Neural Networks
and
Computation Graphs

This lecture

- What is a neural network?

- Computation Graphs

- Algorithms over computation graphs
  - The forward pass
  - The backward pass

Where are we?

- What is a neural network?

- Computation Graphs

- Algorithms over computation graphs
  - The forward pass
  - The backward pass

Three computational questions

1. Forward propagation
    –   Given inputs to the graph, compute the value of the function expressed by the graph
    –   Something to think about: Given a node, can we say which nodes are inputs? Which nodes are outputs?

2. Backpropagation
    –   After computing the function value for an input, compute the gradient of the function at that input
    –   Or equivalently: *How does the output change if I make a small change to the input?*

3. Constructing graphs
    –   Need an easy-to-use framework to construct graphs
    –   The size of the graph may be input dependent
        •   A templating language that creates graphs on the fly
    –   Tensorflow, PyTorch are the most popular frameworks today

Forward propagation

Three computational questions

1. **Forward propagation**
    –   Given inputs to the graph, compute the value of the function expressed by the graph
   –   Something to think about: Given a node, can we say which nodes are inputs? Which nodes are outputs?

2. Backpropagation
    –   After computing the function value for an input, compute the gradient of the function at that input
    –   Or equivalently: *How does the output change if I make a small change to the input?*

3. Constructing graphs
    –   Need an easy-to-use framework to construct graphs
    –   The size of the graph may be input dependent
        •   A templating language that creates graphs on the fly
    –   Tensorflow, PyTorch are the most popular frameworks today
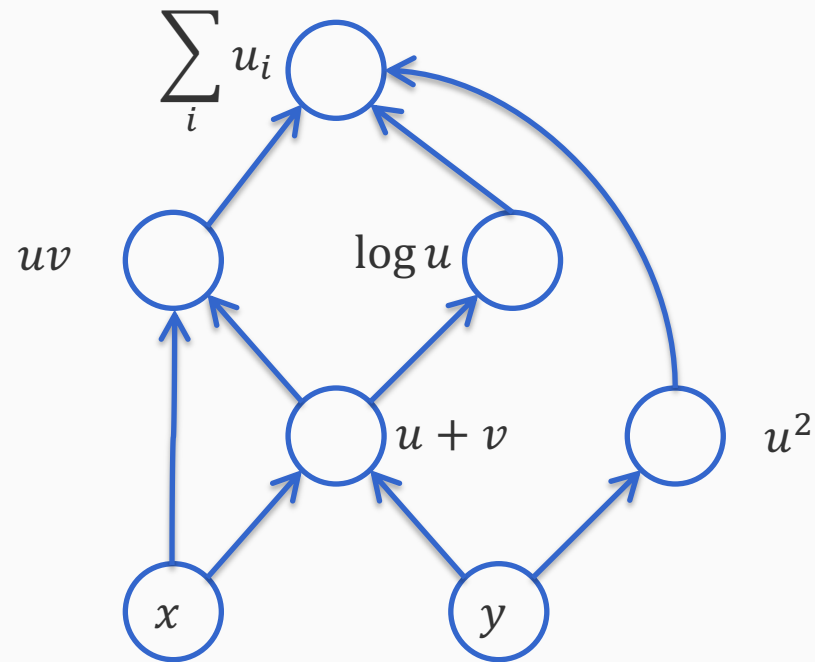
Forward pass: An example
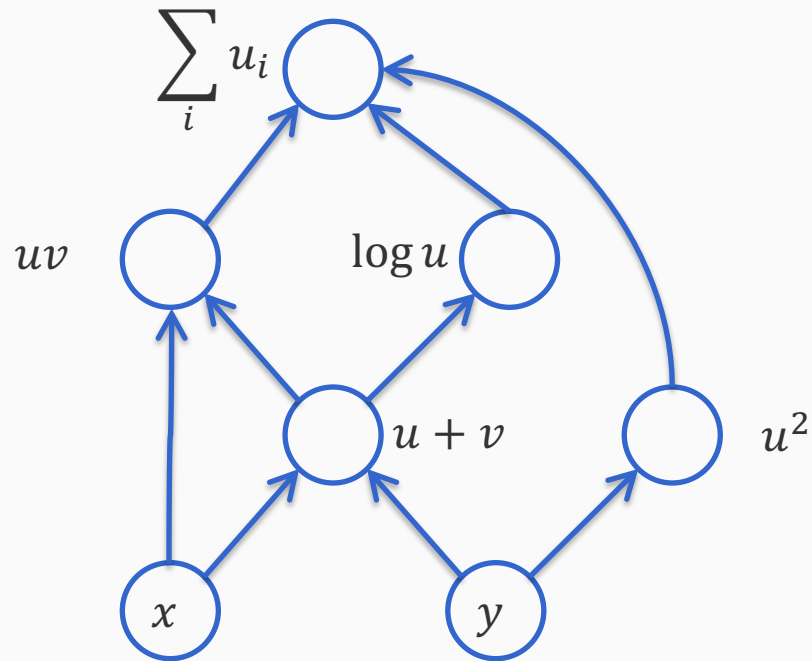
Conventions:

1. Any expression next to a node is the function it computes
2. All the variables in the expression are inputs to the node from left to right.
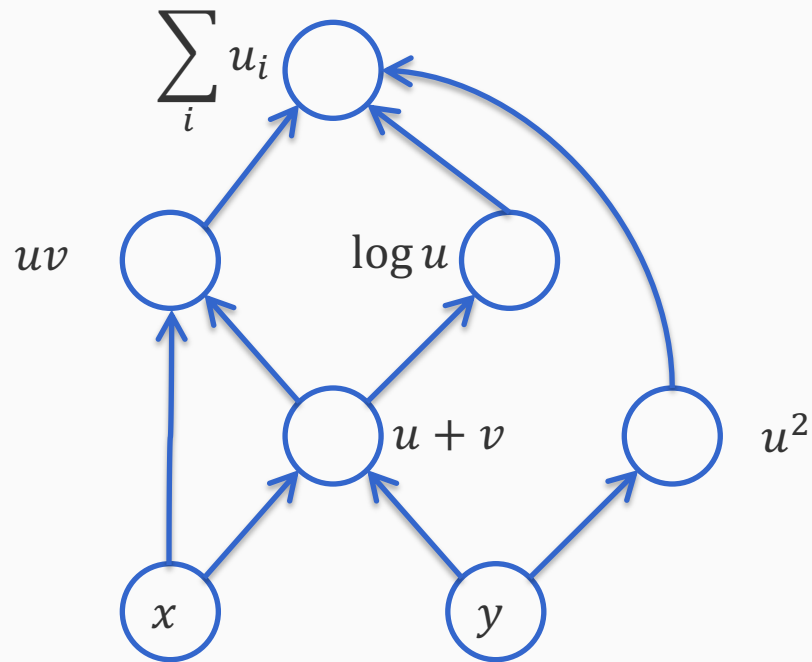
Forward pass

