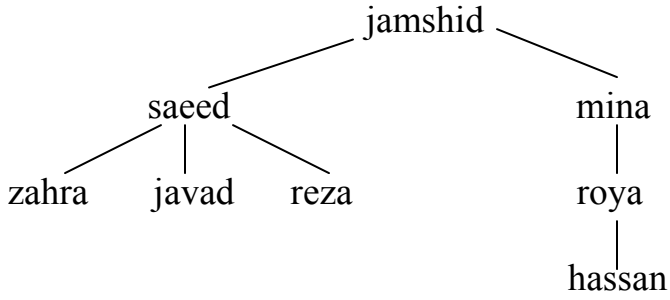
باسمه تعالی

پرولوگ
قسمت دو

A Classic Example

یکی از مسائل کلاسیکی که معمولاً در ابتدای آموزش پرولوگ مطرح می کنند مساله شجره نامه خانوادگی می باشد.

برای مثال شجره نامه زیر را درنظر بگیرید:



این شجره نامه را به زبان پرولوگ توصیف می نماییم. برای این کار سه محمول پایه یعنی male، female و parent را اختیار می کنیم و با یک سری fact درخت را نمایش می دهیم.

```
male(jamshid).
male(saeed).
male(javad).
male(reza).
male(hassan).
female(zahra).
female(mina).
female(roya).
parent( jamshid, saeed).
parent( jamshid, mina).
parent( saeed, javad).
parent( saeed, zahra).
parent( saeed, reza).
parent( mina, roya).
parent( roya, hassan).
```

پس از ساختن پایگاه دانش یک سری query به سیستم می دهیم.

- Is hassan the parent of saeed?

Query: `parent( hassan, saeed).`

- Who is saeed's parent?

Query: `parent( X, saeed).`

- Who were the *children* of saeed?

Query: `parent( saeed, X).`

اکنون می خواهیم سوالاتی هوشمندانه تر از پرولوگ بپرسیم، از قبیل اینکه :

مادر X چه کسی است ؟

آیا Y خواهر X است؟

عموهای X را نام ببر ...

برای پرسیدن چنین سوالاتی بایستی تعریف مادر، خواهر و یا از این دست را به پرولوگ بدهیم.

بنابراین باید به نحوی KB را گسترش بدهیم. یک راه بدیهی این است که برای تمامی افراد موجود در شجره نامه این رابطه را تعریف کنیم، که منطقی به نظر نمی رسد. بنابراین از یک گزاره منطقی استفاده می کنیم.

$$\forall X,Y : parent(X,Y) \wedge female(X) \rightarrow mother(X,Y)$$

کلاز متناظر در پرولوگ به این صورت خواهد بود:

mother(X,Y) :- parent(X, Y), female(X).

این نشان دهنده ی نوع دوم ساختارهایي است که در KB های به زبان پرولوگ می تواند وجود داشته باشد(rule). به قسمت سمت چپ علامت -: اصطلاحاً head وبه قسمت سمت راست آن body می گویند.

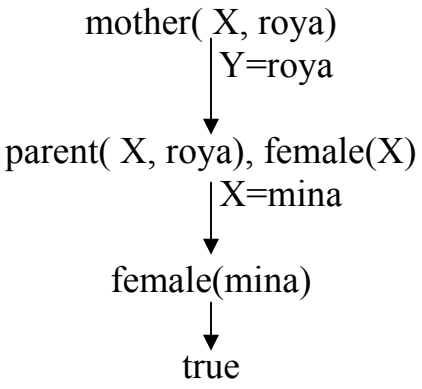
دقت اکید: توجه کنید که رابطه $\rightarrow$ به صورت برعکس و یا goal-driven نوشته شده است.
Prolog finds the proof sequence in the inverse order to that which we have just used. Instead of starting with simple facts given in the program, Prolog starts with the goals and, using rules, substitutes the current goals with new goals, until new goals happen to be simple facts.

با فرض اضافه شدن این rule به KB، فرض کنید که این سوال پرسیده شده است:

mother(X, roya).

و به عبارت دیگر: "مادر رویا کیست؟"

برای این پرسش درخت زیر را می توان بنا کرد.



درمنطق درجه اول به استنتاج $\frac{\alpha \vee \neg\beta, \beta}{\alpha}$ اصطلاحاً قیاس و یا Resolution می گویند. این استنتاج هم از کارهای Robinson محسوب می شود، و جالب است رابینسون فلسفه نام گذاری آنرا فراموش کرده است.

