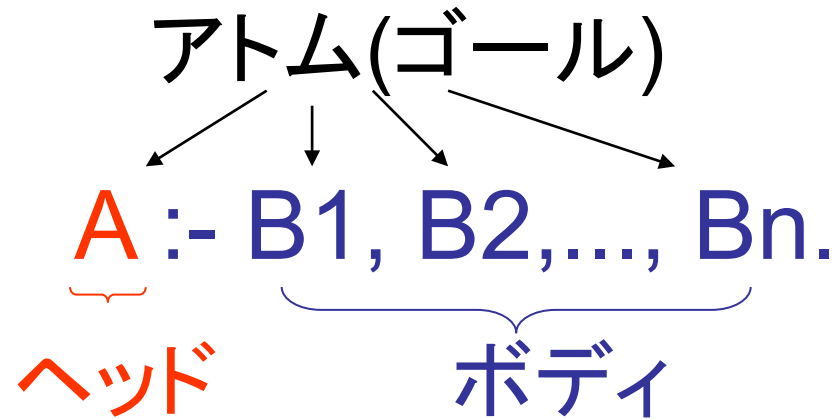


Prolog の実行過程

ホーン節



論理的意味(対応する論理式)

$$\forall x_1 \dots \forall x_k (A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n)$$

PROLOGとホーン節

- 単位節 事実の記述

B.

fr(apple).

fr(banana).

- 確定節 ルール

A :- B1,B2,...,Bn.

式に出現する任意の変数について成立

sweet(X) :- fr(X).

- ゴール節 質問

:- B1,...,Bn.

:- sweet(apple).

例 1

PROLOGの実行と単一化

ロードされている節

```
:- fr(apple).
```

```
fr(apple).  
fr(banana).  
sweet(X) :- fr(X).
```

単一化の候補となるゴール

ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

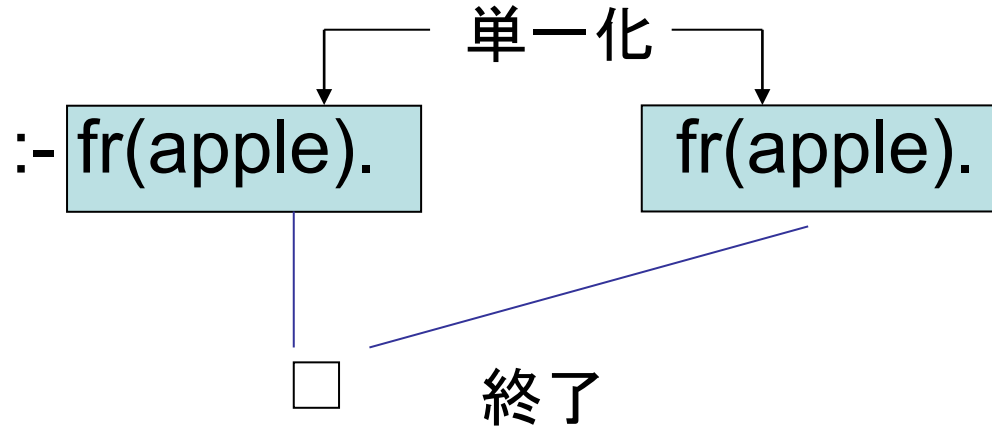
結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする



失敗

別の節で試みる(バックトラック)

PROLOGの実行(1)

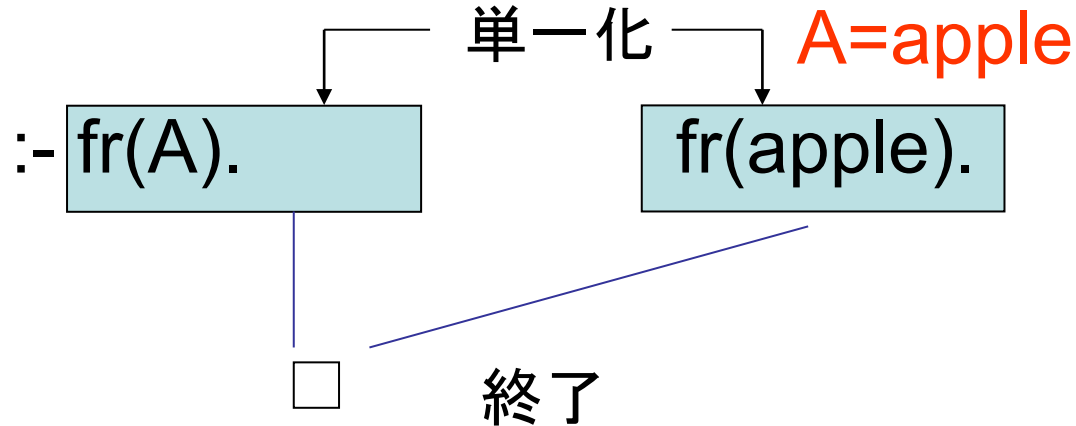


ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行

成功

結果をその節のボディに反映させ、
ボディ部を新たなゴール節とみなして
解こうとする

PROLOGの実行(2)