What function does this compute?
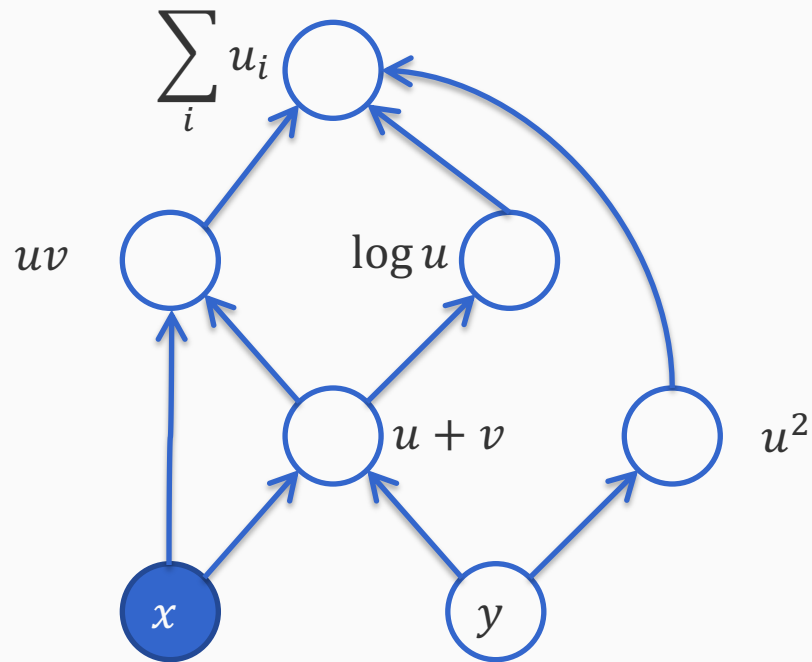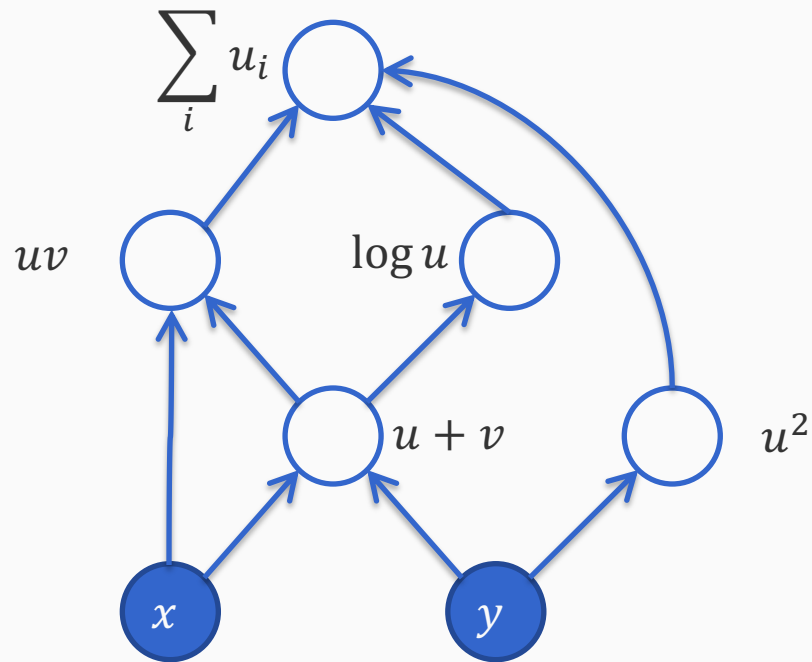
Forward pass

What function does this compute?

Suppose we shade nodes whose values we know (i.e. we have computed).

Forward pass

What function does this compute?

x

Suppose we shade nodes whose values we know (i.e. we have computed).

Forward pass

What function does this compute?
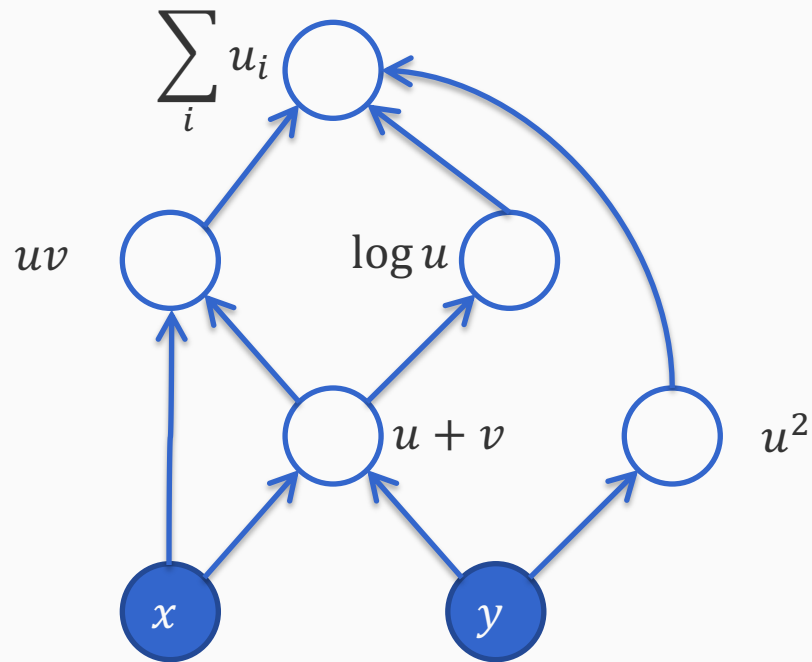
$$x$$

$$y$$

Suppose we shade nodes whose values we know (i.e. we have computed).

Forward pass

What function does this compute?

x

y

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs

Forward pass

What function does this compute?

$$x$$

$$y$$

$$x + y$$

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs

Forward pass

What function does this compute?

x

y

$x + y$

y^2

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs

Forward pass

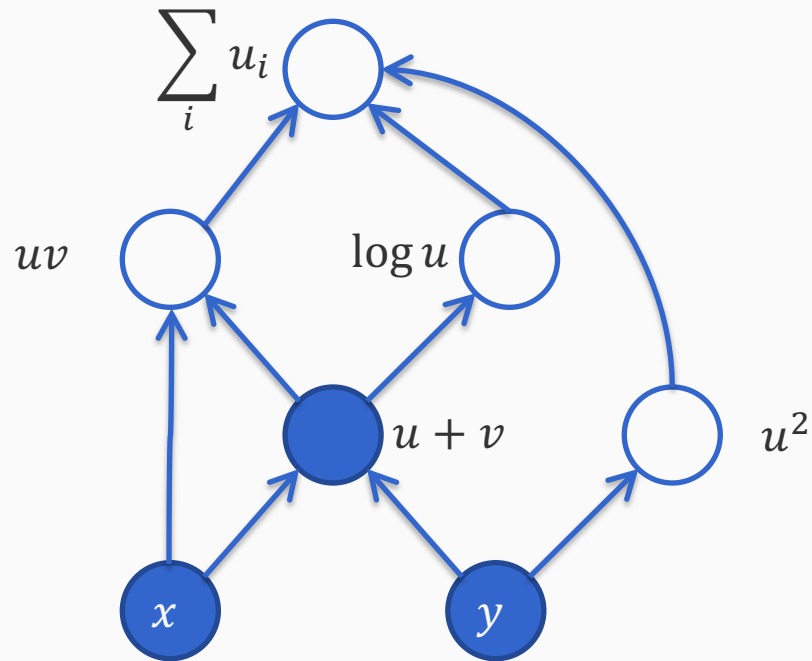What function does this compute?

x

y

$x + y$

y^2

$x(x + y)$

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs

14

Forward pass

What function does this compute?

$$x$$

$$y$$

$$x + y$$

$$y^2$$

$$x(x + y)$$

$$\log(x + y)$$

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs
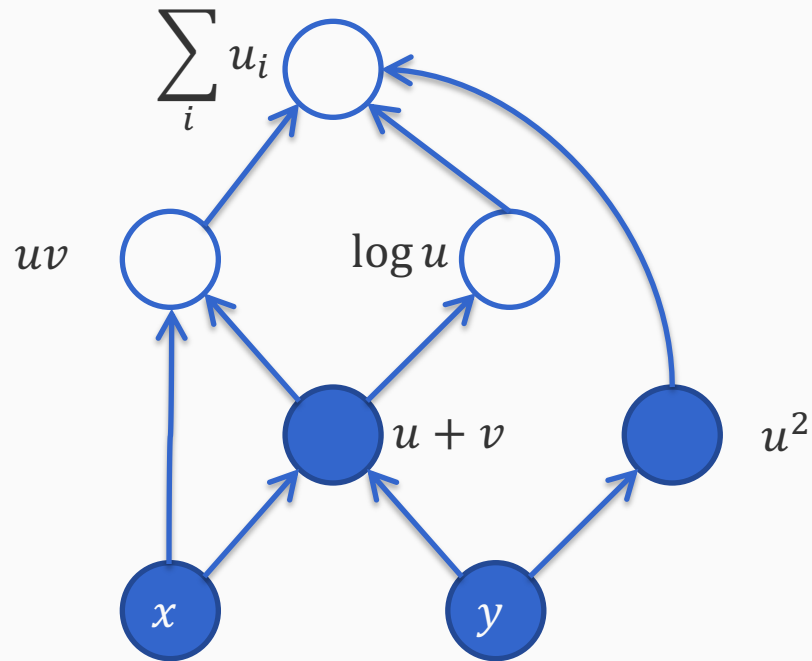
Forward pass

What function does this compute?

x

y

$x + y$

y^2

$x(x + y)$

$\log(x + y)$

$x(x + y) + \log(x + y) + y^2$

Suppose we shade nodes whose values we know (i.e. we have computed).

