4.5 – Signal Flow Graphs

Reading Assignment: ***pp. 189-197***

Q: *Using individual device scattering parameters to analyze a complex microwave network results in a lot of **messy** math! Isn't there an **easier** way?*

A: Yes! We can represent a microwave network with its **signal flow graph**.

HO: SIGNAL FLOW GRAPHS

Then, we can **decompose** this graph using a set of standard **rules**.

HO: SERIES RULE

HO: PARALLEL RULE

HO: SELF-LOOP RULE

HO: SPLITTING RULE

It's sort of a **graphical** way to do algebra! Let's do some examples:

EXAMPLE: DECOMPOSITION OF SIGNAL FLOW GRAPHS

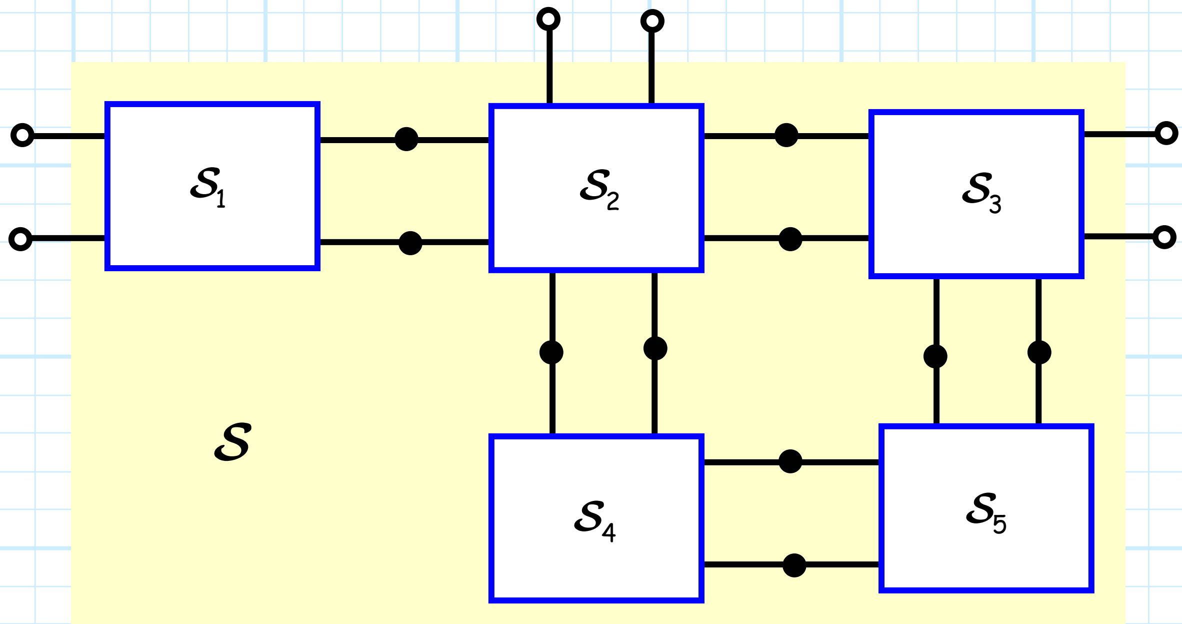
EXAMPLE: SIGNAL FLOW GRAPH ANALYSIS

Signal Flow graphs can likewise help us understand the fundamental **physical behavior** of a network or device. It can even help us **approximate** the network in a way that makes it simpler to analyze and/or design!

HO: THE PROPAGATION SERIES

Signal Flow Graphs

Consider a complex **3-port** microwave network, constructed of **5** simpler microwave devices:



where $\mathcal{S}_n$ is the **scattering matrix** of each device, and $\mathcal{S}$ is the **overall** scattering matrix of the **entire** 3-port network.

Q: *Is there any way to determine this **overall** network scattering matrix $\mathcal{S}$ from the **individual** device scattering matrices $\mathcal{S}_n$?*

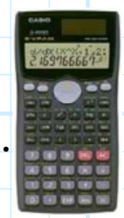
A: **Definitely**! Note the wave **exiting** one port of a device is a wave **entering** (i.e., incident on) another (and vice versa). This is a **boundary condition** at the port connection between devices.

Add to this the scattering parameter equations from each individual device, and we have a **sufficient** amount of math to determine the relationship between the incident and exiting waves of the remaining three ports—in other words, the scattering matrix of the **3-port network**!

Q: *Yikes! Wouldn't that require a lot of* ***tedious*** *algebra!*

A: It sure would! We might use a **computer** to assist us, or we might use a tool employed since the early days of microwave engineering—the **signal flow graph**.

Signal flow graphs are helpful in (count em') **three ways**!



Way 1 - Signal flow graphs provide us with a **graphical** means of **solving** large systems of simultaneous equations.

Way 2 – We'll see the a signal flow graph can provide us with a **road map** of the wave **propagation paths** throughout a microwave device or network. If we're paying attention, we can glean great **physical insight** as to the inner working of the microwave device represented by the graph.

Way 3 - Signal flow graphs provide us with a quick and accurate method for **approximating** a network or device. We will find that we can often replace a rather complex graph with a much **simpler** one that is **almost** equivalent.

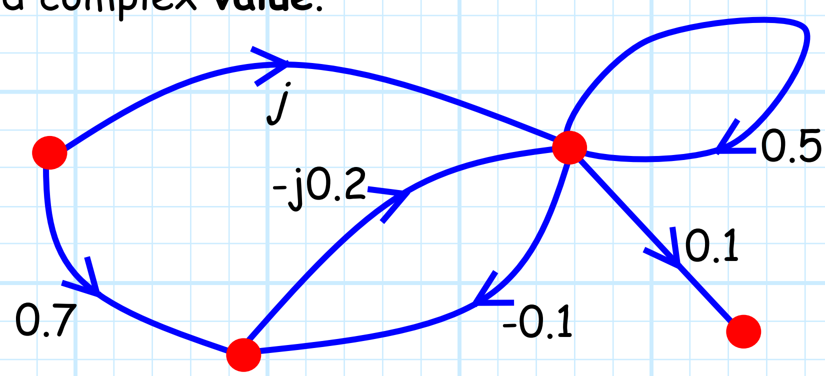
bunny, 64 spheres

We find this to be very helpful when **designing** microwave components. From the analysis of these approximate graphs, we can often determine **design rules** or equations that are tractable, and allow us to design components with (near) optimal performance.

Q: *But what **is** a signal flow graph?*

A: First, some **definitions**!

Every signal flow graph consists of a set of **nodes**. These nodes are connected by **branches**, which are simply contours with a specified **direction**. Each branch likewise has an associated complex **value**.



Q: *What could this possibly have to do with **microwave engineering**?*

A: Each **port** of a microwave device is represented by **two nodes**—the "*a*" node and the "*b*" node. The "*a*" node simply represents the value of the **normalized amplitude** of the wave incident on that port, evaluated **at** the plane of that port:

$$a_n \doteq \frac{V_n^+(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

Likewise, the "*b*" node simply represents the **normalized amplitude** of the wave **exiting** that port, evaluated **at** the plane of that port:

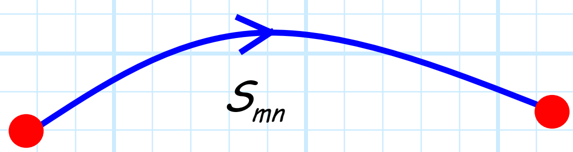
$$b_n \doteq \frac{V_n^-(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

Note then that the **total voltage** at a port is simply:

$$V_n(z_n = z_{nP}) = (a_n + b_n)\sqrt{Z_{0n}}$$

The value of the **branch** connecting two nodes is simply the value of the **scattering parameter** relating these two voltage values:

$$a_n \doteq \frac{V_n^+(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

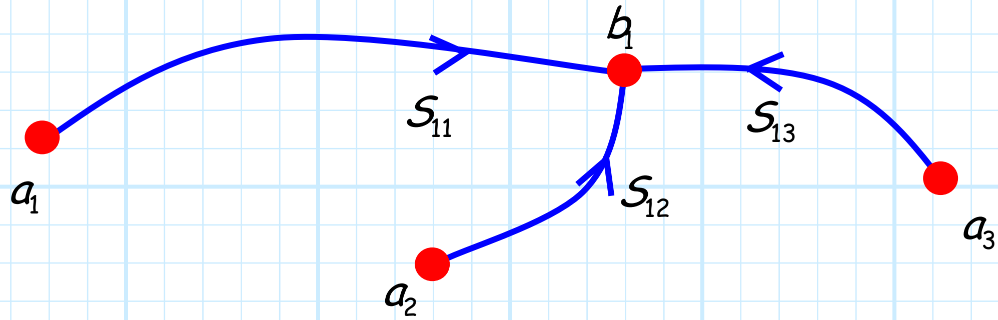


$$b_m \doteq \frac{V_m^-(z_m = z_{mP})}{\sqrt{Z_{0m}}}$$

The signal flow graph above is simply a **graphical** representation of the equation:

$$b_m = S_{mn}\, a_n$$

Moreover, if **multiple** branches enter a node, then the voltage represented by that node is the **sum** of the values from each branch. For example, the signal flow graph:



is a **graphical** representation of the equation:

$$b_1 = S_{11}\,a_1 + S_{12}\,a_2 + S_{13}\,a_3$$

Now, consider a **two-port device** with a scattering matrix $\mathcal{S}$:

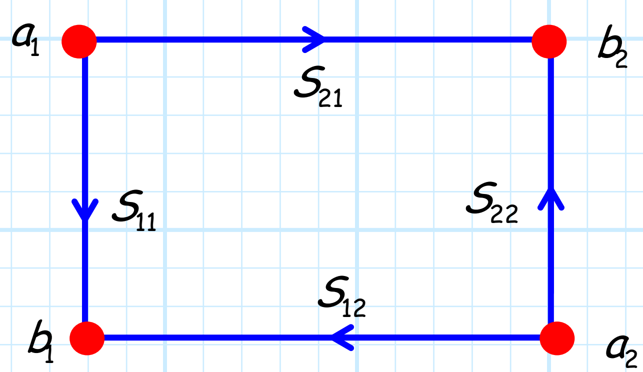
$$\mathcal{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

So that:

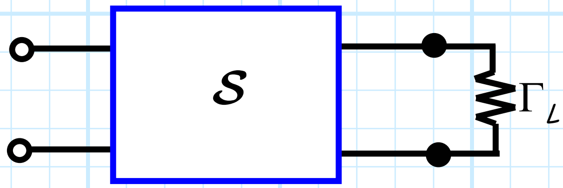
$$b_1 = S_{11}\,a_1 + S_{12}\,a_2$$

$$b_2 = S_{21}\,a_1 + S_{22}\,a_2$$

We can thus **graphically** represent a **two-port device** as:



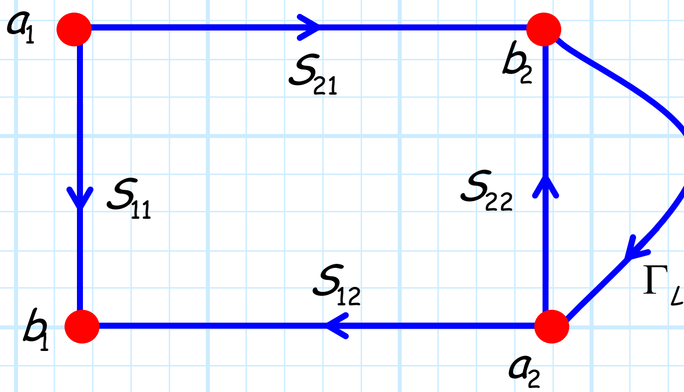
Now, consider a case where the second port is **terminated** by some load Γ_L:



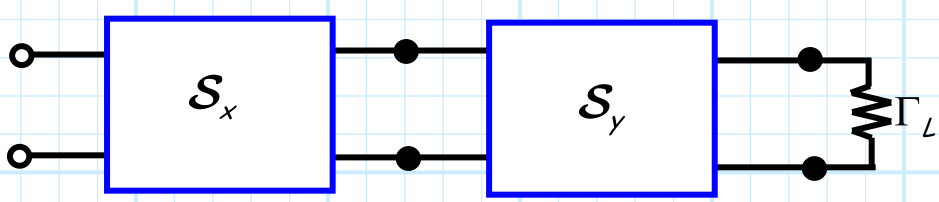
We now have yet **another** equation:

$$V_2^+(z_2 = z_{2P}) = \Gamma_L \, V_2^-(z_2 = z_{2P})$$
$$a_2 = \Gamma_L \, b_2$$

Therefore, the signal flow graph of this **terminated** network is:



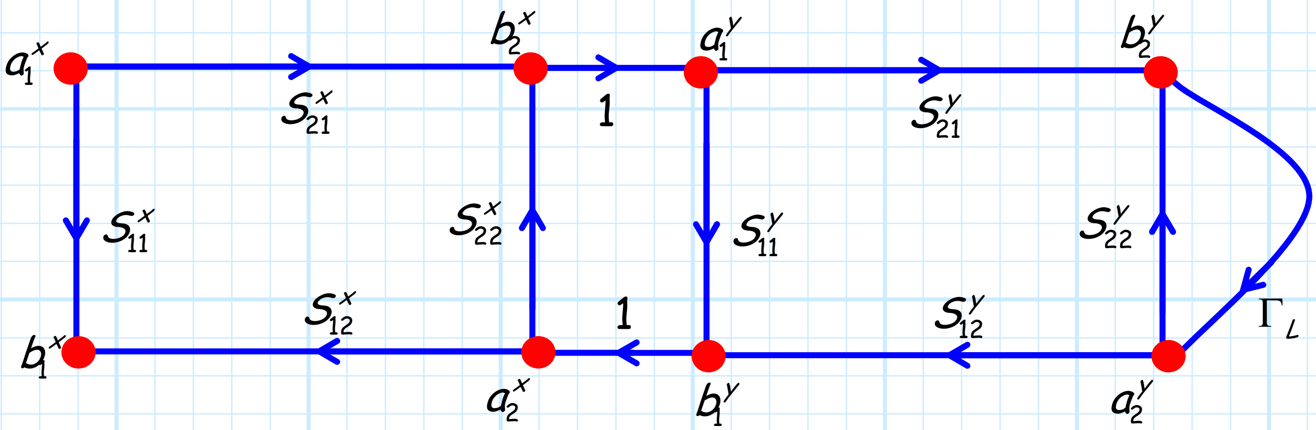
Now let's cascade **two different** two-port networks



Here, the output port of the first device is **directly** connected to the input port of the second device. We describe this mathematically as:

$$a_1^y = b_2^x \qquad \text{and} \qquad b_1^y = a_2^x$$

Thus, the signal flow graph of this network is:

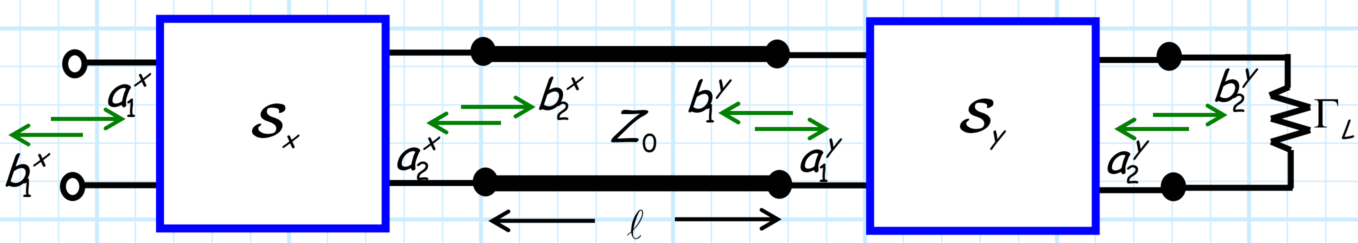


Q: *But what happens if the networks are connected with **transmission lines?***

A: Recall that a length ℓ of transmission line with characteristic impedance Z_0 is likewise a **two-port** device. Its scattering matrix is:

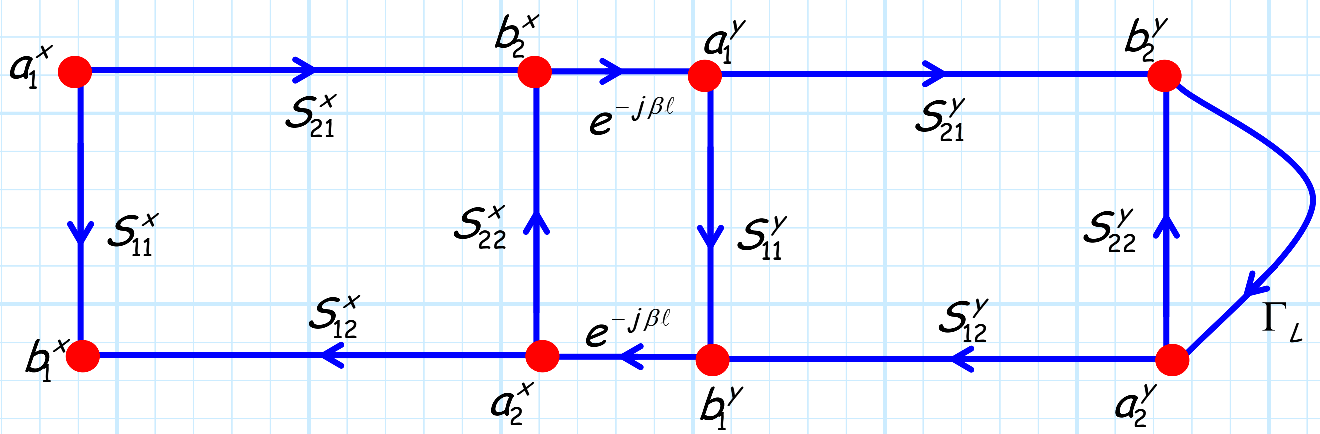
$$\mathcal{S} = \begin{bmatrix} 0 & e^{-j\beta\ell} \\ e^{-j\beta\ell} & 0 \end{bmatrix}$$

Thus, if the two devices are connected by a length of **transmission line**:



$$a_1^y = e^{-j\beta\ell}\, b_2^x \qquad\qquad a_2^x = e^{-j\beta\ell}\, b_1^y$$

so the signal flow graph is:



Note that there is **one** (and only one) **independent variable** in this representation.

