. این شاخه در هدف ۰<2> رد خواهد

شد و پرولوگ باشون! شاخه های دیگر را تست نمیکند.

نتیجه: کار ایپی برنامه با افزودن برش بپیو بخشیده میشود.

آزمایش ۲:

فرض کنید بپرسیم:

۷-۱(۷,۲).

### اول ۷۲۳ + رابررس ملند

- امتحان قانون اول: ۷<3> مردود است، برش اجرا نمیشود، پس برگشت.

- امتحان قانون دوم: ۷<۳> قبول است ولی ۷<6> مردود است و لذا برش اجرا نمیشود و برگشت.

۷ بازهم cut

- امتحان قانون سوم: ۷<۶> قبول است. (منشود مستعین به سرانجام مازن ۳ برد)

نکته: این آزمایش هم منبع دیگری از عدم کار ایپی را نشان میدهد.  
مهندسی صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

7

بن آگر نتیط دعم بیکار بتوانیم رور سرانجام نتیط سوم و حریم نمایم رود.

لزد من نوار دبه خط سوم برو در این حالت بروط به هفتم بدل است  
حالا به سوال بالا رسما

وقت داشت - بازیم تریق بلوسن توانیم بسیار هم ایست در دهن

بروکوں از تها صیز که جلسه استفاده می کند خواست بازیم ایست

شده از این استفاده می کند که طرف را درست - ربا این طبق

85

اگر دشی هر جای لیست قدرت

**biggerList**

برای به لیست اضافه کنیم  $\leftarrow$  insert

بزرگتر بسیار  $\leftarrow$  biggerList به مانع دفعه ب طور سه بارز بطریت صرف کنیم به ما List بدهد. با استفاده از حذف هم جا insert کنیم. می توان کرد.

تا زه من کون member راهم با استفاده del نویسی. یعنی هم معرفی و این در اینجا را از لیست حذف کنیم به حین خود بوده.

حذف کرن یک عضو در لیست  $\leftarrow$  این می توان با این del نویسی (تفصیل ۴۶)

می توانیم  $\leftarrow$  حذف کرن این ایتم آنرا هم را از لیست حذف کنیم لیست هم بر بوده

اضافه کردن یک عنصر به هر جای لیست

برای اضافه کردن یک ایتم جدید به هر جایی در لیست تولید یک لیست بزرگتر می کند که به شکل زیر قابل تعریف است :

`isert(X, List, BiggerList):- del(X, BiggerList, List).`

نکته : از ترکیب این عملگرها میتوانیم عملگرهای دیگر را ایجاد کنیم  
برای مثال متوالیم عملگر عضویت را بر اساس عملگر حذف  
بنویسیم مانند قانون زیر:

`member(X, List):- del(X, List, _).`

عملگر زیر لیست

عملگر زیر لیست sublist میتواند وجود یک زیر لیست در یک لیست

بزرگتر را بررسی کند.

add کرن به ابتدا لیست ایتم  $\leftarrow$  insert به هر جای لیست هم کوئن باشد

و را از لیست insert کردم یک لیست بزرگتر به مانع دهد به نام biggerList

این biggerList حظی دارد؟ آنرا هم را از List خارج کنیم دلیل کن حذف لیست

را به می برد. (با برداز  $\leftarrow$  biggerList حذف پیشود) یعنی insert را بر اساس del

ابتدا در قانون اول مشخص میشود که  $3 < x \leq 7$  درست نیست و لذا قانون دوم را بررسی میکند و به هدف  $x = 3$  میرسد که چون  $3 < 3$  است برقرار است حال با توجه به قانون سوم نیازی به چک کردن قسمت دوم قانون دوم نیست و لذا برنامه فوق را میتوانیم به صورت اصلاح کنیم:

$$F(X,0) := X < 3, !.$$

$$F(X,2) := 3 \leq X.$$

$$F(X,4) := \rightarrow \quad \begin{array}{l} \text{شرط نویسیم} \\ \text{مرحله دوست} \end{array}$$

**مرحله دوست در**

سوال: ایا این برنامه خوب عمل میکند؟ ✓

این برنامه ممکن است جندها راه حل تولید کند که برخی از آنها درست نیست، مثلا:

?- f(1,Y).

Y=0;

Y=2;

Y=4;

No

مثال اساتد:

?- insert([a, b, c, d, e], 1, L).  
 L = [a, c, d, e, b];

آخر و بذایم هم حالت حاره ای دارد  
 L = [b, a, c, d, e] فری L = [b, c, a, d, e], ...  
 ؟ - sublist(

~~شکل~~ : sublist برای

برنامه زیر از اینجا  $L_1$  سوند بزیم  $L_2$  شود که  $L_1$  بزیم بزیم بزیم بزیم باشد.  
 درین صورت  $L_2$  است. (آغاز منبعیان بین فاصله های عویض نداشت)

برنامه برو لوگ برای زیر لیست می توان به صورت زیر فرموله شود:

لیست  $L$  میتواند به دو لیست  $L_1$  و  $L_2$  تجزیه شود،

$L_2$  میتواند به دو لیست  $S$  و  $L_3$  تجزیه شود

از عملگر conc میتوانیم برای تجزیه لیستها استفاده کنیم:

sublist(S,L):- conc(L1,L2,L), conc(S,L3,L).