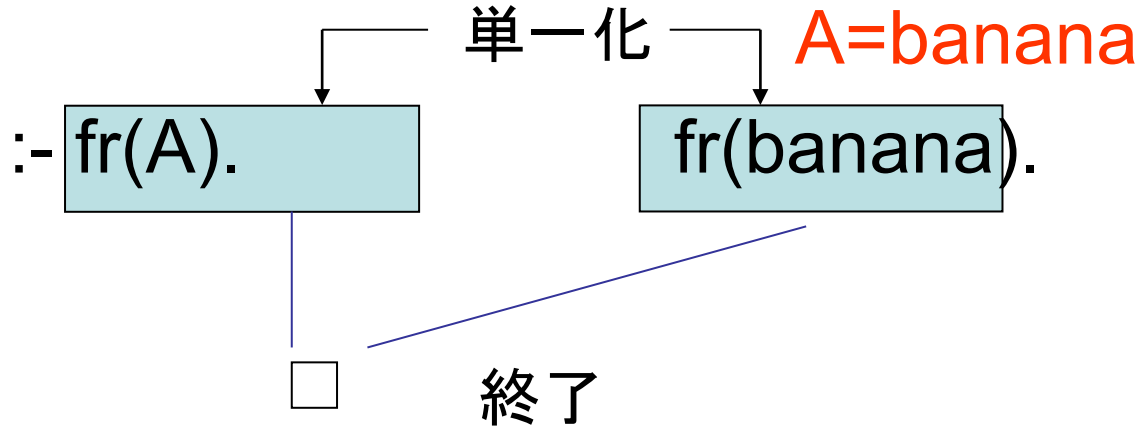
ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行

成功

結果をその節のボディに反映させ、
ボディ部を新たなゴール節とみなして
解こうとする

PROLOGの実行(3)

別解を求めるとバックトラックし、別の節と単一化を試みる



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行

成功

結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする

PROLOGの実行と単一化

ロードされている節

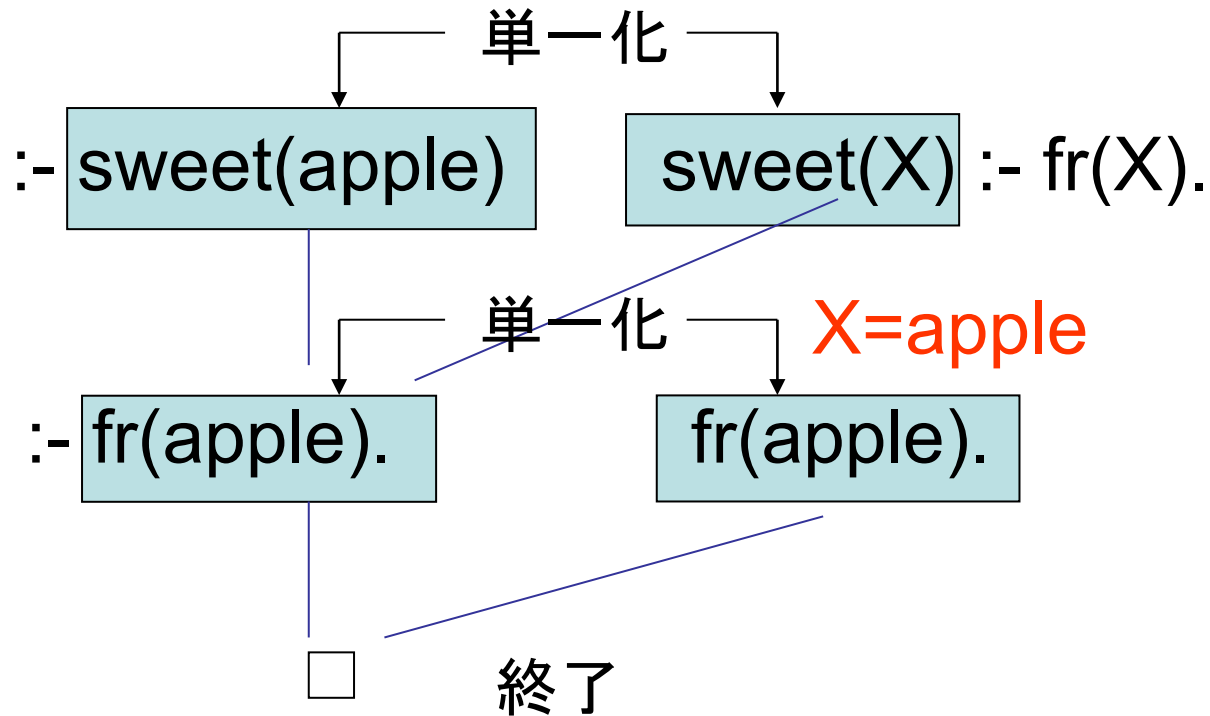
`:- sweet(apple).`

```
fr(apple).  
fr(banana).  
sweet(X) :- fr(X).
```

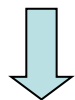
単一化の候補となるゴール

PROLOGの実行(4)

