Operator Interaction

Right now there is no end place for our station keeping plan. Let's add an end place that is triggered by the operator.

We'll do this with the "→OperatorApprove" output event. When activated, the operator will see a "Yes / No" dialog box on the GUI.

- If the operator clicks Yes, the dialog box closes and an "←OperatorApproved" input event is generated
- If the operator clicks No, the dialog box closes and an "←OperatorRejected" input event is generated
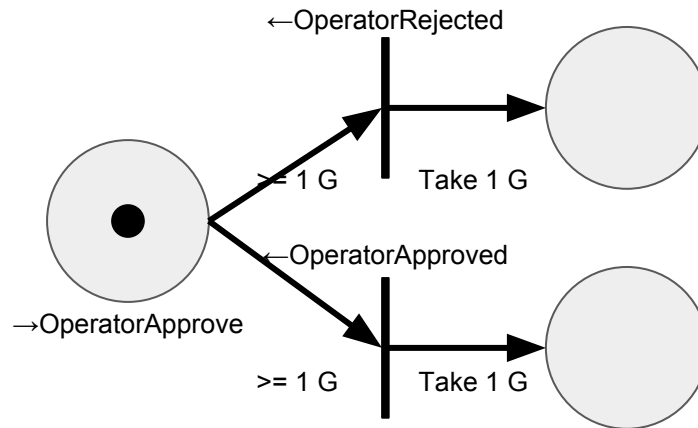
Note that this interaction does not depend on any Robots, which means two things:

- The generated input event will not have any Relevant Tokens
- We can use a Generic token to activate the →OperatorApprove output event

A Robot token can also be used to activate →OperatorApprove, but the generated ←OperatorApproved or ←OperatorRejected input event still will not have any Relevant Tokens.

Remember that the inverse is not true: an output event that uses robots (→RobotGotoLocation) can only be activated by Robot tokens and not Generic tokens, while output events that don't use robots can be activated by Robot or Generic tokens.

Below is an example sequence activating a →OperatorApprove with a Generic token. Note that instead of ">= 1" and "Take 1" as in previous lessons, we now specify the type of token, such as Generic (G), Robot, or Relevant Token (RT).

←OperatorRejected

\>= 1 G   Take 1 G

←OperatorApproved

→OperatorApprove

\>= 1 G   Take 1 G

Here is what the SPN would look like if the the operator chooses "No"

←OperatorRejected

>= 1 G   Take 1 G

←OperatorApproved

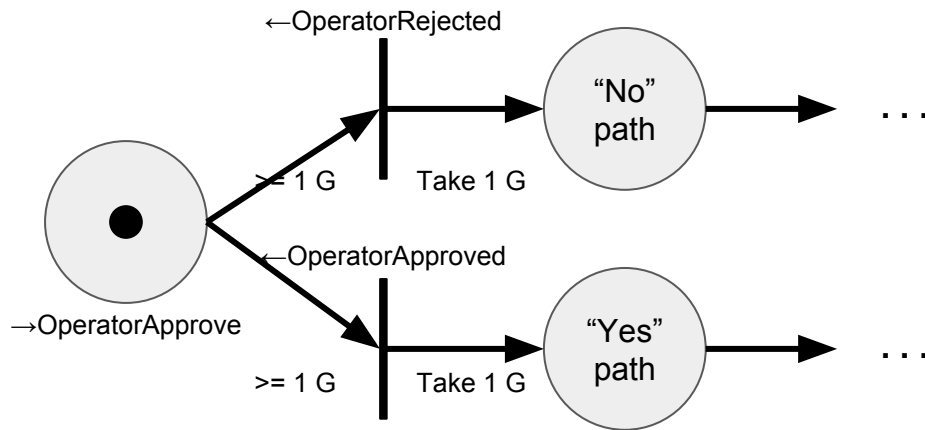→OperatorApprove

>= 1 G   Take 1 G
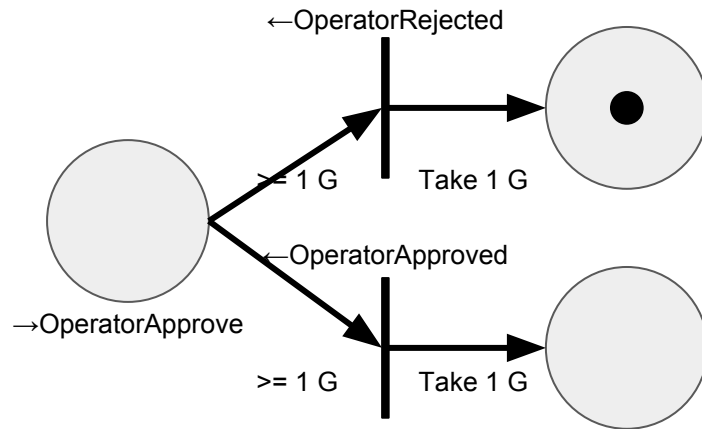
Here is what the SPN would look like if the the operator chooses "Yes"
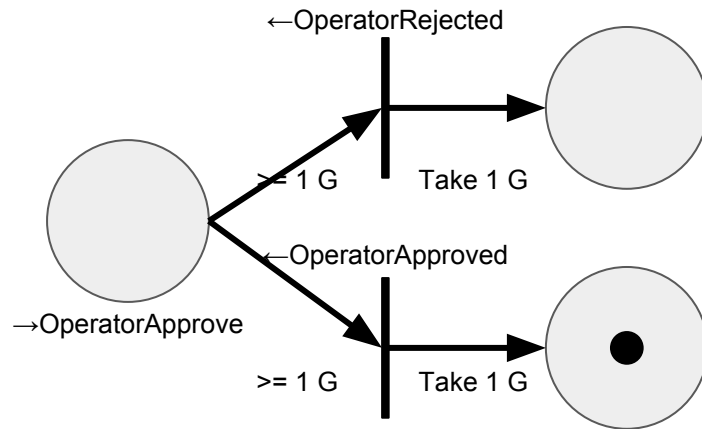
Notice that there are two different paths: the "No" path and the "Yes" path, which split out of the "→OperatorApprove" place. This is called branching.

In this case, we will only enter one of the two branches, depending on what the operator chooses.

←OperatorRejected

"No" path . . .

>= 1 G   Take 1 G

→OperatorApprove

←OperatorApproved

"Yes" path . . .

>= 1 G   Take 1 G

Now we need to decide how to use the sequence in our station keep SPN.

We can use →OperatorApprove to ask the operator to approve ending the SPN

- If they answer yes, the plan should end
- If they answer no, the operator should be asked again later

Let's look at adding the →OperatorApprove output event to an existing place in the SPN and consider when the dialog box will pop up. Remember that we can have multiple output events on a place and multiple input events on a transition.
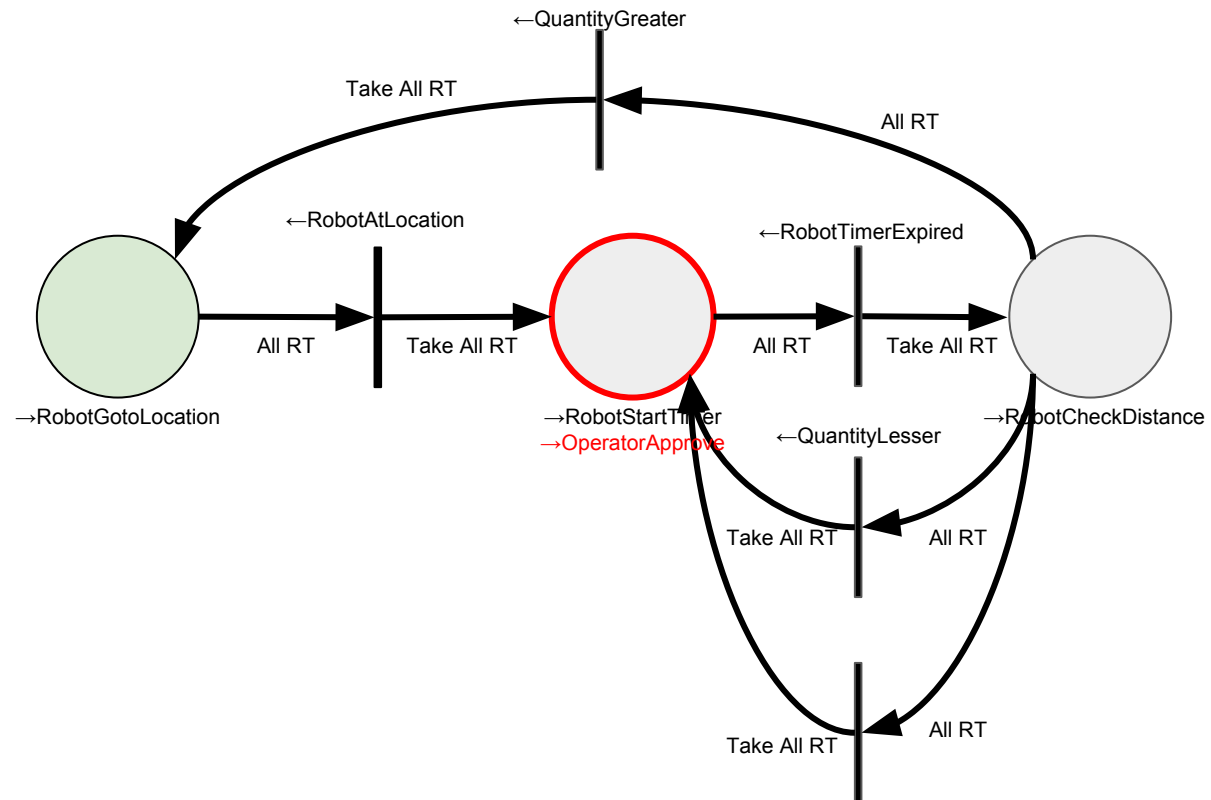
QUIZ 7-1: When will the dialog box appear if we add →OperatorApprove to the list of output events on the first place?
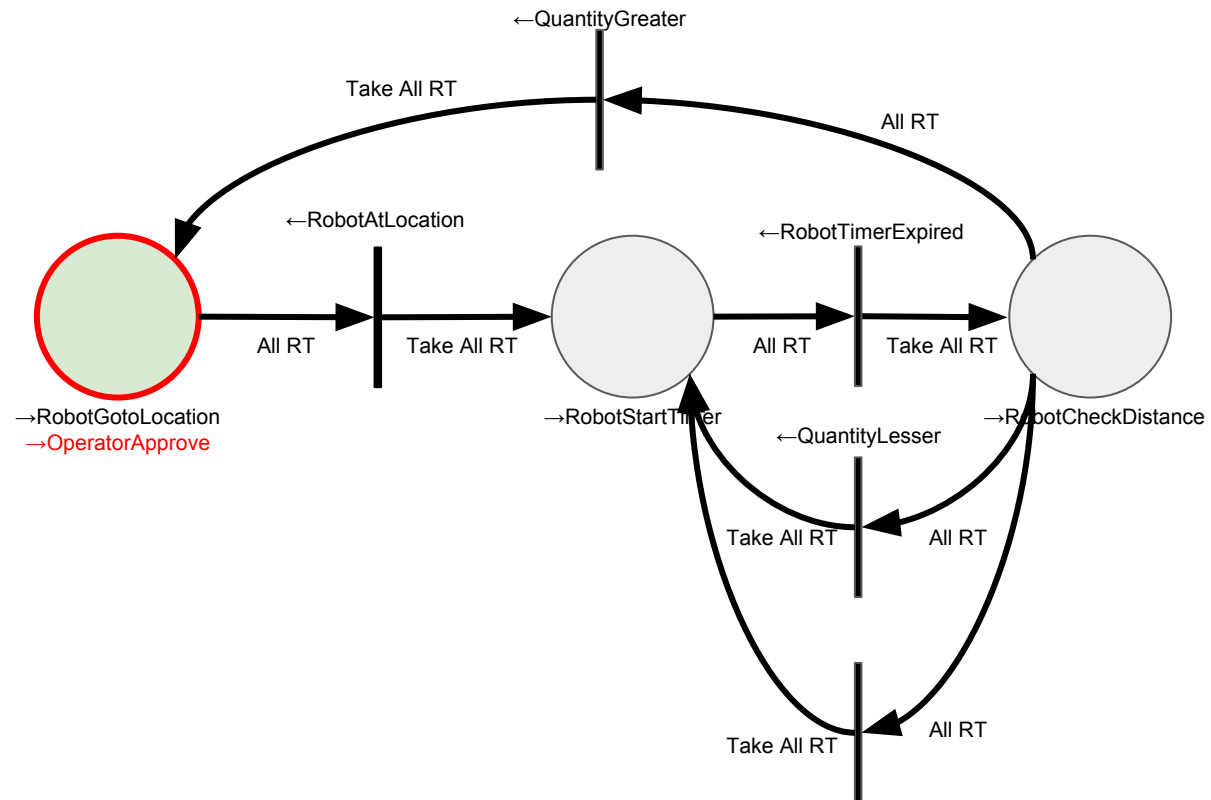
Quiz 7-1 Solution