This **independent** variable is node a_1^x.

This is the only node of the *sfg* that does **not** have any **incoming** branches. As a result, its value depends on **no other** node values in the *sfg*.

→ From the standpoint of a *sfg*, independent nodes are essentially **sources**!

Of course, this likewise makes sense physically (do **you** see why?). The node value a_1^x represents the complex amplitude of the wave **incident** on the one-port network. If this value is **zero**, then **no power** is incident on the network—the rest of the nodes (i.e., wave amplitudes) will likewise be **zero**!

Now, say we wish to determine, for example:

1. The **reflection coefficient** Γ_{in} of the one-port device.

2. The **total current** at port 1 of second network (i.e., network y).

3. The **power absorbed** by the load at port 2 of the second (y) network.

In the first case, we need to determine the value of dependent node b_1^x:

$$\Gamma_{in} = \frac{b_1^x}{a_1^x}$$

For the second case, we must determine the value of wave amplitudes a_1^y and b_1^y:

$$I_1^y = \frac{a_1^y - b_1^y}{\sqrt{Z_0}}$$

And for the third and final case, the values of nodes a_2^y and b_2^y are required:

$$P_{abs} = \frac{\left|b_2^y\right|^2 - \left|a_2^y\right|^2}{2}$$

Q: *But just how the heck do we* ***determine*** *the values of these wave amplitude "nodes"?*

A: One way, of course, is to solve the **simultaneous equations** that describe this network.

From network x and network y:

$$b_1^x = S_{11}^x a_1^x + S_{12}^x a_2^x \qquad b_1^y = S_{11}^y a_1^y + S_{12}^y a_2^y$$

$$b_2^x = S_{21}^x a_1^x + S_{22}^x a_2^x \qquad b_2^y = S_{21}^y a_1^y + S_{22}^y a_2^y$$

From the transmission line:

$$a_1^y = e^{-j\beta\ell}\, b_2^x \qquad a_2^x = e^{-j\beta\ell}\, b_1^y$$

And finally from the load:

$$a_2 = \Gamma_L\, b_2$$

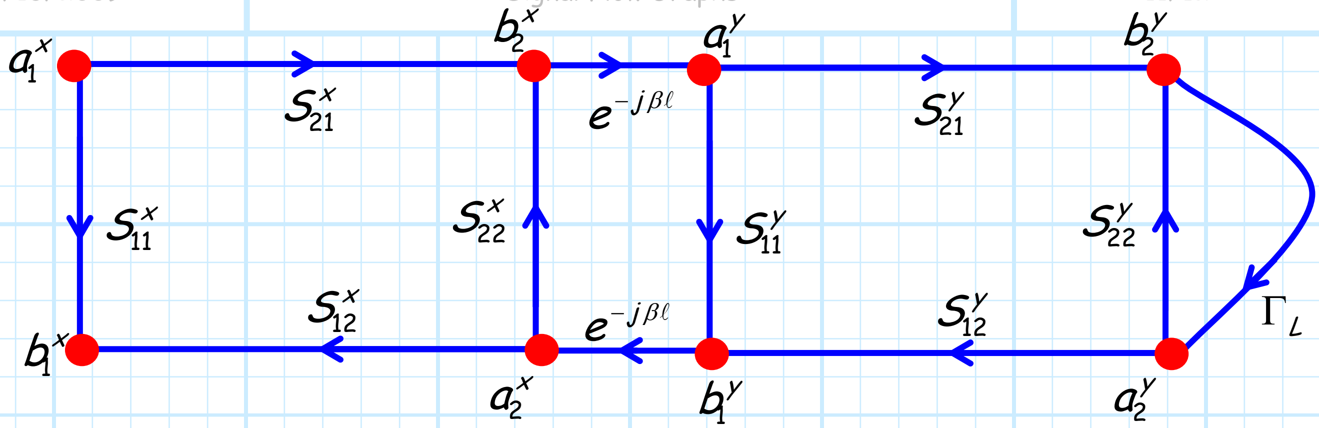
But another, **EVEN BETTER** way to determine these values is to **decompose (reduce)** the signal flow graph!

Q: *Huh?*

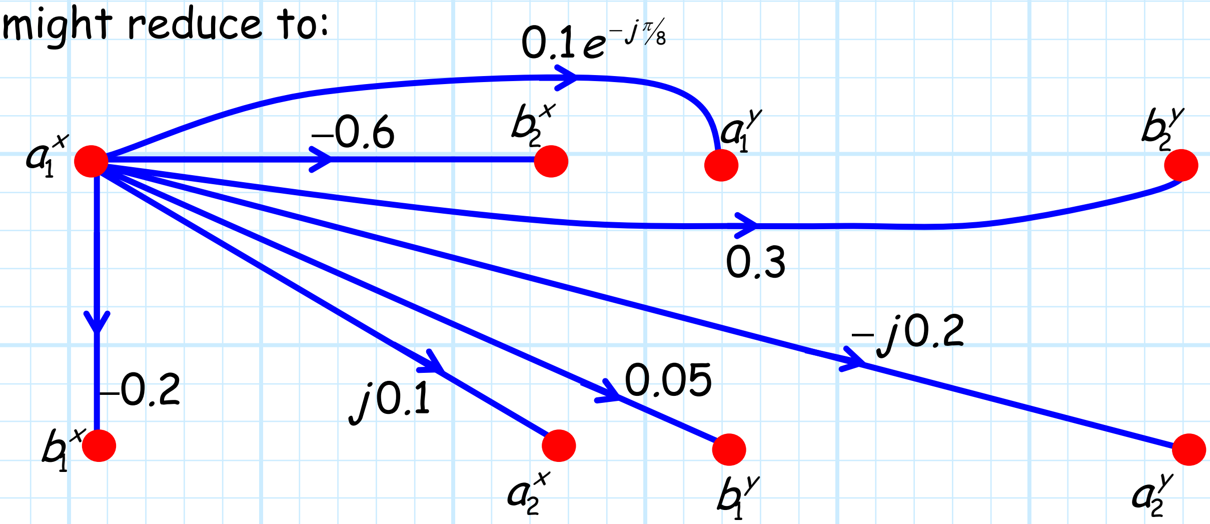
A: Signal flow graph **reduction** is a method for **simplifying** the **complex** paths of that signal flow graph into a more **direct** (but equivalent!) form.

Reduction is really just a **graphical** method of **decoupling** the simultaneous equations that are **described** by the *sfg*.

For instance, in the example we are considering, the *sfg* :



might reduce to:



From **this** graph, we can **directly** determine the value of each node (i.e., the value of each wave amplitude), in terms of the one independent variable a_1^x.

$$b_1^x = -0.2\, a_1^x$$

$$b_2^x = -0.6\, a_1^x \qquad a_2^x = j\,0.1\, a_1^x$$

$$b_1^y = 0.05\, a_1^x \qquad a_1^y = 0.1\, e^{-j\pi/8} a_1^x$$

$$b_2^y = 0.3\, a_1^x \qquad a_2^y = -0.2\, a_1^x$$

And of course, we can then determine values like:

1. $$\Gamma_{in} = \frac{b_1^x}{a_1^x} = \frac{-0.2\, a_1^x}{a_1^x} = -0.2$$

2. $$I_1^y = \frac{a_1^y - b_1^y}{\sqrt{Z_0}} = \frac{0.1\,e^{-j\pi/8} - 0.05}{\sqrt{Z_0}}\, a_1^x$$

3. $$P_{abs} = \frac{\left|b_2^y\right|^2 - \left|a_2^y\right|^2}{2} = \frac{(0.3)^2 - (0.2)^2}{2}\left|a_1^x\right|^2$$

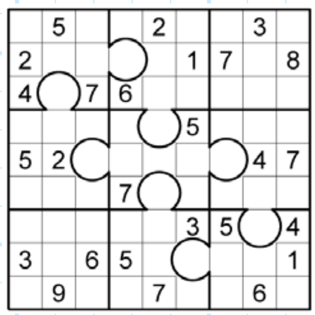
Q: *But **how** do we reduce the sfg to its simplified state? Just what is the **procedure**?*

A: Signal flow graphs can be reduced by sequentially applying one of **four simple rules**.

Q: *Can these rules be applied in **any order**?*

A: **No!** The rules can only be applied when/where the structure of the *sfg* allows. You must **search** the *sfg* for structures that allow a rule to be applied, and the *sfg* will then be (a little bit) reduced. You then search for the **next** valid structure where a rule can be applied. Eventually, the *sfg* will be **completely reduced**!

Q: *????*



A: It's a bit like solving a **puzzle**. Every *sfg* is different, and so each will require a different reduction procedure. It requires a little **thought**, but with a little practice, the reduction procedure is ***easily*** mastered.

You may even find that it's kind of **fun**!

Series Rule

Consider these two complex equations:

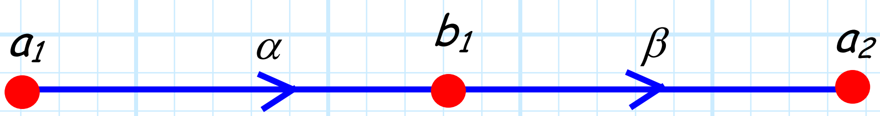
$$b_1 = \alpha\, a_1 \qquad\qquad a_2 = \beta\, b_1$$

where α and β are **arbitrary** complex constants. Using the **associative property** of multiplication, these two equations can combined to form an **equivalent set** of equations:

$$b_1 = \alpha\, a_1 \qquad\qquad a_2 = \beta\, b_1 = \beta\left(\alpha\, a_1\right) = \left(\alpha\beta\right) a_1$$

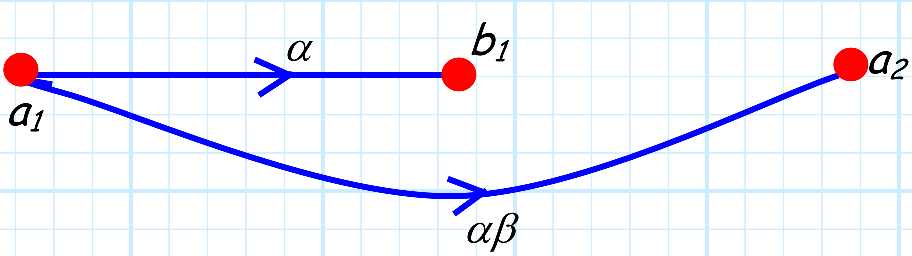
Now let's express these two sets of equations as **signal flow graphs**!

The first set provides:



$$b_1 = \alpha\, a_1$$
$$a_2 = \beta\, b_1$$

While the second is:



$$b_1 = \alpha\, a_1$$
$$a_2 = \alpha\beta\, a_1$$

Q: *Hey wait! If the two sets of equations are* ***equivalent****, shouldn't the two resulting signal flow graphs* ***likewise*** *be equivalent?*

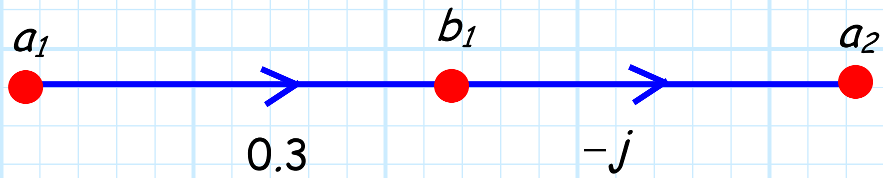
A: Absolutely! The two signal flow graphs are indeed **equivalent**.

This leads us to our **first** signal flow graph **reduction rule**:

Rule 1 - Series Rule

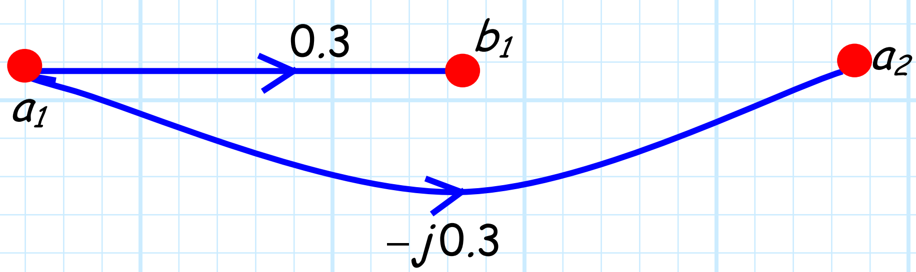
If a node has ***one*** *(and only one!) incoming branch, and* ***one*** *(and only one!) outgoing branch, the node can be eliminated and the two branches can be combined, with the new branch having a value equal to the product of the original two.*

For **example**, the graph:



$$b_1 = 0.3\,a_1$$
$$a_2 = -j\,b_1$$

can be reduced to:



$$b_1 = 0.3\,a_1$$
$$a_2 = -j0.3\,a_1$$

Parallel Rule

Consider the complex equation:

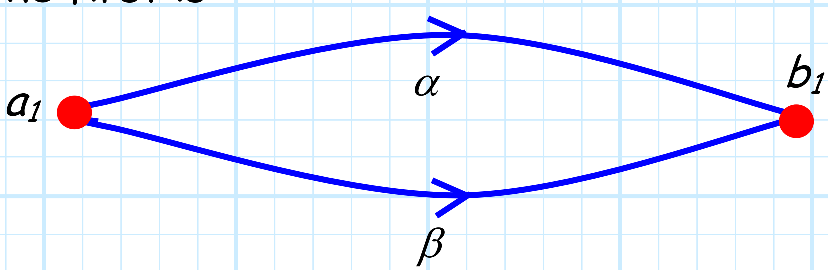
$$b_1 = \alpha\, a_1 + \beta\, a_1$$

where α and β are **arbitrary** complex constants. Using the **distributive property**, the equation can equivalently be expressed as:

$$b_1 = (\alpha + \beta)\, a_1$$

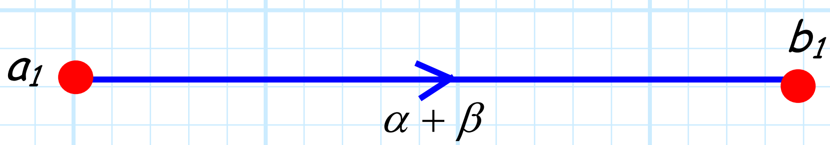
Now let's express these two equations as **signal flow graphs**!

