First Example of Requirement Set that is Complete but Conflicting

R1: When [Left_Input_Device is active], then the SOI shall activate Middle_Output_Device.

R2: When [Left_Input_Device is inactive] OR [Right_Input_Device is inactive], then the SOI shall de-activate Middle_Output_Device.

Step 1:

L = Left_Input_Device is Active R1: $L \rightarrow M$

R = Right_Input_Device is Active R2: $\neg L \vee \neg R \rightarrow \neg M$

M = Middle_Output_Device is Active

				Step 2:		**Step 3:**	*(Completeness)*
L	R	¬L	¬R	L	¬L ∨ ¬R	(∃x A(x)) ∧ (∀x (∀y ((A(x) ∧ A(y)) → (x = y)))	*(∃x A(x))*
0	0	1	1	0	1	1	*1*
0	1	1	0	0	1	1	*1*
1	0	0	1	**1**	**1**	**0**	*1*
1	1	0	0	1	0	1	*1*

Step 4:

Passed Uniqueness Test for Tautology? **FALSE**

Conclusion: **Requirement set is NOT Unique**

Second Example of Requirement Set that is Complete but NOT Unique

R1: When [Left_Input_Device is active] AND [Right_Input_Device is active], then the SOI shall activate Middle_Output_Device.

R2: When [Left_Input_Device is active] AND [Right_Input_Device is inactive], then the SOI shall activate Middle_Output_Device.

R3: When [Left_Input_Device is inactive] OR [Right_Input_Device is inactive], then the SOI shall de-activate Middle_Output_Device.

Step 1:

L = Left_Input_Device is Active
R = Right_Input_Device is Active
M = Middle_Output_Device is Active

R1: $L \wedge R \rightarrow M$
R2: $L \wedge \neg R \rightarrow M$
R3: $\neg L \vee \neg R \rightarrow \neg M$

Step 2: **Step 3:** *(Completeness)*

L	R	¬L	¬R	L ∧ R	L ∧ ¬R	¬L ∨ ¬R	(∃x A(x)) ∧ (∀x (∀y ((A(x) ∧ A(y)) → (x = y)))	*(∃x A(x))*
0	0	1	1	0	0	1	1	*1*
0	1	1	0	0	0	1	1	*1*
1	0	0	1	0	**1**	**1**	**0**	*1*
1	1	0	0	1	0	0	1	*1*

Step 4:

Passed Uniqueness Test for Tautology? **FALSE**

Conclusion: **Requirement set is NOT Unique**

Example of Requirement Set that is incomplete

R1: When [Left_Input_Device is active] AND [Right_Input_Device is active], then the SOI shall activate Middle_Output_Device.
R2: When [Left_Input_Device is inactive] AND [Right_Input_Device is inactive], then the SOI shall de-activate Middle_Output_Device.

Step 1:

L = Left_Input_Device is Active
R = Right_Input_Device is Active
M = Middle_Output_Device is Active

R1: $L \wedge R \rightarrow M$
R2: $\neg L \wedge \neg R \rightarrow \neg M$

				Step 2:		**Step 3:**	*(Completeness)*
L	R	¬L	¬R	L ∧ R	¬L ∧ ¬R	(∃x A(x)) ∧ (∀x (∀y ((A(x) ∧ A(y)) → (x = y)))	*(∃x A(x))*
0	0	1	1	0	1	1	*1*
0	1	1	0	0	0	**0**	***0***
1	0	0	1	0	0	**0**	***0***
1	1	0	0	1	0	1	*1*

Step 4:

Passed Uniqueness Test for Tautology? **FALSE**

Conclusion: **Requirement set is NOT Unique**

Example of Requirement Set that is Complete and Unique

R1: When [Left_Input_Device is active] AND [Right_Input_Device is active], then the SOI shall activate Middle_Output_Device.
R2: When [Left_Input_Device is inactive] OR [Right_Input_Device is inactive], then the SOI shall de-activate Middle_Output_Device.

Step 2:

L = Left_Input_Device is Active
R = Right_Input_Device is Active
M = Middle_Output_Device is Active

R1: $L \wedge R \rightarrow M$
R2: $\neg L \vee \neg R \rightarrow \neg M$

				Step 3:		**Step 4:**	*(Completeness)*
L	R	¬L	¬R	L ∧ R	¬L ∨ ¬R	(∃x A(x)) ∧ (∀x (∀y ((A(x) ∧ A(y)) → (x = y)))	*(∃x A(x))*
0	0	1	1	0	1	1	*1*
0	1	1	0	0	1	1	*1*
1	0	0	1	0	1	1	*1*
1	1	0	0	1	0	1	*1*

Step 4:

Passed Uniqueness Test for Tautology? **TRUE**

Conclusion: **Requirement set is Unique**