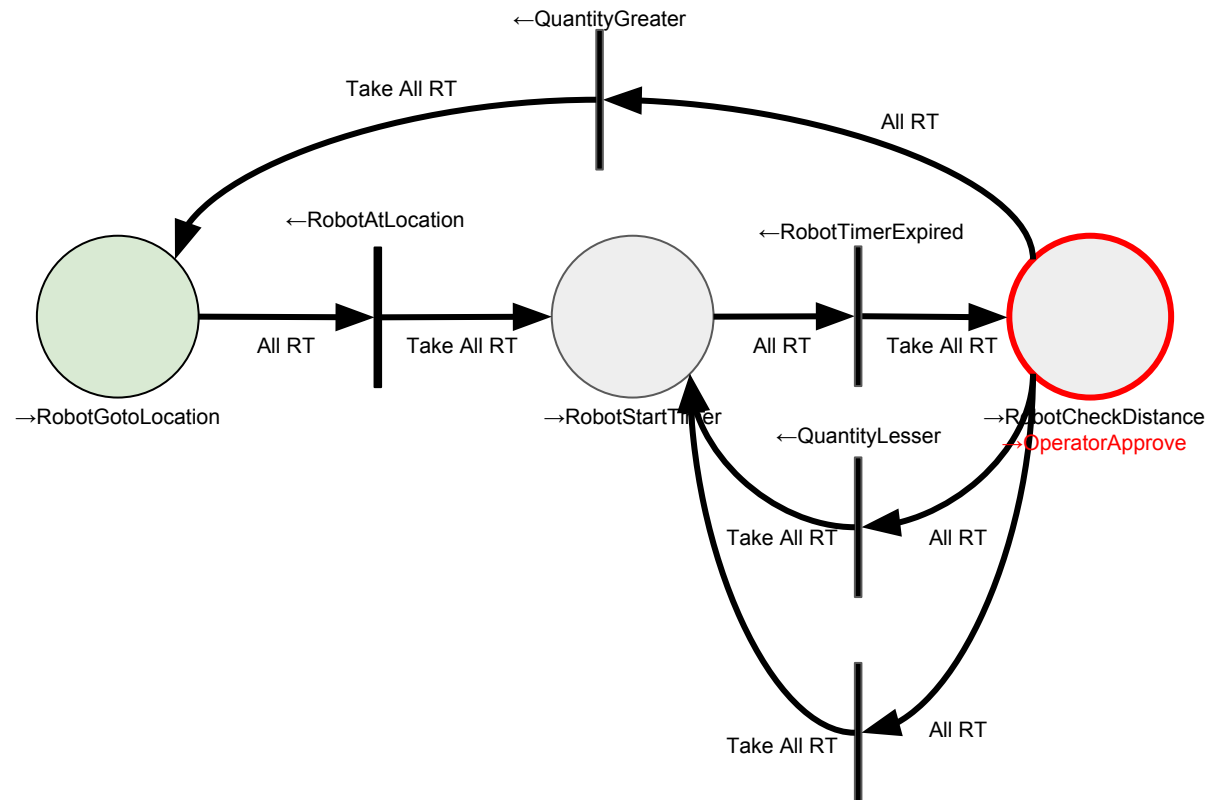Every time any robot is told to move to the station keep location.

QUIZ 7-2: When will the dialog box appear if we add →OperatorApprove to the list of output events on the second place?

←QuantityGreater

Take All RT

All RT

←RobotAtLocation

←RobotTimerExpired

All RT

Take All RT

All RT

Take All RT

→RobotGotoLocation

→RobotStartTimer
→OperatorApprove

→RobotCheckDistance

←QuantityLesser

Take All RT

All RT

Take All RT

All RT

Quiz 7-2 Solution

Every time any robot reaches the station keep location/a robot starts waiting.

QUIZ 7-3: When will the dialog box appear if we add →OperatorApprove to the list of output events on the third place?

Quiz 7-3 Solution

Every time any robot finishes waiting/checks its distance to the station keep location.