We can only compute the value of a node if we know the values of all its inputs

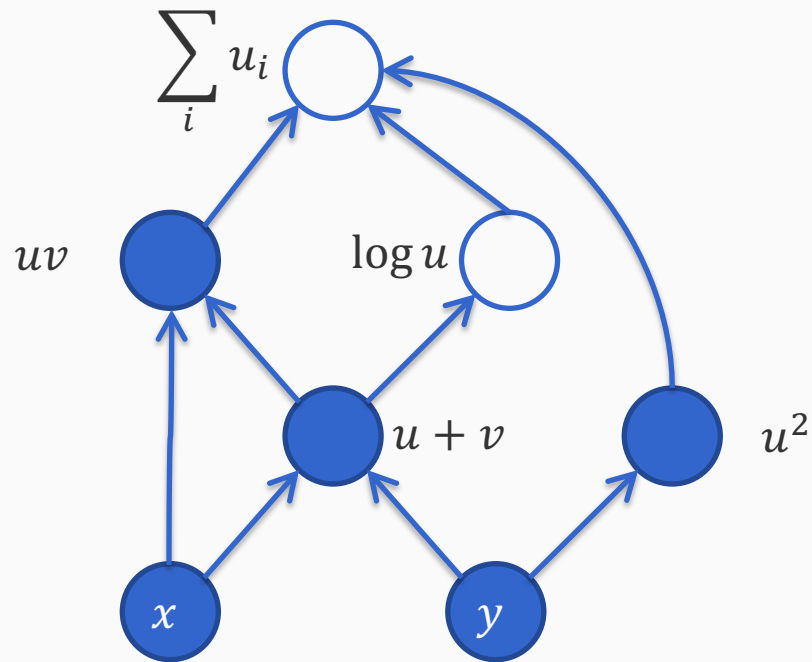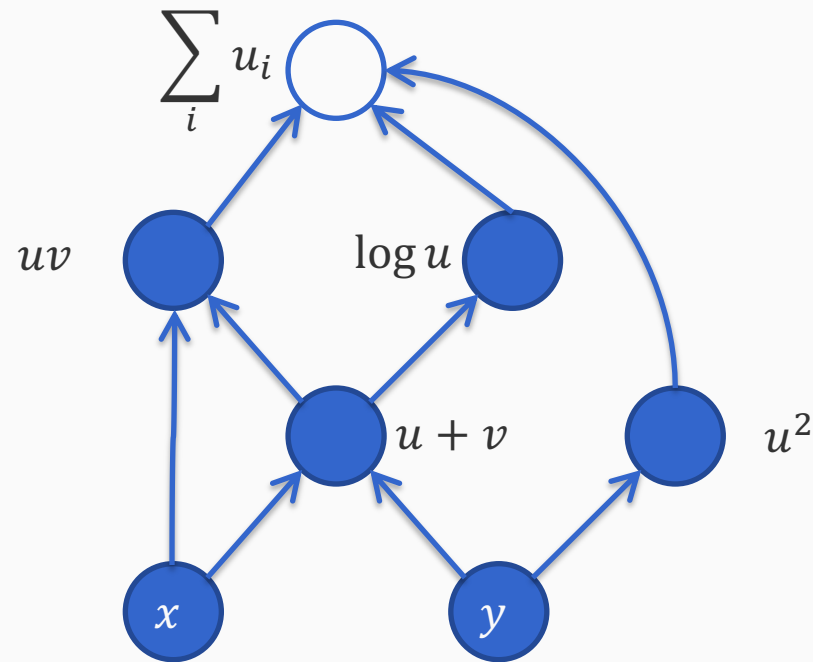# Forward pass

What function does this compute?

$$x$$

$$y$$
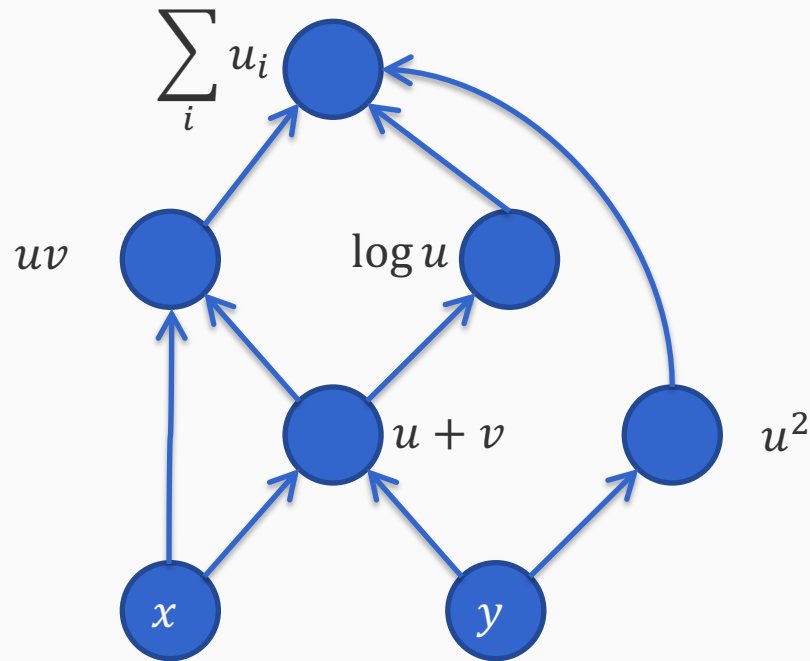
$$x + y$$

$$y^2$$

$$x(x + y)$$

$$\log(x + y)$$

$$x(x + y) + \log(x + y) + y^2$$

This gives us the function

Suppose we shade nodes whose values we know (i.e. we have computed).

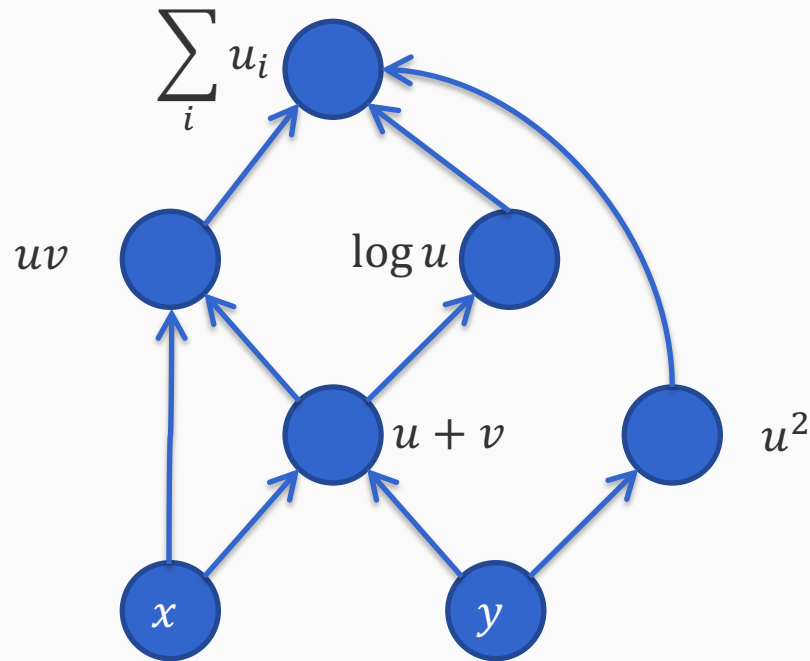We can only compute the value of a node if we know the values of all its inputs

