For network E, one approach is to condition on whether or not edge S fails, then use total probability.

- When edge S is absent, network E reduces to network C.
- When edge S is present, network E is equivalent to network D.

So \( P(F) = p_S P(\text{edge in network C}) + (1-p_S) P(\text{outage in network D}). \)

Working things through, this simplifies to the expression we have.

Another approach is to directly get the expression in rules:

\( \rightarrow \) If some four links used in network D used along with link S, then network E fails whenever network D does.

\( \rightarrow \) Two more ways network E can fail when network D does not:

1. links 1, 4, 5 or only one to fail \( \Rightarrow P_1 P_4 P_5 \)
2. links 2, 3, 5 or only one to fail \( \Rightarrow P_2 P_3 P_5 \).

The \( P_1 P_2 \) operation for parallel paths is pretty straightforward.

The \( P_1 + P_2 - P_1 P_2 \) operation for series paths is a little complicated, but we can use union bond to simplify without too much loss in accuracy:

\[ P_1 + P_2 - P_1 P_2 \leq P_1 + P_2. \]

For example, if \( p = 0.01 \), then \( p^2 = 0.0001 \)

\[ \Rightarrow P_1 + P_2 - P_1 P_2 = 0.0199 \quad \text{vs.} \quad P_1 + P_2 = 0.02. \]

In general, union bond is useful for simplifying series path computations, without losing too much accuracy.
Another approach to computing output probabilities is to sum over all possible present/absent edges and add together probabilities.

Distribution of flow over network:

Network $F$:

$P_x(0) = p_1p_2$
$P_x(1) = p_1p_2$
$P_x(2) = p_1p_2$
$P_x(3) = p_1p_2$

What about network $G$?