L

L1

S

L3

L2

مح ترا تصریفی باز

5 زیر لیست می است آن را  $L_1$  با  $L_2$  سوند بزیم 1 بده حالا چه خواست?

حاصل سوند 3 و 5 " است  $L_1$  با شروع، می تواند آن را خدای بسیغی و 1 آغاز

رفت کنید بالذاتی برخی مفهوم برنامه‌گذاری می‌شود. لذا این بالذاتی است.  
به مفهوم برنامه دقت می‌کنیم در دلیل رسمیت ممکن است در نظرالایم  
ممکن است مرا به این متعلق بگیرم به سایر آسیاه حم باشد.

$$F(X,0) : - X < 3.$$

$$F(X,2) : - 3 = < X.$$

$$F(X,4).$$

سوال: ایا این برنامه خوب عمل می‌کند؟ ✓

این برنامه ممکن است جندها راه حل تولید کند که برخی از آنها درست نیست، مثل:

$$\therefore f(1,Y).$$

$$Y=0;$$

$$Y=2;$$

$$Y=4;$$

No

نکته: این بار عدم وجود برآوردها بر رفتار برنامه تأثیر می‌گذارد و نتایج برنامه را تغییر میدهد.

دانشگاه صنعتی امیرکبیر - دانشکده ریاضی و علوم کامپیوتر

کیا  $a, b$  سبب است که لست  $L$  شود؟ ( $[a, b] \in L$  و  $c \in L$ )

|       |     |       |
|-------|-----|-------|
| $L_1$ | $S$ | $L_3$ |
|-------|-----|-------|

آمر  $L_1$  و  $L_3$  مفهوم شود.

$$L_1 = [ ] \quad \text{و} \quad S = [a, b] \quad \text{و} \quad L_3 = [c, d]$$

آخر طراز با  $L_1$  یوند بزیم می توان  $L_2$  را با  $L$  ( $\bar{L}$ ) یوند بزیم می شود  
چنانکه  $\bar{L}$  مفهومی درست نباشد. اگر  $L$  استفاده شود. بعداز آن  $\bar{L}$  مفهوم  
عدم شود -- -- --

نکته: عملگر زیر لیست برای این طراحی شده که نتواند وقوع لیستی را در  
لیست دیگری بررسی کند. این عملگر دارای انعطاف پیشتری است و  
از آن میتوان برای کاربردهای دیگری هم استفاده نمود مثلاً از آن  
میتوان برای یافتن تمام زیر لیستهای یک لیست استفاده نمود برای  
مثال:

?- sublist(S,[a,b,c]).

$S = [ ];$

$S = [a];$

$S = [a, b];$

$S = [a, b, c];$

$S = [ ];$

$S = [b];$

.....

(البته اینجا هیچ حالت حرام نماید بجز از  $[ ]$  می شود.)

هم را دریابی و لوگ اجر کنید.

(این آنای سطراچی است در اکس)

نمایه برای حذف del( $x$   $\in$   $L$ )

حالات خواهیم برخانی مردر انتشار دشال بیاریم.

اگر اینها ملا مذکورده حذف نند.

از سایه های دیگر دیگر  $L_1 =$

(63)  $\xrightarrow{\text{del}} [y | L]$

کتابخانه باهاشت سری احصایه دریم جالا آلان به حدود مثال میر داشتم. صفحه ۱۴

\* ختم از اینجا هم آید

عدد  $x$  و  $y$  لا برابر و متشابه نشود اگر  $x > y$  باشد  
اگر  $x$  کوچکتر باشد آن وقت  $\max(x, y) = y$  برابر  $y$  باشد اگر  $x < y$  باشد

بخدمت مثال پاسخگویی از پرسش

لطفاً پاسخ

### \* محاسبه مکریم دو عدد

فرض کنید بخواهید رویه ای به شکل  $\max(X, Y, \text{Max})$  برای محاسبه مکریم دو عدد  $X$

و  $Y$  بنویسید. اثرا بصورت زیر میتوانیم بنویسیم:

$$\max(X, Y, X) : - X >= Y.$$

$$\max(X, Y, Y) : - X < Y.$$

توجه داریم که دو قانون بالا انحصاری دو طرفه است اگر اولین عبارت درست باشد دوستی

رد میشود بنابراین با استفاده از برش رویه بالا را میتوانیم بصورت زیر بنویسیم:

$$\max(X, Y, X) : - X >= Y, !.$$

$$\max(X, Y, Y).$$

\* ارادی نشدنی شرط برمیگردید

نه ختم درست برقرار است

در غیر این صورت  $\max$  و  $y$  و  $y$  برابر نیست.

به این صورت هم بزایم کار سردد هم ساده تر.