A second example
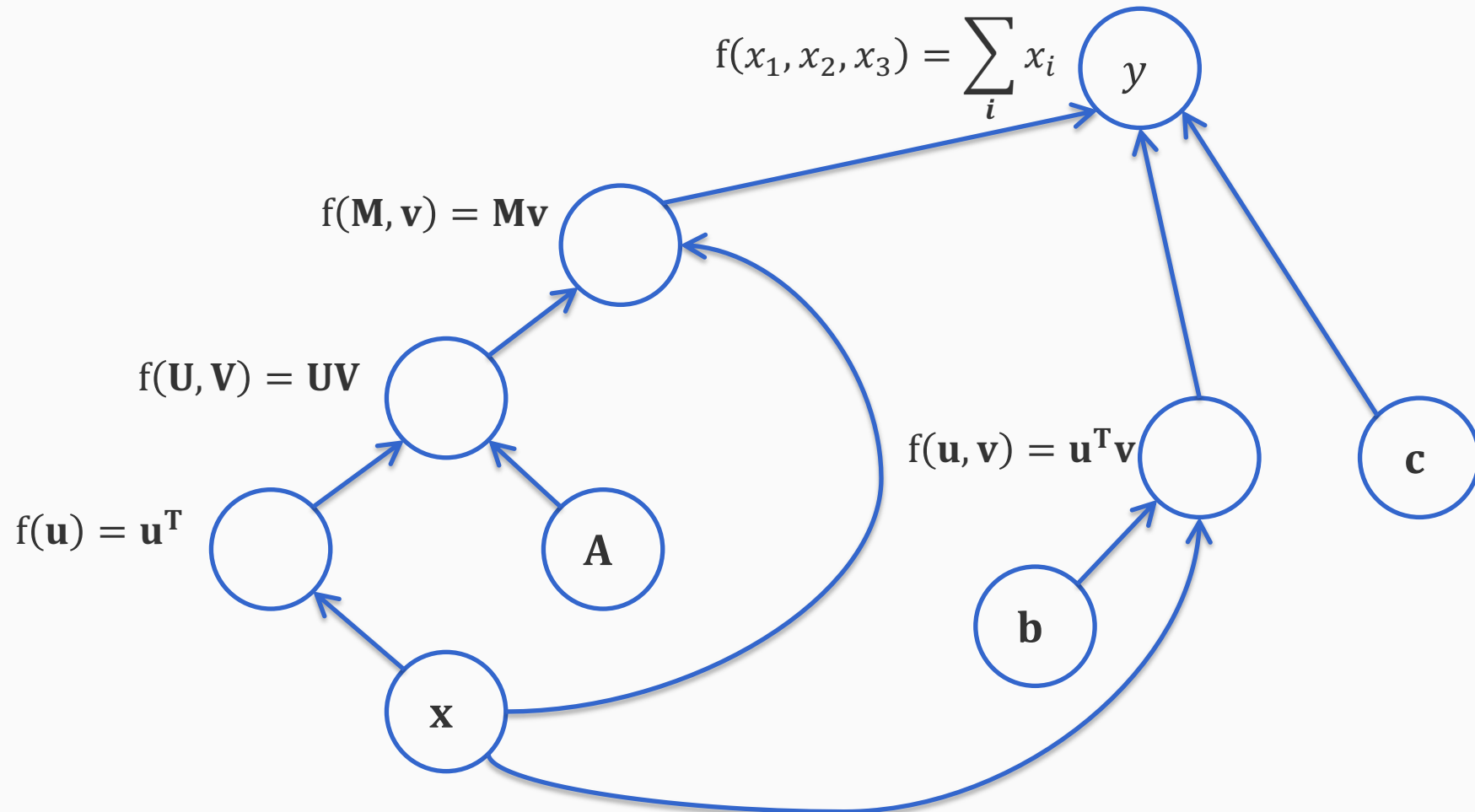
$$f(x_1, x_2, x_3) = \sum_i x_i$$

y

$f(\mathbf{M}, \mathbf{v}) = \mathbf{M}\mathbf{v}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{U}\mathbf{V}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathbf{T}}\mathbf{v}$

$f(\mathbf{u}) = \mathbf{u}^{\mathbf{T}}$

$\mathbf{A}$

$\mathbf{c}$

$\mathbf{b}$

$\mathbf{x}$

A second example

To compute the function, we need the values of the leaves of this DAG

$$f(x_1, x_2, x_3) = \sum_i x_i$$

$f(\mathbf{M}, \mathbf{v}) = \mathbf{Mv}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{UV}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u^T v}$

$f(\mathbf{u}) = \mathbf{u^T}$

y

$\mathbf{c}$

$\mathbf{A}$

$\mathbf{b}$

$\mathbf{x}$

A second example

To compute the function, we need the values of the leaves of this DAG

$$f(x_1, x_2, x_3) = \sum_i x_i$$

$f(\mathbf{M}, \mathbf{v}) = \mathbf{M}\mathbf{v}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{U}\mathbf{V}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathbf{T}}\mathbf{v}$