To avoid the dialog box from popping up continuously, we need the operator decision to be outside of the station keep loop.
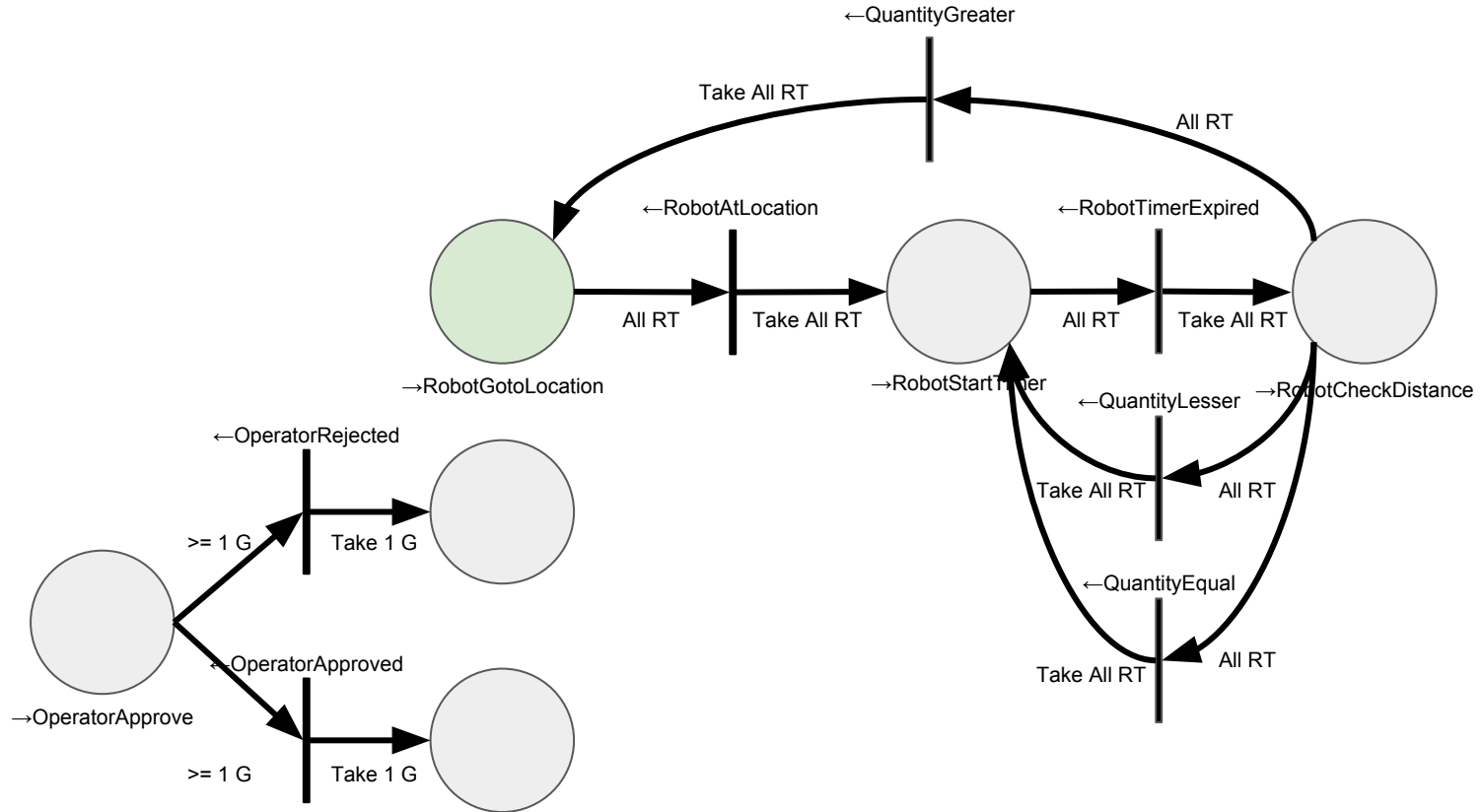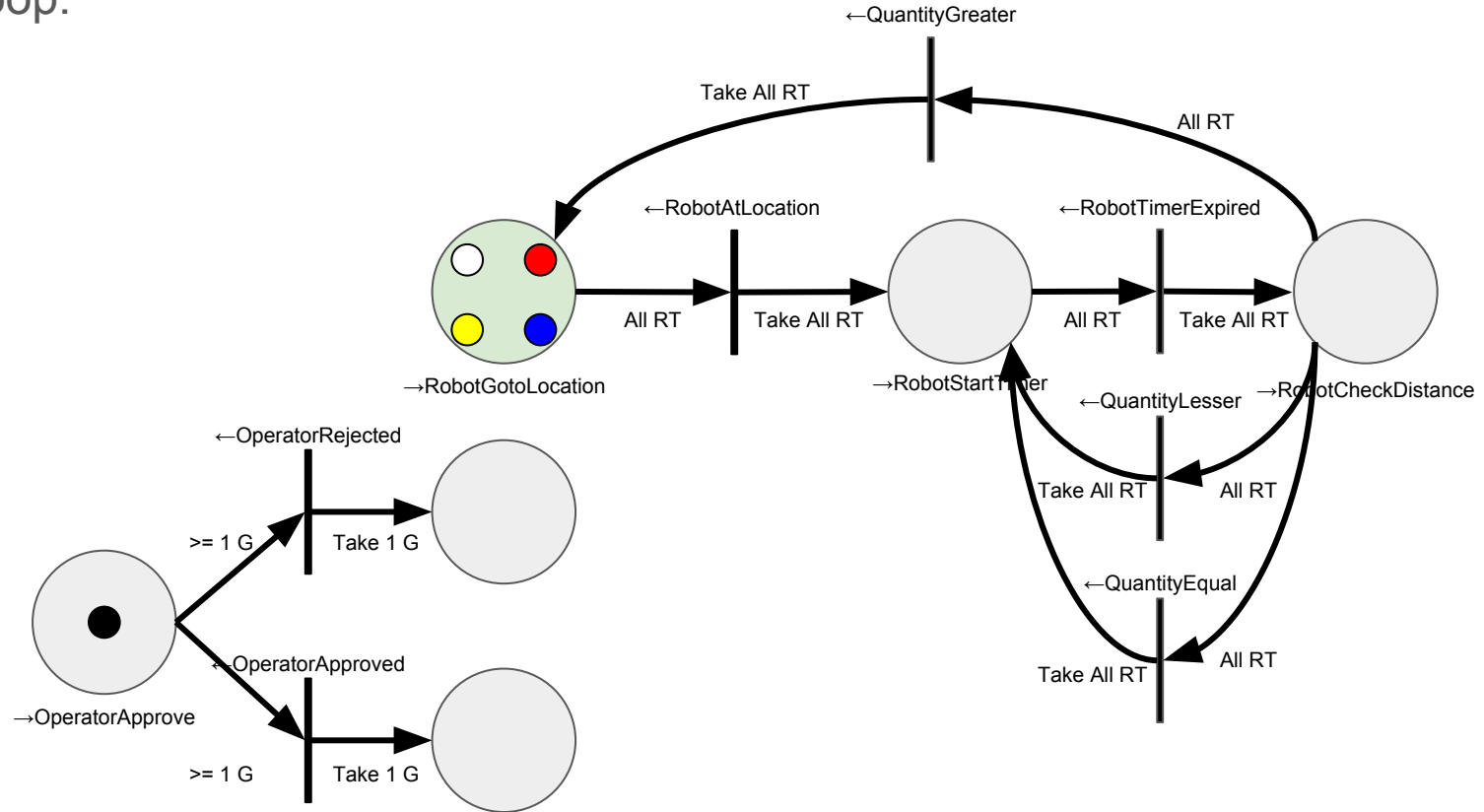
If we use the "branching" concept, we can separate the operator question from the station keep movement. When the plan starts, we will want the robot tokens put in the station keep branch and a generic token put in the question branch.
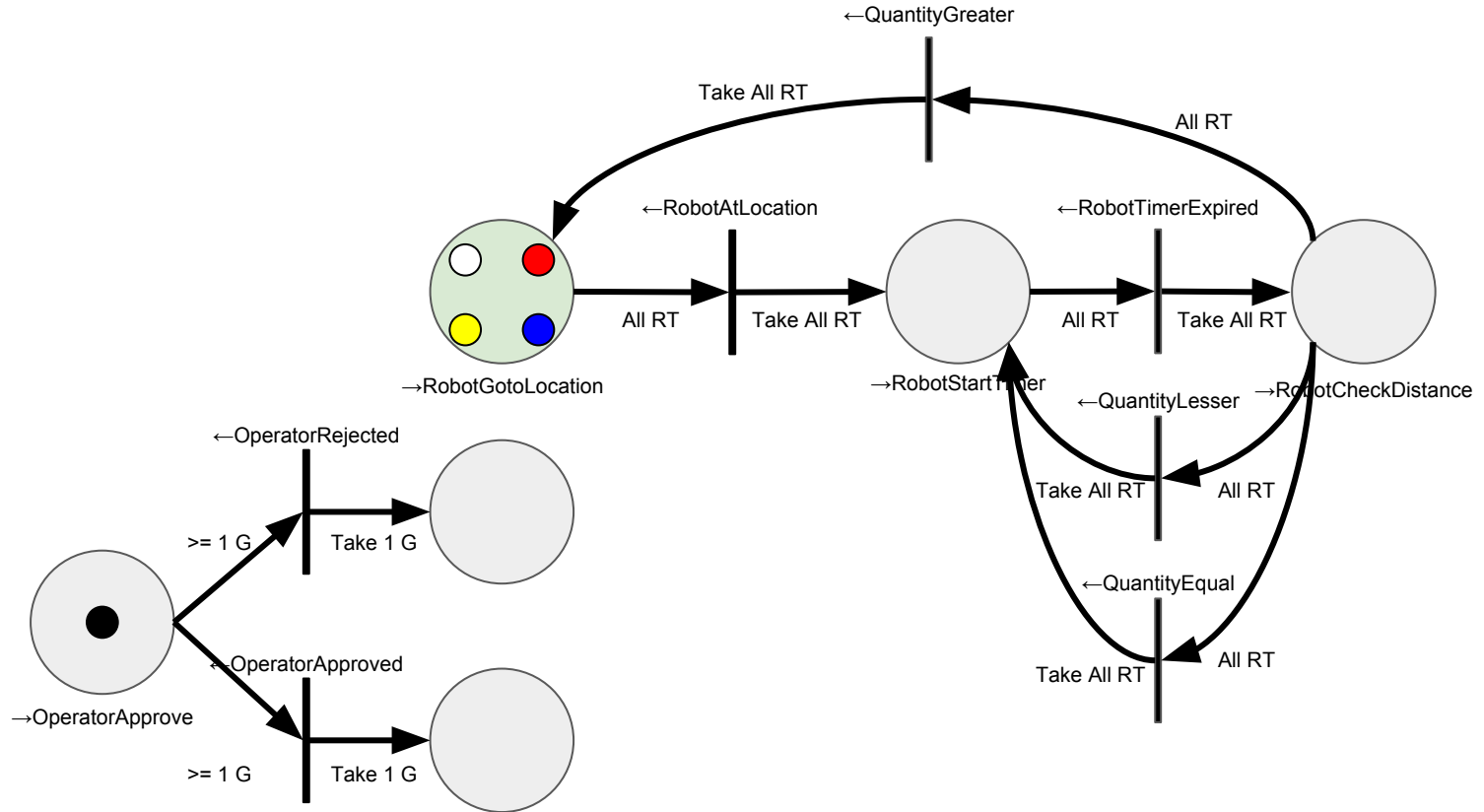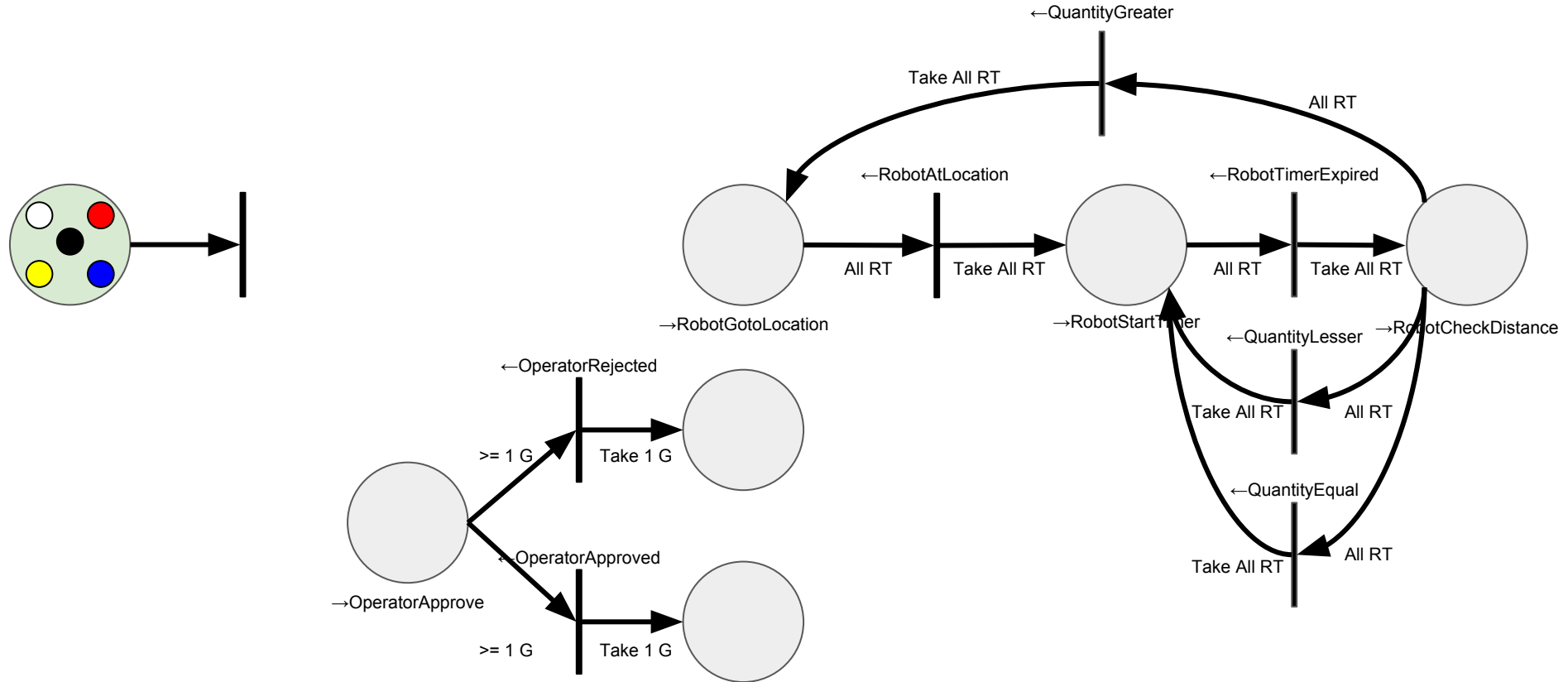
We can only have one start place, which is inside the the station keeping loop. We need to move the start place out of the loop to avoid it getting triggered each station keep loop.

←QuantityGreater

Take All RT

All RT

←RobotAtLocation

←RobotTimerExpired

All RT   Take All RT   All RT   Take All RT

→RobotGotoLocation

→RobotStartTimer

←QuantityLesser

→RobotCheckDistance

Take All RT   All RT

←OperatorRejected

>= 1 G   Take 1 G

←QuantityEqual

Take All RT   All RT

→OperatorApprove

←OperatorApproved

>= 1 G   Take 1 G

Here we add the new start place.

