1. [Start of transcript. Skip to the end.](https://courses.edx.org/xblock/block-v1:ColumbiaX+CSMM.101x+3T2020+type@vertical+block@effdb232521a4d8ea6ec2dca6de5a1bb?show_title=0&show_bookmark_button=0#transcript-end-629a396be97442369350b5bf343fcbcc)
2. We just saw that proof by resolution
3. is a way to address completeness in inference rules.
4. It is a powerful method and it is proven
5. to be sound and complete.
6. However, it is not always needed because it is doing extra work.
7. In fact, it is exponential in a number of closes.
8. In some knowledge base, we have a restricted form
9. of the sentences that actually allow for simpler
10. and more efficient algorithms.
11. And this form could be the conjunction of horn clauses.
12. So I said that there is two ways to address completeness--
13. either use resolution algorithms or use backward
14. chaining, forward chaining on horn clauses.
15. So what is a horn clause?
16. We saw that earlier.
17. It's logic propositions of the form p1 and p2, pn implies q.
18. So we have this formulation here.
19. And we use Modus Ponens as inference rules
20. on horn clauses.
21. That has proven to be sound and complete.
22. So if we have p1, p2, pn, and we have p1 and p2
23. and pn implies q, then we can infer q.
24. So this is by--
25. simply by Modus Ponens on horn clauses.
26. So given this formation--
27. and if we can express the knowledge base
28. as a conjunction of such horn clauses,
29. then we are going to be able to use methods that
30. are linear in time such as forward chaining and backward
31. chaining.
32. Let's start first with forward chaining.
33. The idea of forward chaining is to fire any rule whose premises
34. are satisfied in the knowledge base,
35. add its conclusions to the knowledge base
36. until you reach the query that we need to reach.
37. OK, for example, if we have this knowledge base here
38. and you want to show that KB entails the formula q,
39. then we're going to start with the facts
40. and then move forward by adding all the conclusions based
41. on this fact until we reach q.
42. So we're going to represent this with this graph
43. to make it easy to follow.
44. So we are going to have this indication p implies q here.
45. Then we have-- we have two facts-- a and b.
46. We have l and m implies p.
47. So these are all horn clauses formula.
48. We have l and m implies p.
49. So if we have l and we have m-- so we have this conjunction
50. here that leads to p.
51. If we have b and l, then we have m. b and l leads to m.
52. a and p-- so you could see here a and p leads to l.
53. So if I have a and I have p-- so this is just written this way--
54. the conjunction of a and p leads to l
55. and then the conjunction of a and b leads to l.
56. And then we have the fact a and the fact b.
57. So we're going to see how we're actually
58. going to fire those premises until we reach q.
59. So we're going to start from the fact a and b all the way up
60. until we reach q.
61. So this is how forward chaining works on this example.
62. We're going to first put numbers on each of these conjunctions.
63. We need two elements in the premise to fire the conjunction
64. a and b implies l.
65. So we need two.
66. We need two to fire the rule a and p implies l.
67. But we need only one to fire the rule p implies q.
68. Same thing for l and b implies m and l and m implies p.
69. OK, so going to put this number first.
70. This will help us keep track of the number of elements needed
71. or the proposition needed to fire the rule--
72. horn clause rule.
73. Next, so we observe a, its effect in the knowledge base.
74. This will reduce the number from two to one
75. because we only need to fire b to fire
76. the rule a and b implies l.
77. So we're going to keep going.
78. We observe b.
79. The number of elements to needed is zero.
80. So we're going to fire the rule a and b implies l.
81. From a and b implies l, we observe that actually now we
82. have l and we have b.
83. Then we can fire the rule l and b implies m.
84. So we keep going.
85. From that, you could deduce the rule n and m implies p.
86. And from p, we could either-- we could both fire
87. the rule p and a implies l.
88. So we double fire this rule.
89. So this is the second time.
90. And from p-- and actually we only
91. needed one item to reach q.
92. So from p, we fire q.
93. And therefore, we found q.
94. So this is how forward chaining works.
95. How about backward chaining?
96. Backward chaining works backwards from the query down
97. to the facts.
98. In other words, to prove q by backward chaining, what
99. we need to do is first check if q is known already.
100. If it's known already, there's nothing to do.
101. It's proven.
102. Otherwise, you're going to prove backwards
103. chaining all premises of some rules concluding in q.
104. We're going to try to avoid loops
105. because we need to check if a new sub goal is already
106. on the goal stack.
107. And we avoid repeated work.
108. In other words, check if a new sub
109. goal has already been proven true or has already failed.
110. So we're going to show that again
111. on the example on the same example.
112. Let's see backward chaining now through an example.
113. The idea of backward chaining is to start from the query q
114. and find the implications in the knowledge base that lead to q.
115. The idea is then to go down in the graph
116. until it reaches the facts here, a and b,
117. and bring that information up.
118. All right, so we start with q.
119. We identify that for q, we need p.
120. To get p, we need both lm.
121. To get l, we need both a and p.
122. To get-- et cetera.
123. So you're going to keep going with this process
124. until we find a and b.
125. From a and b, we get l.
126. From b and l, we get m.
127. From m and l, we get p, et cetera, and we recover q.
128. So the process is really targeted
129. toward finding the query q.
130. And the way it compares to forward chaining is as follows.
131. So first of all, forward chaining is data driven.
132. In other words, it's automatic and unconscious processing.
133. It may do lots of work until it finds eventually the query q.
134. Backward chaining is considered as a goal
135. driven method that actually is appropriate for problem
136. solving.
137. In fact, the complexity of bc or backward chaining
138. can be much less than linear time
139. in the size of the knowledge base.