線形導出



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする

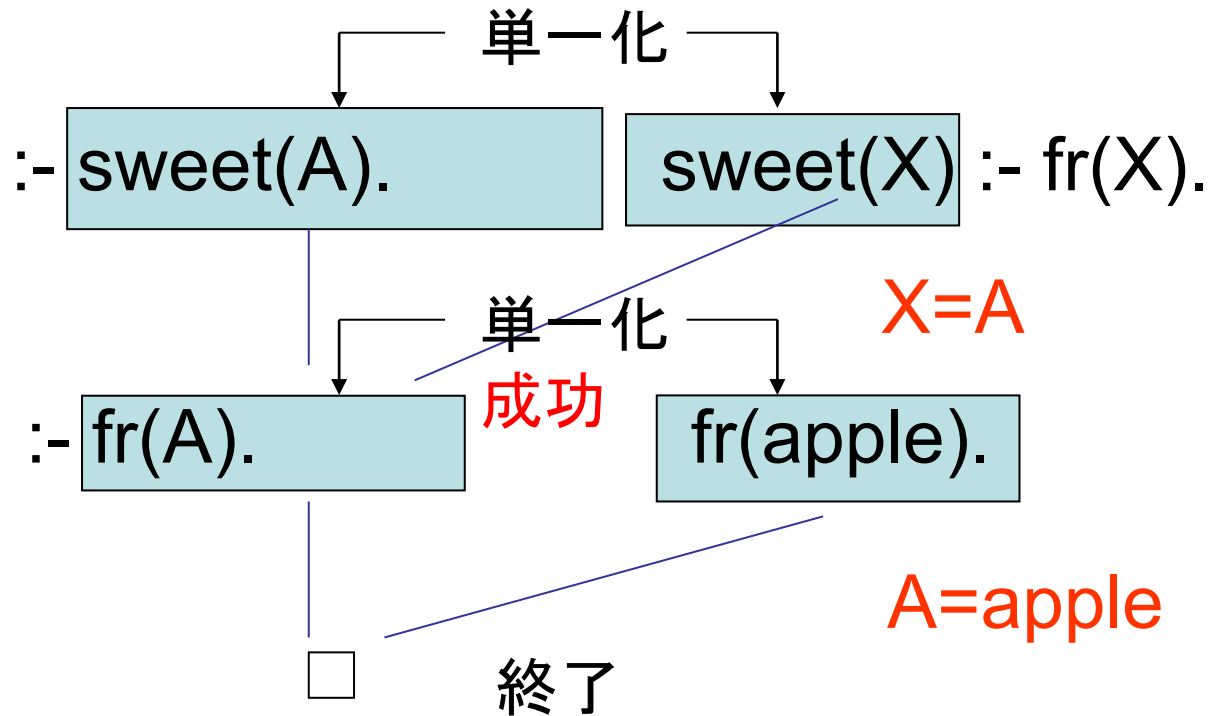


失敗

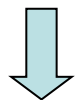
別の節で試みる(バックトラック)

PROLOGの実行(5)

線形導出



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

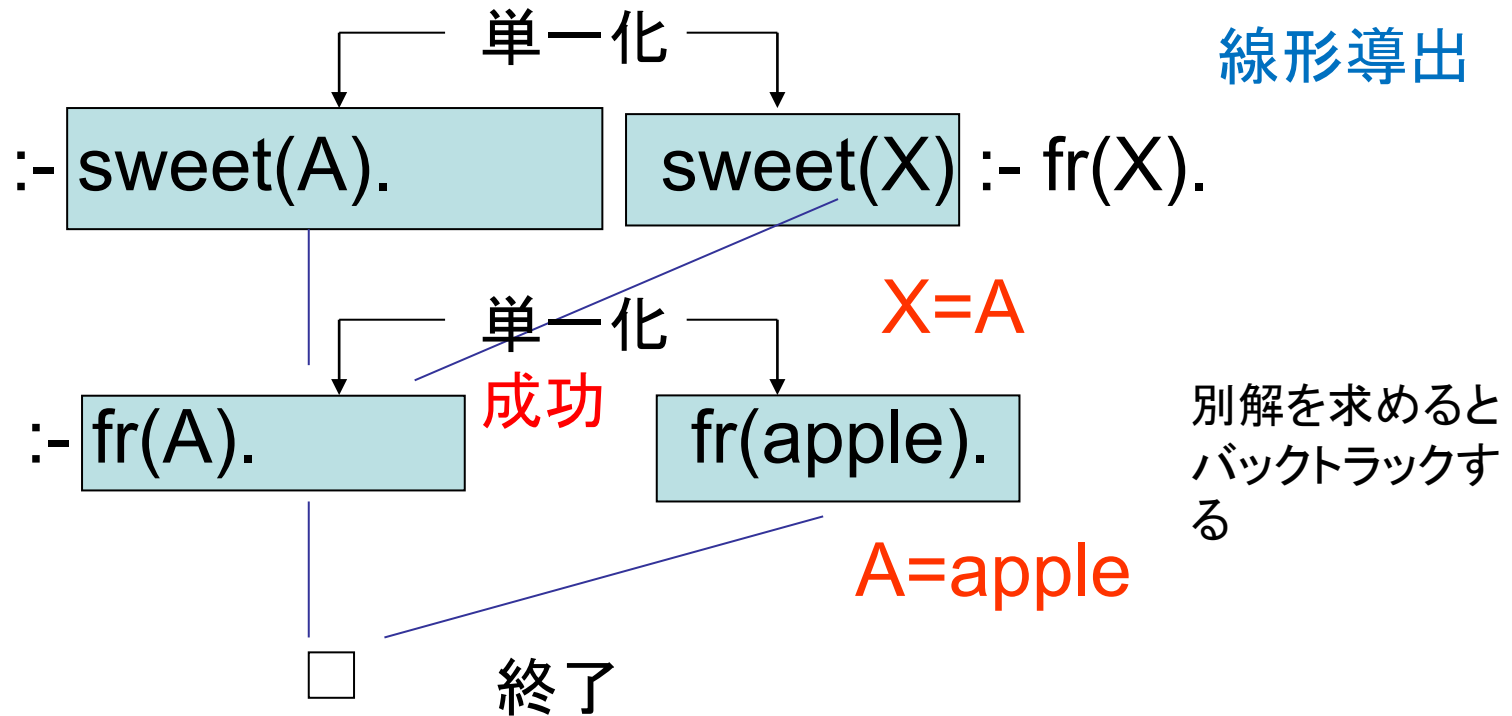
結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする



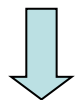
失敗

別の節で試みる(バックトラック)

PROLOGの実行(6)



