1. [Start of transcript. Skip to the end.](https://courses.edx.org/xblock/block-v1:ColumbiaX+CSMM.101x+3T2020+type@vertical+block@2318aad15f6140f89f062f87a85f4700?show_title=0&show_bookmark_button=0#transcript-end-88cc8681f4974b918172972cc914927f)
2. So now let's show how we can convert a propositional logic
3. formula into a CNF form.
4. So suppose we have breeze in 1,1,
5. if and only if a pit in 1,2 or a pit in 2,1, all right?
6. So the way to simplify this into a CNF,
7. or rewriting this into a CNF is first to eliminate
8. the double implication, or the if and only.
9. And the reason is that we only want
10. to use the connectives, the conjunction, the disjunction,
11. and the negation, right?
12. We are not allowed to use any double implication
13. or implication in the form.
14. All right, that's easy.
15. We know that B1,1, if and only if this term here,
16. could be written as the conjunction of the implication
17. from B1,1 to the right term.
18. And P1,1 or P2,1 implies B1,1.
19. So the first is to eliminate the double implication.
20. The second step is to then eliminate the implication.
21. We know, for example from the previous slides,
22. that alpha implies beta is logically
23. equivalent to not alpha or beta.
24. In other words, if you look at their truth tables--
25. so if you do the truth table of alpha implies beta,
26. and not alpha or beta, you will find that actually they
27. have the same truth values.
28. So these two are logically equivalent.
29. All right, so if we have this, then we
30. are going to replace this implication
31. by not alpha or beta and this implication by not alpha
32. or beta.
33. OK, so this implication would be not B1,1 or P1,2 or P2,1.
34. And this one would be not P1,2 or P2,1.
35. This is the negation alpha, or beta which would be B1,1,
36. all right?
37. So the next step after-- so now we
38. are happy that we got rid of the implication,
39. and of the double implication.
40. Now we have only disjunction conjunction.
41. But we want to write this as a conjunctive normal form, which
42. is a conjunction of disjunction of literals.
43. And it's not the case yet, right?
44. Because we have the negation of our brackets.
45. And we know that actually we need
46. to use DeMorgan's Law to introduce the negation
47. over these literals here.
48. So the next step would be then to move the negation inwards
49. using DeMorgan's law and double negation.
50. So this would lead us to-- this term is ready, right?
51. So it's the disjunction of literals.
52. This term here would be converted into not P1,2 and not
53. P2,1, or B1,1.
54. So from this step to step, it's DeMorgan's Law, all right?
55. So now we have disjunction of literals.
56. And here we don't have disjunction of literals.
57. So we're going to distribute the disjunction
58. over the conjunction to be able to present that in form we
59. want.
60. The next step is to apply distributivity
61. of the disjunction over the conjunction and to flatten out.
62. All right, so this term is in the desired form.
63. This one would become not P1,2, or B1,1,
64. by distributing this guy over to the conjunction,
65. and B1,2 or not P2,1, all right?
66. So and now we have the conjunction of three clauses
67. that are actually disjunction of literals
68. which are positive or negative.
69. So this is how we transform by these four steps.
70. We eliminate the double implication.
71. We eliminate the implication.
72. We use DeMorgan's Law to introduce the negation inside.
73. And we flatten and distribute the disjunction
74. over the conjunction to obtain the final result that
75. is actually the conjunction's normal form.
76. Let's now examine together the resolution algorithm
77. that actually adopts a proof by contradiction.
78. In other words, in order to show that KB entails alpha,
79. it's going to show that actually KB and the negation of alpha
80. far is unsatisfiable.
81. OK, so we have initially a knowledge base.
82. We have alpha, and this function,
83. PL resolution will return either true, if KB eentails alpha,
84. falls otherwise, all right?
85. OK, so initially we have KB expressed
86. in propositional logic, alpha expressed
87. in propositional logic.
88. We build KB a negation of alpha that we transform
89. into a CNF representation.
90. So it's important that the clauses used are in CNF form.
91. So after this conversion, we are going to create new clauses.
92. We have a set, new, equal to the empty set.
93. And we are going to repeat the process of applying
94. the resolution rule.
95. In other words, the function here that we use
96. is called PL resolve.
97. We'll take two clauses, Ci and Cj, and we'll resolve them.
98. In other words, it will return the set
99. of all possible clauses obtained by resolving the inputs.
100. All right, so each pair containing
101. complementary literals will be resolved
102. and reproduce new resolvents.
103. If the resolvents contain the empty clause--
104. empty clause means actually that we reached a contradiction.
105. And actually means that two of the clauses resolve to nothing.
106. And this is a contradiction.
107. This means that we need to return true
108. because KB and the negation of alpha is unsatisfiable.
109. In this case, we're going to return true.
110. Otherwise, if the resolvent does not contain the empty clause,
111. and we are not creating any new resolvent,
112. then we are going to return false,
113. means that KB does not entail alpha,
114. or there are no new clauses added to KB, all right?
115. So these are the two outcomes.
116. We're going to continue the process of resolving
117. each pair of clauses, until either we come up
118. with the empty clause, nothing to do anymore,
119. there is a contradiction, then KB infers alpha.
120. Or we're not generating new clauses anymore,
121. which means that we are done.
122. KB and negation of alpha is actually satisfiable.
123. So this is how the resolution algorithm works.
124. So let's see the resolution algorithm through this example.
125. Suppose we have a knowledge base in which we
126. have rule R2 and rule R4.
127. So this constitutes the conjunction of these two terms
128. here, negation of B1,1 and B1,1 equivalent to P1,2 or P2,1.
129. We want to prove that actually this knowledge base entails
130. the formula not P1,2 .
131. To do so, we are going to consider KB and the negation
132. of not P1,2, which would be the negation of the negation
133. of P1,2, which is equivalent to KB and P1,2.
134. So we take PKB and P1,2.
135. This is what we want to prove contradiction on.
136. In other words, we want to show that this is actually not
137. satisfiable, right?
138. So we're going to first start by writing KB and the negation
139. of alpha, which is KB and P1,2 in CNF form.
140. This will lead us to the proposition not P2,1 or P1,1,
141. and this second clause, and this second clause, and et cetera.
142. So it is the conjunction of disjunction of literals.
143. All right, so P1,2 comes from actually the negation of alpha.
144. All right, so then PL resolve will take pairs of clauses
145. and resolve them.
146. So for example, if I take this clause here and this clause
147. here and resolve, let's say, for P2,1.
148. So we have here P2,1 and P2,1 here.
149. So this is the negation of this literal here.
150. These two will resolve to bring me to the resolution form
151. of negation of B1,1, or P1,2, or B1,1.
152. So these two would resolve and we obtain this new resolvent
153. here.
154. We're going to do the same thing for--
155. you obtain this one by resolving these two terms here.
156. And here we are simplifying.
157. So we have P1,2, P2,1, and P1,2.
158. So you're going to resolve based on the B1,1.
159. So if we actually resolve B1,1 and the negation of B1,1.
160. We're going to get rid of that and we obtain only these three
161. terms here.
162. And so and so forth, we're going to take all pairs.
163. And it happened that actually as we kept going, resolving.
164. The process group is going on.
165. We're going to reach a point in which we have the resolving
166. P1,2 and negation of P1,2, which would bring us to the empty
167. clause.
168. The empty clause means actually that KB and the negation
169. of alpha is not satisfiable.
170. in other words, KB actually does entail alpha, all right?
171. So this is how the resolution algorithm works.