$f(\mathbf{u}) = \mathbf{u}^{\mathbf{T}}$

y

A

b

c

x
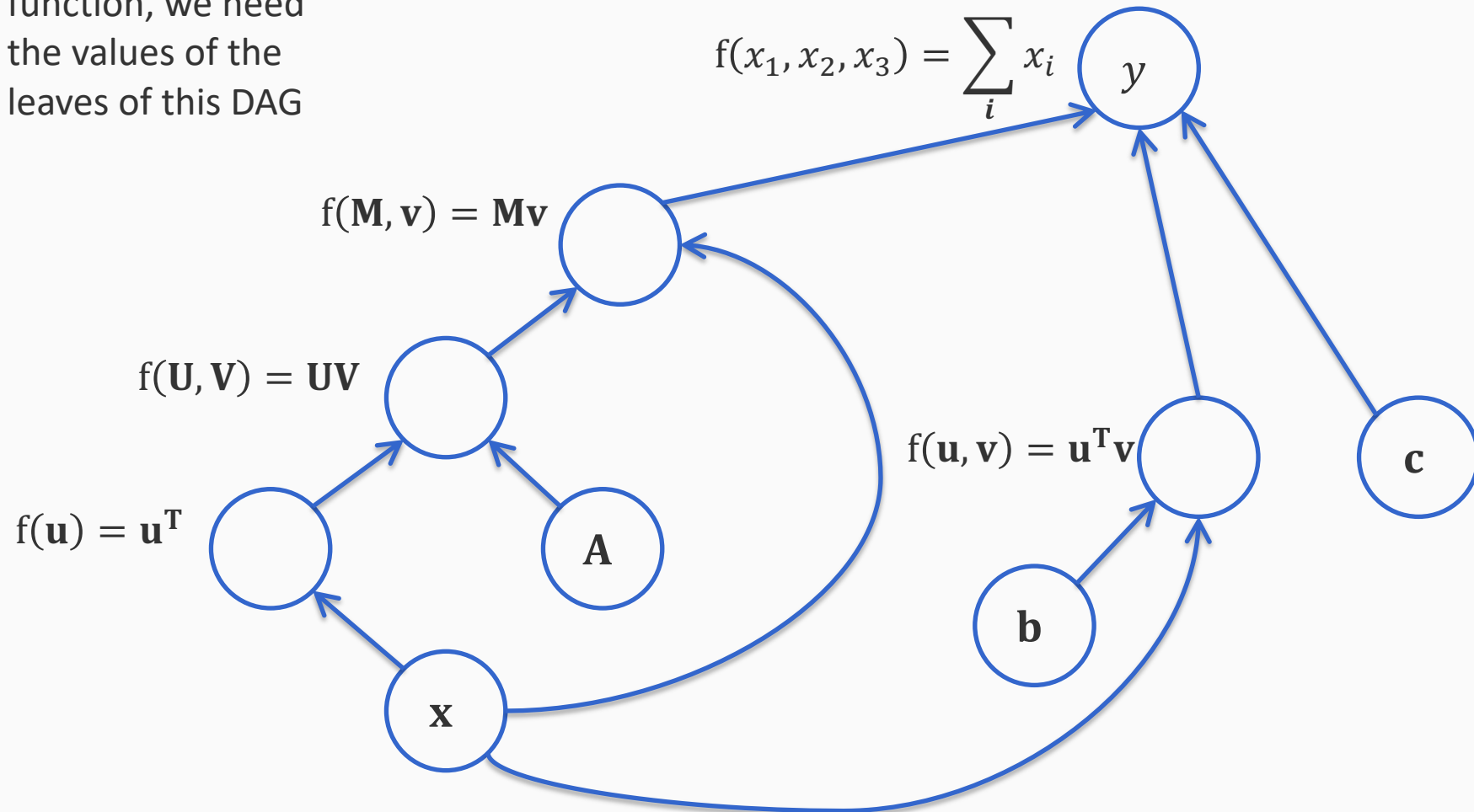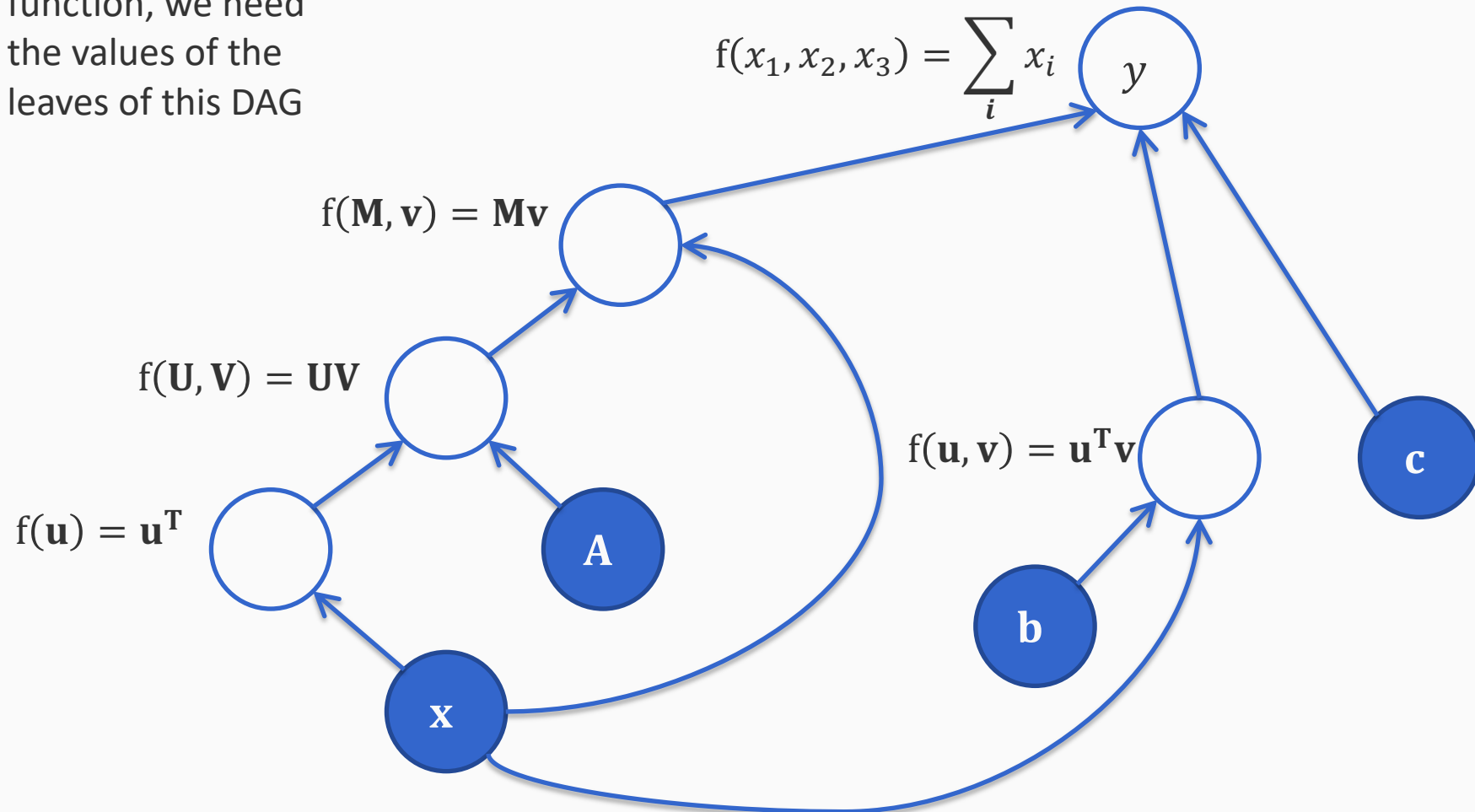
A second example

