The Double-Not Arrow and Negative Sufficient Conditions

The use of double-not arrows in conjunction with single arrow statements that contain a negated condition is pretty standard. It most frequently appears in Logic Games, but also occasionally in Logical Reasoning Formal Logic questions.

As stated in the *PowerScore LSAT Logical Reasoning Bible*, double-not arrows ($\leftarrow\!\!+\!\!\rightarrow$) are "introduced by conditional statements where exactly one of the terms is negative, or by statements using words such as "no" and "none" that imply the two variables cannot 'go together.' " For example:

> If B is selected, then C is not selected.
>
> Diagram: $B \longrightarrow \not{C}$

In this case, the diagram above represents the statement as given. However, from the contrapositive, we also know that $C \longrightarrow \not{B}$, and combining the two statements yields the inference that B and C can never be selected together. Because writing both of those statements out separately can be time-consuming, they can be combined using the double-not arrow, which results in the following diagram:

> Diagram: $B \leftarrow\!\!+\!\!\rightarrow C$

Essentially, then, the double-not arrow means that the variables on each end cannot be selected together in that form (thus, in the diagram above, "B" and "C" cannot both be selected. Under the rule, only "B and C" is ruled out above, and any other combination is acceptable; so, "B and $\not{C}$," "$\not{B}$ and C," and "$\not{B}$ and $\not{C}$" would each be acceptable outcomes under that rule.
Here's another example that leads to the same diagram and set of possible outcomes:

> No Bs are Cs.
>
> Diagram: $B \leftarrow\!\!+\!\!\rightarrow C$

Ultimately, when the sufficient condition is positive and the necessary condition is negative (as is the case in $B \longrightarrow \not{C}$), the result is a double-not arrow relationship like the one above, where both terms are positive and the negative is contained in the symbol itself. But, this result changes when the sufficient condition is negative and the necessary condition is positive, so let's examine what happens there.

Let's start with a simple single arrow conditional statement:

If B is *not* selected, then C is selected.

Diagram: ~~B~~ ⟶ C

In this case, the diagram above represents the statement as given. However, from the contrapositive, we also know that ~~C~~ ⟶ B, and combining the two statements yields the inference that when one is *not* selected, the other *must* be selected. In other words, one of B or C must always be selected and both can never be absent. This is a tough concept, and if you choose to use the double-not arrow to represent this relationship, it appears as follows:

Diagram: ~~B~~ ←+→ ~~C~~

This diagram means that the variables on each end cannot be selected together in that form (thus, in the diagram above, "~~B~~" and "~~C~~" cannot both occur. But, only "~~B~~ and ~~C~~" is ruled out above, and any other combination is acceptable; so, "B and ~~C~~," "~~B~~ and C," and "B and C" would each be acceptable under that rule.

The above usage can be a bit confusing, and so with statements that have a negative sufficient condition our advice is that you should only use the double-not arrow if you are *very* comfortable with it. If you aren't, simply write out the statement and its contrapositive separately, keep in mind the impact that the negative sufficient condition has, and move on from there.

In summary, negative sufficient conditions are tricky, because they revolve around the idea of something not occurring being sufficient. That "absence of occurrence" can be hard to grasp, so when you see statements like this, make sure you carefully track what's occurring, especially in Logic Games where negative sufficient condition rules always play a big rule when they appear.