با استفاده از عمل طایف باعثون لفظ در استادی برین مار زان (D):

جاییست: یعنی های بجا بین عناصر لیست است. (برنامه دفعی است)  
اگر لیست خوبی باشد جاییست هر دوست و اگر خالی نباشد می تسود P طبق

حالات زیر. لیست باشد tail و

مفرض لیست جاییست که باشد و مبود P. آنرا در اب از این که کارک باشد P را به مابعد  
یعنی ععنو اول را حذف کنیم و این را باید جا می کنیم و معملاً اول را insert کنیم

هر جا هم مه تواند insert بشود. جاییست یک لست خالی ممکن است  
هالا P خطو برس می آید؟ ععنواول را حذف کن و این جاییست روسی L برده و این را لذت

### جایگشتها

یکی دیگر از عملگرهای مفید عملگر جایگشت permutation میباشد

این عملگر را فیتوان براساس ملاحظه دو حالت و بر اساس لیست

اول داشت:

اگر لیست اول خالی باشد، بنابراین لیست دوم نیز باید خالی  
باشد.

اگر لیست اول خالی نباشد، بنابراین به شکل  $[X|L]$  است، و  
جایگشتی از چنین لیستی می تواند ساخته شود: ابتدا جایگشت  
L را از L1 بدست می اوریم و سپس X را به هر مکانی در L1  
اضافه می کنیم.

permutation([ ], [ ]).

permutation([X|L], P):-permutation(L, L1), insert(X, L1, P).

L1

[a, b, c, d, e]

حالا اگر را به L1 اضافه کنیم باشد P بود.

مثلاً  $a$  را در ترتیبی داشتیم. این باقیت را باز از L.

شناختی L بود لیست  $[c, d, e]$  باشد و این را در این

عرضه جا می شود لیست  $[c, a, b, d, e]$  باشد  $\leftarrow$  insert

در مثال زیر عمل  $\max$  را برداشته

$$\max(5, 4, 6) :- \text{out}(6).$$

نطا بین فورت  $\max(5, 4, 6)$ .

نمای نیز در مثال زیر

آر برگشتم بذارم چون ترد طبق فوارس  
جواب  $\max$  می‌دهد.

مثال زیر کاربرد نادرستی از این رویه را نشان میدهد.

$$\max(3, 1, 1).$$

Yes.

نکته: استفاده از برش در برنامه‌ها نیاز به دقت دارد و شاید لازم شود فرمولاسیون برنامه تغییر یابد.

فرمولاسیون زیر برای  $\max$  برای مشکل غلبه می‌کند:

$$\max(X, Y, Max) :- X >= Y, !, Max = X$$

;

$$Max = Y.$$

$$\max(X, Y, M) :- X >= Y, !, M = X$$

؛

$$Max = Y$$

برای بازدید cut

بلای خالص بعد از شرط بار

در نمایش نتیجه باشد

$$Max = Y$$

اگر هر خط اول بشد خطا در می‌شود

$$\max(X, Y, M) :- X >= Y, !, M = X \quad \text{or}$$

برای خط روم یعنی هر آن خط لازم را برگزینید (یاد نمایم این نتیجه نمایش نموده است)

گفتنیست صفحه قبل، آرایه اول خانه باشد جایلست خانه است رها آرخانه بخاستد head را برمی کند. وی tail جایلست هی زنده بعد از insert می کند.

آرایه اول خانه باشد نتیجه دوم یا همچنان جایلست آن خانه است. آرایه اول خانه بنابر head را برمی کند از tail را باید از من زنده بعد از insert می کنم.

مثال زیر را مطالع کنید و سومین متنوع شوید رهبر مرده یک جا به جای انجام داده راه را نشان جایلست استفاده از del است.

اصل برای بار اول امروزه عنصر اول را هدف می کند برای دفعات بعد را لست غامر را حذف می کند و هر بار یکی

یکی روش دیگر برای نوشتن عملگر جایگشت با استفاده از عملگر del:

permutation([ ], [ ]).

permutation(L, [X|P]) :- del(X, L, L1), permutation(L1, P).

?- permutation([a, b, c], P).

P=[a,b,c];

P=[a,c,b];

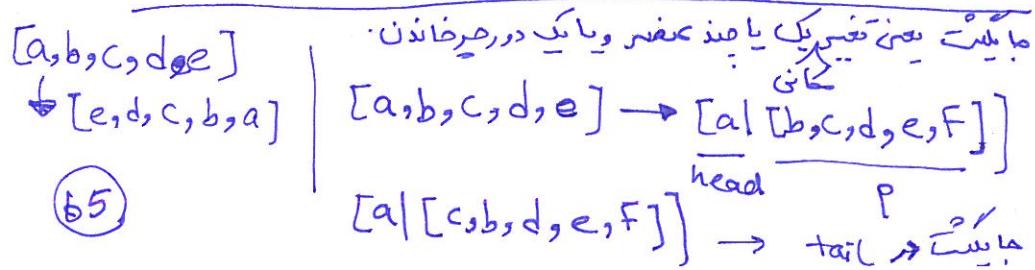
P=[b,a,c];

....

