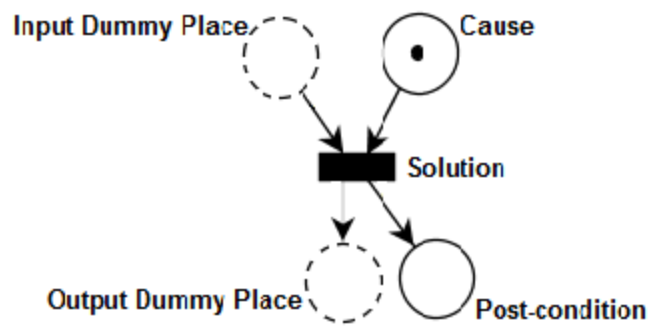
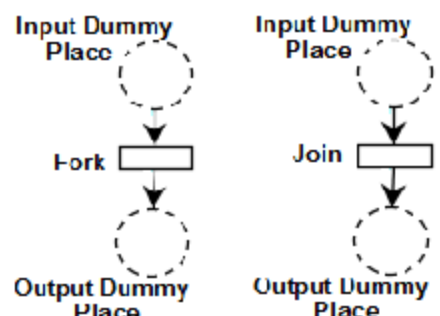
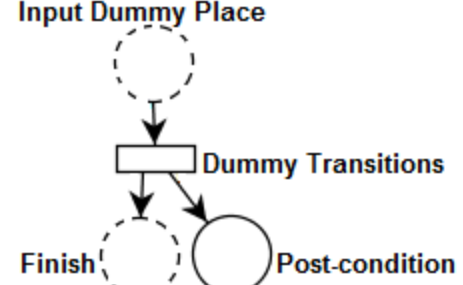


Rule No.	RUCM Element	Petri-Net Element
1.	Initial state Title, Resource, Initial Context: -{Pre-condition}	
2.	Simple Basic Flow <Id><Basic Flow Sentence> -{Pre-condition} -{Post-condition}	
3.	Conditional Basic Flow <Id> IF {<Condition>} THEN <Basic Flow Sentence> -{Pre-condition} -{Post-condition}	
4.	Loop Basic Flow <Id>DO {<Condition>} WHILE <Basic Flow Sentence> -{Pre-condition} -{Post-condition}	

5.	<p>Specific Alternative Flow</p> <p><Id>IF {<Cause>} THEN <Solution> - {Post-condition}</p>	 <p>The diagram illustrates the 'Specific Alternative Flow' construct. It features an 'Input Dummy Place' (dashed circle) and a 'Cause' (solid circle with a black dot). Both have arrows pointing to a solid black rectangular 'Solution' node. From the 'Solution' node, two arrows point out: one to an 'Output Dummy Place' (dashed circle) and another to a 'Post-condition' (solid circle).</p>
6.	<p>Concurrency Construct</p> <p># {Basic Flow Series} #</p>	 <p>The diagram shows two parallel flow paths for the 'Concurrency Construct'. The left path starts with an 'Input Dummy Place' (dashed circle) leading to a 'Fork' node (solid rectangle), which then leads to an 'Output Dummy Place' (dashed circle). The right path starts with an 'Input Dummy Place' (dashed circle) leading to a 'Join' node (solid rectangle), which then leads to an 'Output Dummy Place' (dashed circle).</p>
7.	<p>Final State</p> <p>Context: - {Post-condition}</p>	 <p>The diagram illustrates the 'Final State' construct. It begins with an 'Input Dummy Place' (dashed circle) leading to a 'Dummy Transitions' node (solid rectangle). From this node, two arrows point out: one to a 'Finish' node (dashed circle) and another to a 'Post-condition' node (solid circle).</p>