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする

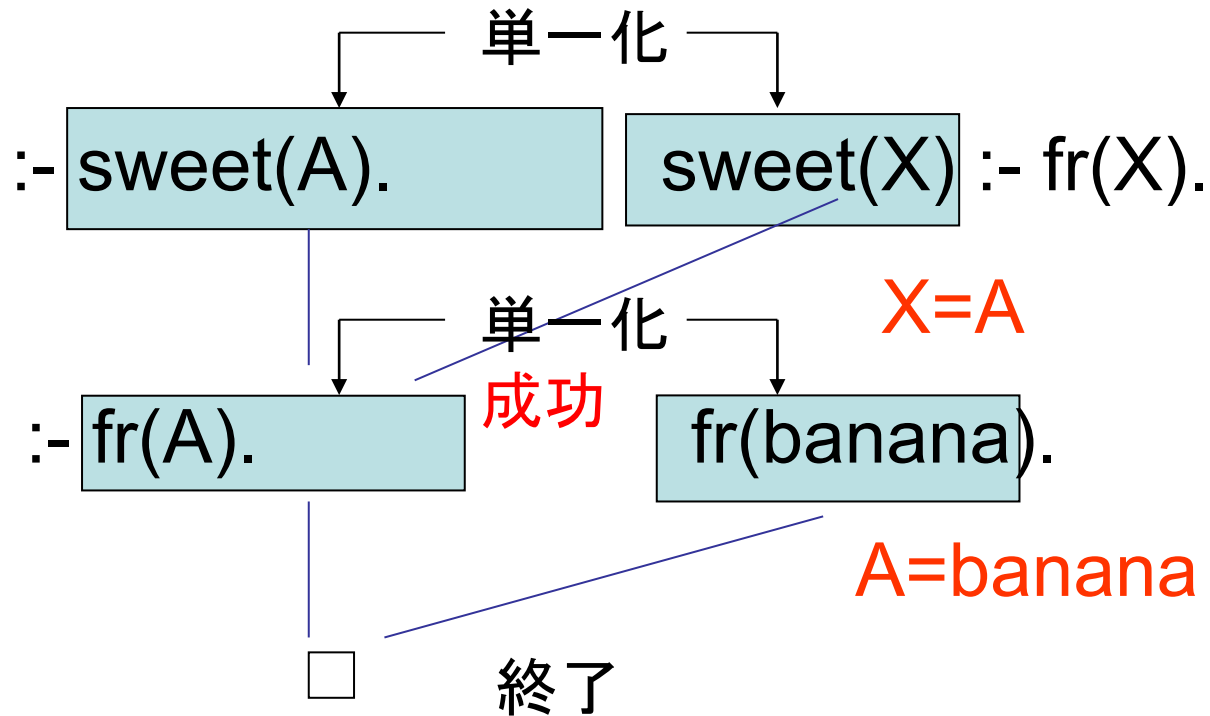


失敗

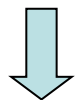
別の節で試みる(バックトラック)

PROLOGの実行(7)

線形導出



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする



失敗

別の節で試みる(バックトラック)

例2

PROLOGの実行と単一化

ロードされている節

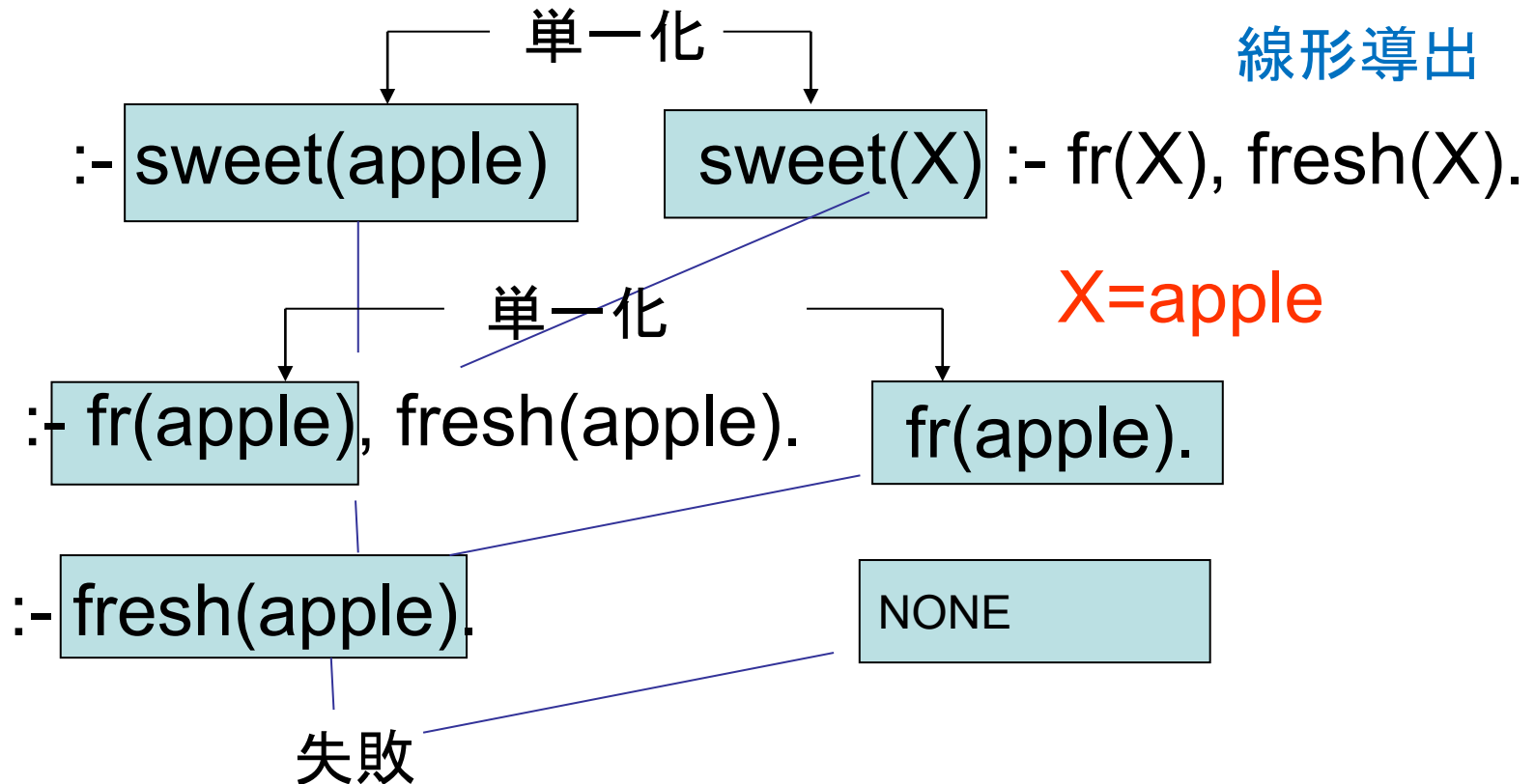
`:- sweet(apple).`

```
fr(apple).  
fr(banana).  
sweet(X) :- fr(X), fresh(X).
```