یعنی لست بعد از اعمال جایلست

جایلست یک لیست هی شود و آن را از ۱، هدف کنید من شود را.

جایلست ۱، هی شود چون تو بر head افزوده شد.



تمدن بر نامه ای بفونیند که بزرگترین مقصود علیه مشترک دو هدف را دارد  
با استفاده همان روشنی های ریاضی : اینجا با مردوک همراه سازی کنید.  
مسئله عضویت برای تابع عضویت داشته . برای آن اگر قدرت لذاتیم حصر عضویت  
را برای مایید ام کرد بقلا دریک لیست .

حالا همچنان خواهیم بر نامه ای بفرمایی عضویت کرد اگر یک عضویت داشت به مراعت بعدی  
نرود و نزیر هم تباه ندارم برو خوشبختان و مطلع نمایم .

### \* عضویت تک راه حلی

از رابطه member(X,L) برای بررسی وجود X در لیست L استفاده کردیم برنامه به شکل

زیر بود:  $\text{head} \neq \text{بُعد در tail}$  لشـ

member(X,[X | L]).

member(X,[Y | L]):-member(X,L).

این برنامه قطعی نیست یعنی اگر X چندین بار روی دهد، هر رویدادی ممکن است پیدا

شود. با استفاده از برش میتوانیم برنامه را طوری تغییر دهیم که فقط یک رویداد را پیدا

کند به صورت زیر:

member(X,[X | L]):-!.

member(X,[Y | L]):-member(X,L).

تابع عضویت مبتنی  
را اصلاح نمود و تک  
راه حلی شد .

[a, b, c, d, e]

مکال بـ جـاـبـجـانـ اـزـیـتـ هـرـ بـهـسـیرـ

۱) کـلـهـ صـدـفـ ۱۸ـتـ هـاـقـیـ رـایـنـ جـاـبـلـیـتـ بـزـنـ ↪ [c, d, e, b, a]

۲) هـلـاـحـهـ بـوـسـمـ ۱۹ـ رـاـدـرـلـیـتـ نـصـیـ insertـ نـهـ مـطـبـیـ: [c, e, a, d, b] ↪ [c, e, a, d, b, f]

آنـ بـرـیـامـ اوـلـ بـوـرـ.

عملگرهای از پیش تعریف شده

فصل سوم

برخی از عملگرهای از پیش تعریف شده در پرولوگ که میتوان از آنها در عملیات اساسی حساب استفاده کرد عبارتند از:

|     |                      |
|-----|----------------------|
| +   | جمع                  |
| -   | تفريق                |
| *   | ضرب                  |
| /   | تقسيم                |
| **  | توان                 |
| //  | تقسيم صحيح           |
| mod | باقيمانده تقسيم صحيح |

→ هـرـ لـمـ دـعـیـ بـمـاـ بـهـدـ

حالات نامددم الالا ۱۷ هین درس است  
برنام سوم الالا ۱۸ هین درس است

در الالا ۱۸ از صرف عنصر کاظمه ازاله برتر آمده در حان ۱۸ است

وست del هـرـ کـمـ حـمـلـاـ عـصـرـ اـولـ هـذـفـ هـمـ تـورـ

[a, b, c, d, e]

↓  
[a, b, c, d, e] → [c, d, e, b, a] → [a, c, d, e]

اگر حیزی افنا فهم دریم آن در لیست بود باز هم ممکن برداشته  
فرق لیست و مجموعه؟ مجموعه عنصر تکراری ندارد اما لیست ممکن است داشته باشد بنابراین  
مکن از افراد عنصره را لیست در قدرت وجود داشت باز هم انتروه من ممکن  
حالات خدا هم این را املاح کنند.  
یعنی هرمی تا حالا لفته روزخانه بود که رفت. خوب نتفع حمیود رسیده من را داری.  
النافه کردن عنصر بصر بدون تکرار

مثال:

 $\exists\text{-member}(X,[a,b,c]).$ 

X=a;

no

\* اضافه کردن یک عنصر بدون تکرار به لیست

قبلاب نامه اضافه کردن یک عنصر به لیست (در واقع به ابتدای یک لیست) را به صورت  
زیر نوشتیم:

add(X,L,[X | L]).

این برنامه در صورتی که عنصر X قبلاب در لیست L پاشد مجدداً افزایی این را به ابتدای لیست اضافه میکند. اغلب  
قصد داریم یک ایتم X را به یک لیست L اضافه کنیم به طوریکه این ایتم در لیست نباشد در صورتی که  
X در لیست پاشد، L بدون تغییر باقی بماند. این مطلب به این صورت برنامه نویسی میشود:

add(X,L,L):-member(X,L),!.

add(X,L,[X | L]).

