CISC 301

Handout: Conversion to CNF/Clausal Form

A literal is either an atomic formula (positive literal) or the negation of an atomic formula (negative literal). A clause contains literals and is interpreted as their disjunction. If a clause contains the literals $l_1, \ldots, l_k$ then we will write it as $[l_1, \ldots, l_k]$ rather than the notation $\{l_1, \ldots, l_k\}$ used in the textbook. Thus, $[l_1, \ldots, l_k]$ is interpreted the same way as the formula $(l_1 \vee \ldots \vee l_k)$.

Here is a (*non-deterministic*) algorithm to convert a formula, F, (assumed to not include $\leftrightarrow$) into a set of clauses. Start with $\{[F]\}$.

At any point, we will have a set that has the form, $\{C_1, \ldots, C_i, \ldots, C_k\}$, where the $C's$ have the form $[F_1, \ldots F_n]$, and the $F's$ are formulae. We are finished with a C if every formula in C is already a literal.

While not done do (i.e., some C has a non-literal)

Let C_i include a non-literal.

WLOG, we can express C_i as $[F_1, \ldots F_{n-1}, F_n]$ where F_n is a non-literal.

Case 1: $F_n = \neg\neg G$, for some formula G.

Replace C_i by $[F_1, \ldots F_{n-1}, G]$.

Case 2: (disjunctive case).

Case 2a: $F_n = (F \vee G)$ for some F, G.

Replace C_i by $[F_1, \ldots F_{n-1}, F, G]$.

Case 2b: $F_n = (F \rightarrow G)$ for some F, G.

Replace C_i by $[F_1, \ldots F_{n-1}, \neg F, G]$.

Case 2c: $F_n = \neg(F \wedge G)$ for some F, G.

Replace C_i by $[F_1, \ldots F_{n-1}, \neg F, \neg G]$.

Case 3: (conjunctive case).

Case 3a: $F_n = (F \wedge G)$ for some F, G.

Replace C_i by C_i^1 and C_i^2, where

$C_i^1 = [F_1, \ldots F_{n-1}, F]$, and

$C_i^2 = [F_1, \ldots F_{n-1}, G]$

Case 3b: $F_n = \neg(F \rightarrow G)$ for some F, G.

Replace C_i by C_i^1 and C_i^2, where

$C_i^1 = [F_1, \ldots F_{n-1}, F]$, and

$C_i^2 = [F_1, \ldots F_{n-1}, \neg G]$

Case 3c: $F_n = \neg(F \vee G)$ for some F, G.

Replace C_i by C_i^1 and C_i^2, where

$C_i^1 = [F_1, \ldots F_{n-1}, \neg F]$, and

$C_i^2 = [F_1, \ldots F_{n-1}, \neg G]$

Example: Converting $F = (\neg A \vee B) \rightarrow (C \vee A)$ into a set of clauses:

$\{[(\neg A \vee B) \rightarrow (C \vee A)]\}$ (starting with $\{[F]\}$)

$\{[\neg(\neg A \vee B), (C \vee A)]\}$ (applying rule 2b)

$\{[\neg(\neg A \vee B), C, A]\}$ (applying rule 2a)

$\{[\neg\neg A, C, A], [\neg B, C, A]\}$ (applying rule 3c)

$\{[A, C], [A \neg B, C]\}$ (applying rule 1) Note we write $[A, C]$ rather than $[A, C, A]$ and that $[\neg B, C, A]$ can also be written as $[A, \neg B, C]$.