Now we can create the branches to the station keep and operator question.
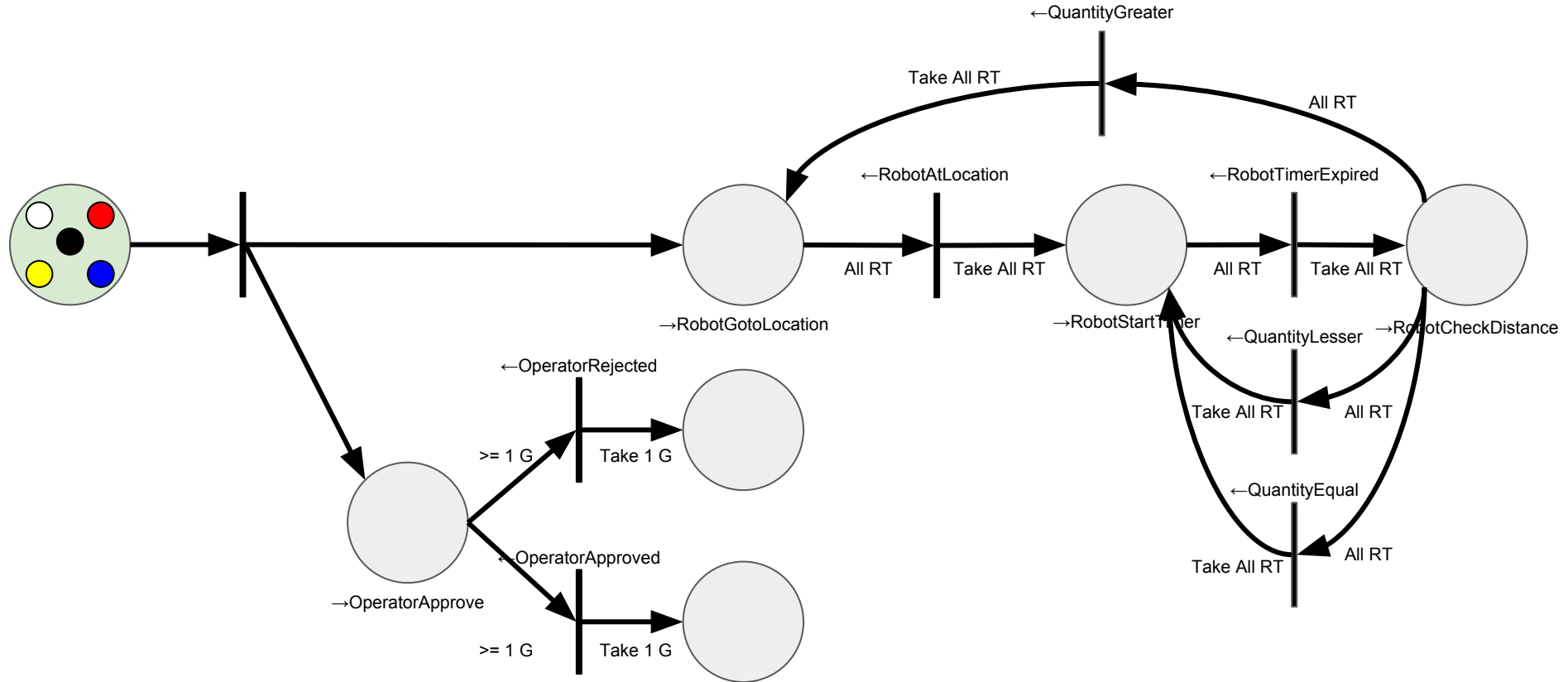
Now we need to figure out the edge requirements for this new section so that the White, Red, Yellow, and Blue robot tokens are in the station keep branch and the (black) generic token is in the operator question branch.
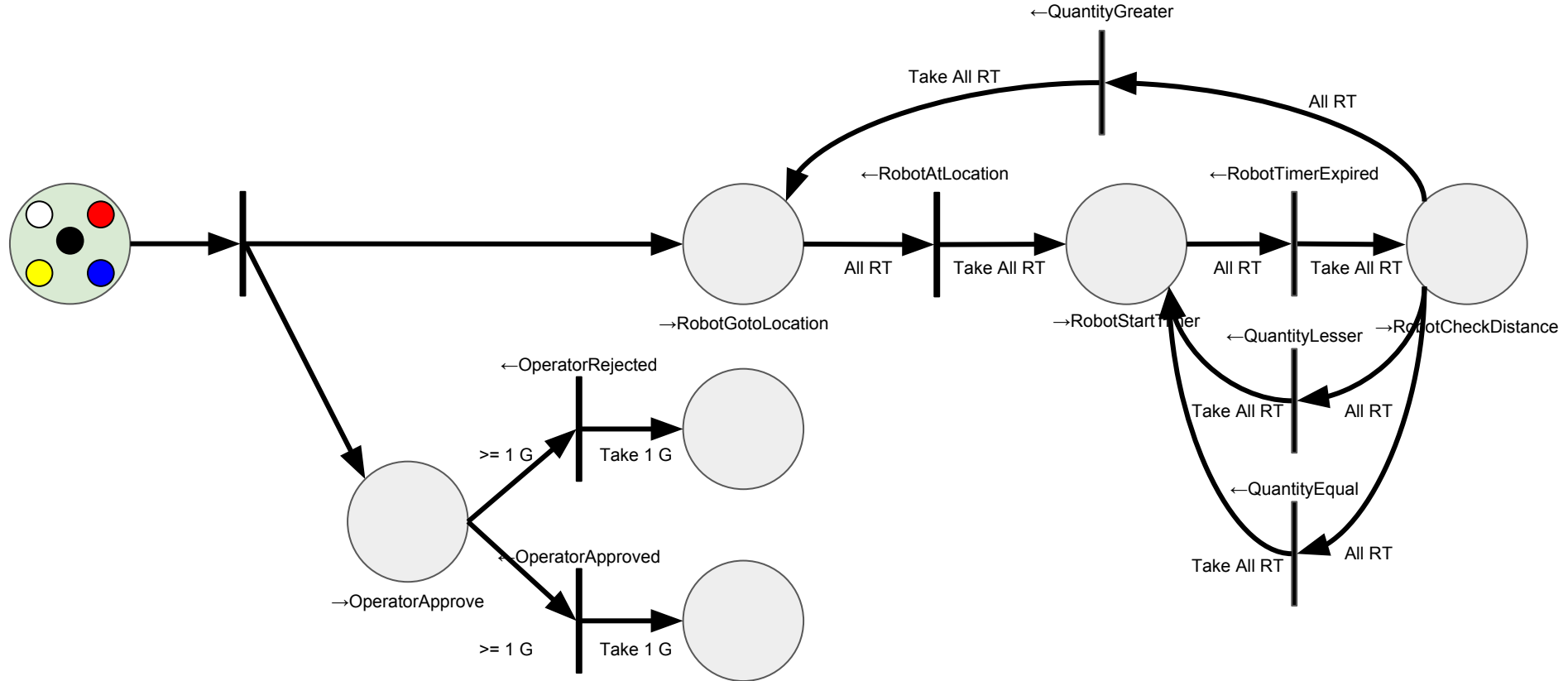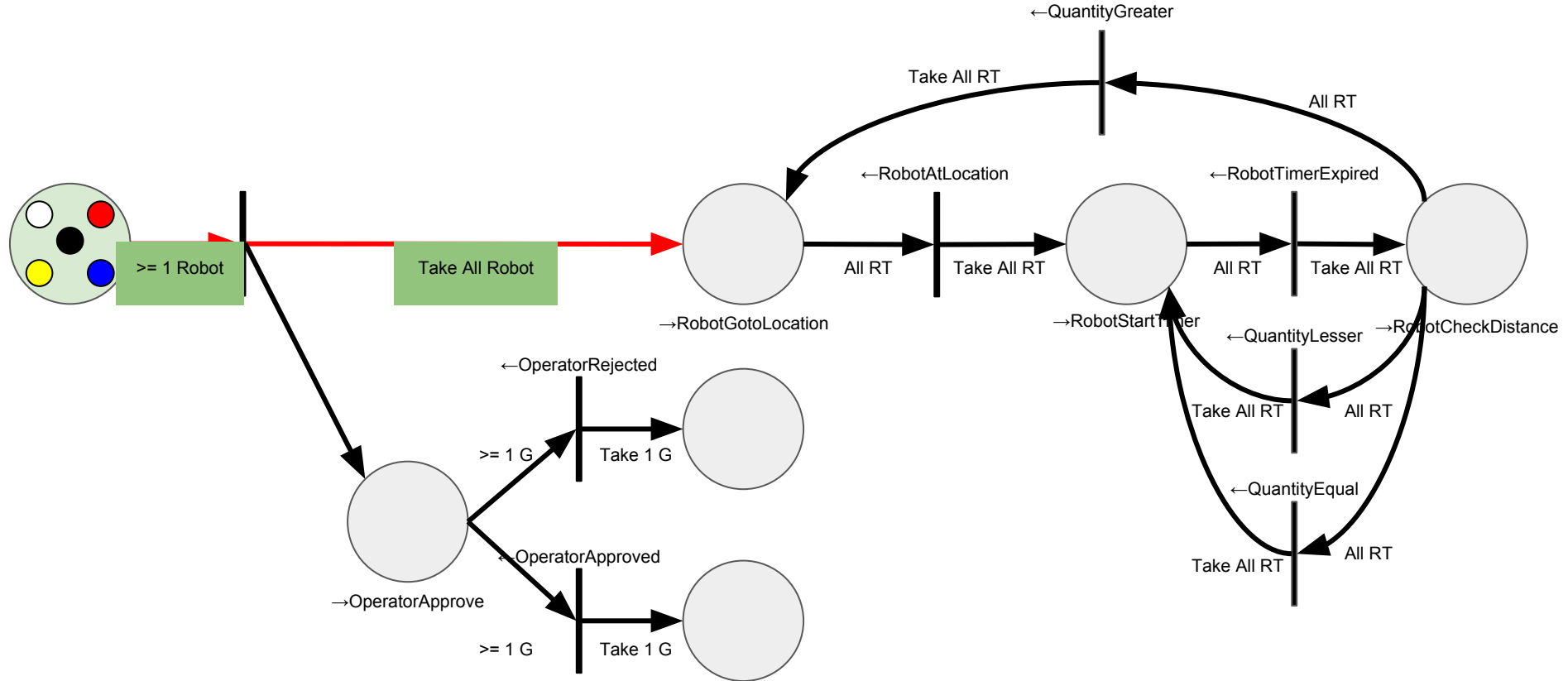