13

دانشگاه صنعتی آمیر کبیر - داشکده ریاضی و علوم کامپیوتر

**هرابه ا افنا کن به طوریکه خود  
ک را بحداکثر عضوی از**

**ل بود خود را ببرد.**  
در عین این صورت ما را بسریت النافه کن که به هم باشد  
**و ل ا tail باشد. (دعیه ۷۰)**

مثال جاییست: (ترفن لینج جاییست است  
 $\text{perm}(L, [x|p])$  ماتریس  $M$   
 $\text{perm}(L, M)$  هست؟  
 $\downarrow$  حرف حالا  $M$  هست؟  
 $[x|p]$  حالا  $p$  هست؟ (ترننچه هارو؟)

باشه کاره نداشته باشید چون آوردمیم  $\rightarrow L = [x | \dots]$

بقیه اس را هابایست نزن به اینه دلیل اهل در کد جاییست داریم که تبرانشم

برهه  
راتئور  
بدمع

مثال: چند نمونه مثال برای آشنازی با عملگرهای حساب:

?- X is 5/2,

Y is 5//2,

Z is 5 mod 2.  $\rightarrow$  باعثها نده

حوال برولوگ چنین است:

X=2.5

Y=2

Z=1

آرگومان سمت چپ عملگر is یک شی ساده است و آرگومان سمت راست آن

یک عبارت محاسباتی است که مرکب از عملگرهای حسابی، هعداد و

متغیرها می باشد.

نکته 1: با توجه باینکه عملگر is باعث ارزیابی میشود، تمام متغیرها در این

عبارت پاسخی در زمان اجرای هدف با اعداد نمونه سازی شوند.

$\text{permutation}([], []).$   $\text{perm} \Leftarrow 65$

$\text{permutation}(L, [x|p]): - \text{del}(x, L, L_1), \text{permutation}(L_1, p).$   
 این راهنمایی دارد و درباره می خواهیم در این  
 لهستانی و راه حذف درین  $(L, p).$

شدن  $L \leftarrow$  جاییست  $\leftarrow$  شده است  $\leftarrow$  بوده است

(67)

$L \rightarrow [x, \dots]$

$L \rightarrow [\dots]$

(چه ذوقی درستاد  
 و بیره می کنم نه روی املای های نویس)

رفتار این رویه با مثالهای زیر تشریح میشود:

?-add(a,[a,b,c],L).

L=[a,b,c].

?-add(a,[b,c],L).

L=[a,b,c]

?-add(X,[b,c],L).

L=[b,c]

X=b

?-add(a,[b,c,X],L).

L=[b,c,a]

X=a

پرلوگ مغایب از کا استفاده نمی‌شود.

مثال:  $X = ?$  یعنی  $X = 2 + 1$  آنها دارد.

نکته: در پرولوگ از عملگر خاص 'is' برای ارزیابی استفاده می‌شود و نه عملگر '=' که برای تطابق اشیاء بکار گرفته می‌شود. برای محاسبه مثال: اگر از پرولوگ سوال شود:

?-  $X \text{ is } 1+2.$

جواب چنین است:

$X=3$

در حالی که اگر چنین سوال شود:

?-  $X=1+2.$

بروتوگ جواب معادل را میدهد :

$X=1+2$

جمع‌تر برآورده بوده است!!  
کسی در همه بازیها را برنده شده Winner است.

### • طبقه‌بندی در دسته‌ها

فرض کنید که پایگاه داده ای از نتایج بازی تنیس اعضاي یک باشگاه را داریم. بازی‌ها دونفره به صورت زیر ثبت شده‌اند:

برنده ترد  
beat(tom,jim).

beat(ann, tom).

beat(pat,jim).

میخواهیم رابطه‌ای به شکل class(Player, Category) تعریف کنیم که بازیکنان را به

دسته‌هایی درجه بندی کنیم. سه دسته به شکل زیر داریم:

برنده (winner): بازیکنی که تمام بازیهاش را برده از این نوع می‌باشد. ①

مبارز (fighter): بازیکنی که بعضی از بازیها را برد و بعضی را باخته ②

ورزشکار (sportsman): بازیکنی که تمام بازیهاش را باخته ③

باشگاه صنعتی نیز کمتر . ادشکده ریاضی و حلوم کامپیوتر

ظاهراً هماری همه درین هزاریم تر معرفه اینها را چه بلذاریم؟  
ـ لفون اساد زنگ زدن آمای شکاری بود. بنن ان طرحارا!!!  
اساد حواب نده سزا مارا ازیت می‌کند. هنوز سایلنت کردن اخیراً شده  
شکاری زند تدن. شکاری رسم اهرما راشکار ندرد. یا

# (رسانیدن مهارتی در این تم تفابرتن؟! خدمت منویلین از دست آین)

نکته 2: نقدم عملگرهای محاسباتی مانند نقدم طبیعی آنها بعنوان عملگرهای ریاضی است و با استفاده از برآنتر میتوان نقدم ها را تغیر داد.

