Problem: Rule-Based Book Recommendations

Part A: Forward chaining

You need book recommendations for two of your friends, so you decide to use your forward-chaining book recommender.

Here's what you know.

Database of assertions:

(Max lives-in WashingtonDC)
(Jane lives-in SanFrancisco)
(Max likes science-fiction)
(Jane likes PhilipKDick)
(Pat likes TheThreeStigmataOfPalmerEldritch)
(PhilipKDick is-author-of Ubik)
(PhilipKDick is-author-of TheManInTheHighCastle)
(PhilipKDick is-author-of ThePenultimateTruth)

Rules:

R1 if (?x likes PhilipKDick)
then (?x likes science-fiction)

R2 if (?x likes Ubik)
then (?x likes alternate-realities)

R3 if (?x lives-in SanFrancisco)
(?x likes science-fiction)
then (?x likes alternate-realities)

R4 if (?x lives-in WashingtonDC)
then (?x likes politics)

R5 if (?x likes politics)
(?x likes science-fiction)
then (ThePenultimateTruth is-recommended-for ?x)

R6 if (?x likes alternate-realities)
then (TheManInTheHighCastle is-recommend-for ?x)

Fill out the following table to show the details of running the forward chainer. Use rule ordering for the conflict resolution strategy. Assume new assertions are added after already existing ones. Terminate when no further assertions can be made. You may abbreviate clauses as long as there is no ambiguity. (Note: there may be more lines in the table than you need.) **NOTE in solution: how we use the strategy of "pick the first rule <u>in the database</u> to resolve conflicts (e.g., R3 > R4 in step 2).**

Step	Triggered Rule(s)	Rule Instance Binding(s)	Rule Fired	Database Assertion(s) Added
1	R1	?x = Jane	R1	(Jane likes science-fiction)
	R4	?x = Max		
2	R3	?x = Jane	R3	(Jane likes alternate-realities)
	R4	?x = Max		
3	R4	?x = Max	R4	(Max likes politics)
	R6	?x = Jane		
4	R5	?x = Max	R5	(ThePenultimateTruth is-recommended-for Max)
	R6	?x = Jane		
5	R6	?x = Jane	R6	(TheManInTheHighCastle is-recommended-for Jane)

Part B: Backward chaining

One of your friends suggests that Pat might like TheManInTheHighCastle, but you want your backward chainer to help you prove whether or not that statement is true.

You use the same assertions as in your forward chaining system, plus a new assertion:
(Pat lives-in SanFrancisco)

You also use the same 6 rules as in your forward chaining system, plus a new rule:

R7 if (?x likes TheThreeStigmataOfPalmerEldritch)
then (?x likes PhilipKDick)

You then ask your backward chainer to prove the following assertion:
(TheManInTheHighCastle is-recommend-for Pat)

Using this assertion as the root node, draw the goal (and/or) tree that your system uses to prove the assertion. (The root node is provided below.) Assume that your system uses rule ordering as a conflict resolution strategy. Also assume that if an assertion cannot be proven via rules or existing assertions, that it fails. (In other words, your system does not query you for an answer.) Label each branch of the tree with the name of the rule (e.g. R1) that it represents.