We need the robot tokens in this part of the plan. Similar to the previous lesson where an out edge requirement could manipulate all RT, we can also instruct an out edge requirement to manipulate all G or Robot tokens.
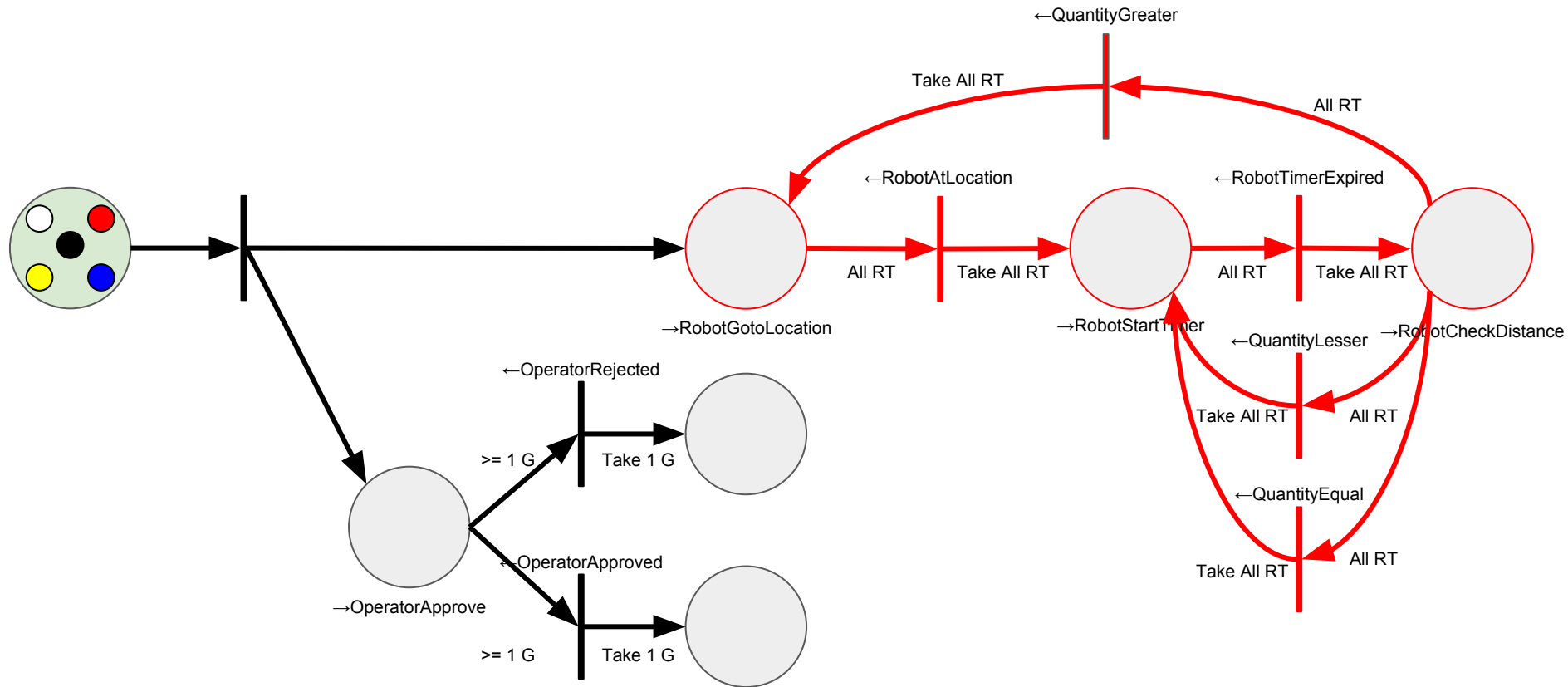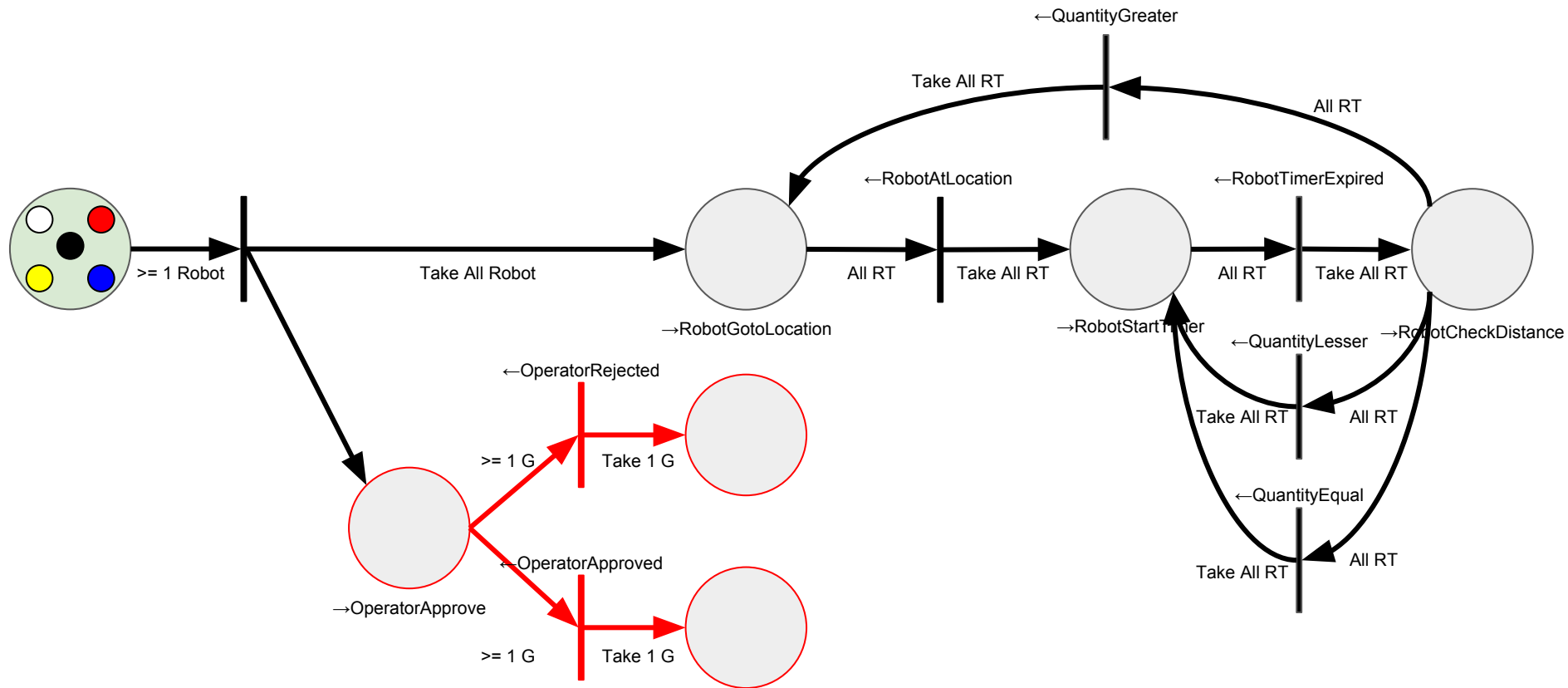
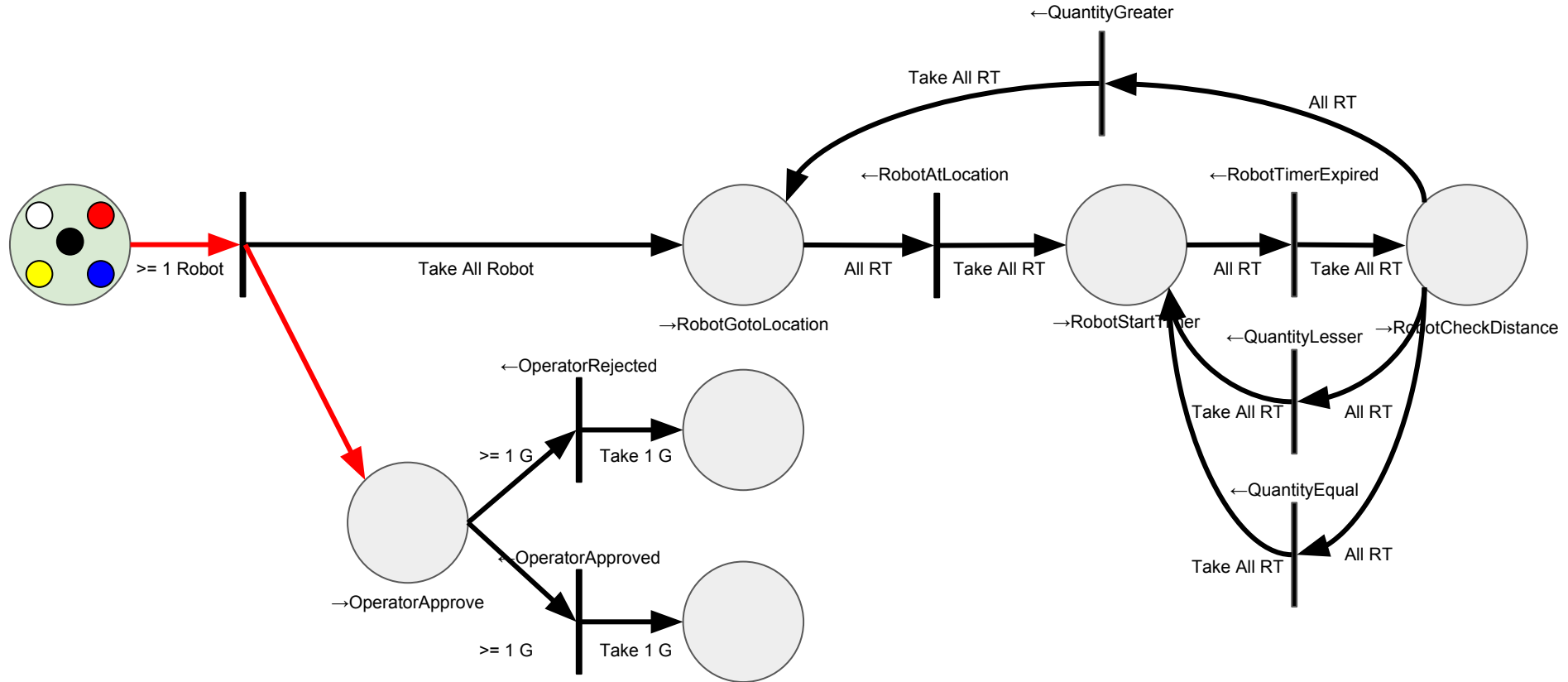# Job 7-1: Add the edge requirements.
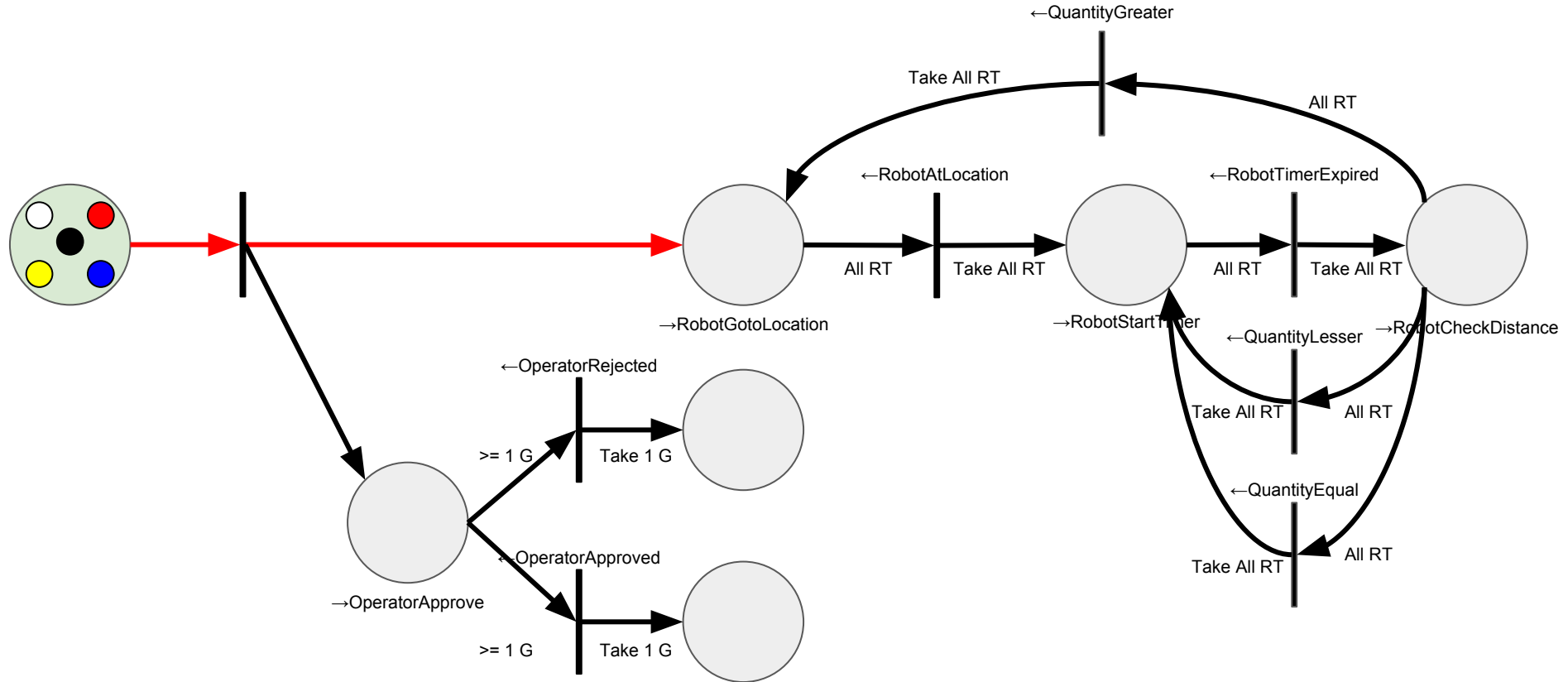
Job 7-1 Solution