نکته 3: عملگرهای دیگری که باعث ارزیابی میشوند عملگرهای مقایسه ای می باشند برای مثال:

$$?- 277*37 > 10000.$$

Yes

عملگرهای مقایسه ای عبارتند از:

$$X \text{ بزرگتر از } Y \quad X > Y$$

$$X \text{ کوچکتر از } Y \quad X < Y$$

$$X \text{ بزرگتر از یا مساوی با } Y \quad X \geq Y$$

$$X \text{ کوچکتر از یا مساوی با } Y \quad X \leq Y$$

$$\text{مقادیر } X \text{ و } Y \text{ برابرند} \quad X = Y$$

$$\text{مقادیر } X \text{ و } Y \text{ برابر نیستند} \quad X \neq Y$$

وَمَنْ لِرَحْمَةِ رَبِّكَ اسْتَهْدَى مَنْ لِنِعْمَةِ رَبِّكَ شَرِّبَ حَيْثُ شِئَ . جلسه ۱۴

لَعْنَهُمْ كُسَانٌ رَابِرٌ وَهُمْ كُسَانٌ نَى تَسْلِيَتْ دَارِنْشَ كَيْنِي زَانِشَ هَاسَ .

وَهُمْ كُسَانٌ رَابِرٌ أَمَا هِيجَ كَسَنْ مَهْ رَانِبَرْ .

البَهْ كَيْنِي اسْلَانْ دَارِرْ خَوْبَانْ بُلْوِيمْ Tom (Sportman ) است

جَوَابَيْ Yes بِهَامِ دَهْر . حَوْنْ جَاهِيزَارِي مَهْ لَهْ ( مثل مَتَالْ قَبَهْ دَهْ اسْتَدَهْ )  
بِهَبرِلَشَنْ نَارِنْد : ( )

این برنامه را به صورت زیر میتوانیم بنویسیم:

class(X , fighter):-beat(X,\_), beat(\_,X),!.

حَوْنْ جَاهِيزَارِي بُورْ

class(X , winner):-beat(X,\_),!.

مَسْتَقِيَّ بِهَ حَالَهْ سَرْمَهْ زَفَرَتْ

class(X , sportsman):-beat(\_,X).

وَجَوَابَيْ باَصْكَلْ سَواْجَهْ حَمَدْ .

حل اگر سوال زیر را بپرسیم کلاس بازیکن مربوطه را مشخص میکند:

?-class(tom, C).

C=fighter

نکته: فرآخوانی ای رویه مطمئن است اگر ارگومان دوم نموده سازی نشده باشد. در غیر

اینصورت ممکن است نتیجه غیرمنتظره ای حاصل شود مانند مثال زیر:

?-class(tom, sportsman).

yes

16

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

نکته 2: به تفاوت بین عملگر  $=$  و  $\equiv$  توجه کنید:

?-  $1+2=:=2+1.$

Yes

?-  $1+2=2+1.$

No

?-  $1+A=B+2.$

$A=2$

$B=1$

yes

تساوی

اصطلاح نظریه

هو خانم خیری صلیح از هارمه تردد.  
استار: شنیدم باش دختر.

صریح خانم هم حیوانات را درست نارد به بجز ماز

#### \* نفي در هنگام ناکامي

“ماري تمام حيوانات را دوست دار بجز مار ” اين عبارت را چگونه  
ميتوان در پرولوگ بيان كرد؟

`likes(mary , X):-snake(X),!,fail.`

`likes(mary, X):-animal(X).`

اين دو عبارت را ميشود با يك عبارت به صورت ساده تر نوشت:

`likes(mary, X):-snake(X),!,fail; animal(X).`

ميتوانيم از ايده مشابهی برای تعریف رابطه different(X,Y) استفاده کنیم که اگر  $X$  و  $Y$  متفاوت باشند این رابطه درست است. برای اینکار میتوانیم مجدداً از ترکیب برش و fail استفاده کنیم:

جلسه نازدیم ۹۳. ۱۰. ۳

در رياضي برای بدست آوردن مقسوم علیه مشترک دو عدد از هم بزرگ‌تر ممکن و تعیین خواهد شد.

- ۱) اگر  $X = Y$  باشد بزرگ‌ترین مقسوم علیه مشترک خواهد بود شرطی شود.
- ۲) اگر  $X > Y$  باشد  $D$  خواهد بود  $\Leftrightarrow$  بزرگ‌ترین تفاضلی خواهد بود که بزرگ‌تر است را تعادل داشته باشد و مقسوم علیه مشترک خواهد بود.
- ۳) دو عدد طوری در یک جمله به صورت بازگشتی  $(a_1 - a_2)$  داشته باشند  $\Rightarrow X = a_1 - a_2$  به مررت خط سوم در می‌شود.