In 1960 an extraordinary thing happened. The Swedish logician Dag Prawitz independently rediscovered unification, the powerful secret which had been buried for 30 years in the obscurity of the Property A section of Herbrand's thesis. This turned out to be an important moment in the history of computational logic. Ironically, Prawitz' paper (reprinted in [8], Volume 1) too was rather inaccessible. William Davidon, a physicist at Argonne, drew my attention to Prawitz' paper in 1962, two years after it had first appeared. I wish I had known about it sooner. It completely redirected my own increasingly frustrated efforts to improve the computational efficiency of the Davis-Putnam Herbrand Property B proof procedure. Once I had managed to recast the unification algorithm into a suitable form, I found a way to combine the Cut Rule with unification so as to produce a rule of inference of a new machine-oriented kind. It was machine-oriented because in order to obtain the much greater deductive power than had hitherto been the norm, it required much more computational effort to apply it than traditional human-oriented rules typically required. In writing this work up for publication, when I needed to think of a name for my new rule, I decided to call it "resolution", but at this distance in time I have forgotten why. This was in 1963. The resolution paper took more than another year to reach print, finally coming out in January 1965 (it is reprinted in [8], Volume 1).

The trick of combining a known inference rule with unification can of course be applied to many other rules besides the cut rule. At Argonne, George Robinson and Larry Wos quickly saw this, and they applied it to the rule of Equality Substitution, producing another powerful new machine-oriented rule which they subsequently exploited with much success. They called their new rule "paramodulation". Their paper is reprinted in [8], Volume 2. It and resolution have been mainstays of the famous Argonne theorem provers, such as McCune's OTTER, ever since. See, for example, the 1991 survey ([12], pp. 297 ff.) by Larry Wos.

تمرین: رابطه های عمه، عمو، نوه و مادربزرگ را به پرولوگ بنویسید.

قوانین بازگشتی (recursive rules):

فرض کنید که می خواهیم رابطه predecessor را به KB اضافه کنیم.

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

قانون فوق می گوید که X از پیشینیان Z است اگر X پدر Z باشد. این قانون تنها تا یک سطح رابطه predecessor رابیان می کند.

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

و این قانون تنها تا دوسطح.

```
predecessor( X, Z) :-
        parent( X, Y1),
        parent( Y1, Y2),
        parent( Y2, Z).
```

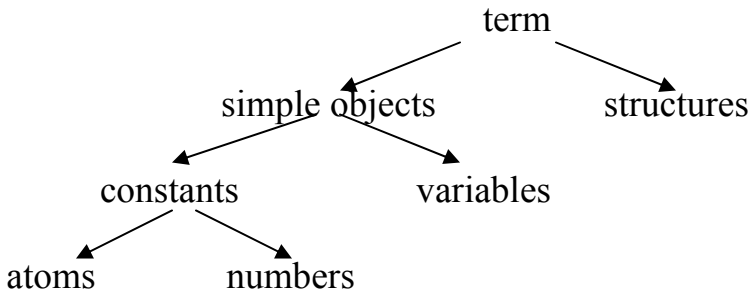
این روش پیاده سازی انتها ندارد، و طولانی است. راه بهتر پیاده کردن از روش بازگشتی است، یعنی بگوییم:

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

Horn Clause : همان طور که توجه کردید کلازهای پرولوگ تنها از یک نوع خاص هستند، که به آنها کلازهای Horn می گویند.(به افتخار معرفی کننده ی آنها Alfred Horn). این کلازها قابلیت بیان اکثرکلازهای منطق درجه اول رادارند. بااین که پیروی کردن از تنها یک نوع کلاز برنامه سازی را قدری مشکل می سازد، ولی ثابت می شود که از نظر قدرت برنامه سازی تاثیری در کار ندارد.

Prolog Syntax

یک برنامه پرولوگ از تعدادی ترم تشکیل یافته است.



Atom : برسه قسم هستند:

- رشته ای از حروف، ارقام و underscore که با یک حرف کوچک شروع می شود:
  ali, x_123,y_ _ _ p,ali_Alavi

- رشته ای از حروف ویژه:

-: ، -?

- رشته ای داخل single quote

Numbers : اعداد می توانند صحیح و یا اعشاری باشند. چون پرولوگ اساساً برای کارهای symbolic و غیر محاسباتی استفاده می شود زیاد اعداد اعشاری کاربردی ندارند.

Variables : متغیرها رشته هايي ازحروف ،ارقام و underscore هستند که با یک حرف upper-case شروع می شوند.

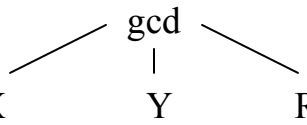
متغیر بی نام : در پرولوگ متغیر بی نام با under-score نمایش داده می شود.برای متغیر هايي که تنها در یک جمله به کار برده شده اند نیازی به ابداع نام جدید نیست.

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

در رابطه ی "بچه ای دارد" مهم نیست که بچه چه کسی باشد، بنابراین از متغیر بی نام استفاده کردیم.

Structures : تمامی ساختارها در پرولوگ به صورت درخت هستند. این اشیا دارای یک سری اجزاء می باشند. این اجزا هم به نوبه خود می توانند structure باشند.

gcd(X, Y, Res).



بنابراین از نظر پرولوگ نوشتنa*b به صورت prefix یعنی (a,b)* کاملاً مجاز است.