We need a generic token in this part of the plan. Remember that multiple requirements can be listed on a single edge.
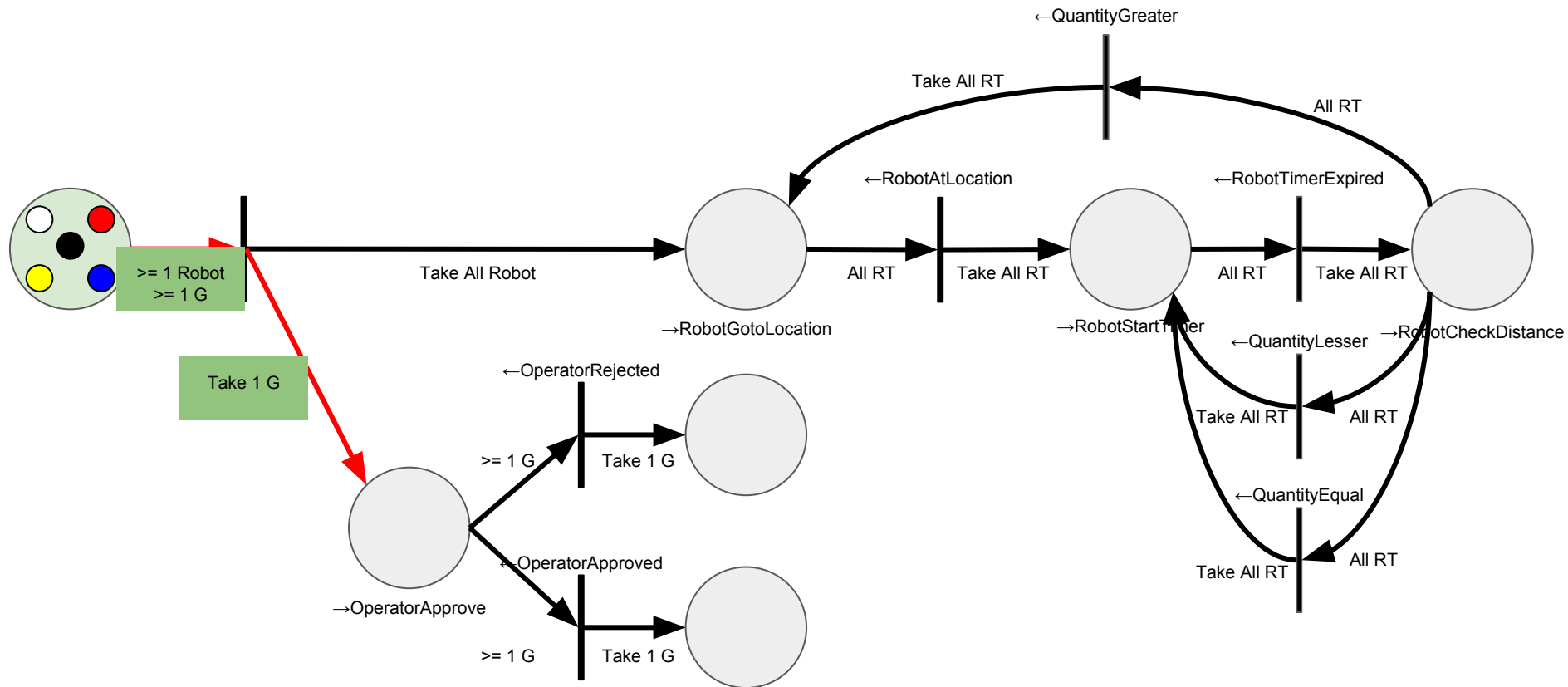
Job 7-2: Add the edge requirements.