جند مثال:

مثال 1: محاسبه بزرگ‌ترین مقسوم علیه مشترک دو عدد

چگونه متواتر از بزرگ‌ترین مقسوم علیه مشترک دو عدد  $X$  و  $Y$  را بیندا کیم:

اگر  $X = Y$  باشد بزرگ‌ترین مقسوم علیه مشترک  $D$  برابر  $X$  است. ۱

اگر  $X > Y$  باشد بزرگ‌ترین  $D$  با بزرگ‌ترین مقسوم علیه مشترک  $X - Y$  است. ۲

برابر است

اگر  $X < Y$  باشد آنگاه همانند ۲ با تغیر جای  $X$  و  $Y$  ۳

$\text{gcd}(X, X, X).$

$\text{gcd}(X, Y, D) :- X < Y, Y1 \leftarrow Y - X, \text{gcd}(X, Y1, D).$

$\text{gcd}(X, Y, D) :- Y < X, \text{gcd}(Y, X, D).$

دست داشتم با شیر بزرگ‌تر حسنه یاری داشتم گویا هاست و از خواندن بازیسته استفاده کردم  
لوب هم توسط همین الگوریتم حسنه ایجاد می‌شود

روابطی میان دو عبارت باشند که هم‌توافق نباشند different:

سازمانی دو عبارت باشند که هم‌توافق نباشند. باید میان آنها اختلاف باشد.

فهرست احتمالات:

حیثیت دو عبارت باشند (درست):  
حیثیت دو عبارت نباشند (نادرست):

$\{ \text{different}(X, X) :- !, \text{fail}. \quad \rightarrow \text{different}$

$\text{different}(X, Y). \quad \text{براساس cut}$

تعریف گزاره 'not': این گزاره را میتوان بصورت زیر تعریف کرد:

$\{ \text{not}(P) :- P, !, \text{fail}; \text{true}.$

در غیر این حالت

که true هدفی است که همیشه پذیرفته است.

حال میتوانیم از این گزاره 'not' در برنامه خود بجای برش استفاده نمائیم.

در زیر برنامه های قبلی با استفاده از مفهوم گزاره 'not' مجدداً نوشته شده است:

$\{ \text{likes(mary, X) :- animal(X), not(snake(X)).}$

$\text{Different}(X, Y) :- \text{not}(X = Y).$

هنا لقبید

طول لیست خالی برابر لفکرات شرط توقف برنامه. (خط اول مقدم)

حد همینزی باشد و tail داشته باشد طول آن N است.

اگر طول باشه N باشد پس  $N \leftarrow M$  است

هدمی کوئند چند عضوی هم باشد آما جوئن به صورت بازگشایی اسیداگ حرفی هم نمی برد و قاعده  
tail می باشد (بمقدار استاندارد این طوری عبارت هم نمی باشد) ۱) همین علت ۱) اضافه کنید

حند مثال:

مثال ۲: محاسبه طول یک لیست.

چگونه متواترین طول یک لیست L را حساب کنیم:

اگر لیست خالی باشد طول آن برابر صفر است. ۱

اگر لیست خالی نباشد بنابراین  $L = [Head | Tail]$  و طول آن برابر است با ۲

Tail بعلاوه طول ۱

`length([], 0).`

`Length( [ _ | Tail], N):- length(Tail, N1), N is 1+N1.`

حال اگر سوال زیر را از برولوج بپرسیم جواب آن به صورت زیر است:

?- `length([a,b,[c,d],e], N).`

N=4

1 2 3 4

## صلحات بعد راهم بازی خبریم

تاریخ: ۹۳.۱۰.۲۴ ساعت: ۲:۲۸ am  
کامپیوٹر: جا برای پایان ترم

Class(X, fighter):- beat(X, \_), beat(\_, X).

Class(X, winner):- beat(X, \_), not(beat(X, \_)).

Class(X, sportsman):- beat(\_, X).

نکته: از این به بع فرض خواهیم کرد که 'not' گزاره تعییه شده پرولوگ است و به صورتی که اینجا تعریف میکند، رفتار میکند. همچنین فرض میکنیم که 'not' یک عملگر پیشوندی تعریف میشود. در واقع بعضی از پیاده ساریهای پرولوگ این نمادگذاری را پشتیبانی می کنند.

شال حاصل است حتماً بخوبی در اینجا آید  
در خواسته استاد درون حیزن وقتی تمام شد  
خانم هرگیز

شروع حوس ۴ : هلس ۱۱ (صلوات ختم لیند) پاییزه داده استاجه

(ادام این مطالع در  
نیت هفته ۴۹) نوی خانی از پاییزه داده است . در پاییزه داره مجموعی ما دارم ولی اینها را ن

بلویم پاییزه دست استاجه .

به همان پاس P.74 وی داشت بذارم در پاییزه داده قدرت اندالی واستاجه رشید سری دارد .

در پاییزه های مجموعی قدرت استاج ذرا منقطع عملیات (جامد) دارد .

## موفق باشید

پاییزه داده استاجی نوچی پاییزه داده هوشمند است . یعنی از راستا مجبور می شوند به اطلاعات و داشت های رسیده دست یافته . (دانش مدیری استخراج هر شعده همچنین است از پاییزه بنویسید)

چون داشت درین استفاده هم ترد کاربردهای زیادی خواهد داشت با این مقصود هم شویم که حیرکاری هم را از انجام برده

قتلا اطلاعاتی راچیم به خانواره های مختلف را روی پاییزه داده تعداد پر هم (73) هم تکلیف سوالات زیادی در می دارد .