Let's also highlight which nodes can be computed using what we know so far
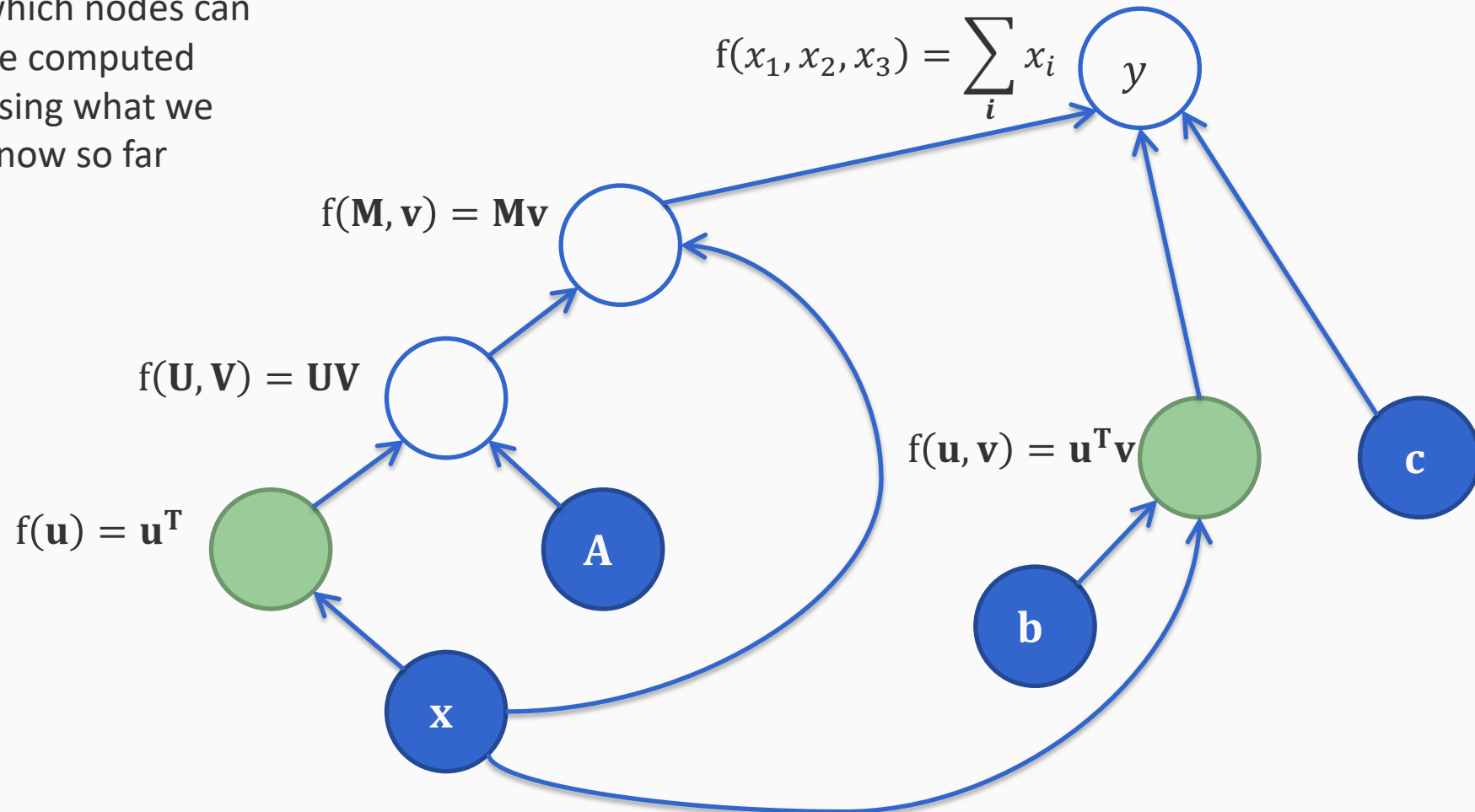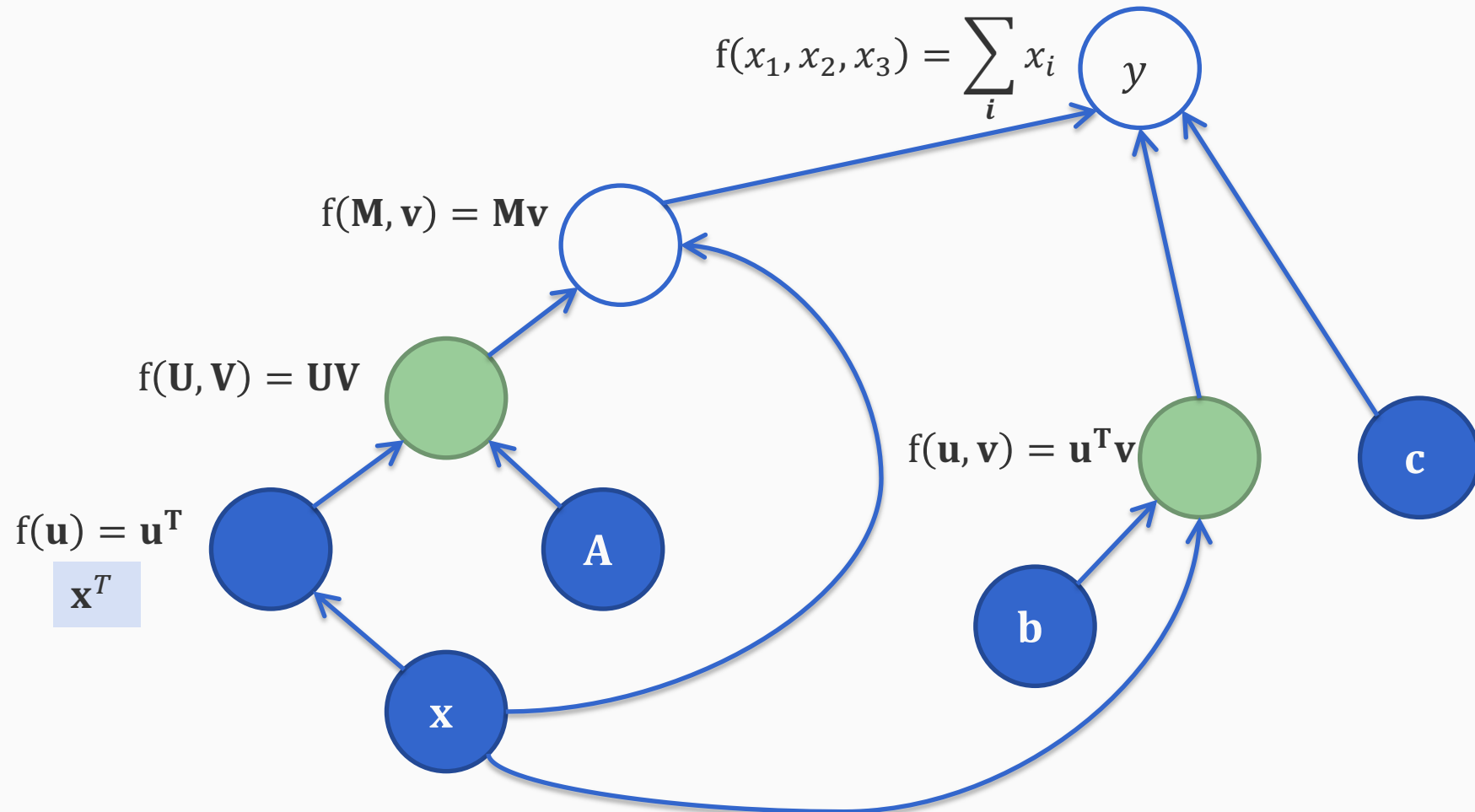