単一化の候補となるゴール

PROLOGの実行(8)

線形導出



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

結果をその節のボディに反映させ、
ボディ部を新たなゴール節とみなし
て解こうとする

例3

PROLOGの実行と単一化

ロードされている節

`:- sweet(candy).`

```
fr(apple).  
kashi(candy).  
sweet(X) :- fr(X).  
sweet(X) :- kashi(X).
```

単一化の候補となるゴール

ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする

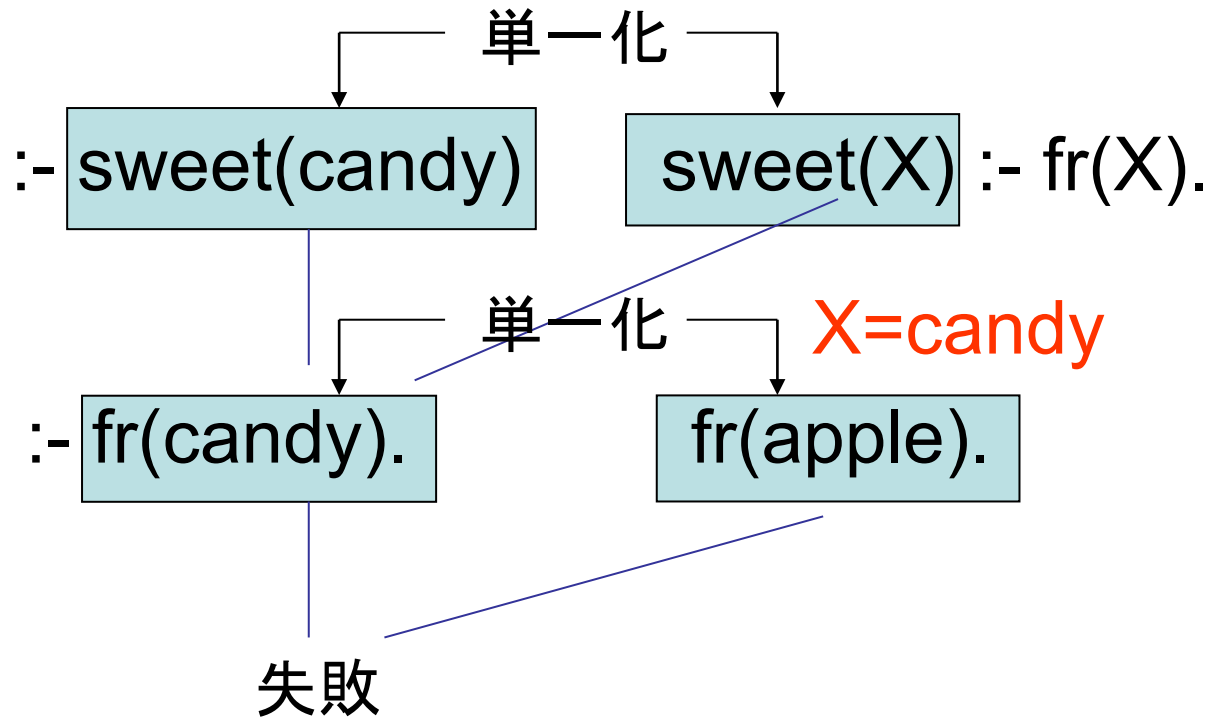


失敗

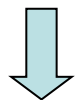
別の節で試みる(バックトラック)

PROLOGの実行(9)