The first is:



$$b_1 = \alpha\, a_1 + \beta\, a_1$$

With the second:



$$b_1 = (\alpha + \beta)\, a_1$$

Q: *Hey wait! If the two equations are **equivalent**, shouldn't the two resulting signal flow graphs **likewise** be equivalent?*

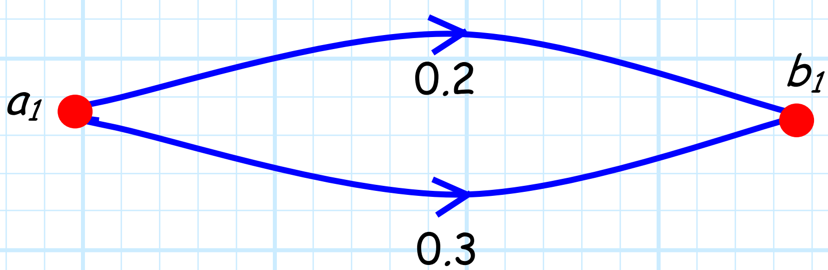
A: Absolutely! The two signal flow graphs are indeed **equivalent**.

This leads us to our **second** signal flow graph reduction rule:

Rule 2 - Parallel Rule

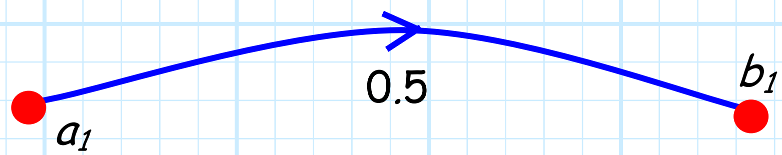
*If two nodes are connected by parallel branches—and the branches have the **same direction**—the branches can be combined into a single branch, with a value equal to the **sum** of each two original branches.*

For example, the graph:



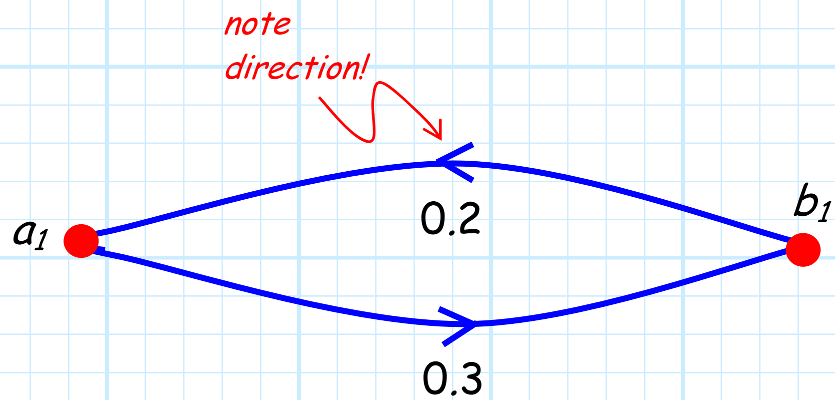
$$b_1 = 0.3\,a_1 + 0.2\,a_1$$

Can be reduced to:

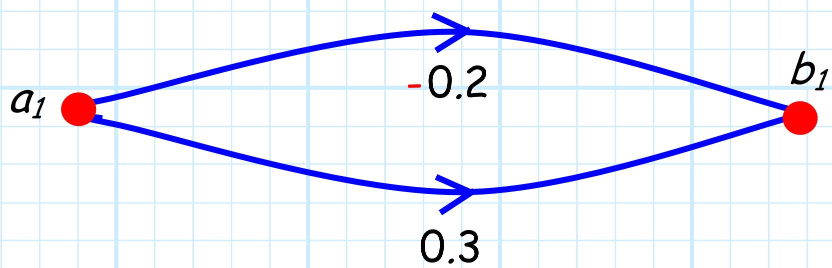


$$\begin{aligned} b_1 &= (0.3 + 0.2)\,a_1 \\ &= 0.5\,a_1 \end{aligned}$$

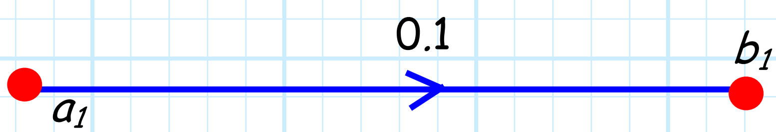
Q: *What about **this** signal flow graph?*



***Can** I rewrite this as:*



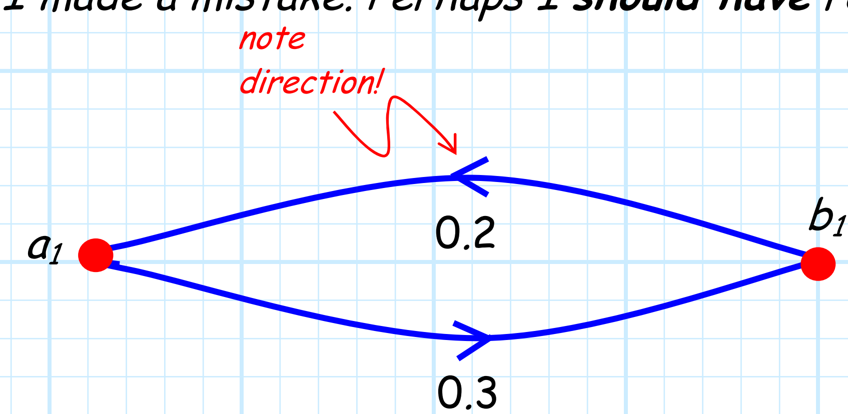
so that (since 0.3-0.2=0.1):



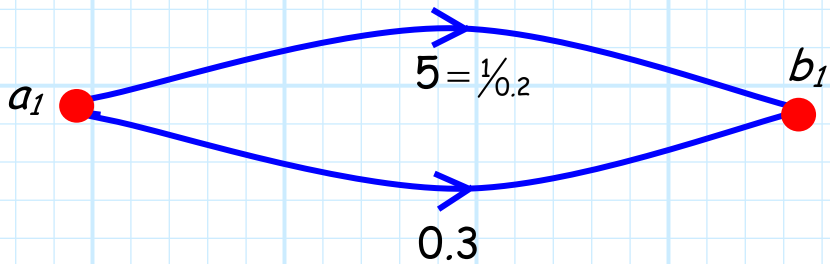
???

A: Absolutely not! **NEVER DO THIS!!** 

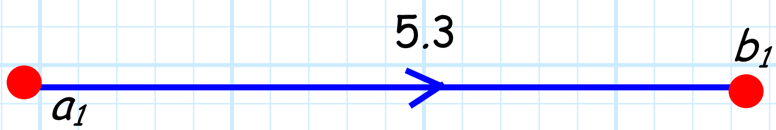
Q: *Maybe I made a mistake. Perhaps I **should have** rewritten:*



as this:



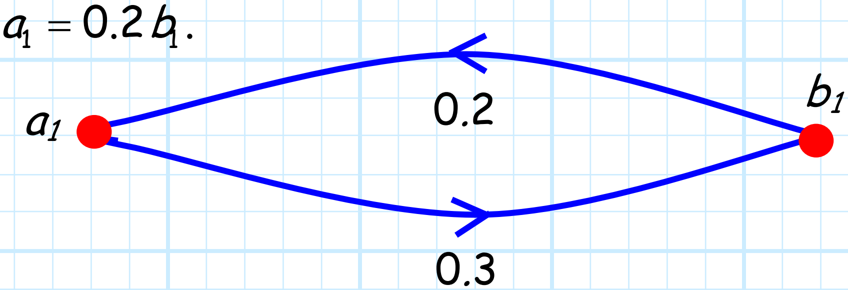
so that (since 5.0+0.3=5.3):



???

A: Absolutely not! **NEVER DO THIS EITHER!!** 

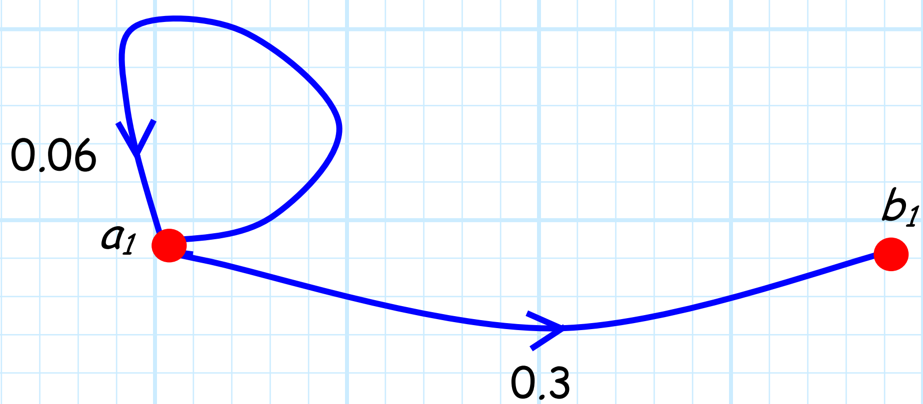
From the signal flow graph below, we can **only** conclude that $b_1 = 0.3\,a_1$ and $a_1 = 0.2\,b_1$.



Using the **series rule** (or little bit of algebra), we can conclude that an **equivalent** signal flow graph to this is:

$a_1 = 0.06\,a_1$

$b_1 = 0.3\,a_1$



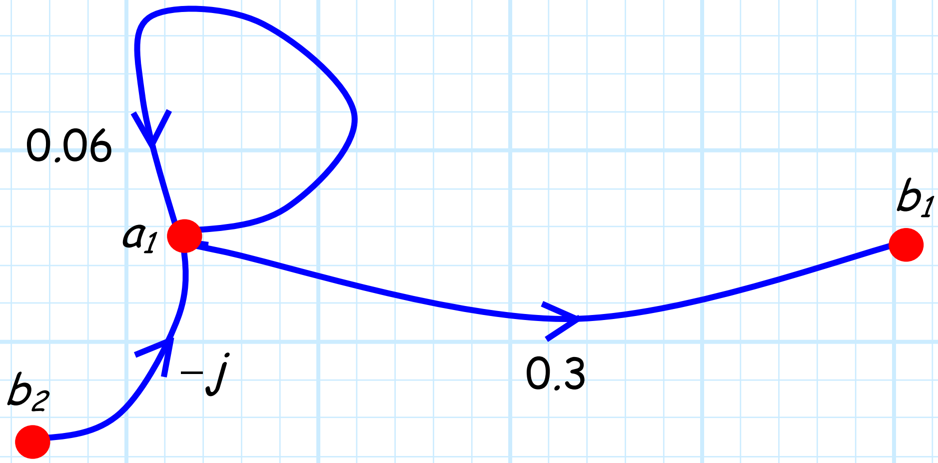
Q: *Yikes! What kind of **goofy** branch begins and ends at the same node?*

A: Branches that begin and end at the same node are called **self-loops**.

Q: *Do these self-loops actually **appear** in signal flow graphs?*

A: **Yes**, but the self-loop node will **always** have at least **one other incoming branch**. For example:

$$a_1 = 0.06\,a_1 - j\,b_2$$
$$b_1 = 0.3\,a_1$$



Q: *But how do we **reduce** a signal flow graph containing a **self-loop**?*

A: See **rule 3** !

Self-Loop Rule

Now consider the equation:

$$b_1 = \alpha\, a_1 + \beta\, a_2 + \gamma\, b_1$$

A little dab of **algebra** allows us to determine the value of node b_1:

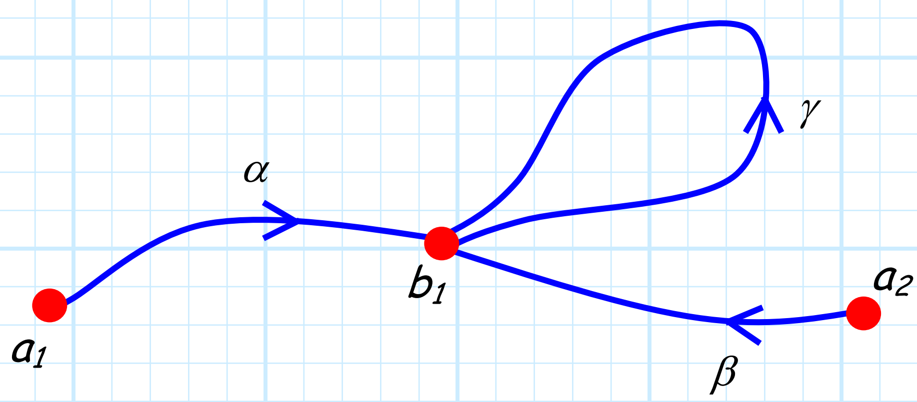
$$b_1 = \alpha\, a_1 + \beta\, a_2 + \gamma\, b_1$$

$$b_1 - \gamma\, b_1 = \alpha\, a_1 + \beta\, a_2$$

$$(1-\gamma)\, b_1 = \alpha\, a_1 + \beta\, a_2$$

$$b_1 = \frac{\alpha}{1-\gamma} a_1 + \frac{\beta}{1-\gamma} a_2$$

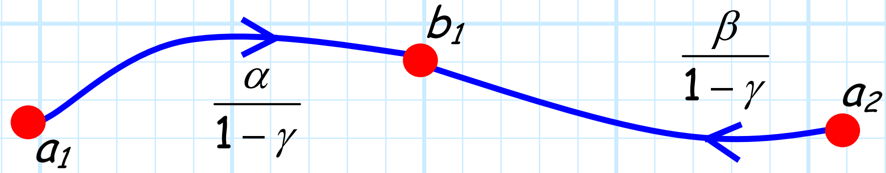
The signal flow graph of the **first** equation is:



$$b_1 = \alpha\, a_1 + \beta\, a_2 + \gamma\, b_1$$

While the signal flow graph of the **second** is:

$$b_1 = \frac{\alpha}{1-\gamma} a_1 + \frac{\beta}{1-\gamma} a_2$$



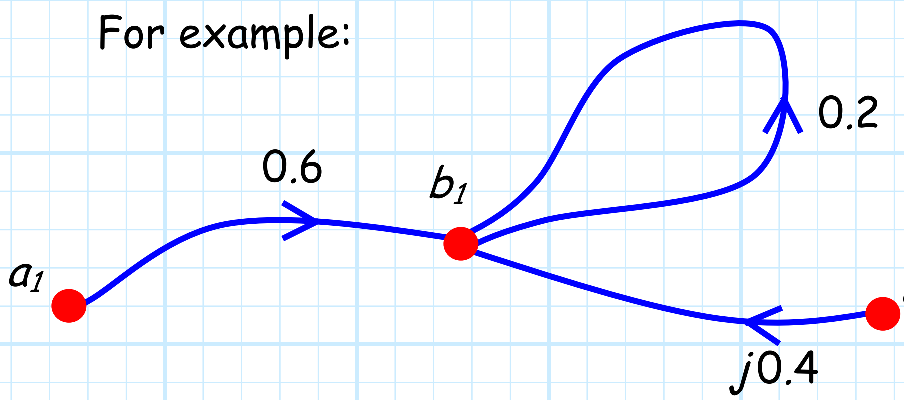
These two signal flow graphs are **equivalent**!

Note the self-loop has been **"removed"** in the second graph. Thus, we now have a method for removing self-loops. This method is **rule 3**.

