Meta–Interpreter 1:

```
prove(true) :- !.
prove((Goal1, Goal2)) :- !, prove(Goal1), prove(Goal2).
prove(Goal) :- clause(Goal, Body), prove(Body).
```

Meta–Interpreter 2:

```
prove(true) :- !.
prove((Goal1, Goal2)) :- !, prove(Goal2), prove(Goal1).
prove(Goal) :- clause(Goal, Body), prove(Body).
```

Meta–Interpreter 3:

```
prove(true,0) :- !.
prove((Goal1,Goal2),N) :- !, prove(Goal1,N1), prove(Goal2,N2),
                             N is N1+N2.
prove(Goal,N) :- clause(Goal, Body), prove(Body,N1),
                 N is N1+1.
```

Classic append:

```
append([],Ys,Ys).
append([X|Xs],Ys,[X|Zs]) :- append(Xs,Ys,Zs).
```

app with difference lists:

```
app(Xs - Ys, Ys - Zs, Xs - Zs).
```

Example grammar $G = (N, T, S, P)$

- $N =$ { Sentence, Nominalphrase, Verbalphrase, Article, Noun, Verb }
- $T =$ { a, the, cat, mouse, scares, hates }
- $S =$ Sentence
- P consists of the following rules:

Sentence	$\rightarrow$	Nominalphrase Verbalphrase
Nominalphrase	$\rightarrow$	Article Noun
Verbalphrase	$\rightarrow$	Verb
Verbalphrase	$\rightarrow$	Verb Nominalphrase
Articlel	$\rightarrow$	a
Article	$\rightarrow$	the
Noun	$\rightarrow$	cat
Noun	$\rightarrow$	mouse
Verb	$\rightarrow$	scares
Verb	$\rightarrow$	hates

$L(G)$ contains: `a cat scares the mouse`, `the mouse hates the cat`,
`a mouse scares a mouse`, `a mouse hates`

Example grammar in Prolog

```
sentence --> nominalphrase, verbalphrase.
nominalphrase --> article, noun.
verbalphrase --> verb.
verbalphrase --> verb, nominalphrase.
article --> [a].
article --> [the].
noun --> [cat].
noun --> [mouse].
verb --> [scares].
verb --> [hates].
```

Transformation into Prolog–clauses

```
sentence(S, R) :- nominalphrase(S, VP), verbalphrase(VP, R)
nominalphrase(NP, R) :- article(NP, N), noun(N, R).
verbalphrase(VP, R) :- verb(VP, R).
verbalphrase(VP, R) :- verb(VP, NP), nominalphrase(NP, R).
article([a | R], R).
article([the | R], R).
noun([cat | R], R).
noun([mouse | R], R).
verb([scares | R], R).
verb([hates | R], R).
```