線形導出



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行



成功

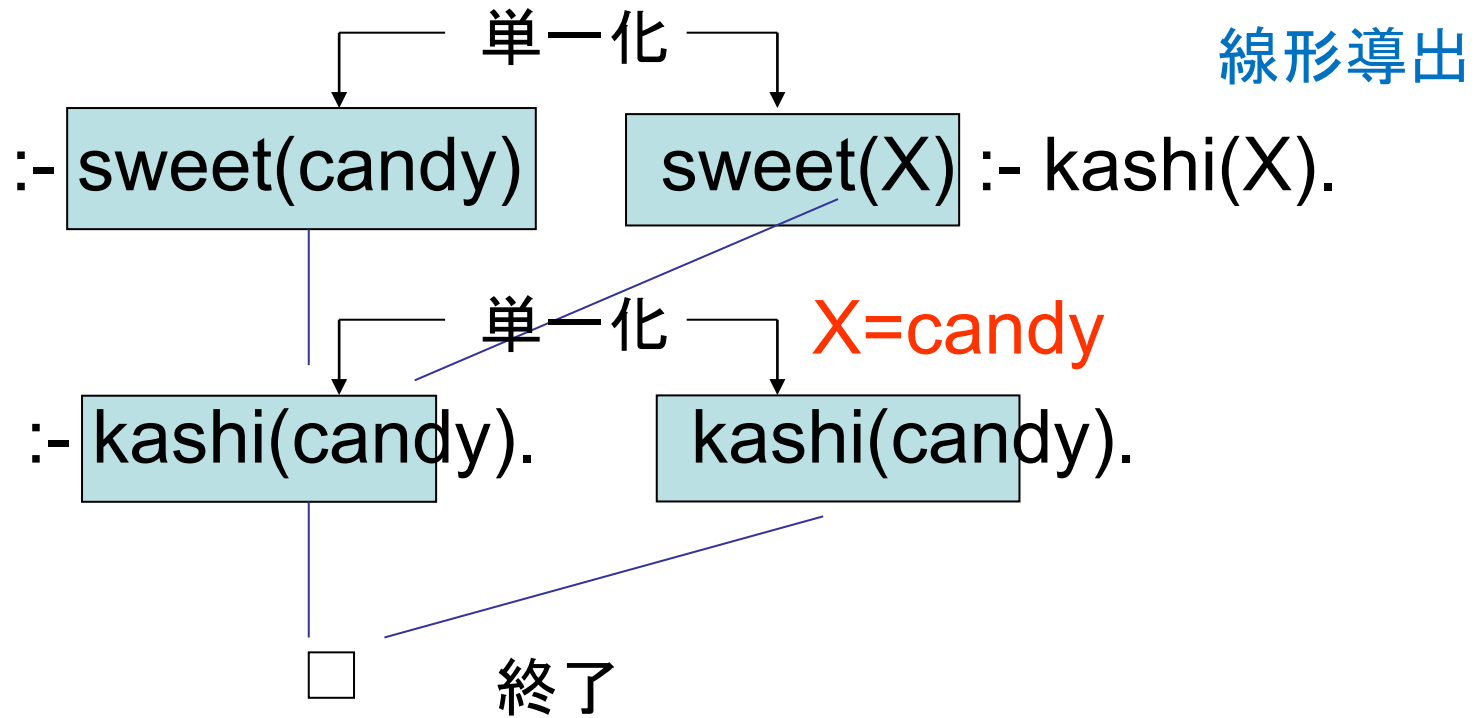
結果をその節のボディに反映させ、ボディ部を新たなゴール節とみなして解こうとする



失敗

別の節で試みる(バックトラック)

PROLOGの実行(10)



ゴール節のゴールと単位節または確定節のヘッドの間で単一化を実行

例4

最初の解を求める(1)

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

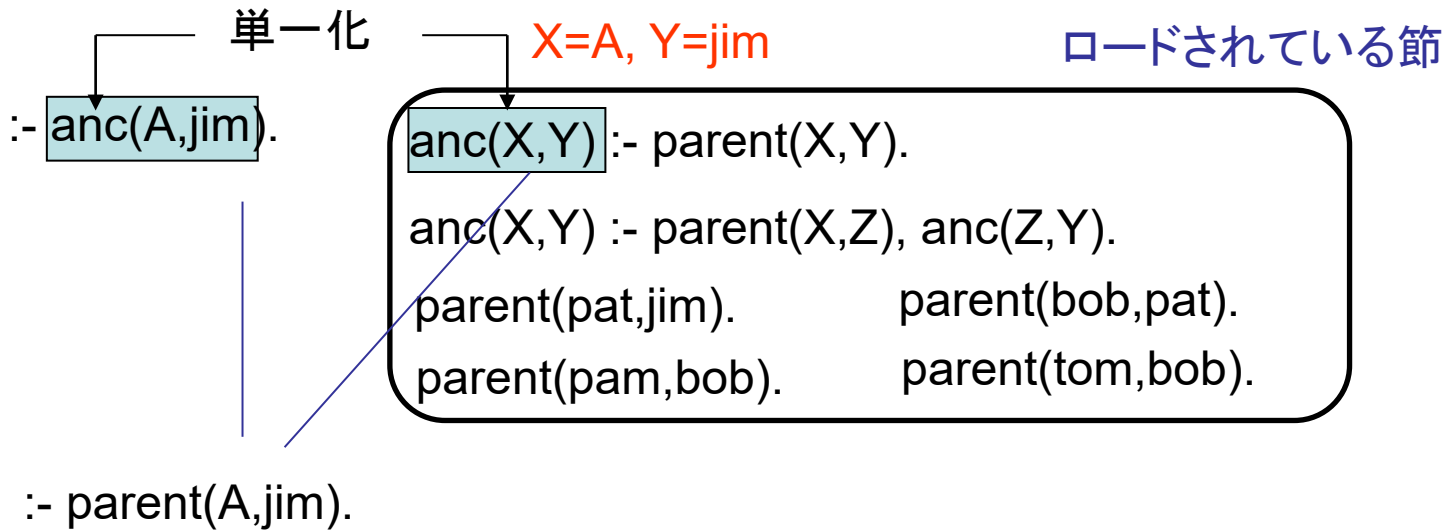
`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

最初の解を求める(2)



最初の解を求める(3)

$X=A, Y=jim$

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`
`anc(X,Y) :- parent(X,Z), anc(Z,Y).`
`parent(pat,jim).` `parent(bob,pat).`
`parent(pam,bob).` `parent(tom,bob).`

単一化

$A=pat$

`:- parent(A,jim).`



終了

`:- anc(A,jim)` に対する解
 $A=pat$ が得られる

強制的にバックトラックさせて(; を打つ)別解を探す

2つ目の解を求める(1)