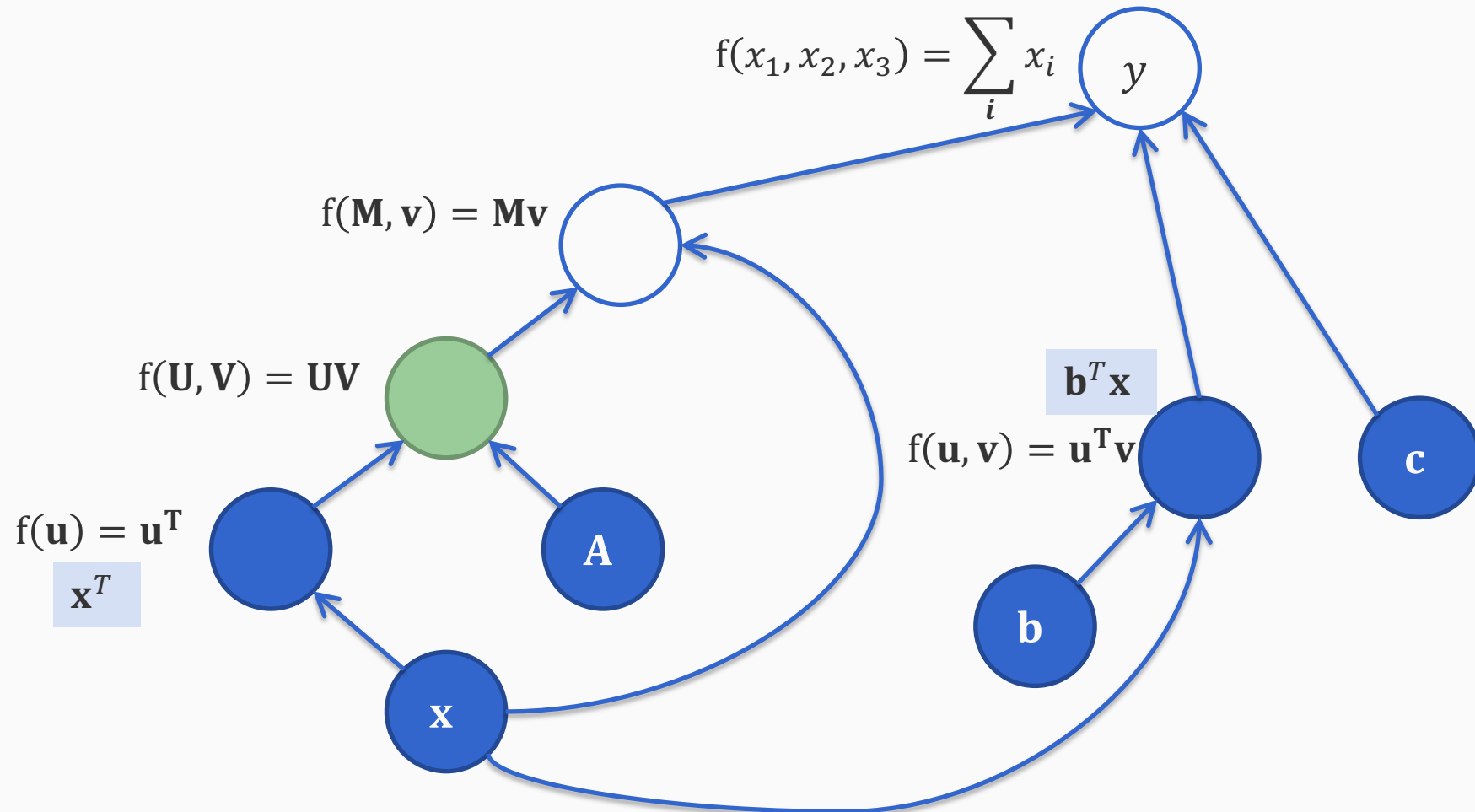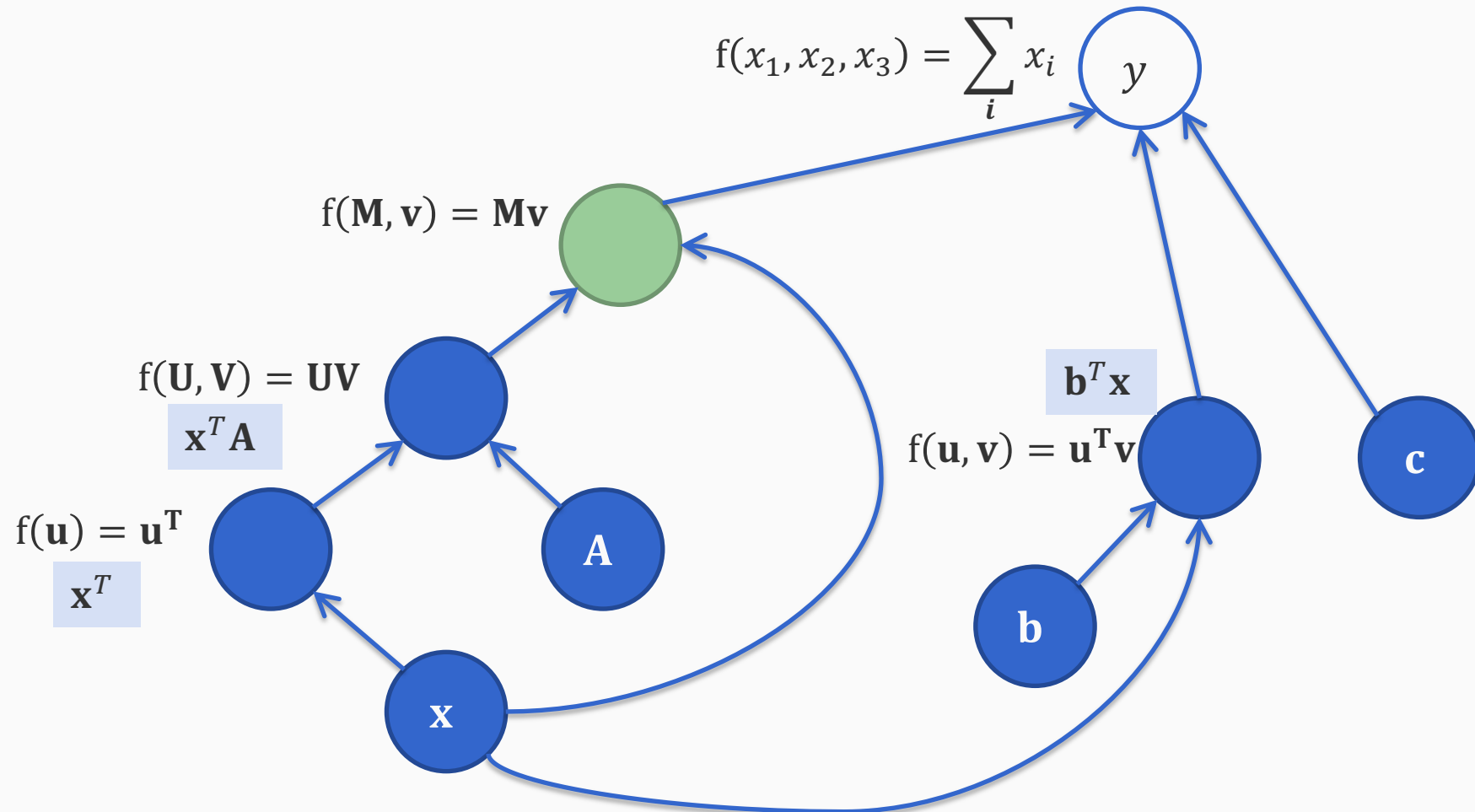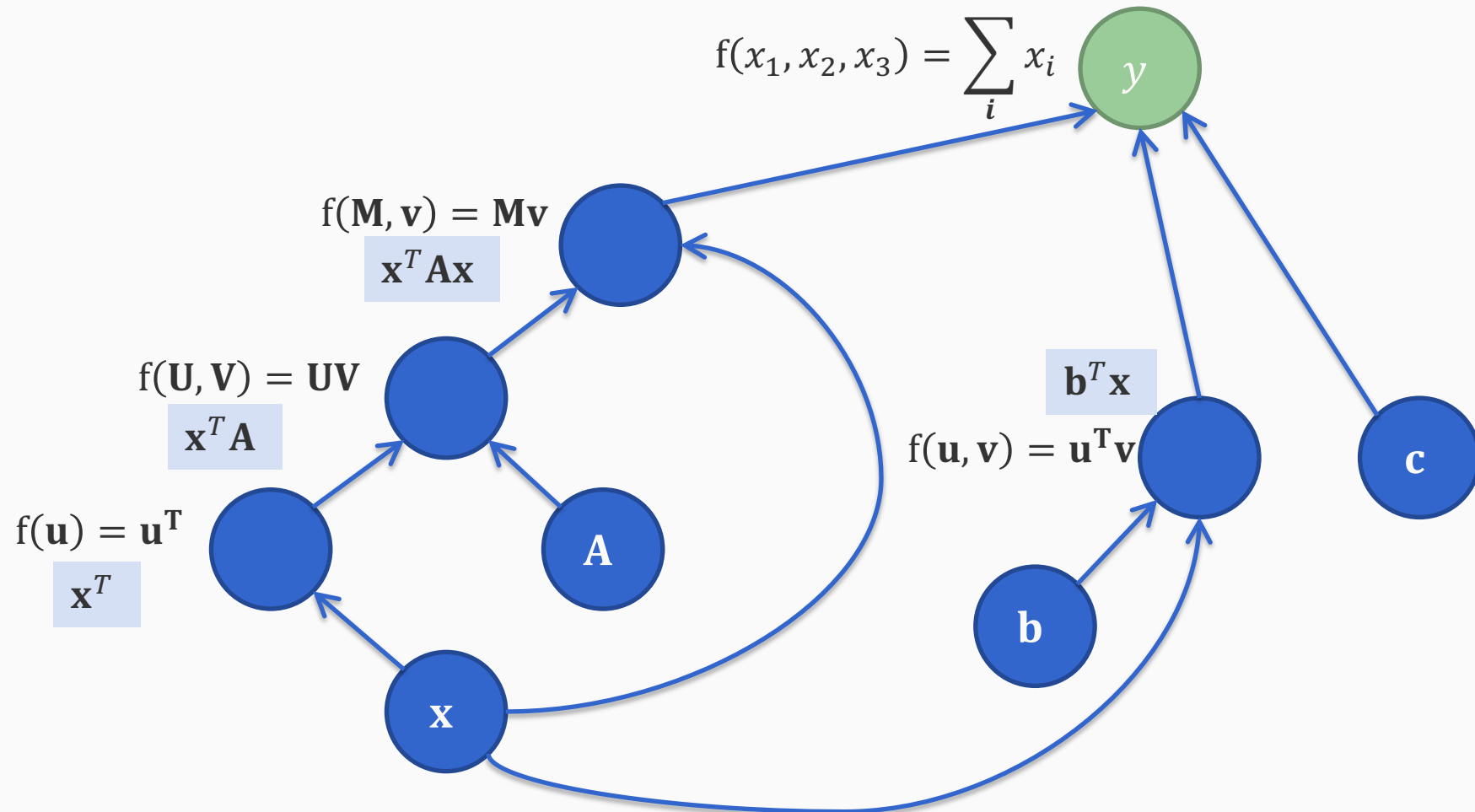
$$f(x_1, x_2, x_3) = \sum_i x_i$$