←QuantityGreater

Take All RT

All RT

←RobotAtLocation

←RobotTimerExpired

>= 1 Robot

Take All Robot

All RT

Take All RT

All RT

Take All RT

→RobotGotoLocation

→RobotStartTimer

→RobotCheckDistance

←QuantityLesser

Take All RT

All RT

←OperatorRejected

>= 1 G

Take 1 G

→OperatorApprove

←OperatorApproved

>= 1 G

Take 1 G

←QuantityEqual

Take All RT

All RT

Job 7-2 Solution

←QuantityGreater

Take All RT

All RT

←RobotAtLocation

←RobotTimerExpired

Take All Robot

All RT

Take All RT

All RT

Take All RT

>= 1 Robot
>= 1 G

→RobotGotoLocation

→RobotStartTimer

→RobotCheckDistance

←QuantityLesser

Take 1 G

Take All RT

All RT

←OperatorRejected

>= 1 G

Take 1 G

←QuantityEqual

Take All RT

All RT

→OperatorApprove

←OperatorApproved

>= 1 G

Take 1 G

Now that we have the edge labels, let's customize what happens depending on the operator's decision.
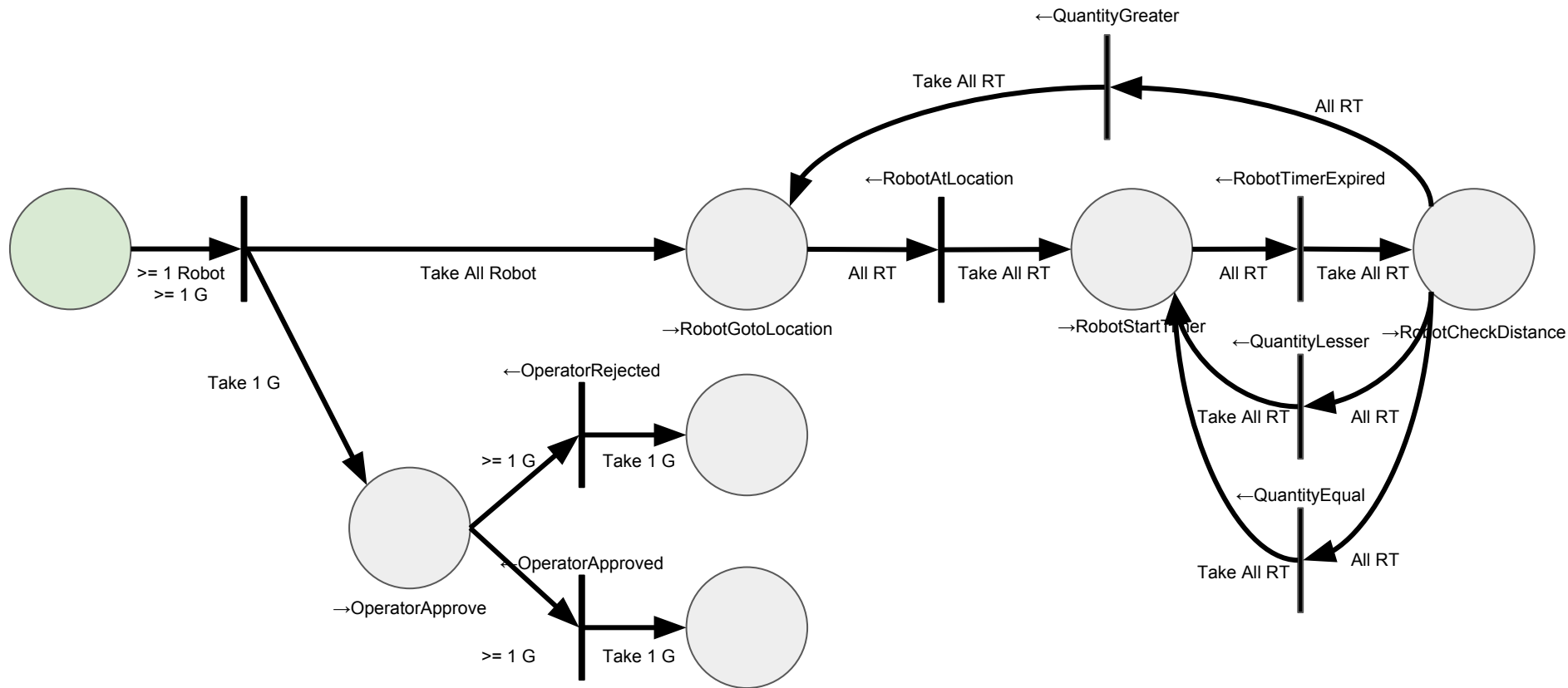
If the operator chooses "Yes", we want the plan to end. So we should make the place connected to the OperatorRejected input event an end place.
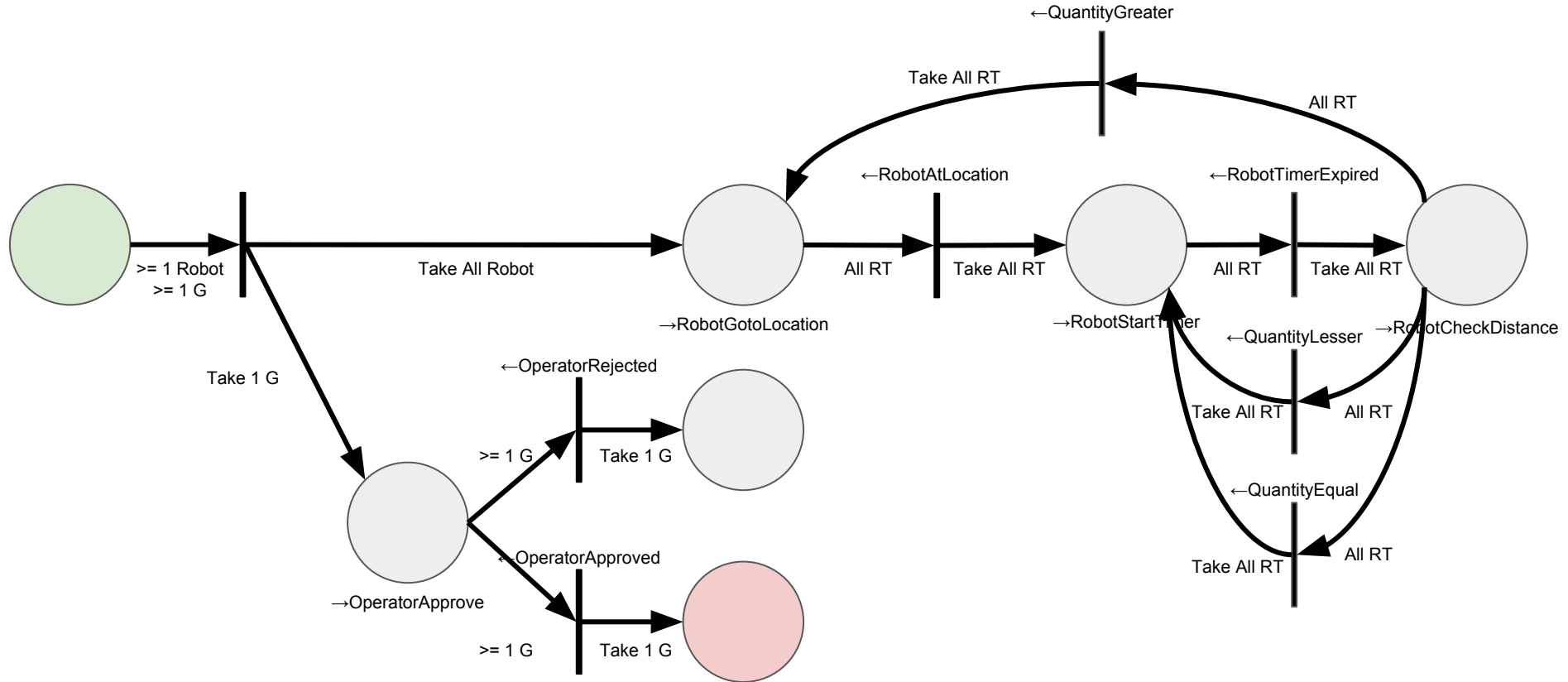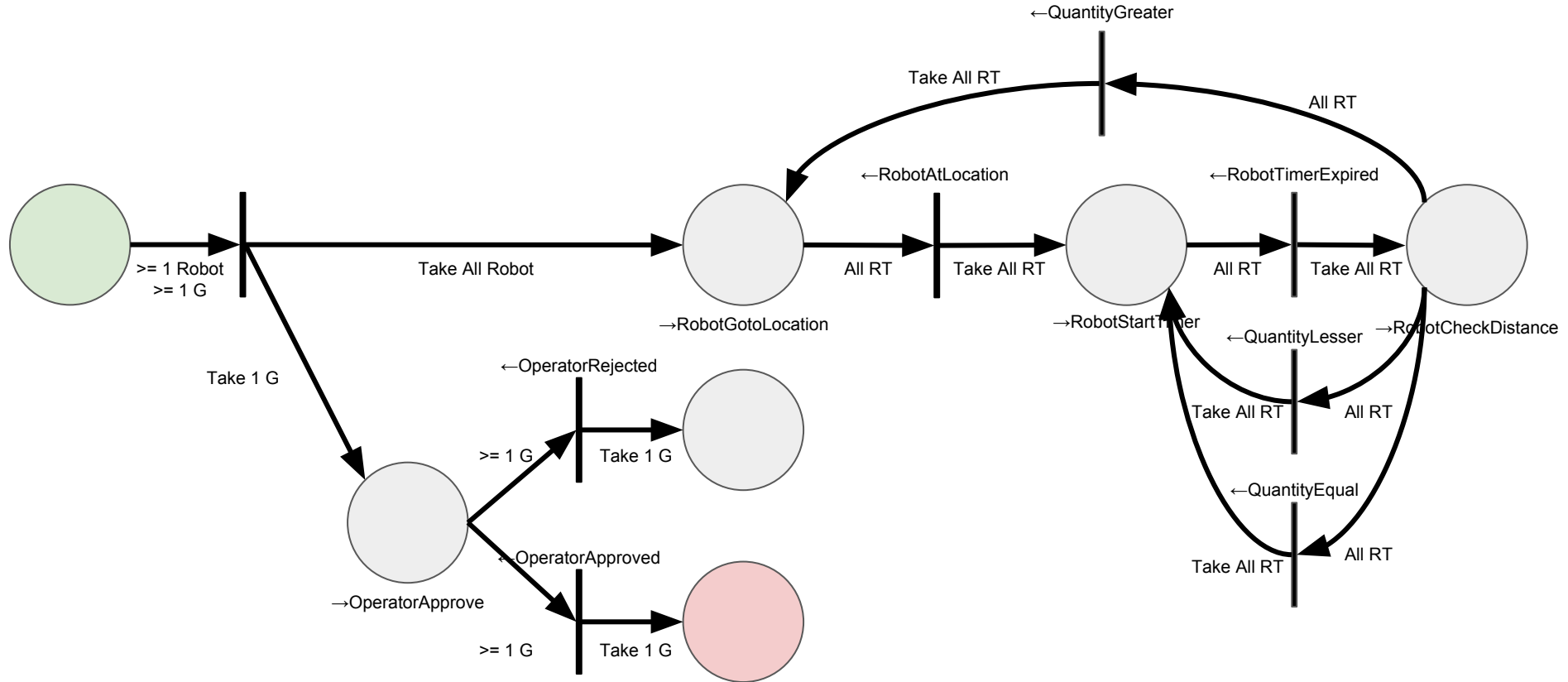
QUIZ 7-4: What will happen in this case if the operator clicks "No"?
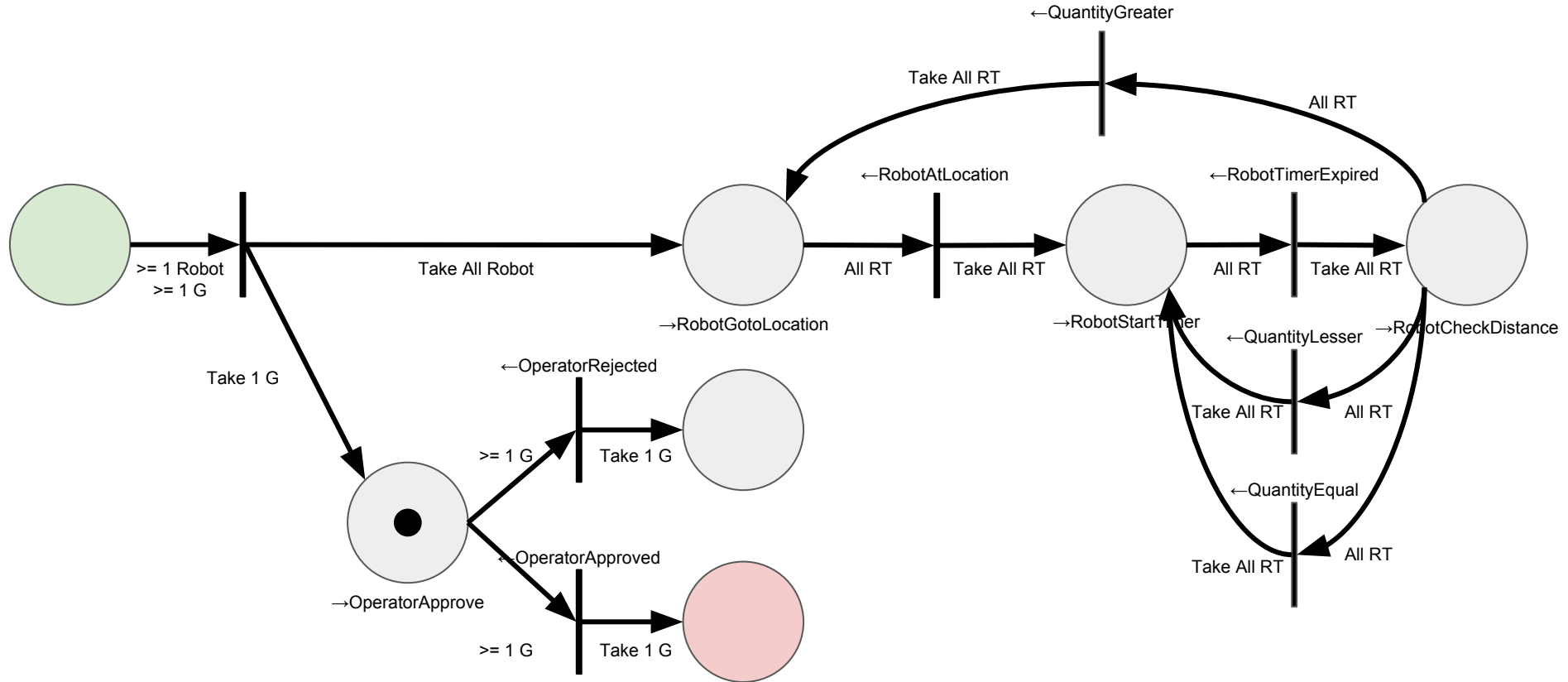
Quiz 7-4 Solution