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

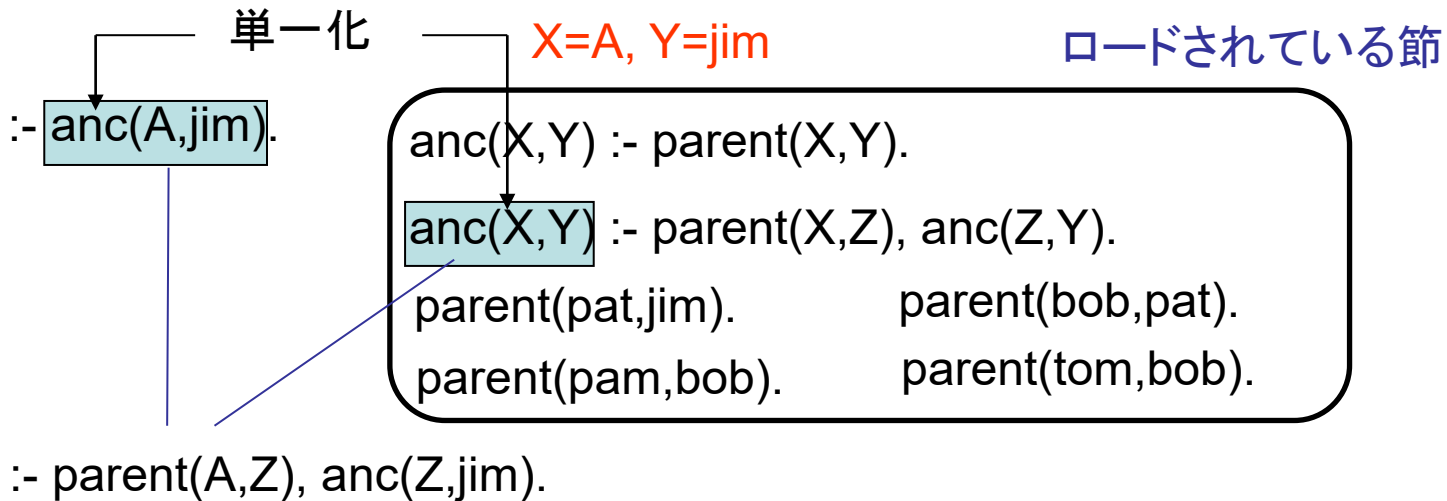
`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

もっとも最近の分岐点に戻る

2つ目の解を求める(2)



anc(A,jim) と単一化可能な節がもう1つあるので、そちらを調べる

2つ目の解を求める(3)

$X=A, Y=jim$

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

単一化

`:- parent(A,Z), anc(Z,jim).`

$A=pat, Z=jim$

`:- parent(pat,jim), anc(jim,jim).`

2つ目の解を求める(4)

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

単一化

`:- parent(A,Z), anc(Z,jim).`

A=pat, Z=jim

`:- parent(pat,jim), anc(jim,jim).`

parent(pat,jim) は true になる

`:- anc(jim,jim).`

2つ目の解を求める(5)

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`
`anc(X,Y) :- parent(X,Z), anc(Z,Y).`
`parent(pat,jim).` `parent(bob,pat).`
`parent(pam,bob).` `parent(tom,bob).`

単一化

`:- parent(A,Z), anc(Z,jim).`

A=pat, Z=jim

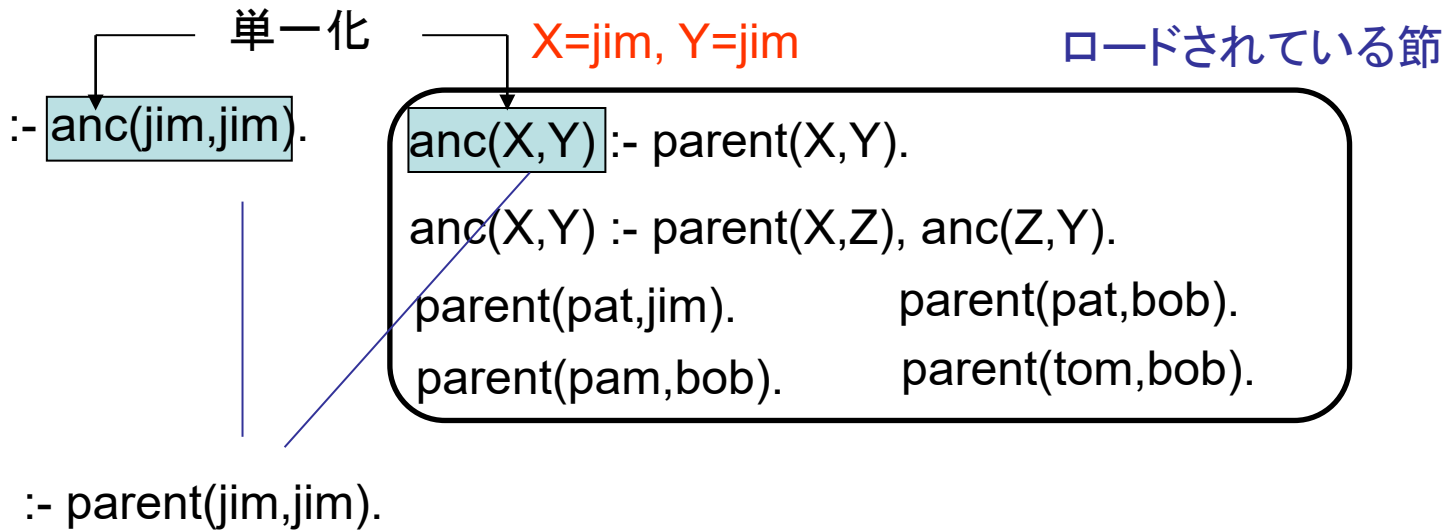
`:- parent(pat,jim), anc(jim,jim).`

parent(pat,jim) は true になる

`:- anc(jim,jim).`

残ったゴールを実行する

残りのゴールの実行(1)