Rule 3 – Self-Loop Rule

*A self-loop can be eliminate by multiplying **all** of the branches "**feeding**" the self-loop node by $1/(1-S_{sl})$, where S_{sl} is the value of the self loop branch.*

For example:

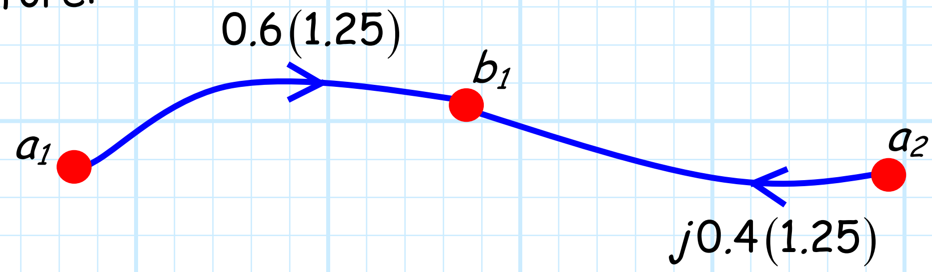


$$b_1 = 0.6\, a_1 + j0.4 a_2 + 0.2 b_1$$

can be simplified by **eliminating the self-loop**. We multiply **both** of the two branches **feeding** the self-loop node by:

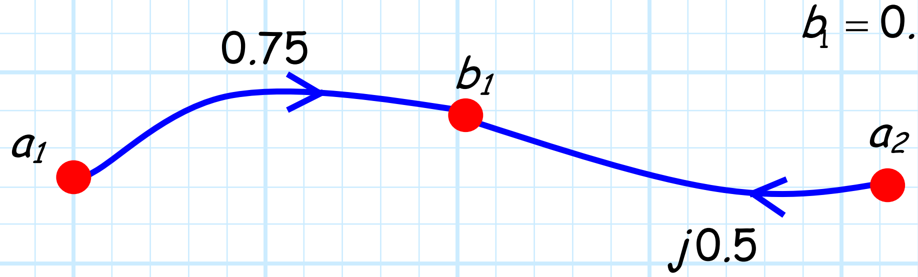
$$\frac{1}{1-S_{sl}} = \frac{1}{1-0.2} = 1.25$$

Therefore:



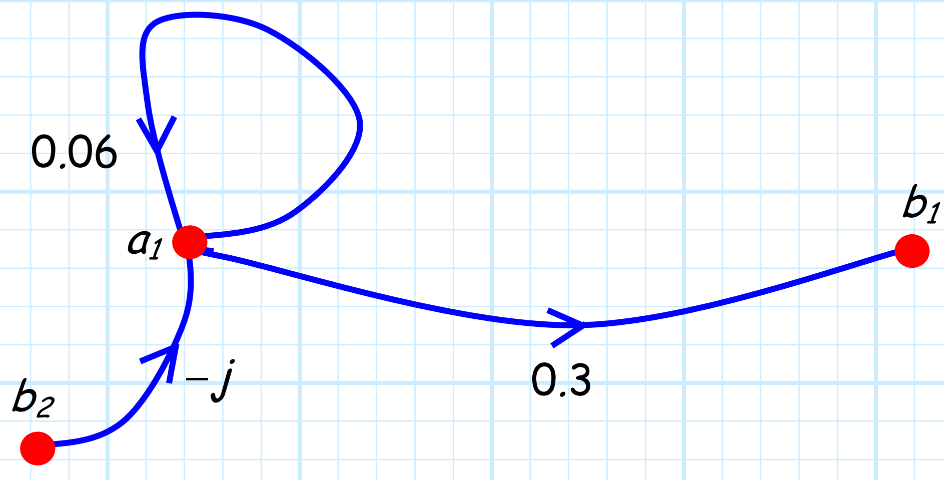
And thus:

$$b_1 = 0.75\,a_1 + j0.5\,a_2$$



Or **another example**:

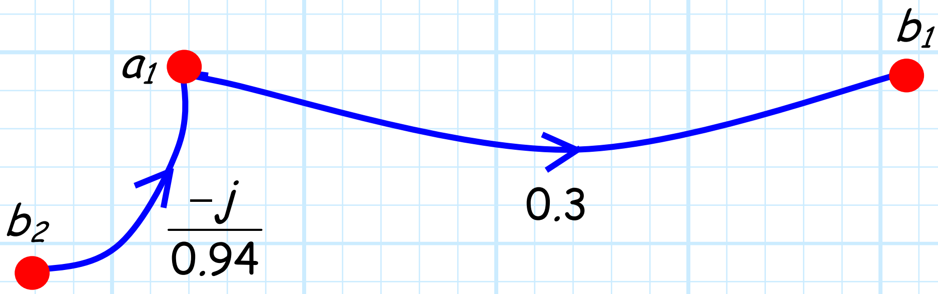
$$a_1 = 0.06\,a_1 - j\,b_2$$
$$b_1 = 0.3\,a_1$$



becomes after reduction using **rule 3**:

$$a_1 = \frac{-j}{0.94} b_2$$

$$b_1 = 0.3\, a_1$$



Q: *Wait a minute! I think you **forgot** something. Shouldn't you **also** divide the 0.3 branch value by* $1-0.06=0.94$ *??*

A: Nope! The 0.3 branch is **exiting** the self-loop node a_1. **Only** incoming branches (e.g., the $-j$ branch) to the self-loop node are modified by the self-loop rule!

Splitting Rule

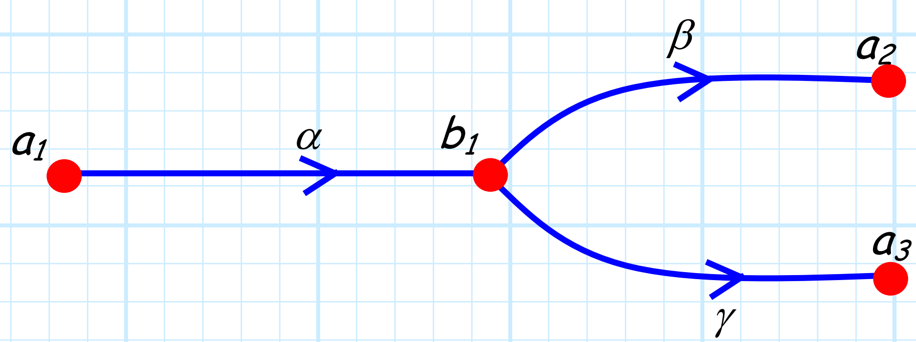
Now consider these **three** equations:

$$b_1 = \alpha \, a_1$$
$$a_2 = \beta \, b_1$$
$$a_3 = \gamma \, b_1$$

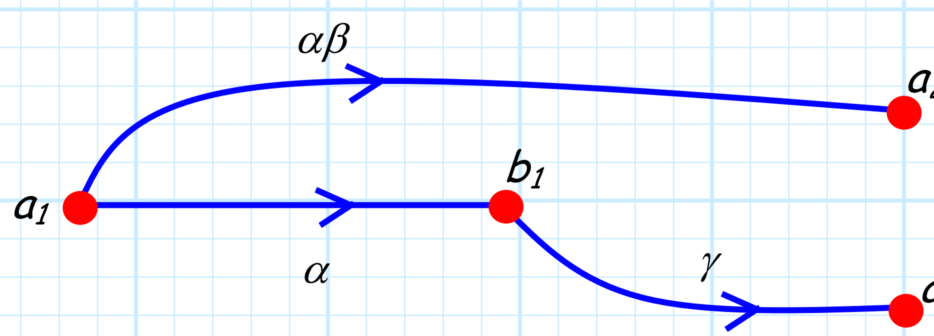
Using the **associative property,** we can likewise write an equivalent set of equations:

$$b_1 = \alpha \, a_1$$
$$a_2 = \alpha\beta \, a_1$$
$$a_3 = \alpha \, b_1$$

The signal flow graph of the **first** set of equations is:



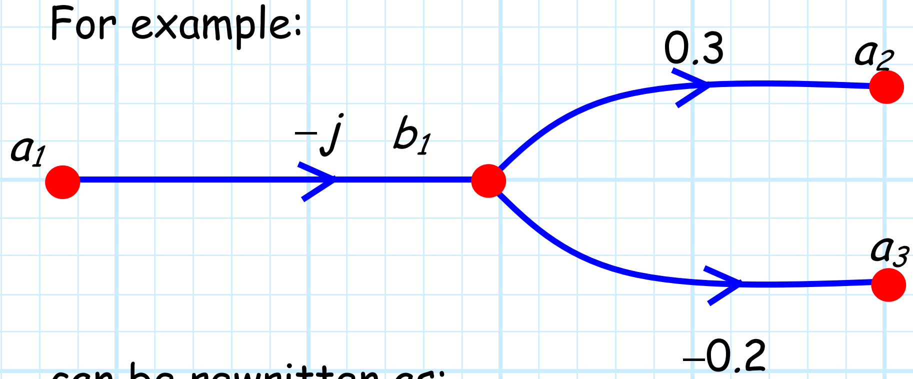
While the signal flow graph of the **second** is:



Rule 4 – Splitting Rule

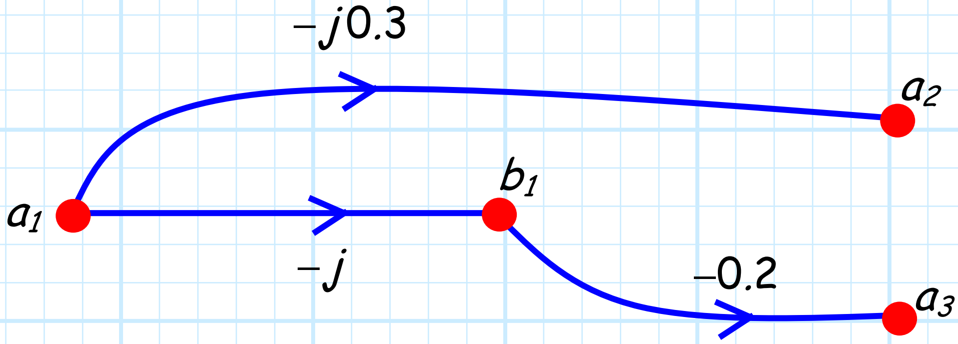
If a node has one (and only one!) incoming branch, and one (or more) exiting branches, the incoming branch can be "split", and directly combined with each of the exiting branches.

For example:



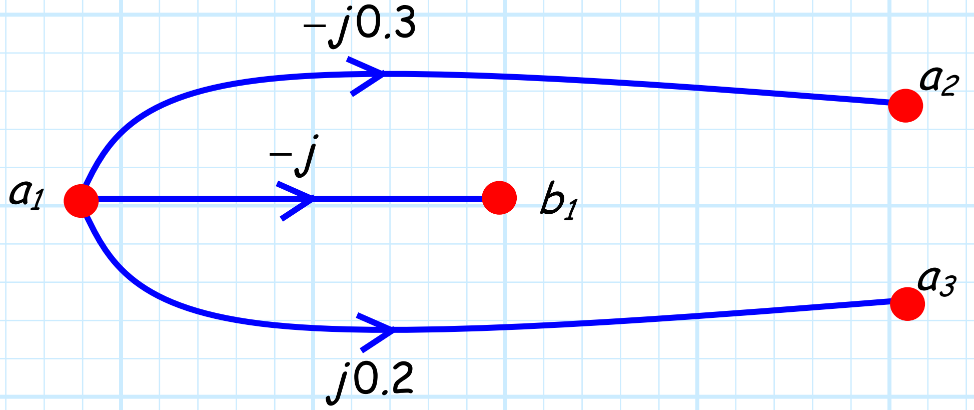
$$b_1 = -j\ a_1$$
$$a_2 = 0.3\ b_1$$
$$a_3 = -0.2\ b_1$$

can be rewritten as:



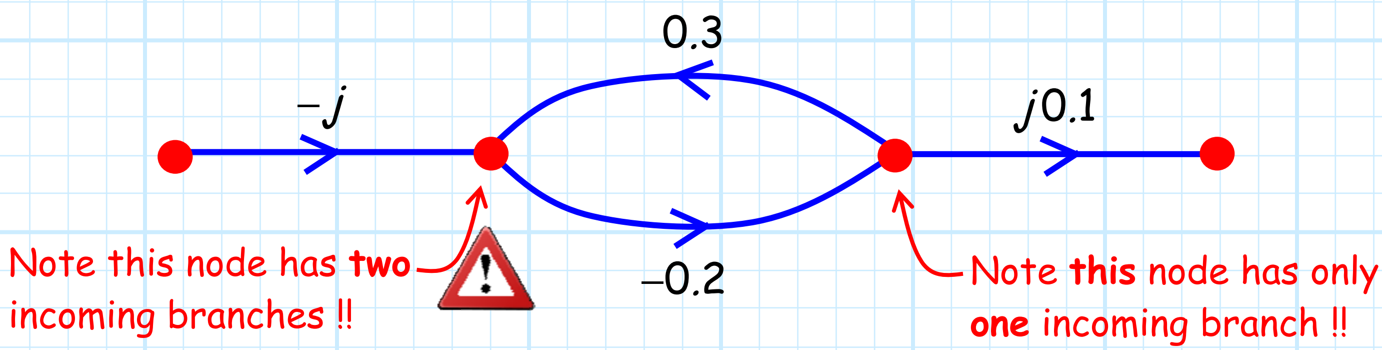
$$b_1 = -j\ a_1$$
$$a_2 = -j0.3\ a_1$$
$$a_3 = -0.2\ b_1$$

Of course, from rule 1 (or from rule 4!), this graph can be **further** simplified as:

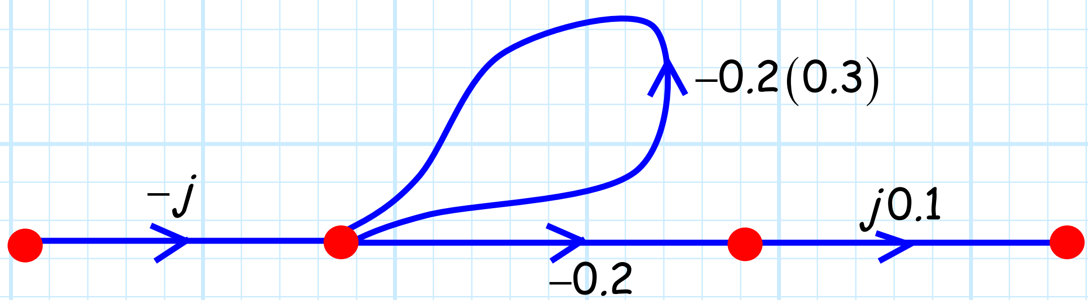


$$b_1 = -j\ a_1$$
$$a_2 = -j0.3\ a_1$$
$$a_3 = j0.2\ a_1$$

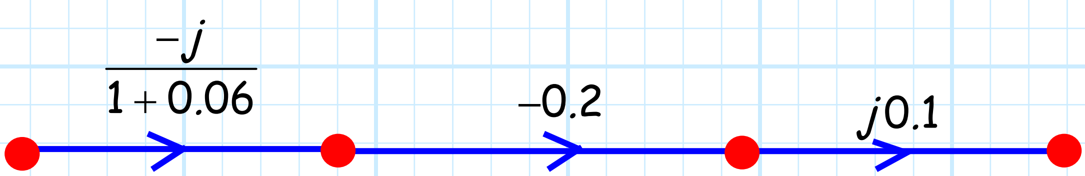
The splitting rule is **particularly useful** when we encounter signal flow graphs of **the kind**:



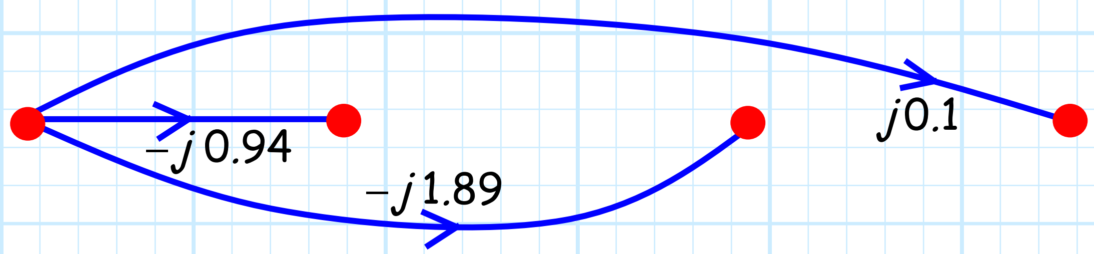
We can **split** the -0.2 branch, and rewrite the graph as:



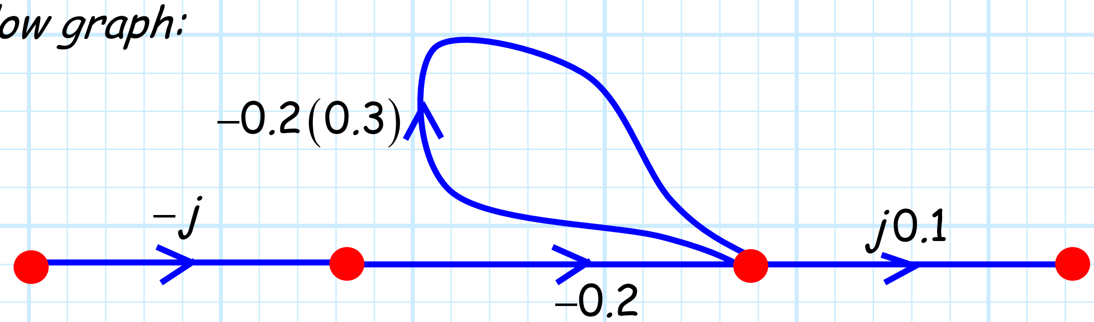
Note we now have a **self-loop**, which can be eliminated using **rule #3**:



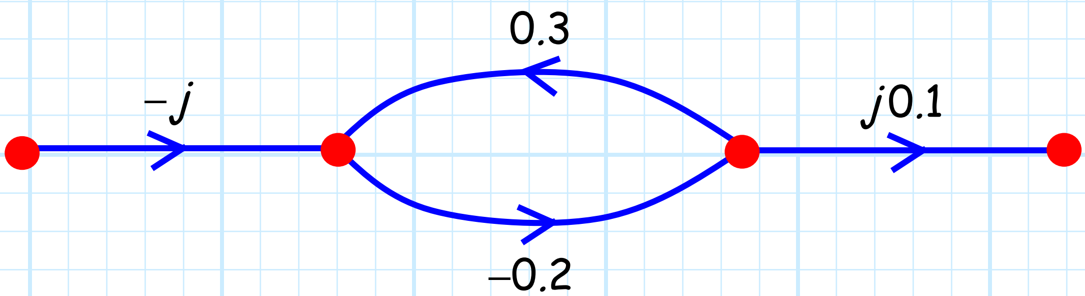
Note that this graph can be further simplified using **rule #1**.



Q: *Can we split the* ***other*** *branch of the loop? Is* ***this*** *signal flow graph:*



Likewise equivalent to this one ??:



A: NO!! Do not make this mistake! We **cannot** split the 0.3 branch because it terminates in a node with **two** incoming branches (i.e., $-j$ and 0.3). This is a **violation** of rule 4.

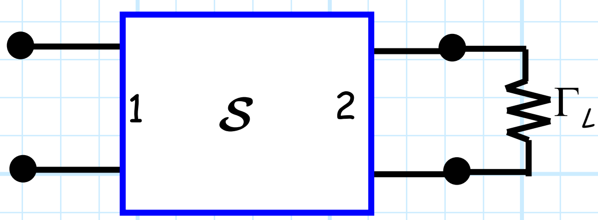
Moreover, the equations represented by the two signal flow graphs are **not** equivalent—they two graphs describe two **different** sets of equations!

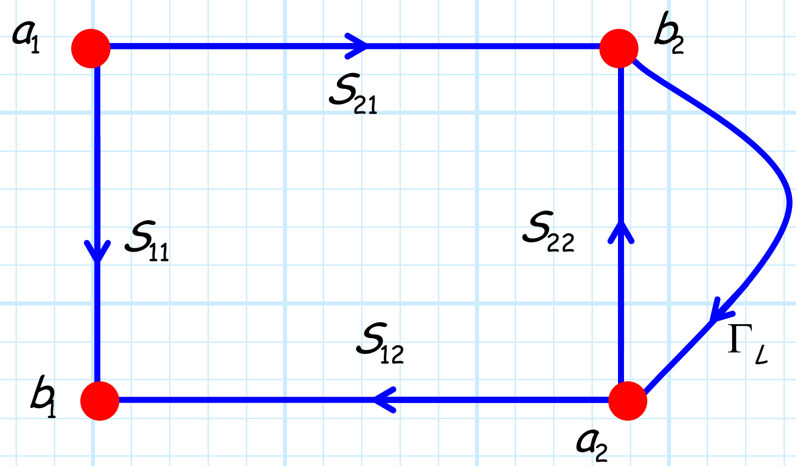
It is important to remember that there is no "magic" behind signal flow graphs. They are simply a **graphical** method of representing—and then solving—a set of linear equations.

As such, the four basic **rules** of analyzing a signal flow graph represent basic **algebraic** operations. In fact, signal flow graphs can be applied to the analysis of **any** linear system, not just microwave networks.

Example: Decomposition of Signal Flow Graphs

Consider the basic 2-port network, terminated with load Γ_L.





Say we want to determine the value:

$$\Gamma_1 \doteq \frac{V_1^-(z=z_{1P})}{V_1^+(z=z_{1P})} = \frac{b_1}{a_1} \quad ??$$

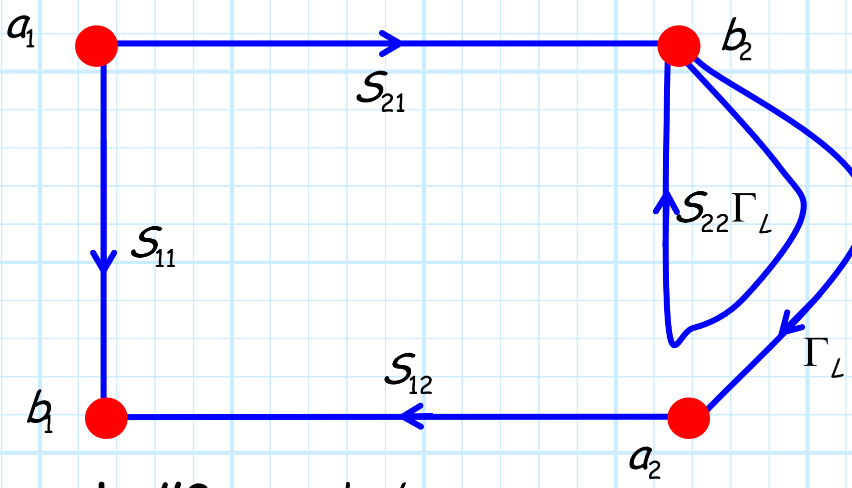
In other words, what is the **reflection coefficient** of the resulting **one-port** device?

Q: *Isn't this simply S_{11} ?*

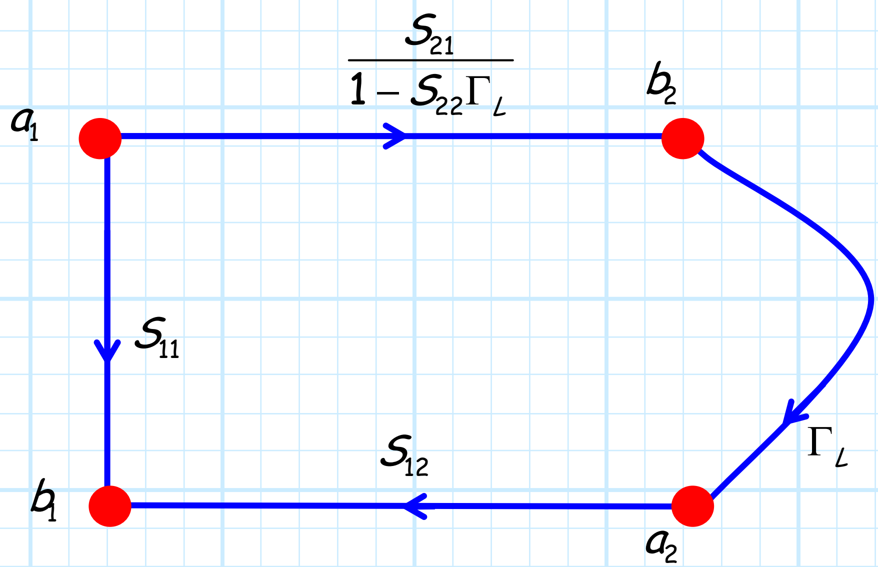
A: **Only** if $\Gamma_L = 0$ (and it's **not**)!!

So let's decompose (simplify) the signal flow graph and find out!

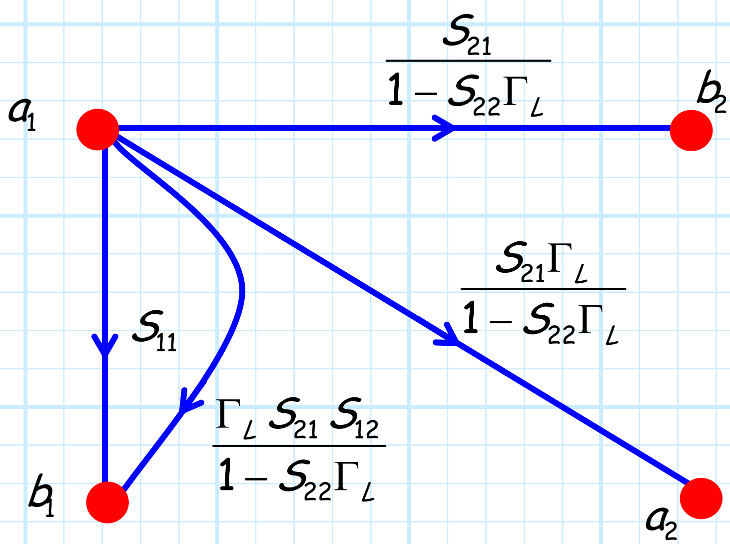
Step 1: Use **rule #4** on node a_2



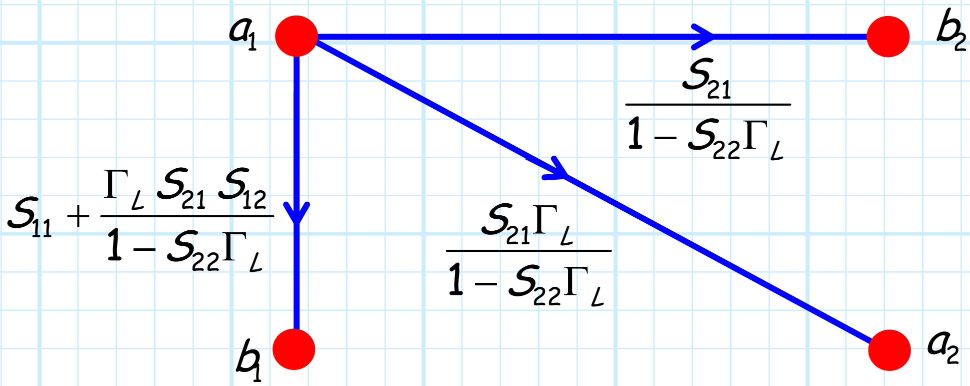
Step 2: Use **rule #3** on node b_2



Step 3: And then using **rule #1**:



Step 4: Use **rule 2** on nodes a_1 and b_1



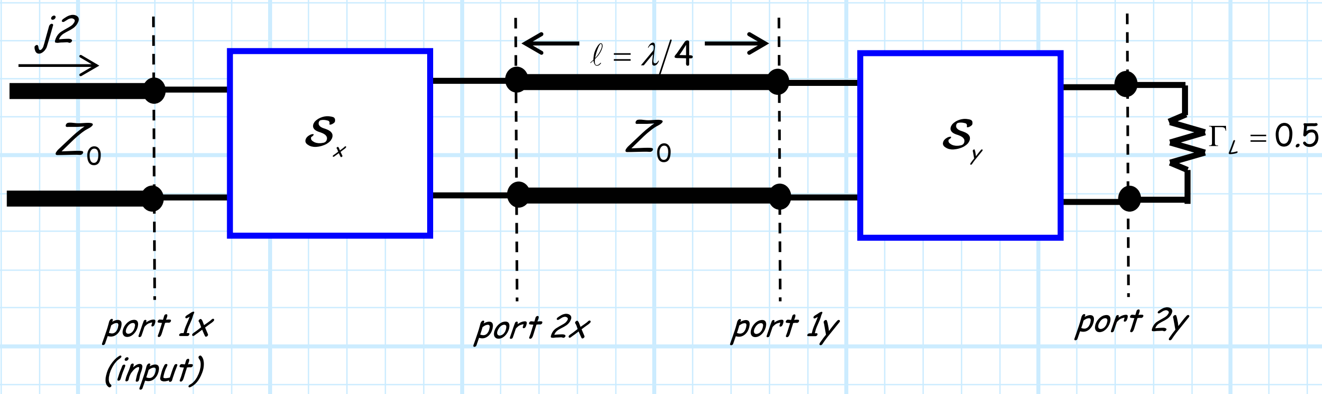
Therefore:

$$\Gamma_1 = \frac{b_1}{a_1} = S_{11} + \frac{\Gamma_L S_{21} S_{12}}{1-S_{22}\Gamma_L}$$

Note if $\Gamma_L = 0$, then $b_1/a_1 = S_{11}$!

Example: Analysis Using Signal Flow Graphs

Below is a **single**-port device (with **input** at port 1a) constructed with two two-port devices ($\mathcal{S}_x$ and $\mathcal{S}_y$), a quarter wavelength transmission line, and a load impedance.



Where $Z_0 = 50\Omega$.

The scattering matrices of the two-port devices are:

$$\mathcal{S}_x = \begin{bmatrix} 0.35 & 0.5 \\ 0.5 & 0 \end{bmatrix} \qquad \mathcal{S}_y = \begin{bmatrix} 0 & 0.8 \\ 0.8 & 0.4 \end{bmatrix}$$

Likewise, we know that the value of the voltage wave **incident** on port 1 of device $\mathcal{S}_x$ is:

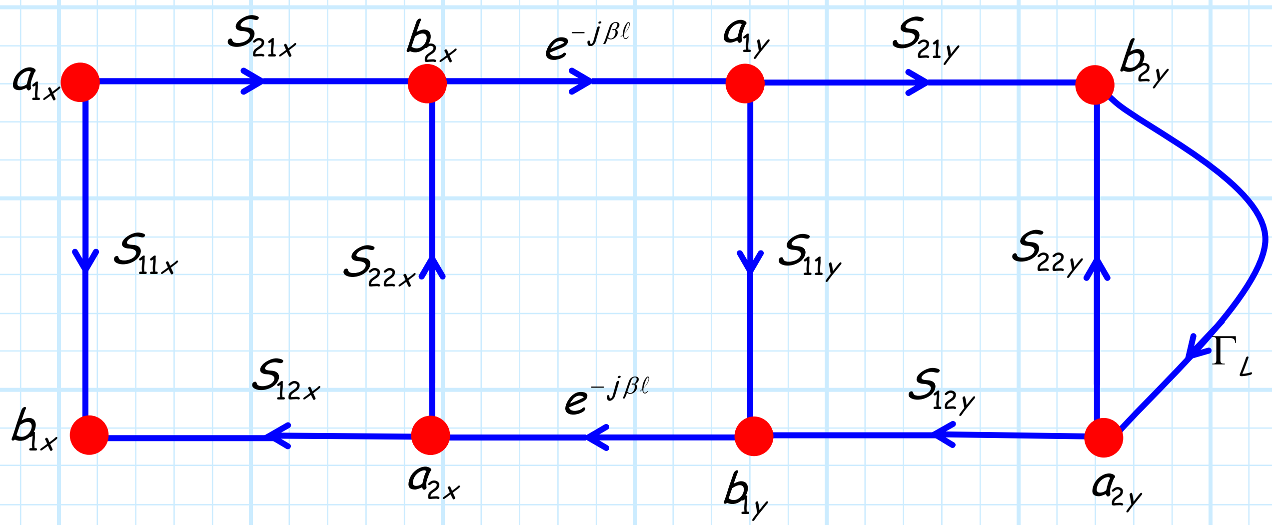
$$a_{1x} \doteq \frac{V_{01x}^{+}\left(z_{1x} = z_{1xP}\right)}{\sqrt{Z_0}} = \frac{j2}{\sqrt{50}} = \frac{j\sqrt{2}}{5} \quad V$$

Now, let's draw the complete **signal flow graph** of this circuit, and then reduce the graph to determine:

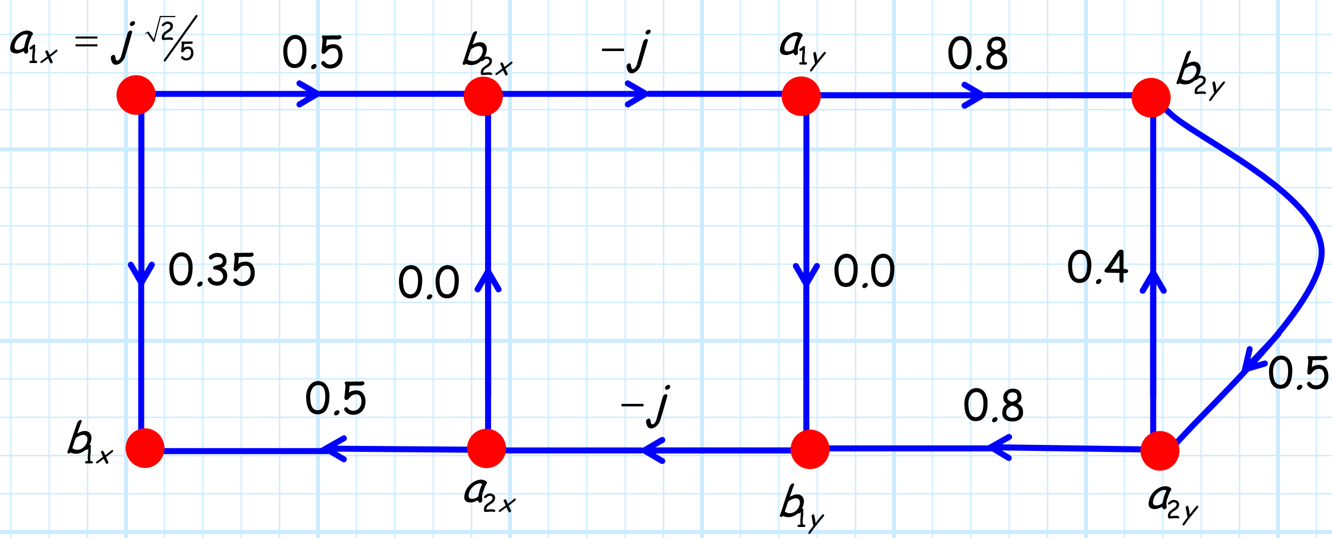
a) The total current through load Γ_L.

b) The power delivered to (i.e., absorbed by) port *1x*.

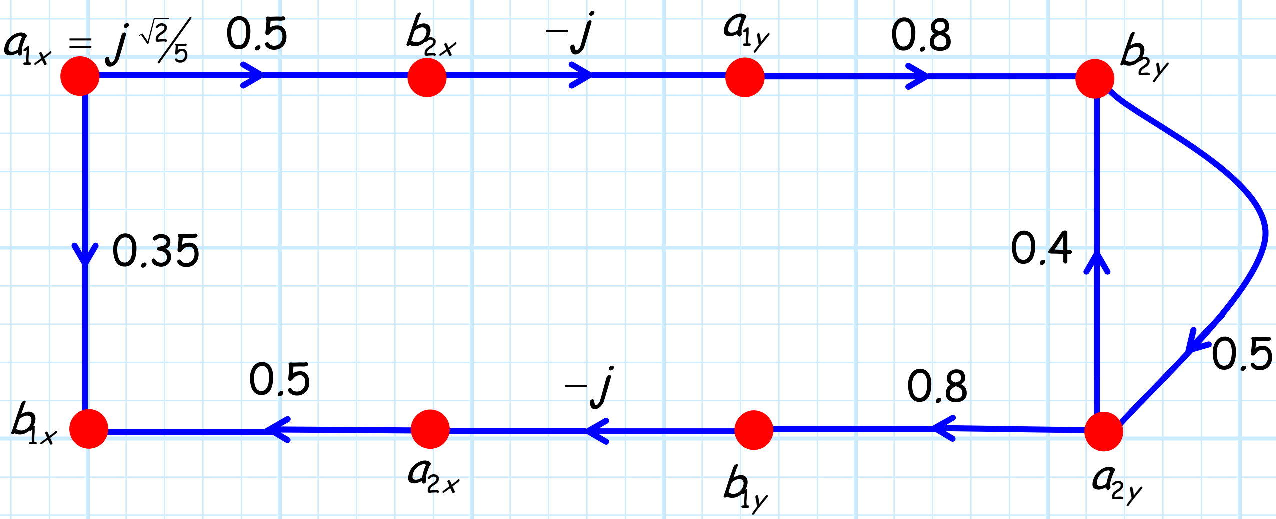
The signal flow graph describing this network is:



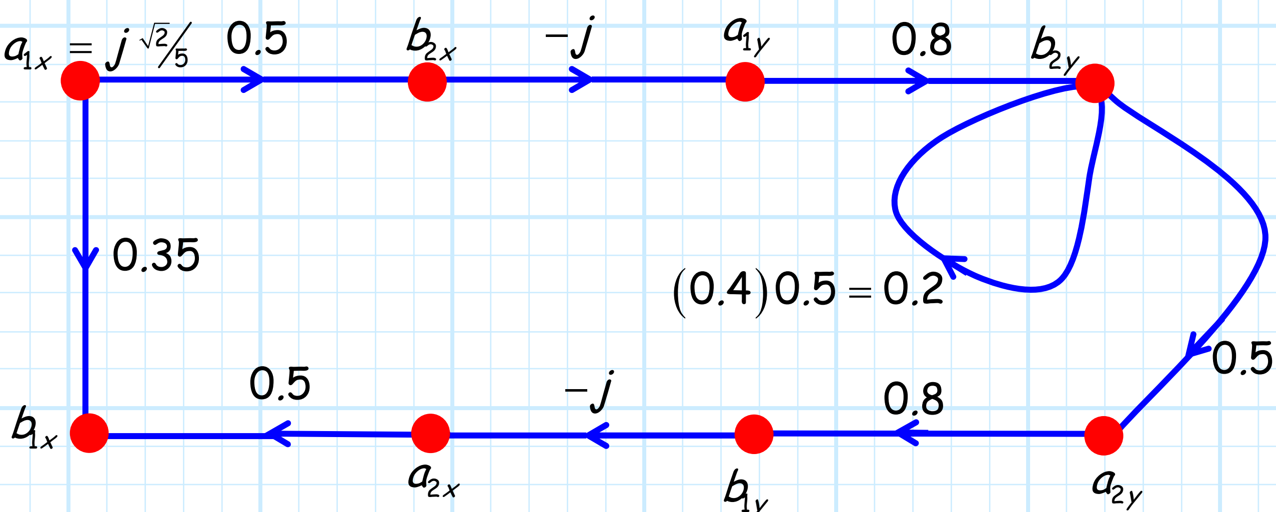
Inserting the numeric values of branches:



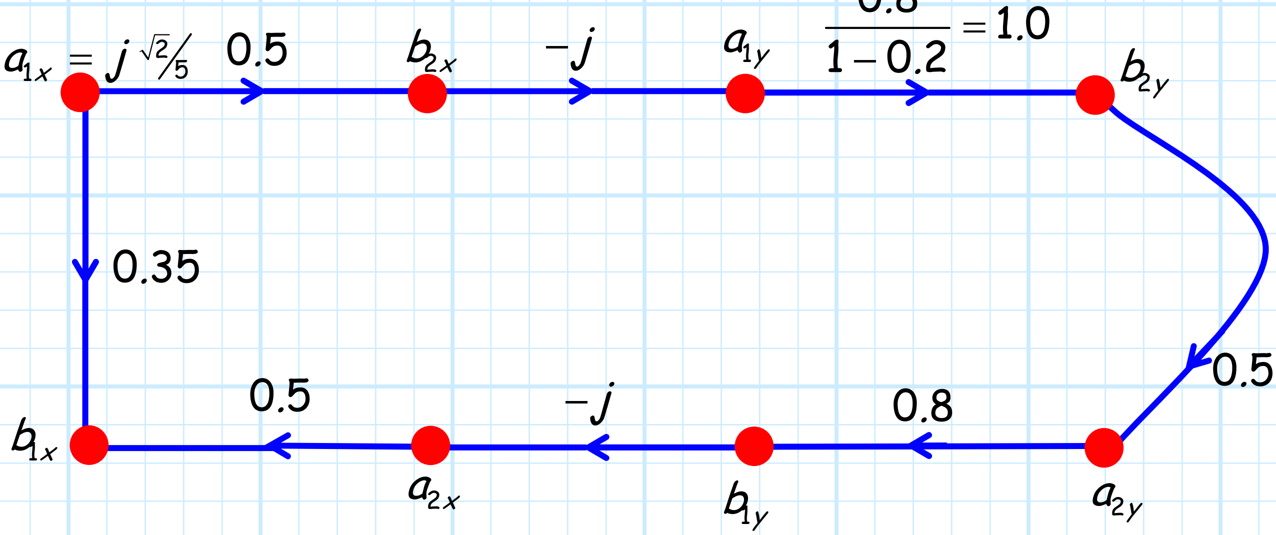
Removing the zero valued branches:



And now applying "splitting" rule 4:



Followed by the "self-loop" rule 3:



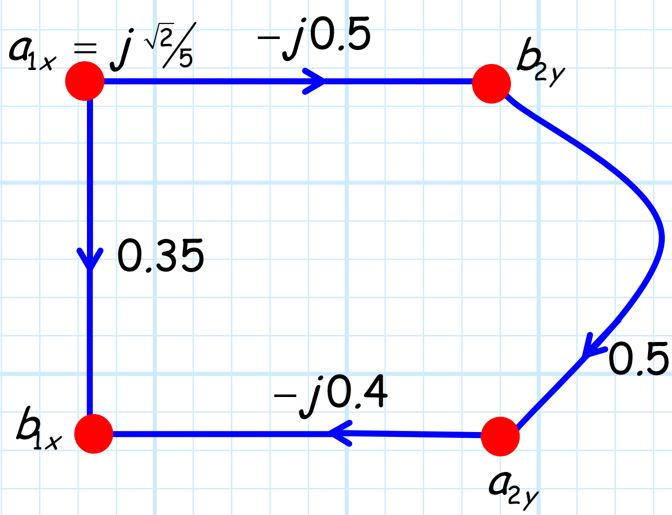
Now, let's used this simplified signal flow graph to find the solutions to our questions!

a) The total current through load Γ_L .

The total current through the load is:

$$
\begin{aligned}
I_L &= -I\left(z_{2y} = z_{2yP}\right) \\
&= -\frac{V_{02y}^{+}\left(z_{2y} = z_{2yP}\right) - V_{02y}^{-}\left(z_{2y} = z_{2yP}\right)}{Z_0} \\
&= -\frac{a_{2y} - b_{2y}}{\sqrt{Z_0}} \\
&= \frac{b_{2y} - a_{2y}}{\sqrt{50}}
\end{aligned}
$$

Thus, we need to determine the value of nodes a_{2y} and b_{2y}. Using the "series" rule 1 on our signal flow graph:



Note we've simply **ignored** (i.e., neglected to plot) the node for which we have **no interest**!

From this graph we can conclude:

$$b_{2y} = -j0.5\, a_{1x} = -j0.5\left(\frac{j\sqrt{2}}{5}\right) = 0.1\sqrt{2}$$

and:

$$a_{2y} = 0.5\, b_{2y} = 0.5\left(0.1\sqrt{2}\right) = 0.05\sqrt{2}$$

Therefore:

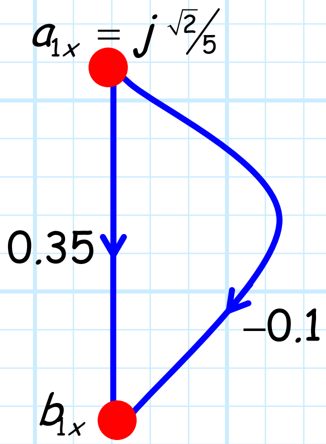
$$I_L = \frac{b_{2y} - a_{2y}}{\sqrt{50}} = \frac{(0.1 - 0.05)\sqrt{2}}{\sqrt{50}} = \frac{0.05}{5} = 10.0\ mA$$

b) The **power** delivered to (i.e., absorbed by) port *1x*.

The power delivered to port 1x is:

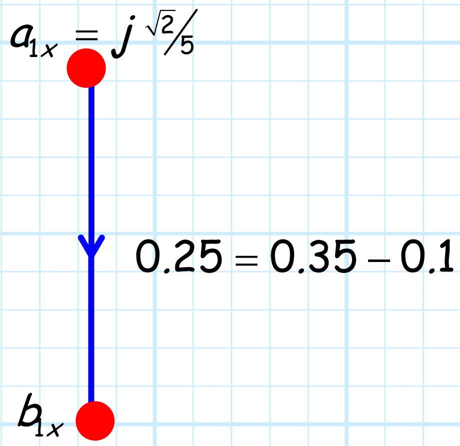
$$\begin{aligned} P_{abs} &= P^+ - P^- \\ &= \frac{\left|V_{1x}^+\left(z_{1x} = z_{1xP}\right)\right|^2}{2Z_0} - \frac{\left|V_{1x}^-\left(z_{1x} = z_{1xP}\right)\right|^2}{2Z_0} \\ &= \frac{\left|a_{1x}\right|^2 - \left|b_{1x}\right|^2}{2} \end{aligned}$$

Thus, we need determine the values of nodes a_{1x} and b_{1x}. Again using the series rule 1 on our signal flow graph:



Again we've simply **ignored** (i.e., neglected to plot) the node for which we have **no interest**!

And then using the "parallel" rule 2:



Therefore:

$$b_{1x} = 0.25\, a_{1x} = 0.25\left(j\,\sqrt{2}/5\right) = j0.05\sqrt{2}$$

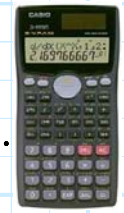
and:

$$P_{abs} = \frac{\left|j\,\sqrt{2}/5\right|^2 - \left|j0.05\sqrt{2}\right|^2}{2} = \frac{0.08 - 0.005}{2} = 37.5\ mW$$

The Propagation Series

Q: *You earlier stated that signal flow graphs are helpful in (count em') **three** ways. I now understand the **first** way:*

Way 1 - Signal flow graphs provide us with a **graphical** means of **solving** large systems of simultaneous equations.



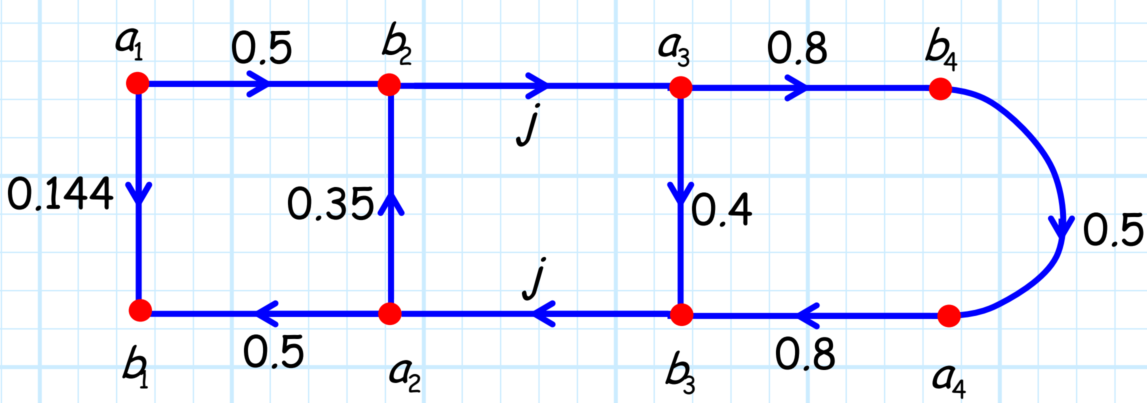
But what about ways 2 and 3 ??

"**Way 2** – We'll see the a signal flow graph can provide us with a **road map** of the wave **propagation paths** throughout a microwave device or network."

"**Way 3** - Signal flow graphs provide us with a quick and accurate method for **approximating** a network or device."

bunny, 64 spheres

A: Consider the *sfg* below:

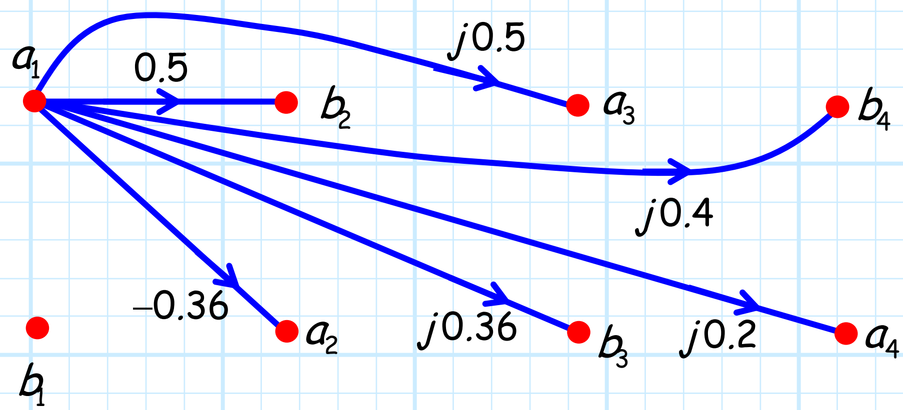


Note that node a_1 is the only **independent** node. This signal flow graph is for a rather complex **single-port** (port 1) device.

Say we wish to determine the wave amplitude **exiting port 1**. In other words, we seek:

$$b_1 = \Gamma_{in}\, a_1$$

Using our four **reduction rules**, the signal flow graph above is simplified to:



Q: *Hey, node b_1 is **not connected** to anything. What does this mean?*

A: It means that $b_1 = 0$—**regardless** of the value of incident wave a_1. I.E.,:

$$\Gamma_{in} = \frac{b_1}{a_1} = 0$$

In other words, port 1 is a **matched load!**

Q: *But look at the **original** signal flow graph; it doesn't **look** like a matched load. How can the exiting wave at port 1 be **zero?***

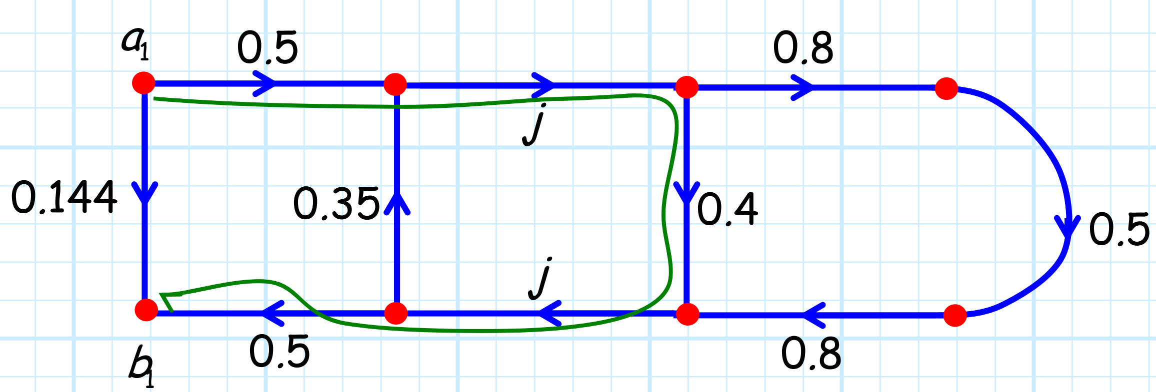
A: A signal flow graph provides a bit of a **propagation road map** through the device or network. It allows us to understand—often in a **very** physical way—the **propagation** of an incident wave once it enters a device.

We accomplish this by identifying from the *sfg* **propagation paths** from an independent node to some other node (e.g., an exiting node). These paths are simply a **sequence of branches** (pointing in the correct direction!) that lead from the independent node to this other node.

Each path has **value** that is equal to the **product** of each branch of the path.

Perhaps this is best explained with some **examples**.

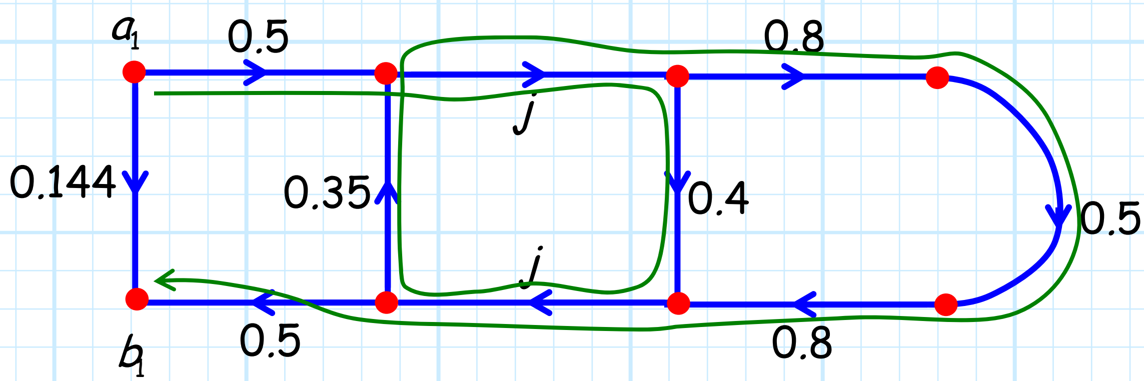
One **path** between independent (incident wave) node a_1 and (exiting wave) node b_1 is shown below:



We'll **arbitrarily** call this path 2, and its value:

$$p_2 = (0.5)\,j\,(0.4)\,j\,(0.5) = -0.1$$

Another propagation **path** (path 5, say) is:



$$p_5 = (0.5)\,j\,(0.4)\,j\,(0.35)\,j\,(0.8)(0.5)(0.8)\,j\,(0.5)$$
$$= j^4\,(0.35)(0.4)(0.8)^2\,(0.5)^3$$
$$= 0.0112$$

Q: ***Why*** *are we doing this?*

A: The exiting wave at port 1 (wave amplitude b_1) is simply the **superposition** of **all** the propagation paths from incident node a_1! Mathematically speaking:

$$b_1 = a_1 \sum_n p_n \qquad \Rightarrow \qquad \Gamma_{in} = \frac{b_1}{a_1} = \sum_n p_n$$

Q: *Won't there be an awful* ***lot*** *of propagation paths?*

A: Yes! As a matter of fact there are an **infinite** number of paths that connect node a_1 and b_1. Therefore:

$$b_1 = a_1 \sum_n^{\infty} p_n \qquad \Rightarrow \qquad \Gamma_{in} = \frac{b_1}{a_1} = \sum_n^{\infty} p_n$$

Q: *Yikes! Does this infinite series **converge**?*

A: Note that the series represents a finite physical value (e.g., Γ_{in}), so that the infinite series **must** converge to the correct **finite** value.

Q: *In this example we found that $\Gamma_{in} = 0$. This means that the infinite propagation series is likewise **zero**:*

$$\Gamma_{in} = \sum_{n}^{\infty} p_n = 0$$

*Do we conclude from this that **all** propagation paths are **zero**:*

$$p_n = 0 \quad ?????$$

A: Absolutely **not**! Remember, we have already determined that $p_2 = -0.1$ and $p_4 = 0.0112$—definitely **not** zero-valued! In fact for this example, **none** of the propagation paths p_n are precisely equal to zero!

Q: *But then **why** is:*

$$\sum_{n}^{\infty} p_n = 0 \;\; ???$$

A: Remember, the path values p_n are **complex**. A **sum** of **non-zero** complex values can equal **zero** (as it apparently does in this case!).

Thus, a **perfectly rational** way of viewing this network is to conclude that there are an **infinite number of non-zero waves exiting port 1**:

$$\Gamma_{in} = \sum_{n}^{\infty} p_n \qquad \text{where } p_n \neq 0$$

It just so happens that these waves **coherently add** together to **zero**:

$$\Gamma_{in} = \sum_{n}^{\infty} p_n = 0$$

—they essentially **cancel each other out**!

Q: *So, I now appreciate the fact that signal flow graphs: 1) provides a **graphical method** for solving linear equations and 2) also provides a method for **physically evaluating** the wave **propagation paths** through a network/device.*

*But what about helpful **Way 3**:*

"**Way 3** - Signal flow graphs provide us with a quick and accurate method for **approximating** a network or device." ??

bunny, 64 spheres

A: The propagation series of a microwave network is very analogous to a **Taylor Series** expansion:

$$f(x) = \sum_{n=0}^{\infty} \left.\frac{d^n f(x)}{dx^n}\right|_{x=a} (x-a)^n$$

Note that there likewise is a **infinite** number of terms, yet the Taylor Series is quite helpful in engineering.

Often, we engineers simply **truncate** this infinite series, making it a finite one:

$$f(x) \approx \sum_{n=0}^{N} \left. \frac{d^n f(x)}{d x^n} \right|_{x=a} (x-a)^n$$

Q: *Yikes! Doesn't this result in **error**?*

A: Absolutely! The truncated series is an **approximation**.

We have **less** error if **more** terms are retained; more **error** if fewer **terms** are retained.

The trick is to retain the **"significant"** terms of the infinite series, and **truncate** those less important "insignificant" terms. In this way, we seek to form an **accurate** approximation, using the **fewest** number of terms.

Q: *But how do we know **which** terms are significant, and which are **not**?*

A: For a Taylor Series, we find that as the **order** ***n*** increases, the significance of the term generally (but **not** always!) decreases.

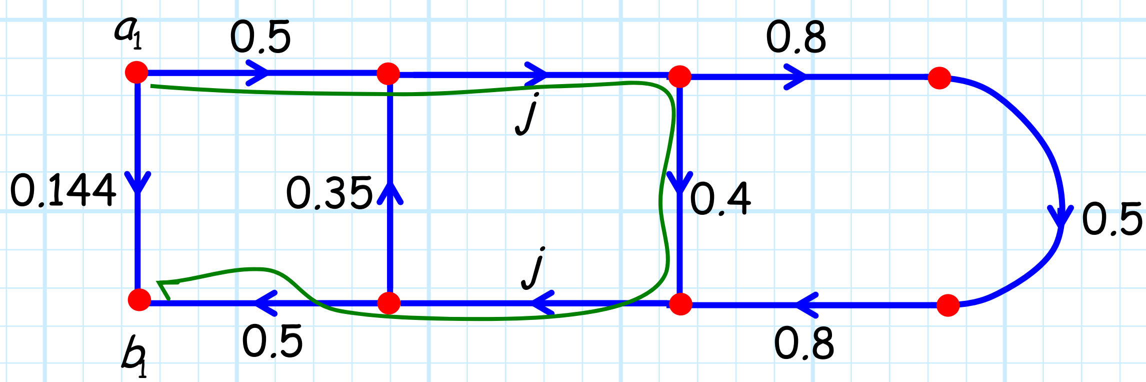
Q: *But what about our **propagation series**? How can we determine which paths are **"significant"** in the series?*

A: Almost always, the most significant paths in a propagation series are the **forward paths** of a signal flow graph.

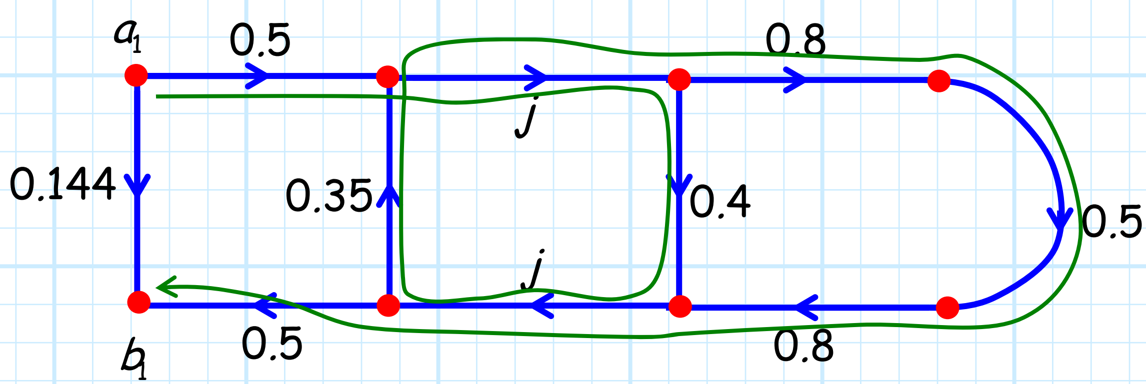
forward path - \'fȯr-wərd' päth\ *-noun*

A path through a signal flow graph that passes through any given node no more than once. A path that passes through any node two times (or more) is therefore *not* a forward path.

In our example, **path 2** is a forward path. It passes through **four** nodes as it travels from node a_1 to node b_1, but it passes through each of these nodes **only once**:



Alternatively, path 5 is **not** a forward path:



We see that path 5 passes through six different nodes as it travels from node a_1 to node b_1. However, it **twice passes** through four of these nodes.

The good news about forward paths is that there are **always** a **finite** number of them. Again, these paths are typically the **most significant** in the propagation series, so we can determine an approximate value for *sfg* nodes by considering **only** these forward paths in the propagation series:

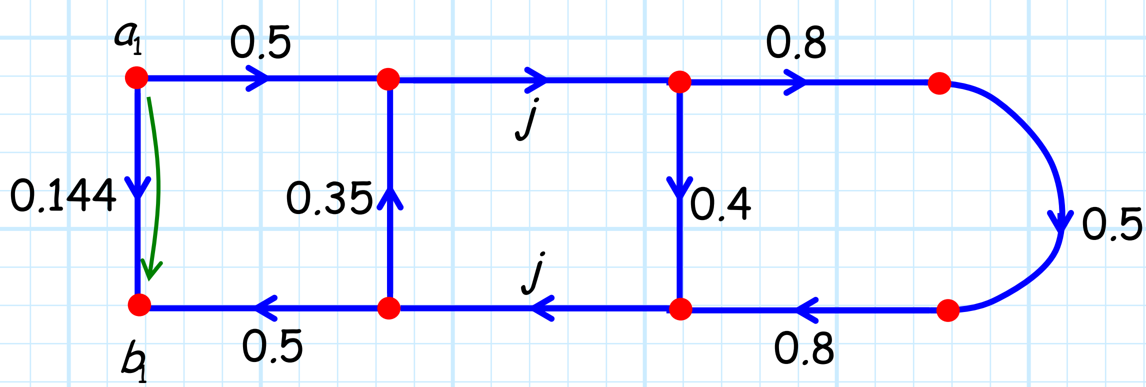
$$\sum_{n}^{\infty} p_n \approx \sum_{n=1}^{N} p_n^{fp}$$

where p_n^{fp} represents the value of one of the N forward paths.

Q: *Is path 2 the **only** forward path in our example* sfg *?*

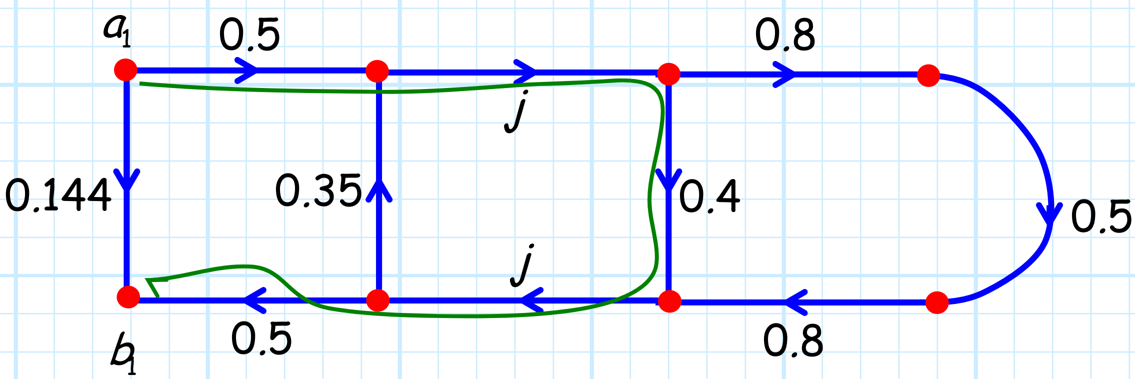
A: No, there are **three**. Path 1 is the most **direct**:

$$p_1 = 0.144$$



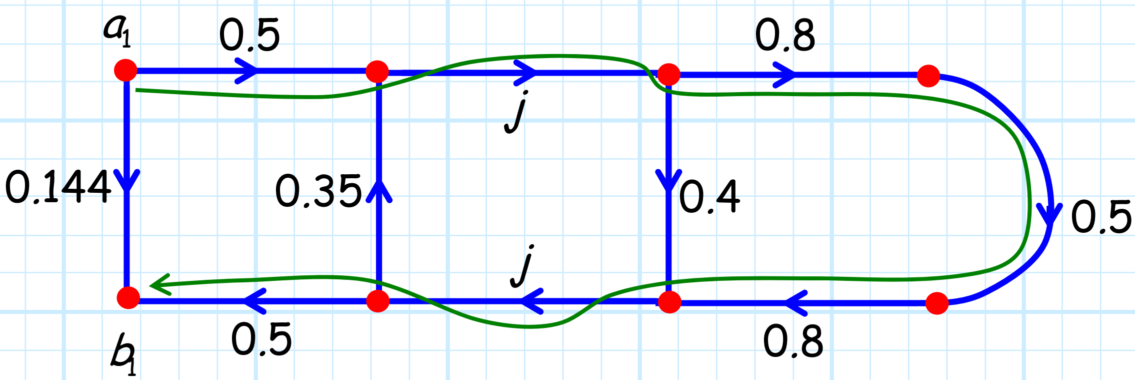
Of course we already have identified path 2:

$$p_2 = -0.1$$



And finally, path 3 is the **longest** forward path:

$$\begin{aligned} p_3 &= (0.5)\,j\,(0.8)(0.5)(0.8)\,j\,(0.5) \\ &= j^2\,(0.8)^2\,(0.5)^3 \\ &= -0.08 \end{aligned}$$



Thus, an **approximate** value of Γ_{in} is:

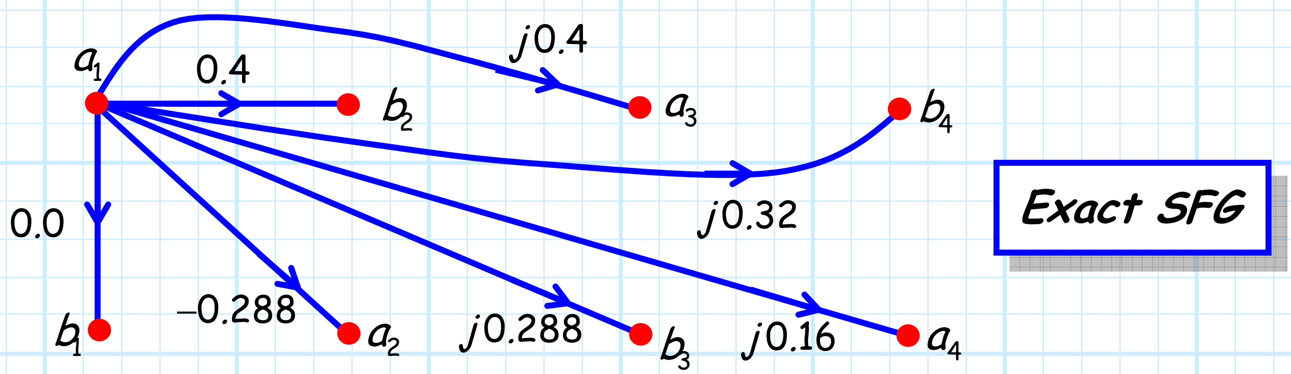
$$\begin{aligned} \Gamma_{in} &= \frac{b_1}{a_1} \\ &\approx \sum_{n=1}^{3} p_n^{fp} \\ &= p_1 + p_2 + p_3 \\ &= 0.144 - 0.1 - 0.08 \\ &= -0.036 \end{aligned}$$

Q: *Hey wait! We determined earlier that $\Gamma_{in} = 0$, but now your saying that $\Gamma_{in} = -0.036$. **Which is correct??***

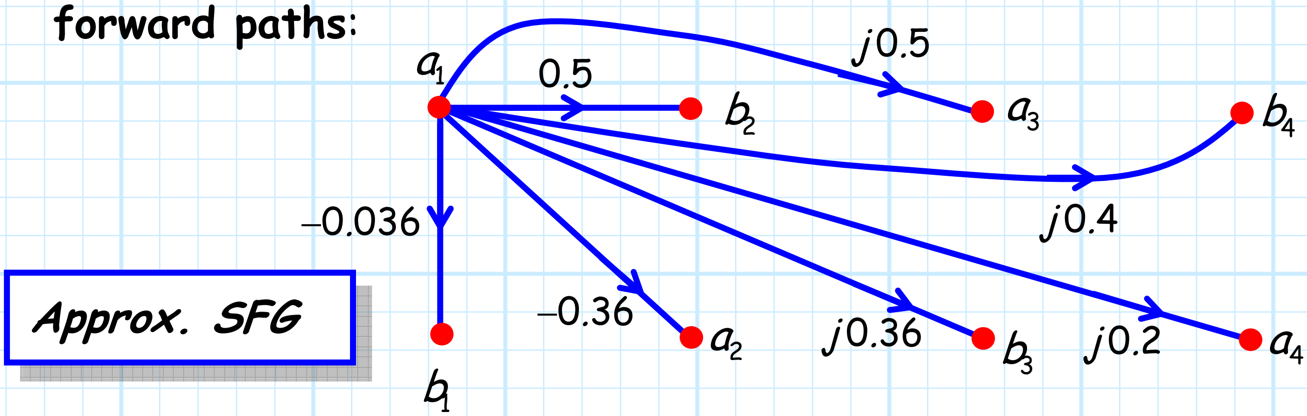
A: The correct answer is $\Gamma_{in} = 0$. It was determined using the four *sfg* reduction rules—**no approximations** were involved!

Conversely, the value $\Gamma_{in} = -0.036$ was determined using a **truncated** form of the propagation series—the series was limited to just the **three** most **significant** terms (i.e., the forward paths). The result is **easier** to obtain, but it is just an approximation (the answers will contain **error**!).

For example, consider the **reduced** signal flow graph (**no** approximation error):



Compare this to the same sfg, computed using only the **forward paths**:



No surprise, the **approximate** *sfg* (using forward paths only) is **not** the same as the **exact** *sfg* (using reduction rules).

The approximate *sfg* contains **error**, but note this error is not **too** bad. The values of the approximate *sfg* are certainly **close** to that of the exact *sfg*.

Q: *Is there any way to **improve** the accuracy of this approximation?*

A: Certainly. The error is a result of truncating the infinite propagation series. Note we **severely** truncated the series—out of an **infinite** number of terms, we retained **only three** (the forward paths). If we retain **more terms**, we will likely get a **more accurate** answer.

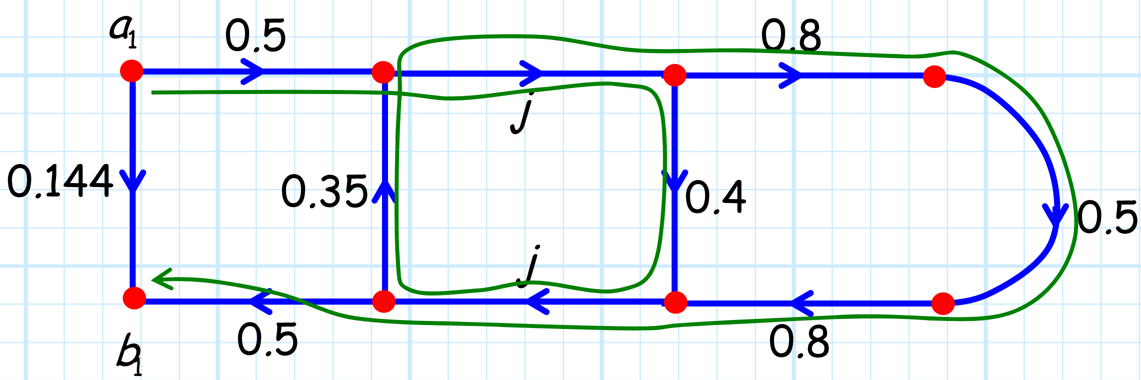
Q: *So why did these approximate answers turn out so **well**, given that we **only** used three terms?*

A: We retained the **three most significant** terms, we will find that the **forward paths** typically have the **largest magnitudes** of all propagation paths.

Q: *Any idea what the **next** most significant terms are?*

A: Yup. The **forward paths** are all those propagation paths that pass through any node no more than **one** time. The next most significant paths are almost certainly those paths that pass through any node no more than **two** times.

Path 4 is an **example** of such a path:



There are **three more** of these paths (passing through a node no more than two times)—see if **you** can find them!

After determining the values for these paths, we can add **4 more terms** to our summation (now we have **seven** terms!):

$$\begin{aligned}\Gamma_{in} &= \frac{b_1}{a_1} \\ &\approx \sum_{n=1}^{7} p_n \\ &= \left(p_1 + p_2 + p_3\right) + \left(p_4 + p_5 + p_6 + p_7\right) \\ &= (-0.036) + (0.014 + 0.0112 + 0.0112 + 0.0090) \\ &= 0.0094\end{aligned}$$

Note this value is **closer** to the correct value of **zero** than was our previous (using only **three** terms) answer of **-0.036**.

As we **add** more terms to the summation, this approximate answer will get **closer** and closer to the correct value of **zero.** However, it will be **exactly** zero (to an **infinite** number of decimal points) **only** if we sum an **infinite** number of terms!

Q: *The **significance** of a given path seem to be inversely proportional to the **number of times** it passes through any node. Is this true? If so, then **why** is it true?*

A: It **is** true (generally speaking)! A propagation path that travels though a node **ten** times is much **less** likely to be significant to the propagation series (i.e., summation) than a path that passes through any node no more than (say) **four** times.

The reason for this is that the significance of a given term in a summation is dependent on its **magnitude** (i.e., $|p_n|$). If the magnitude of a term is **small**, it will have far **less affect** (i.e., significance) on the sum than will a term whose magnitude is large.

Q: *You seem to be saying that paths traveling through **fewer** nodes have larger **magnitudes** than those traveling through **many** nodes. Is that true? If so why?*

A: Keep in mind that a microwave *sfg* relates wave **amplitudes**. The branch values are therefore always **scattering parameters**. One important thing about scattering parameters, their magnitudes (for **passive** devices) are always **less than** or equal to one!

$$|S_{mn}| \leq 1$$

Recall the value of a path is simply the **product** of each branch that forms the path. The more branches (and thus nodes), the more terms in this product.

Since each term has a magnitude **less than one**, the magnitude of a product of **many** terms is **much smaller** than a product of a few terms. For example:

$$\left|-j0.7\right|^{3} = 0.343 \qquad \text{and} \qquad \left|-j0.7\right|^{10} = 0.028$$

→ In other words, paths with **more branches** (i.e., more nodes) will typically have **smaller magnitudes** and so are **less significant** in the propagation series.

Note path 1 in our example traveled along **one** branch only:

$$p_1 = 0.144$$

Path 2 has **five** branches:

$$p_2 = -0.1$$

Path 3 **seven** branches:

$$p_3 = -0.08$$

Path 4 **nine** branches:

$$p_4 = 0.014$$

Path 5 **eleven** branches:

$$p_5 = 0.0112$$

Path 6 **eleven** branches:

$$p_6 = 0.0112$$

Path 7 **thirteen** branches:

$$p_7 = 0.009$$

Hopefully it is **evident** that the magnitude **diminishes** as the path "length" **increases**.

Q: *So, does this mean that we should* ***abandon*** *our four* ***reduction rules****, and* ***instead*** *use a truncated propagation series to* ***evaluate*** *signal flow graphs??*

A: **Absolutely not**! 

Remember, truncating the propagation series always results in some **error**. This error might be sufficiently small **if** we retain enough terms, but knowing precisely **how many** terms to retain is problematic.

We find that in most cases it is simply **not worth the effort**—use the four **reduction rules** instead (it's **not** like they're particularly difficult!).

Q: *You say that in "**most cases**" it is not worth the effort. Are there some cases where this approximation is actually **useful**??*

A: Yes. A truncated propagation series (typically using only the **forward paths**) is used when these **three** things are true:

1. The network or device is **complex** (lots of nodes and branches).

2. We can conclude from our knowledge of the device that the **forward paths** are sufficient for an **accurate** approximation (i.e., the magnitudes of all other paths in the series are almost certainly **very** small).

3. The branch values are **not numeric**, but instead are variables that are dependent on the **physical** parameters of the device (e.g., a characteristic impedance or line length).

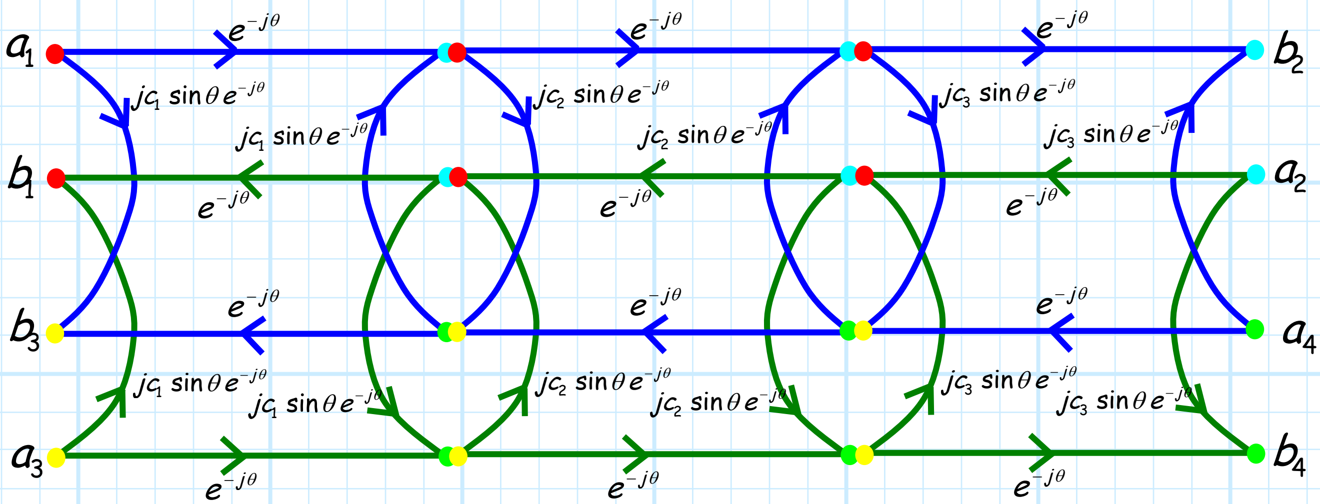
The result is typically a **tractable** mathematical equation that relates the **design variables** (e.g., Z_0 or ℓ) of a complex device to a specific **device parameter**.

For **example**, we might use a truncated propagation series to **approximately** determine some function:

$$\Gamma_{in}\left(Z_{01},\ell_1,Z_{02},\ell_2\right)$$

If we desire a matched input (i.e., $\Gamma_{in}(Z_{01},\ell_1,Z_{02},\ell_2)=0$) we can **solve** this tractable design equation for the (nearly) proper values of $Z_{01},\ell_1,Z_{02},\ell_2$.

We will use this technique to **great effect** for designing **multi-section matching networks** and **multi-section coupled line couplers**.



The signal flow graph of a three-section coupled-line coupler.