The generic token will be moved to the place with ←OperatorRejected and the question will never be asked again

If the operator clicks "No," the plan continues as normal, but the question is never asked again. This means the operator cannot end the plan in the future.
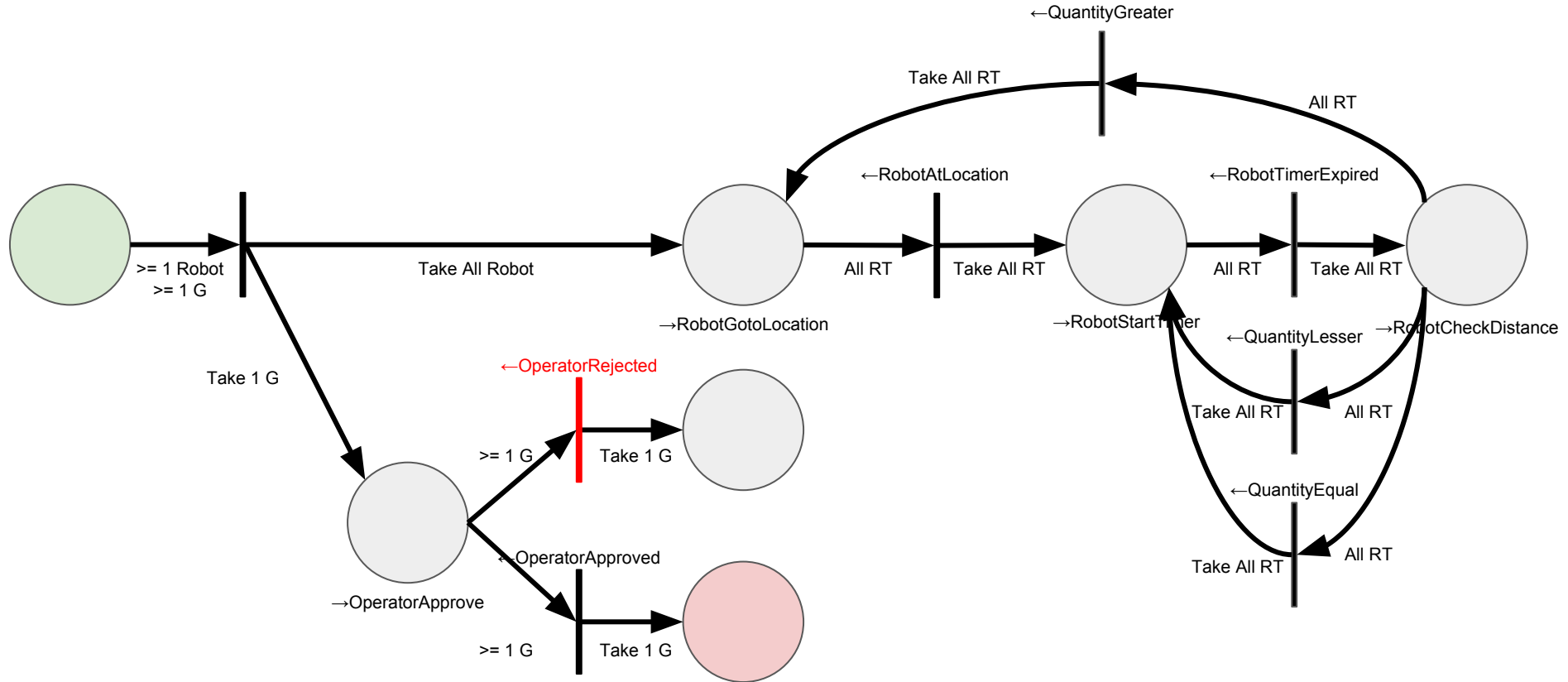
Let's change it so that if the operator accidentally presses "No" they are asked the question again.
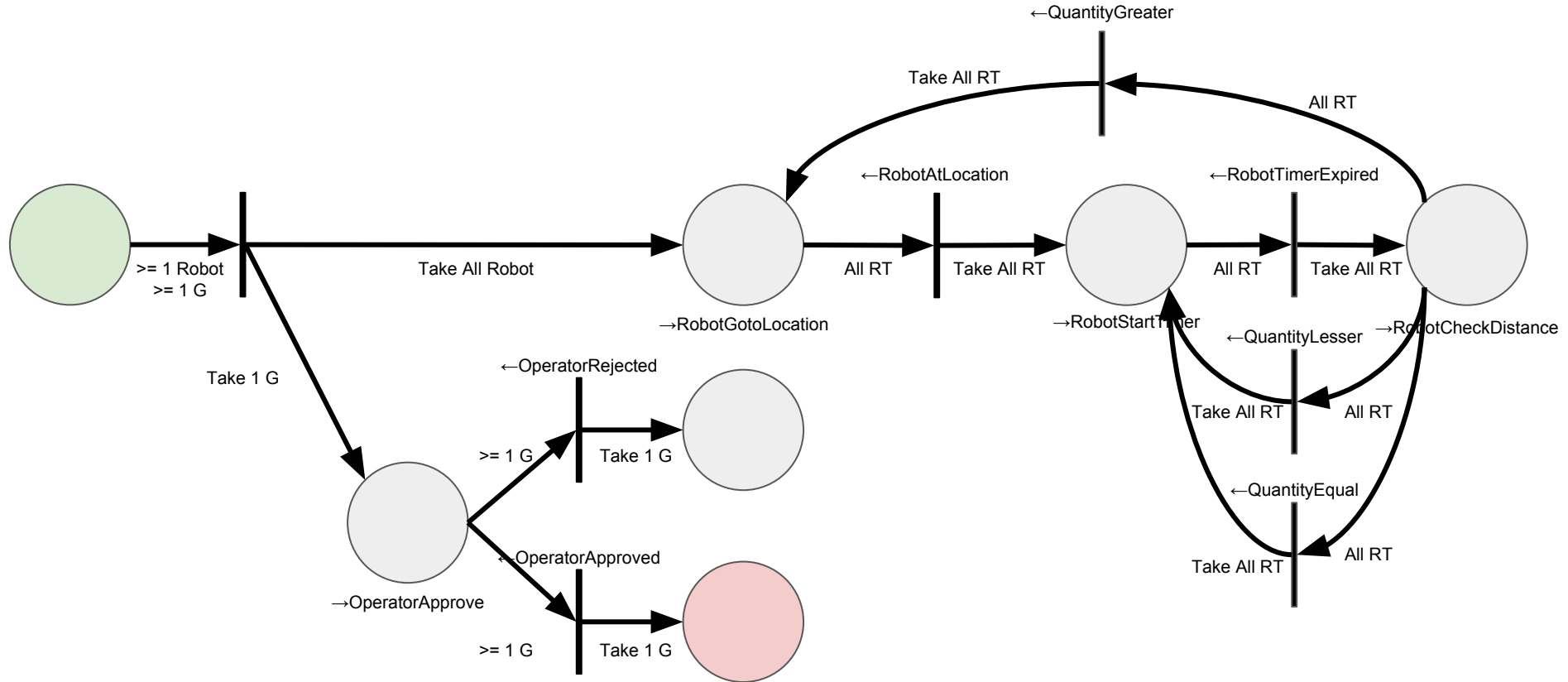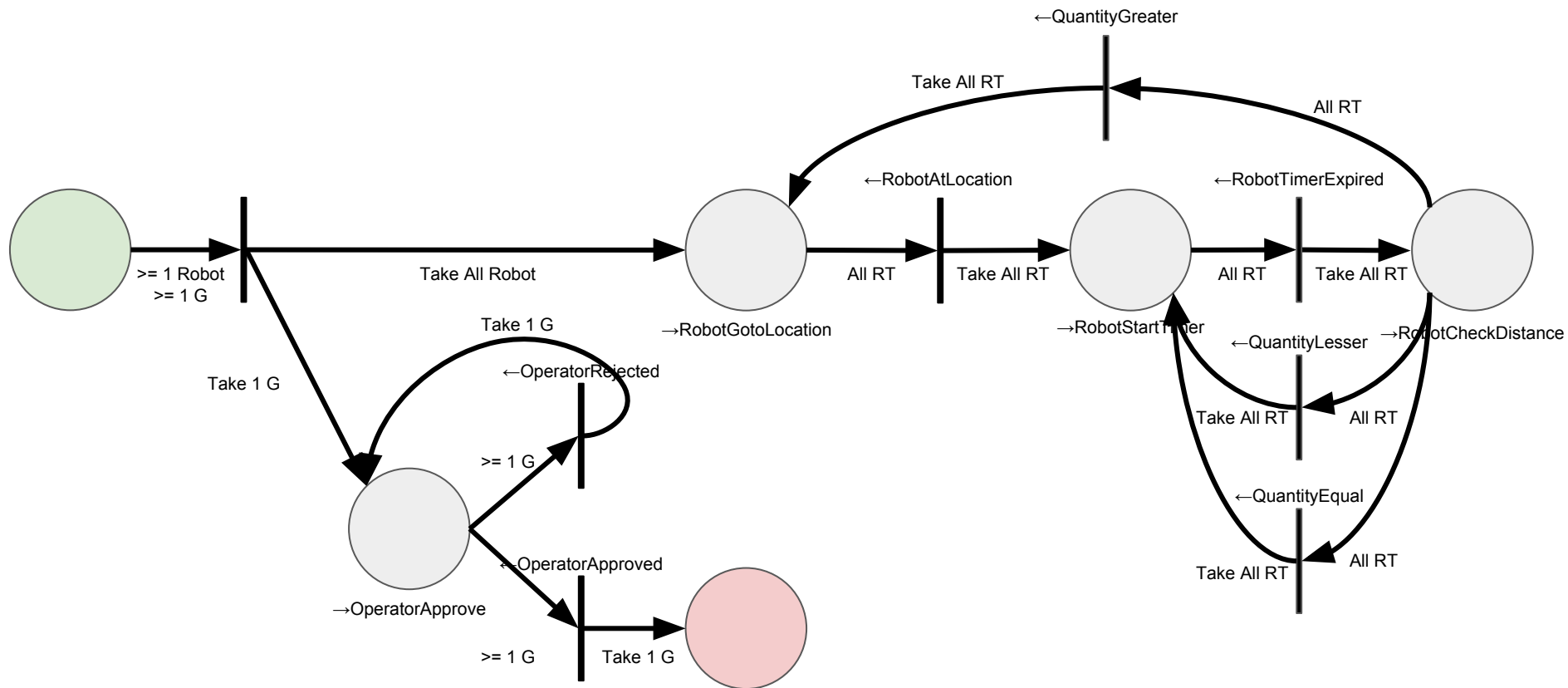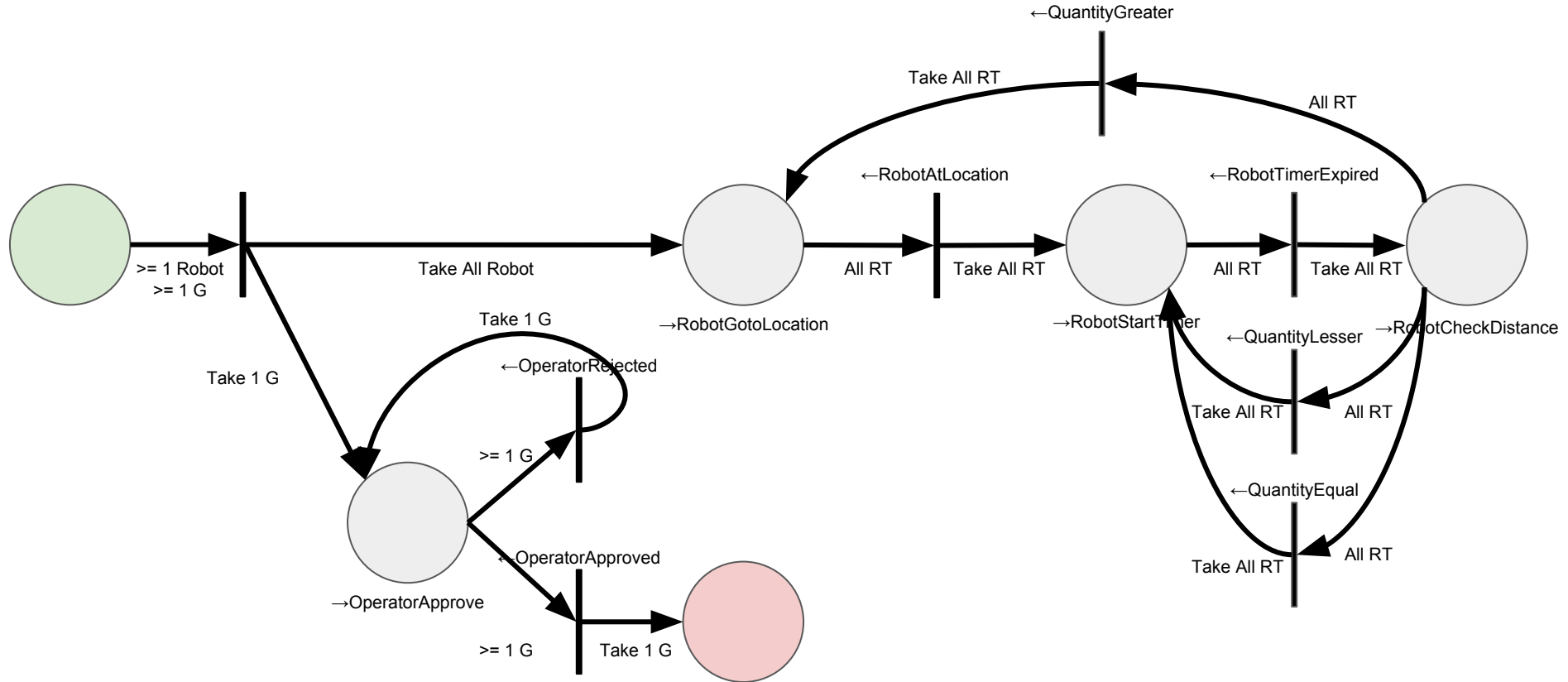
Job 7-3: Change the SPN so that if the operator accidentally presses "No," they are asked the question again.
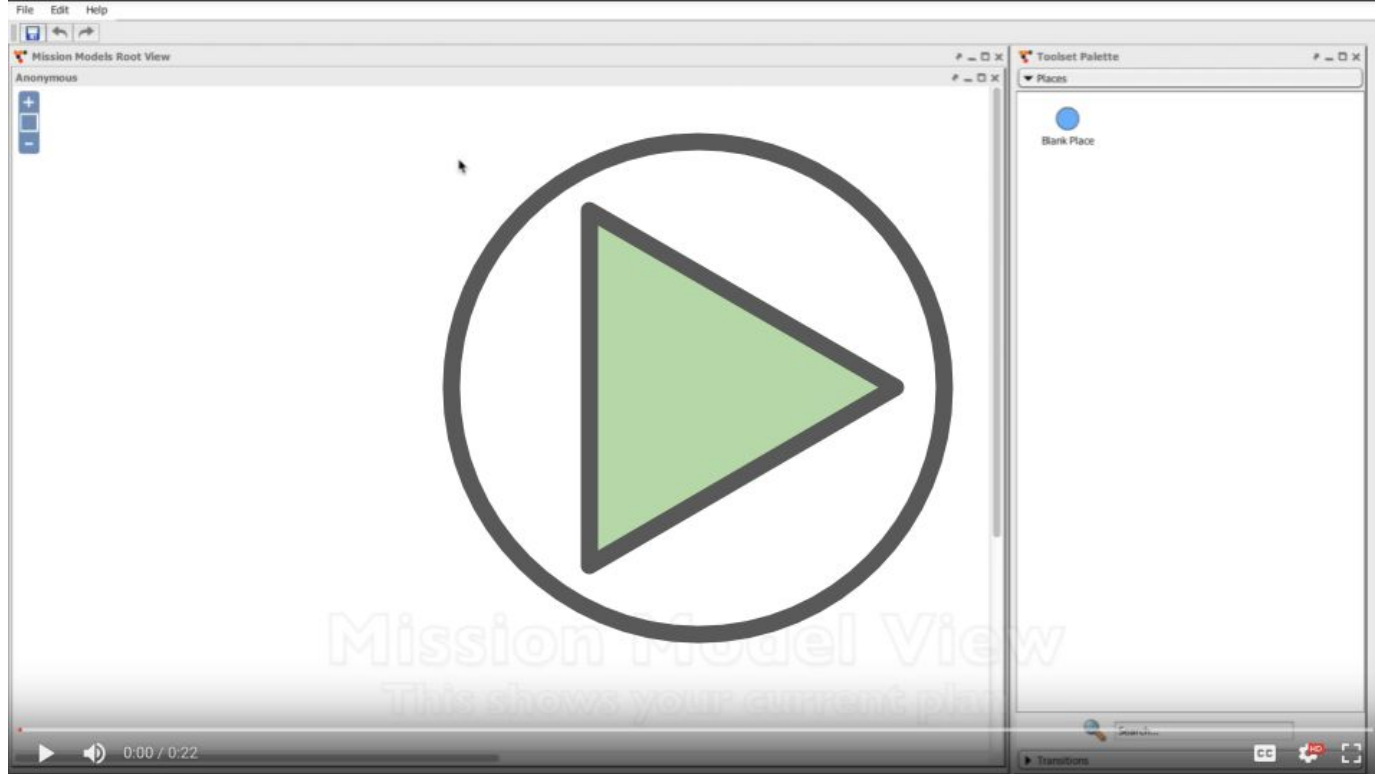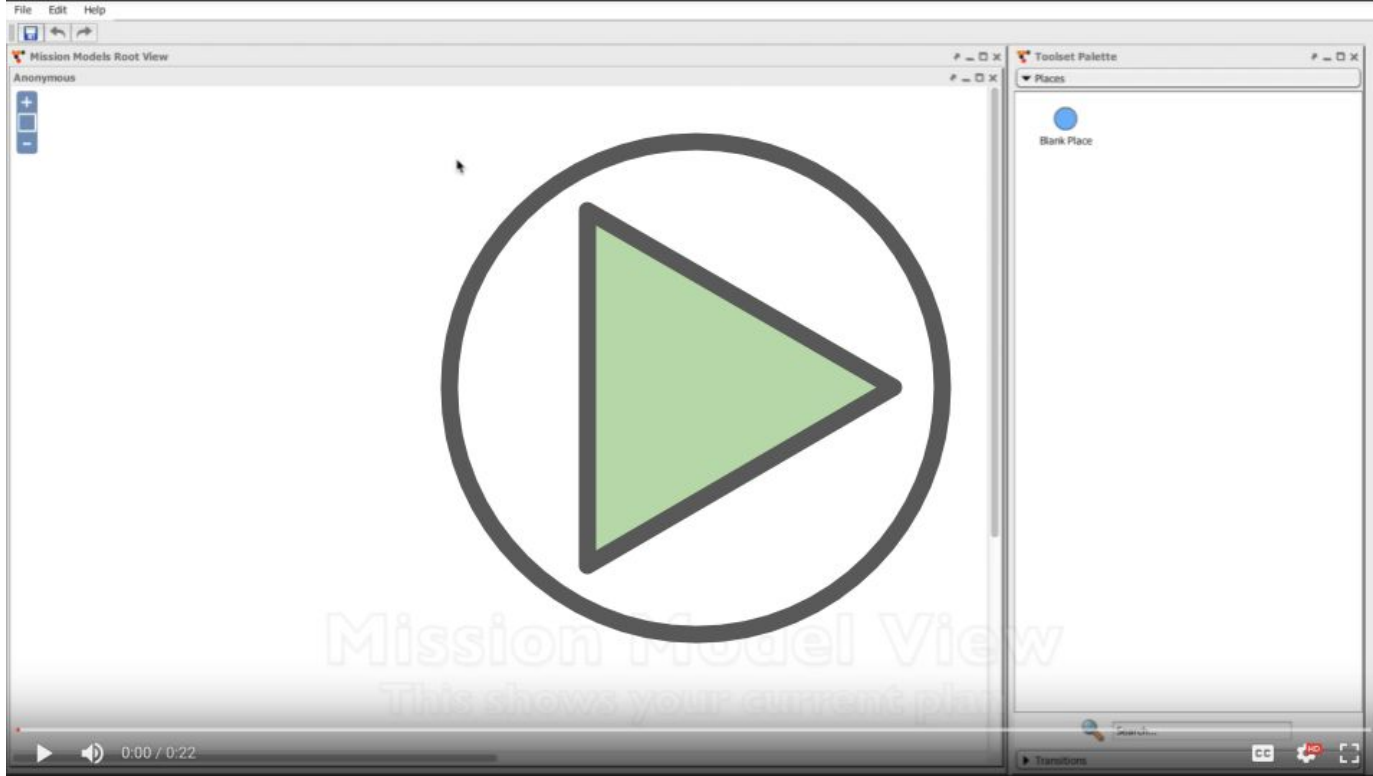
Job 7-3 Solution

Now if the operator accidentally clicks "No," the questions will pop back up again.

Watch "Setting a Start Place": This video will show you how to set a place as a start place.

Watch "Setting an End Place": This video will show you how to set a place as an end place.

Job 7-4: Update the SPN