## مثال حواری‌جوابی

نمایندگی در زبان فارسی

باید توجه داشته باشیم که استفاده از برش برای هر چیزی نیست. بعضی از مزایا و معایب برش در زیر آمده:

۱. با برش اغلب میتوانیم کار ایپی برنامه را بهبود ببخشیم. این ایده دقیقاً به پرولوگ میگوید سعی نکن راههای دیگر را امتحان کنی زیرا محدود به عدم پذیرش شده اند
۲. با استفاده از برش میتوان قوانین انحصار متناسب را تعیین کرد. بنابراین میتوانیم قوانینی به شکل زیر را بیان کنیم:

اگر شرط  $P$  بنابراین نتیجه  $Q$  در غیر اینصورت نتیجه  $R$ .

نتیجه: برش نوان بیانگر زبان را افزایش میدهد.

مثال: بے مثال زیر توجہ کنید:

good\_standard(jeanluis).

expensive(jeanluis).

good\_standard(francesco).

reasonable(Reastaurant):- not(expensive(Reastaurant)).

حال اگر بپرسیم:

?- good\_standard(X), reasonable(X).

پرولوگ جواب میدهد:

X=francesco

اگر این سوال را بپرسیم:

?- reasonable(X), good\_standard(X).

پرولوگ جواب میدهد:

دانشگاہ صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوٹر

نکته: در یک برنامه ترجیح داده میشود بجای ترکیب برش و fail از گزاره 'not' استفاده شود. ولی 'not' خود دارای مشکلاتی است. برای مثال اگر از پرولوگ

بپرسیم:

?- not(human(mary)).

پرولوگ احتمالا جواب 'Yes' را میدهد. و این مسئله بدان معنی نیست که پرولوگ بگوید "ماری انسان نیست" انچه معنی واقعی پرولوگ است این است که در برنامه اطلاعات کافی وجود ندارد که ثابت کند ماری انسان است.

در واقع در هنگام پردازش هدف not پرولوگ سعی نمیکند تا این هدف را مستقیما ثابت کند بلکه سعی میکند تا خلاف انرا ثابت کند و اگر خلاف ان ثابت نشود، پرولوگ فرض میکند که هدف not پذیرفته شده است.

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

پرولوگ جواب میدهد:

No

دلیل پاسخهای متفاوت چیست؟

مشکل اهداف منفی نمونه سازی نشده از تغیر ناجور کمیت متغیرها در نفي هنگام عدم پذیرش بوجود می آيد.

در تفسیر عادی سوال زیر :

?-expensive(X).

بدین معنی است که آیا X ای وجود دارد به قسمی که expensive(X) درست باشد؟ که جواب X=jeanlluid است.اما این سوال:

?-not(expensive(X)).

چنین تفسیر نمیشود. بلکه تفسیر میشود آیا X ای وجود دارد که not(expensive(X)) باشد؟ پاسخ مورد انتظار X=francesco است ولی پرولوگ جواب میدهد

دانشگاه صنعتی امیر کبیر - دانشکده ریاضی و علوم کامپیوتر

ach location. A head can thereby attend sharply to the memory at a single location or  
ikly to the memory at many locations.

## Reading

$M_t$  be the contents of the  $N \times M$  memory matrix at time  $t$ , where  $N$  is the number of memory locations, and  $M$  is the vector size at each location. Let  $w_t$  be a vector of weightings over the  $N$  locations emitted by a read head at time  $t$ . Since all weightings are normalised, the  $N$  elements  $w_t(i)$  of  $w_t$  obey the following constraints:

$$\sum_i w_t(i) = 1, \quad 0 \leq w_t(i) \leq 1, \forall i. \quad (1)$$

The length  $M$  *read vector*  $r_t$  returned by the head is defined as a convex combination of the row-vectors  $M_t(i)$  in memory:

$$r_t \leftarrow \sum_i w_t(i) M_t(i), \quad (2)$$

which is clearly differentiable with respect to both the memory and the weighting.

## 2 Writing

Taking inspiration from the input and forget gates in LSTM, we decompose each write into two parts: an *erase* followed by an *add*.

Given a weighting  $w_t$  emitted by a write head at time  $t$ , along with an *erase vector* whose  $M$  elements all lie in the range  $(0, 1)$ , the memory vectors  $M_{t-1}(i)$  from the previous time-step are modified as follows:

$$\tilde{M}_t(i) \leftarrow M_{t-1}(i) [\mathbf{1} - w_t(i) e_t], \quad (3)$$

here  $\mathbf{1}$  is a row-vector of all 1-s, and the multiplication against the memory location acts point-wise. Therefore, the elements of a memory location are reset to zero only if both the weighting at the location and the erase element are one; if either the weighting or the erase is zero, the memory is left unchanged. When multiple write heads are present, the erasures can be performed in any order, as multiplication is commutative.

Each write head also produces a length  $M$  *add vector*  $a_t$ , which is added to the memory after the erase step has been performed: