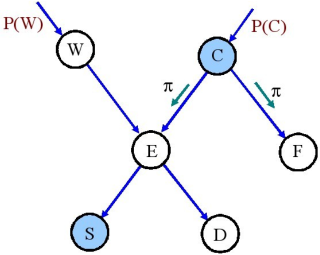
Lecture 4

Probability Propagation in Singly Connected Networks

Probability Propagation

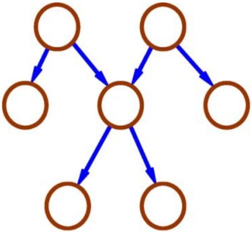
We will now develop a general probability propagation method based on just five operating equations which can be applied independently at any node to process incoming evidence and dispatch messages to neighbours.



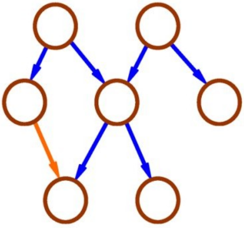
We will first need to introduce the idea of multiple parents. To keep the notation simple we will restrict the treatment to nodes with at most two parents.

Singly Connected Networks

The equations are also limited to singly connected networks. which have at most one path between any two nodes.



Singly Connected Network



Multiply Connected Network

Multiple Parents

- Up until now our networks have been trees, but in general they need not be. In particular, we need to cope with the possibility of multiple parents.

- Multiple parents can be thought of as representing different possible causes of an outcome.

- In our cat example, the eyes in a picture could be caused by other image features:
  - Pictures showing other animals (owls dogs etc),
  - Pictures that have features sharing the same geometric model (bicycles).

How to tell an owl from a cat

How to tell an owl from a cat

How to tell an owl from a cat

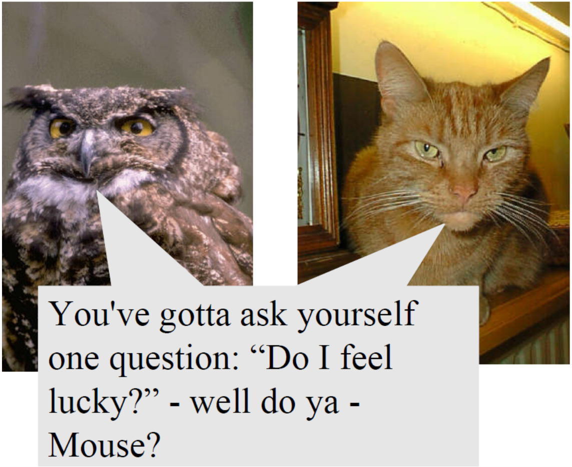
Owl

How to tell an owl from a cat

Owl

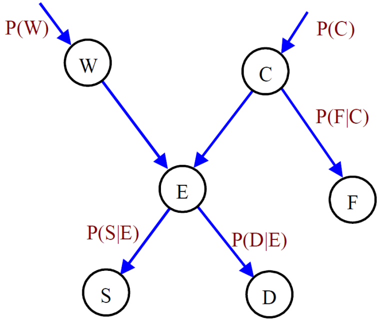
Cat

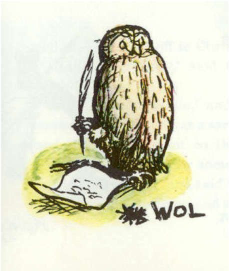
How to tell an owl from a cat



The Owl and the Pussycat

The two causes for the E variable (owl and cat) form multiple parents.





Where W is the “owl” variable.

Conditional Probabilities with Multiple Parents

Conditional probabilities, with multiple parents must include all the joint states of the parents. Thus for the E node we have:

$$P(E|C\&W)$$

Notice that the conditional probability table is associated with the child node and not with the arcs of the networks.

Example Link Matrix

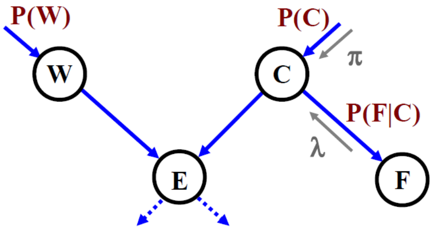
Given that node W has states w_1 and w_2, and similarly C, the conditional probability matrix takes the form:

$$\boldsymbol{P}(\boldsymbol{E}|\boldsymbol{W}\&\boldsymbol{C}) = \begin{bmatrix} P(e_1|w_1\&c_1) & P(e_1|w_1\&c_2) & P(e_1|w_2\&c_1) & P(e_1|w_2\&c_2) \\ P(e_2|w_1\&c_1) & P(e_2|w_1\&c_2) & P(e_2|w_2\&c_1) & P(e_2|w_2\&c_2) \\ P(e_3|w_1\&c_1) & P(e_3|w_1\&c_2) & P(e_3|w_2\&c_1) & P(e_3|w_2\&c_2) \end{bmatrix}$$

π messages with multiple parents

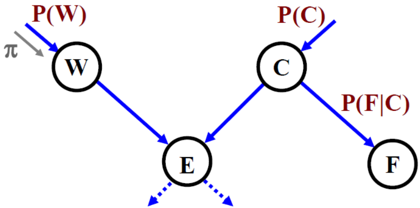
If we wish to calculate the π evidence for eyes, we first calculate the evidence for C, taking into account only the evidence that does not come from E. This is called the π message from C to E.

In this case it includes the prior probability of C and the λ evidence from F.



λ and π messages with multiple parents

Next we calculate the π message from W to E. In this example there is only the prior evidence for W.



Notation for π messages

We write:

- $\pi_E(C)$ for the π message from C to E. It means all the evidence for C excluding the evidence from E.
- $\pi_E(W)$ for the π message from W to E. It means all the evidence for W excluding the evidence from E.

We can also define the π messages using the posterior probability of a node, for example $P'(C)$, as follows:

$$P'(C) = \alpha P(C)\lambda_E(C)\lambda_F(C)$$
$$\pi_E(C) = P'(C)/\lambda_E(C)$$

Remember that it is not necessary to normalise evidence.

Finding a joint distribution over the parents

Having calculated a π message from each parent, we calculate a joint distribution of π messages over all parents. To do this we assume that $\pi_E(C)$ and $\pi_E(W)$ are independent:

$$\pi_E(W\&C) = \pi_E(W) \times \pi_E(C)$$

Remember that this is a scalar equation with variables C and W, for individual states:

$$\pi_E(c_i\&w_j) = \pi_E(c_i) \times \pi_E(w_j)$$

In vector form the joint evidence is:

$$\begin{aligned}\boldsymbol{\pi_E(W\&C)} &= [\pi_E(w_1\&c_1), \pi_E(w_1\&c_2), \pi_E(w_2\&c_1), \pi_E(w_2\&c_2)] \\ &= [\pi_E(w_1)\pi_E(c_1), \pi_E(w_1)\pi_E(c_2), \pi_E(w_2)\pi_E(c_1), \pi_E(w_2)\pi_E(c_2)]\end{aligned}$$

The independence of $\pi_E(C)$ and $\pi_E(W)$

- In this simple example C and W have no other path linking them, and hence if we do not consider the evidence from E then $\pi_E(C)$ and $\pi_E(W)$ must be independent.
- If they had, for example, a common parent, then our assumption about the independence of $\pi_E(C)$ and $\pi_E(W)$ would no longer hold.
- We have therefore made an implicit assumption that there are no loops in our network.

Calculating the π evidence for E

We can now compute the π evidence for E using the link matrix:

$$[\pi(e_1)\ \pi(e_2)\ \pi(e_3)] =$$

$$\begin{bmatrix} P(e_1|w_1\&c_1) & P(e_1|w_1\&c_2) & P(e_1|w_2\&c_1) & P(e_1|w_2\&c_2) \\ P(e_2|w_1\&c_1) & P(e_2|w_1\&c_2) & P(e_2|w_2\&c_1) & P(e_2|w_2\&c_2) \\ P(e_3|w_1\&c_1) & P(e_3|w_1\&c_2) & P(e_3|w_2\&c_1) & P(e_3|w_2\&c_2) \end{bmatrix} \begin{bmatrix} \pi_E(w_1\&c_1) \\ \pi_E(w_1\&c_2) \\ \pi_E(w_2\&c_1) \\ \pi_E(w_2\&c_2) \end{bmatrix}$$

or in vector form:

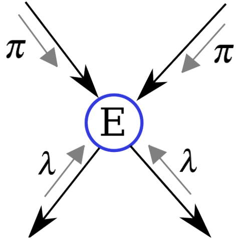
$$\boldsymbol{\pi}(\boldsymbol{E}) = \boldsymbol{P}(\boldsymbol{E}|\boldsymbol{W}\&\boldsymbol{C})\boldsymbol{\pi}_{\boldsymbol{E}}(\boldsymbol{W}\&\boldsymbol{C})$$

Posterior probability of E

We now can finally compute a probability distribution over the states of E. As before all we do is multiply the λ and π evidence together and normalise.

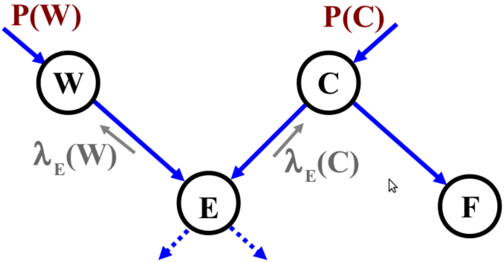
$$P'(e_i) = \alpha\lambda(e_i)\pi(e_i)$$

α is the normalising constant.



λ messages with multiple parents

A node with multiple parents will send an individual λ messages to each.



We must compute each one from the joint conditional probability table $P(E|C\&W)$

Revision on λ messages

The λ message from E to C (before introducing the node W) was written as:

$$\lambda_E(c_1) = \lambda(e_1)P(e_1|c_1) + \lambda(e_2)P(e_2|c_1) + \lambda(e_3)P(e_3|c_1)$$
$$\lambda_E(c_2) = \lambda(e_1)P(e_1|c_2) + \lambda(e_2)P(e_2|c_2) + \lambda(e_3)P(e_3|c_1)$$

or more generally:

$$\lambda_E(c_i) = \sum_j P(e_j|c_i)\lambda(e_j)$$

or in vector form:

$$\boldsymbol{\lambda_E(C)} = \boldsymbol{\lambda(E)}P(E|C)$$

λ messages with joint link matrices

We can calculate a joint λ message using the vector formulation:

$$\boldsymbol{\lambda_E}(\boldsymbol{W}\&\boldsymbol{C}) = \boldsymbol{\lambda}(\boldsymbol{E})P(E|W\&C)$$

but this gives us a joint message to the parents:

$$\boldsymbol{\lambda_E}(\boldsymbol{W}\&\boldsymbol{C}) = [\lambda(w_1\&c_1), \lambda(w_1\&c_2), \lambda(w_2\&c_1), \lambda(w_2\&c_2)]$$

We use the evidence for W to separate out the evidence for C from the joint λ message:

$$\lambda_E(c_i) = \sum_j \pi_E(w_j)\lambda(w_j\&c_i)$$

Scalar equation for the λ message

For any specific $\pi_E(W)$ we can estimate the link matrix $P(E|C)$, from $P(E|W\&C)$. Each element is computed using the scalar equation:

$$P(e_i|c_j) = P(e_i|c_j\&w_1)\pi_E(w_1) + P(e_i|c_j\&w_2)\pi_E(w_2)$$

or more generally:

$$P(e_i|c_j) = \sum_k P(e_i|c_j\&w_k)\pi_E(w_k)$$

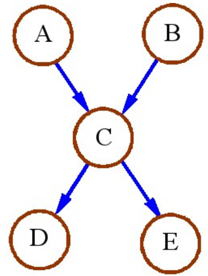
where k ranges over the states of W.

The λ message can be written directly as a single scalar equation:

$$\lambda_E(c_j) = \sum_k \pi_E(w_k) \sum_i P(e_i|c_j\&w_k)\lambda(e_i)$$

Probability Propagation

Each node in a Bayesian Network has six different data items being:



1. The posterior probability
2. The link matrix
3. The λ evidence
4. The π evidence
5. The λ messages
6. The π messages

Operating Equation 1: the λ message

The λ message from C to A in scalar form is given by:

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k|a_i\&b_j)\lambda(c_k)$$

For the case of a single parent this simplifies to:

$$\lambda_C(a_i) = \sum_{k=1}^{n} P(c_k|a_i)\lambda(c_k)$$

Operating Equation 1: the λ message

For the case of the single parent A we have a very simple matrix form:

$$\lambda_{\mathbf{C}}(\mathbf{A}) = \lambda(\mathbf{C})\mathbf{P}(\mathbf{C}|\mathbf{A})$$

The matrix form for multiple parents relates to the joint states of the parents.

$$\lambda_{\mathbf{C}}(\mathbf{A}\&\mathbf{B}) = \lambda(\mathbf{C})\mathbf{P}(\mathbf{C}|\mathbf{A}\&\mathbf{B})$$

It is necessary to separate the λ evidence for the individual parents with a scalar equation of the form:

$$\lambda_C(a_i) = \Sigma_j \pi_C(b_j)\lambda_C(a_i\&b_j)$$

Operating Equation 2: the π message

If C is a child of A, the π message from A to C is given by:

$$\pi_C(a_i) = \begin{cases} 1 & \text{if } A \text{ is instantiated for } a_i \\ 0 & \text{if } A \text{ is instantiated but not for } a_i \\ P'(a_i)/\lambda_C(a_i) & \text{if } A \text{ is not instantiated} \end{cases}$$

The π message to a child contains all the evidence for the parent except that from the child.

Operating Equation 3: the λ evidence

If C is a node with n children $D_1, D_2, ..D_n$, then the λ evidence for C is:

$$\lambda(c_k) = \begin{cases} 1 & \text{if } C \text{ is instantiated for } c_k \\ 0 & \text{if } C \text{ is instantiated but not for } c_k \\ \prod_i \lambda_{D_i}(c_k) & \text{if } C \text{ is not instantiated} \end{cases}$$

Operating Equation 4: the π evidence

If C is a child of two parents A and B the π evidence for C is given by:

$$\pi(c_k) = \sum_{i=1}^{l} \sum_{j=1}^{m} P(c_k|a_i\&b_j)\pi_C(a_i)\pi_C(b_j)$$

This can be written in matrix form as follows:

$$\pi(\mathbf{C}) = \mathbf{P}(\mathbf{C}|\mathbf{A}\&\mathbf{B})\pi_{\mathbf{C}}(\mathbf{A}\&\mathbf{B})$$

where

$$\pi_C(a_i\&b_j) = \pi_C(a_i)\pi_C(b_j)$$

The single parent matrix equation is:

$$\pi(\mathbf{C}) = \mathbf{P}(\mathbf{C}|\mathbf{A})\pi_{\mathbf{C}}(\mathbf{A})$$

Operating Equation 5: the posterior probability

If C is a variable the (posterior) probability of C based on the evidence received is written as:

$$P'(c_k) = \alpha\lambda(c_k)\pi(c_k)$$

where α is chosen to make $\sum_k P'(c_k) = 1$

Belief Propagation

Probability propagation is a form of belief propagation and is achieved by message passing.

- New evidence enters a network when a variable is instantiated, ie when it receives a new value from the input.
- When this happens the posterior probabilities of each node in the whole network may need to be re-calculated.

Belief Propagation

- When the π or λ evidence for a node changes it must inform some of its parents and its children as follows:
- For each parent to be updated:
  - Update the parent's λ message array.
  - Set a flag to indicate that the parent must be re-calculated
- For each child to be updated
  - Update the π message held for each child.
  - Set a flag to indicate that the child must be re-calculated

Termination

- The network reaches a steady state when there are no more nodes to be re-calculated.
- This condition will always be reached in singly connected networks. This is because λ messages will terminate at root nodes and π messages will terminate at leaf nodes.
- Multiply connected networks will not necessarily reach a steady state. This is referred to as loopy belief propagation

We will discuss termination conditons later in the lecture.

Initialisation

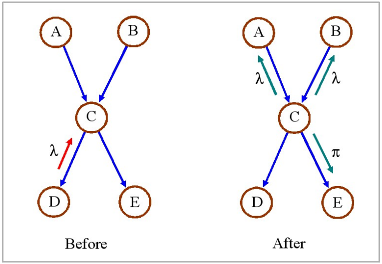
When a network is initialised no nodes have been instantiated and the only evidence comes from the prior probabilities of the roots:

1. All λ message and evidence values are set to 1
2. All π messages are set to 1
3. For all root nodes the π evidence values are set to the prior probabilities, eg, for all states of $R : \pi(r_i) = P(r_i)$
4. Post and propagate the π messages from the root nodes down to the leaf nodes (see downward propagation).

Upward Propagation

if node C receives λ message from a child
if C is not instantiated

1. Compute the new $\lambda(C)$ value (Op. Eq. 3)
2. Compute the new posterior probability $P'(C)$ (Eq. 5)
3. Post a λ message to all C's parents
4. Post a π message to C's other children



Downward Propagation

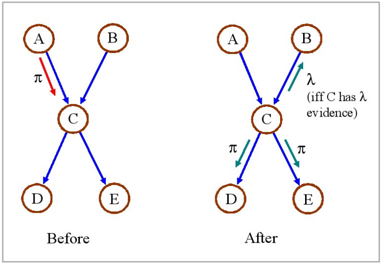
if a variable C receives a π message from one parent:

if C is not instantiated

1. Compute a new value for evidence $\pi(C)$ (Eq.4)
2. Compute a new value for $P'(C)$ (Eq. 5)
3. Post a π message to each child

if (there is λ evidence for C)

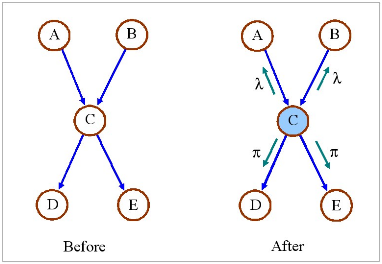
1. Post a λ message to the other parents



Instantiation

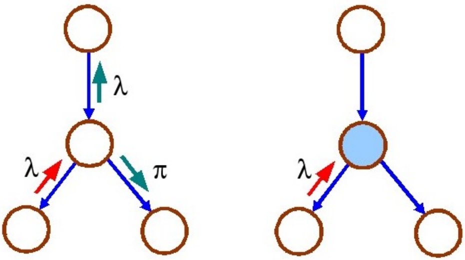
if variable C is instantiated in state c_i

1. Compute $\lambda(C)$ (Eq. 3)
2. Compute $P'(C)$ (=$\lambda(C)$) (Eq. 5)
3. Post a π message (=$\lambda(C)$) to each child of C (Eq. 2)
4. Post a λ message to each parent of C



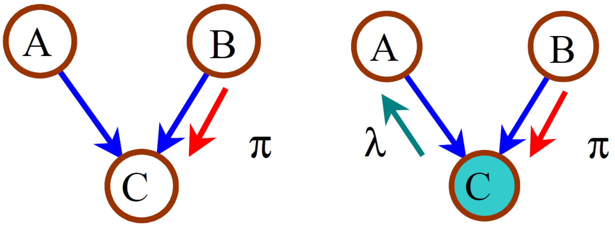
Blocked Paths

If a node is instantiated it will block some message passing. Instantiating the centre node below prevents any messages passing through it.



Converging Paths

However, converging paths are blocked when there is no λ evidence for the child node, but unblocked if the child has λ evidence or is instantiated.



Converging Paths

To understand the behaviour of converging paths we look at operating equaion 1.

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k|a_i\&b_j)\lambda(c_k)$$

Given that node C has no λ evidence we can write $\lambda(C) = \{1, 1, 1, .\ 1\}$ and substituting this into operating equation 1 we get:

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k|a_i\&b_j)$$

Converging Paths

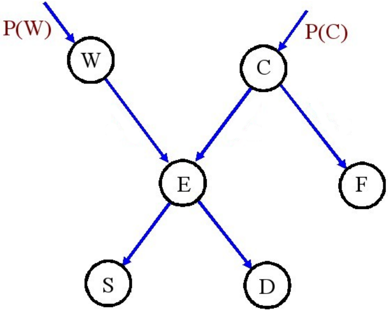
$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k|a_i\&b_j)$$

The second summation now evalutes to 1 for any value of i and j, since we are summing a probability distribution. Thus:

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j)$$

The sum is independent of i and hence the value of $\lambda_c(a_i)$ is the same for all values of i, that is to say there is no λ evidence sent to A.

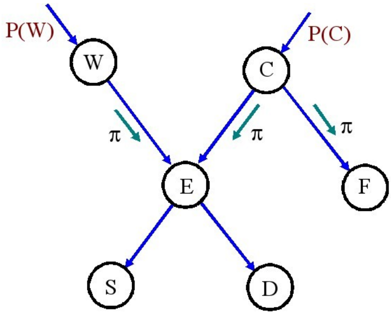
Example: the Owl and the Pussy Cat



Initialisation:

The only evidence in the network is the prior probabilities of the root nodes.

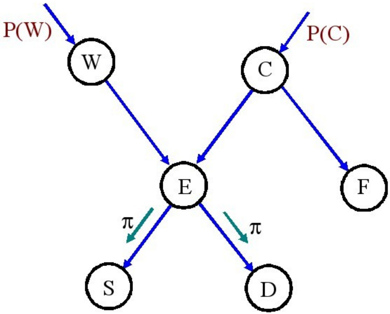
Example: the Owl and the Pussy Cat



Initialisation:

The priors become π messages and propagate downwards.

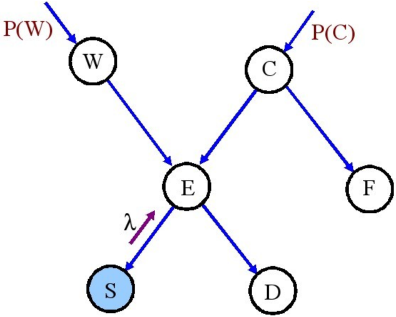
Example: the Owl and the Pussy Cat



Initialisation:

E has no λ evidence so it just sends a π message to its children and the network reaches a steady state.

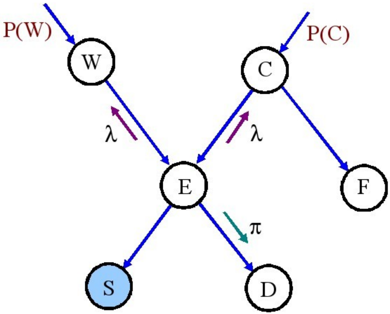
Example: the Owl and the Pussy Cat



Instantiation of S:

A new measurement is made and we instantiate S. A λ message propagates upwards.

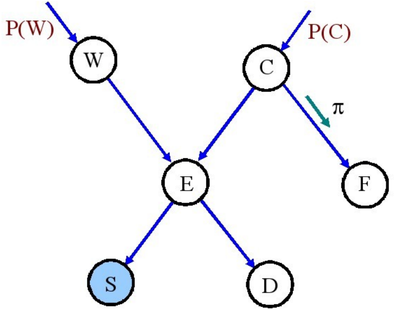
Example: the Owl and the Pussy Cat



Instantiation of S:

E calculates its evidence and sends messages everywhere but to S.

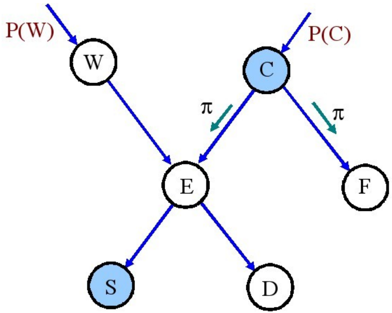
Example: the Owl and the Pussy Cat



Instantiation of S:

W and C recalculate their evidence. W has no messages to send but C will send a π message to F. The network reaches a steady state.

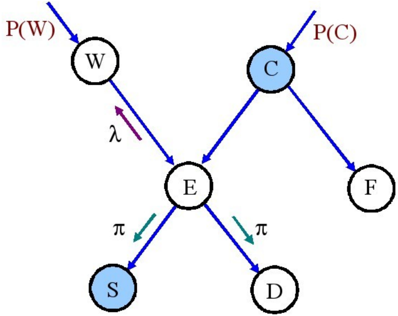
Example: the Owl and the Pussy Cat



Instantiation of C:

C is now instantiated and sends π messages to its children.

Example: the Owl and the Pussy Cat



Instantiation of C:

There is λ evidence for E so it will send a λ message to W. The π message sent to S has no effect. It is blocked since S is already instantiated. The network reaches a steady state.