CSE 142 Computer Programming I

Complex Conditions

© 2000 UW CSE

Overview

Concepts this lecture Complex conditions Boolean operators Negating a condition Truth tables DeMorgan's laws

Complex Conditionals

if I have at least \$15 or you have at least \$15, then we can go to the movies 1-1

if the temperature is below 32 degrees and it's raining, then it's snowing

if it's not the case that it's Saturday or Sunday, then it's a work day

Boolean Operators in C

Complex conditionals often involve words like AND, OR, and NOT, and sometimes TRUE or FALSE

I-2

The Boolean operators AND, OR, and NOT have these symbols in C: & & || 1

and

As we know, TRUE and FALSE are not built-in concepts in C. You can define symbols: #define TRUE 1 #define FALSE 0

or not

Complex Conditionals in C

if I have at least \$15 or you have at least \$15, then we can go to the movies:

if (myMoney>=15.0 <mark>||</mark> yourMoney>=15.0) canGoToMovies = TRUE;

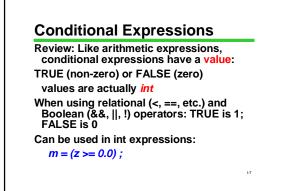
if the temperature is below 32 degrees and it's raining, then it's snowing:

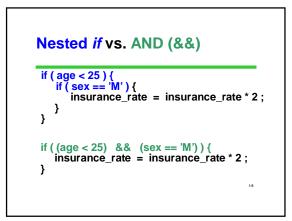
if (temperature<32.0 && raining) snowing = TRUE;

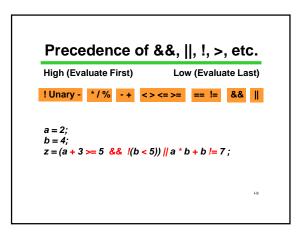
I-5

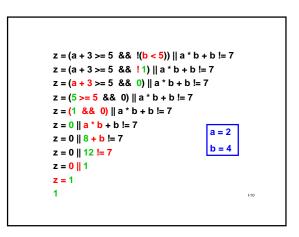
An Example with ! if it's not the case that it's Saturday or Sunday, then it's a work day: weekday = FALSE; if (!(today==6 || today==7)) weekday = TRUE; if (weekday) mustWork = TRUE;

1-6







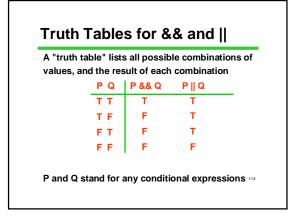


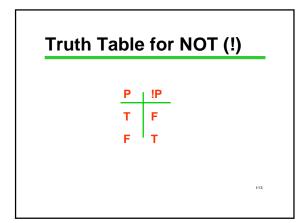


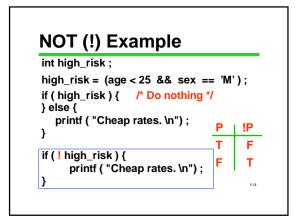
Suppose we want a while loop to terminate as soon as either x is 17 or x is 42 Which is it? while (x!=17 || x!=42) ... while (x!=17 && x!=42) ...

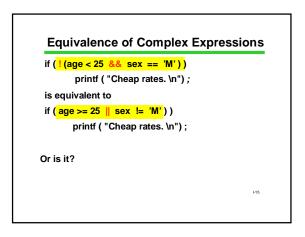
either way? something else?

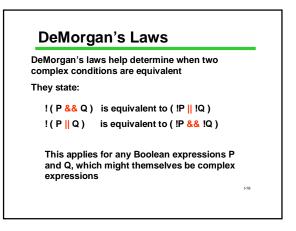
Truth tables and DeMorgan's laws give us tools for answering such questions

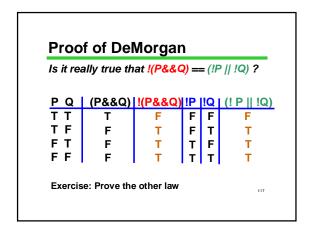








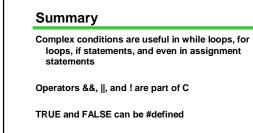




Is it really true that !(P&&Q) == (!P !Q) ?					
ΡQ	(P&&Q)	!(P&&Q)	!P	!Q	(! P !Q)
ТΤ	Т	F	F	F	F
ΤF	F	т	F	Т	т
FΤ	F	Т	т	F	т
FF	F	т	т	т	т

Solution To a Previous Question

We wanted a while loop to terminate as soon as either x is 17 or x is 42. I.e., loop should terminate if (x==17 || x==42) So the loop condition is while (! (x==17 || x==42) ... Using DeMorgan's laws, we can rewrite as while (x != 17 & x != 42) ... A truth table would show that while (x != 17 || x != 42) is wrong!



Truth tables and DeMorgan's laws help evaluate complex expressions