PROLOG NOTES #3

De Morgan Laws with Operators
:-op(800, xfx, <=>).
:-op(700, xfy, v).
:-op(600, xfy, &).
:-op(500, xfx, ~).

Definition

\( \sim (A \& B) \iff \sim A \lor \sim B. \)

Which is

\( \iff (\sim (A \& B)), v (\sim (A), \sim (B)). \)

Using it:

?- C=false, D=true, \( \sim (C \& D) \iff X. \)
C = true,
D = true,
X = (\sim true v \sim true).

Precedence
It is an integer number. Precedence is the rank of the operator, the stronger the higher the rank (closer to 1). You can always use the () to override the precedence. BTW: () have the rank of 0 – the highest possible.

Associativity

Prefix Unary Operators
oper1 oper2 A
is it
oper1 (oper2 (A)) \ OK!
or
oper1(oper2) A \ illegal!

fx what inside f must have higher precedence (lower precedence number)
oper oper A is illegal

fy what inside f can have lower or equal precedence (higher or equal precedence number)
oper oper A is legal
**Postfix Unary Operators**
A oper1 oper2
is it
oper2 (oper1 (A)) OK!
or
A oper2(oper1) illegal!

xf what inside f must have higher precedence (lower precedence number)
A oper oper is illegal

yf what inside f can have lower or equal precedence (higher or equal precedence number)
A oper oper is legal

**Infix Binary Operators**
A oper1 B oper2 C
is it
oper1(A, oper2(B, C))
or
oper2(oper1(A, B), C)

xfy what on the right can have lower or equal precedence (lower precedence number)
A oper1 B oper2 C is equivalent to oper1(A , B oper2 C)

yfx what on the left can have lower or equal precedence (lower precedence number)
A oper1 B oper2 C is equivalent to oper2(A oper1 B, C)

xf what on both sides is evaluated first
A oper1 B oper2 C is illegal!

yfy no such thing exists

**Grammar**
Grammar of a Language – set of rules for specifying what sequences of words are acceptable as sentences, for example CGF (context-free grammar).
Parse Tree – diagram showing the parse structure of a sentence.
Parsing Problem – construct parse tree
Parser – program that produces parse trees.
DCG (definite clause grammar) – formalism for grammar rules
Prolog Grammar Rule Notation – syntactic shorthand for ordinary Prolog code.