y

$$f(\mathbf{M}, \mathbf{v}) = \mathbf{Mv}$$

$$f(\mathbf{U}, \mathbf{V}) = \mathbf{UV}$$

$$f(\mathbf{u}, \mathbf{v}) = \mathbf{u^T v}$$

$$f(\mathbf{u}) = \mathbf{u^T}$$

A

b

c

x

A second example

$$f(x_1, x_2, x_3) = \sum_i x_i$$

$f(\mathbf{M}, \mathbf{v}) = \mathbf{M}\mathbf{v}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{U}\mathbf{V}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathbf{T}}\mathbf{v}$

$f(\mathbf{u}) = \mathbf{u}^{\mathbf{T}}$

$\mathbf{x}^{T}$

A second example

$$f(x_1, x_2, x_3) = \sum_i x_i$$

y

$$f(\mathbf{M}, \mathbf{v}) = \mathbf{Mv}$$

$$f(\mathbf{U}, \mathbf{V}) = \mathbf{UV}$$

$$\mathbf{b}^T\mathbf{x}$$

$$f(\mathbf{u}, \mathbf{v}) = \mathbf{u^Tv}$$

$$f(\mathbf{u}) = \mathbf{u^T}$$

$$\mathbf{x}^T$$

A

c

b

x

A second example

$$f(x_1, x_2, x_3) = \sum_i x_i$$

y

$f(\mathbf{M}, \mathbf{v}) = \mathbf{Mv}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{UV}$

$\mathbf{x}^T \mathbf{A}$

$f(\mathbf{u}) = \mathbf{u^T}$

$\mathbf{x}^T$

$\mathbf{b}^T \mathbf{x}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u^T v}$

$\mathbf{A}$

$\mathbf{b}$

$\mathbf{c}$

$\mathbf{x}$

A second example

$$f(x_1, x_2, x_3) = \sum_i x_i$$

y

$f(\mathbf{M}, \mathbf{v}) = \mathbf{M}\mathbf{v}$

$\mathbf{x}^T \mathbf{A} \mathbf{x}$

$f(\mathbf{U}, \mathbf{V}) = \mathbf{U}\mathbf{V}$

$\mathbf{x}^T \mathbf{A}$

$\mathbf{b}^T \mathbf{x}$

$f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$

$f(\mathbf{u}) = \mathbf{u}^T$

$\mathbf{x}^T$

$\mathbf{A}$

$\mathbf{b}$

$\mathbf{c}$

$\mathbf{x}$

A second example

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c}$$

$$f(x_1, x_2, x_3) = \sum_i x_i$$

$$f(\mathbf{M}, \mathbf{v}) = \mathbf{M}\mathbf{v}$$
$$\mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$f(\mathbf{U}, \mathbf{V}) = \mathbf{U}\mathbf{V}$$
$$\mathbf{x}^T\mathbf{A}$$

$$f(\mathbf{u}) = \mathbf{u}^T$$
$$\mathbf{x}^T$$

$$\mathbf{b}^T\mathbf{x}$$
$$f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}$$

y

$\mathbf{A}$

$\mathbf{c}$

$\mathbf{x}$

$\mathbf{b}$
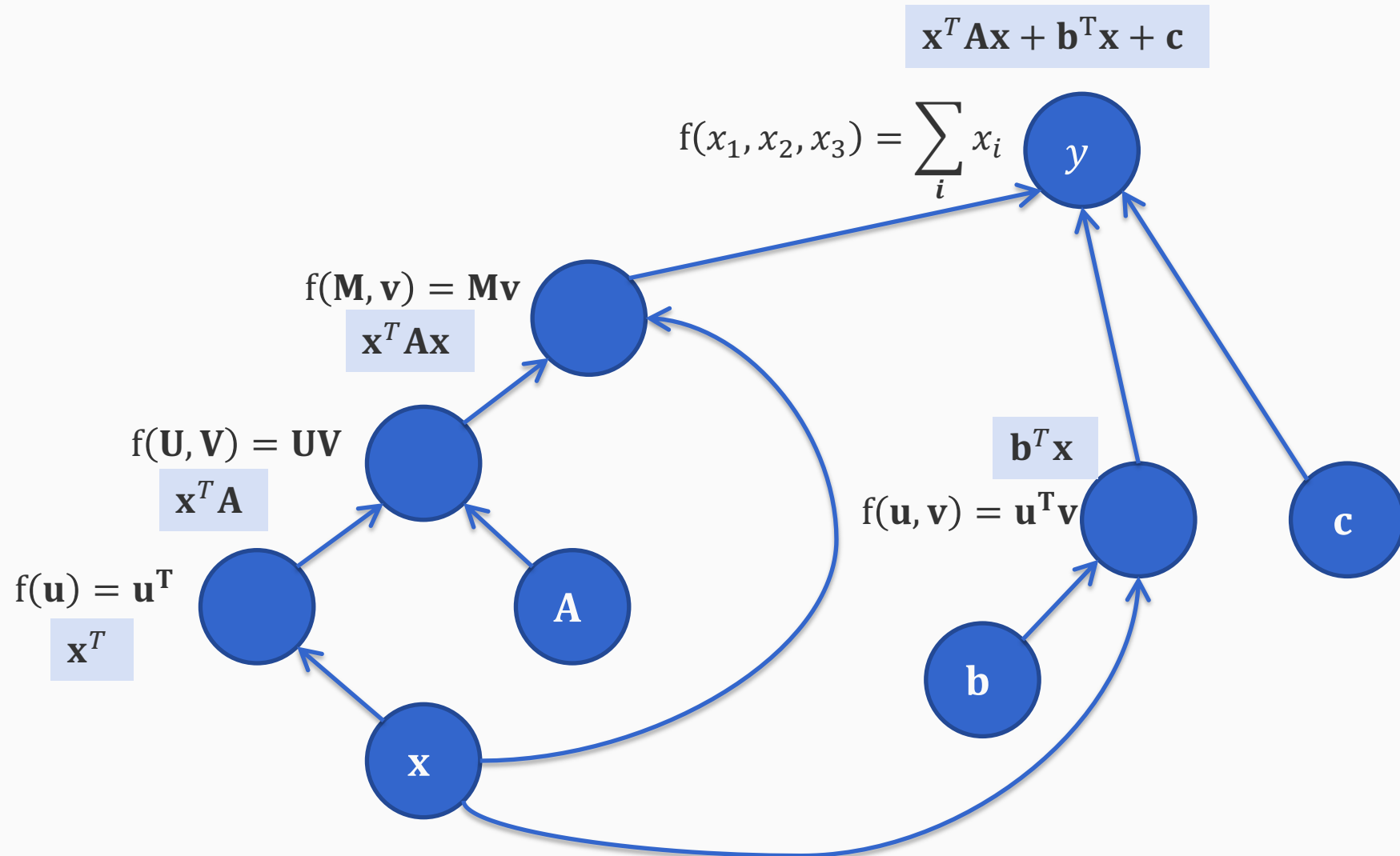
Forward propagation

Given a computation graph G and values of its input nodes:

For each node in the graph, in topological order:

    Compute the value of that node

Forward propagation

Given a computation graph G and values of its input nodes:

For each node in the graph, in **topological order**:

    Compute the value of that node

Why topological order: Ensures that children are computed before parents.