残りのゴールの実行(2)

X=jim, Y=jim

ロードされている節

`:- anc(jim,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(pat,bob).`

`parent(pam,bob).`

`parent(tom,bob).`

`:- parent(jim,jim).`

単一化できるゴールがない

false

失敗

自動的にバックトラックして別解を探す

バックトラック(1)

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

この単一化が悪かった

単一化

`:- parent(A,Z), anc(Z,jim).`

A=pat, Z=jim

`:- parent(pat,jim), anc(jim,jim).`

parent(pat,jim) は true になる

`:- anc(jim,jim).`

もっとも最近の分岐点に戻る

バックトラック(2)

$X=A, Y=jim$

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).` → `parent(bob,pat).`

`parent(pam,bob).` `parent(tom,bob).`

別の節と単一化

単一化

`:- parent(A,Z), anc(Z,jim).`

parent(A,Z) と単一化可能な節が他にもあるので、そちらを調べる

バックトラック(3)

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).` → `parent(bob,pat).`

`parent(pam,bob).` `parent(tom,bob).`

別の節と単一化

単一化

`:- parent(A,Z), anc(Z,jim).`

A=bob, Z=pat

`:- parent(bob,pat), anc(pat,jim).`

バックトラック(4)

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).` → `parent(bob,pat).`

`parent(pam,bob).` `parent(tom,bob).`

別の節と単一化

単一化

`:- parent(A,Z), anc(Z,jim).`

A=bob, Z=pat

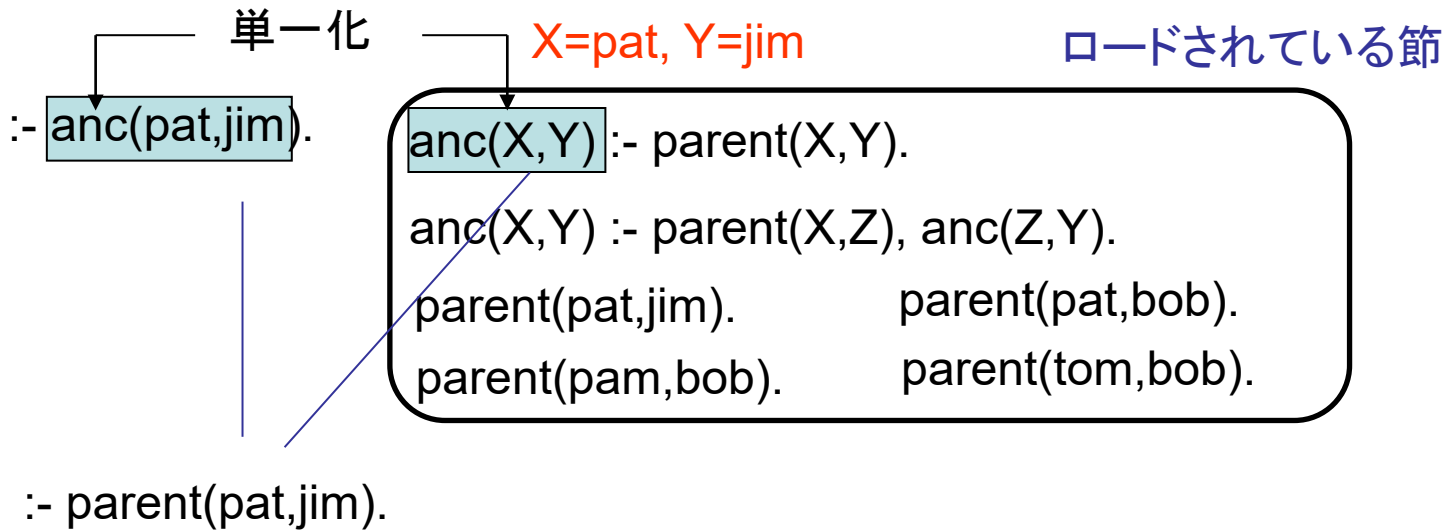
`:- parent(bob,pat), anc(pat,jim).`

parent(bob,pat) は true になる

`:- anc(pat,jim).`

残ったゴールを実行する

後戻り: 残りのゴールの実行(1)



後戻り: 残りのゴールの実行(2)

X=pat, Y=jim

ロードされている節

`:- anc(pat,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

単一化

`:- parent(pat,jim).`



終了

parent(pat,jim) は true になる

後戻りで実行した結果

X=A, Y=jim

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

別の節と単一化

単一化

`:- parent(A,Z), anc(Z,jim).`

A=bob, Z=pat

`:- parent(bob,pat), anc(pat,jim).`

parent(bob,pat) は true になる

`:- anc(pat,jim).`

parent(bob,pat) は true になる

:- anc(A,jim) に対する解
A=bob が得られる

さらに別解を求める

$X=A, Y=jim$

ロードされている節

`:- anc(A,jim).`

`anc(X,Y) :- parent(X,Y).`

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

`parent(pat,jim).`

`parent(bob,pat).`

`parent(pam,bob).`

`parent(tom,bob).`

別の節と単一化

単一化

`:- parent(A,Z), anc(Z,jim).`

$A=bob, Z=pat$

強制的にバックトラックさせて(; を打つ)別解を探す

`:- parent(bob,pat), anc(pat,jim).`

`parent(bob,pat)` は true になる

`:- anc(pat,jim).`

`parent(bob,pat)` は true になる

別解がある(ゴールと単一化可能な節がある)限り解が得られ、なくなると false が返ってくる