Inductive logic programming

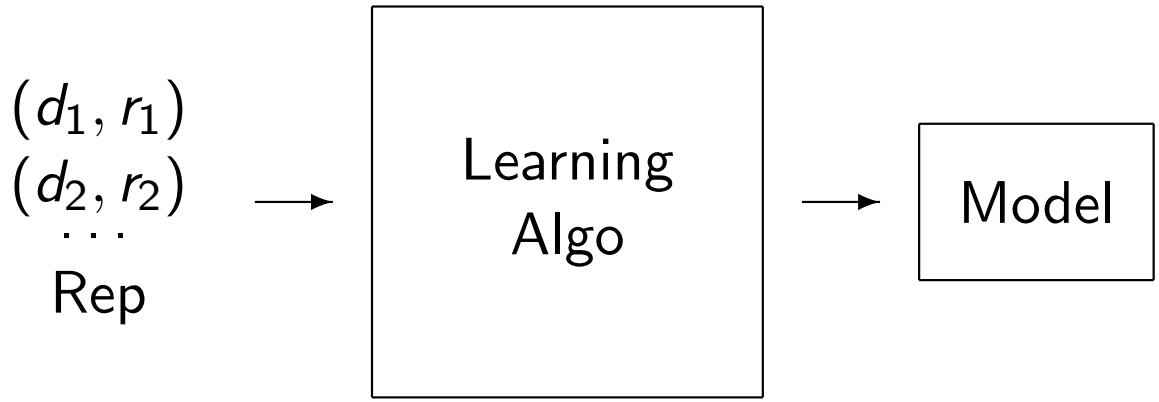
In Federated Learning

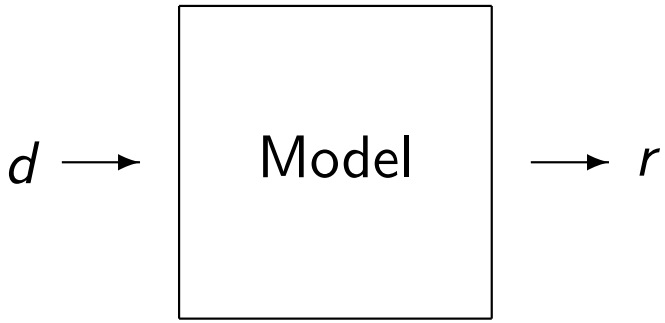
S. Jacquet, J.-M. Jacquet, I. Linden, W. Vanhoof

University of Namur

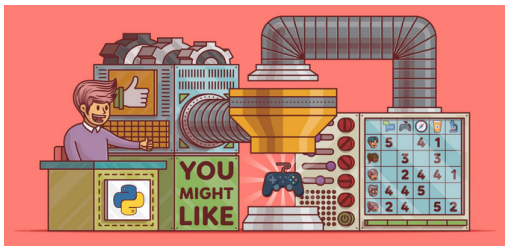
February 21st 2022

Machine learning in a picture

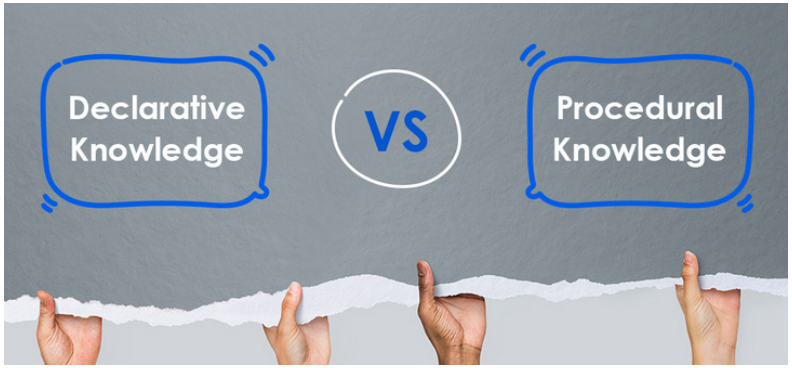


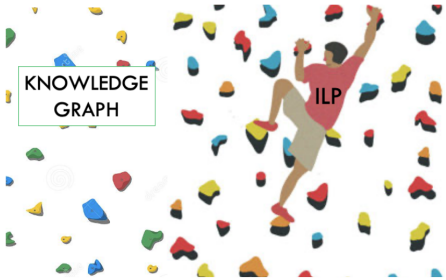


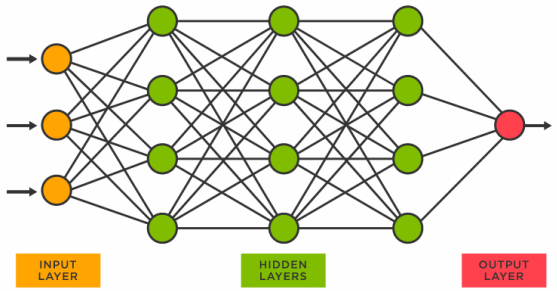
Typical applications

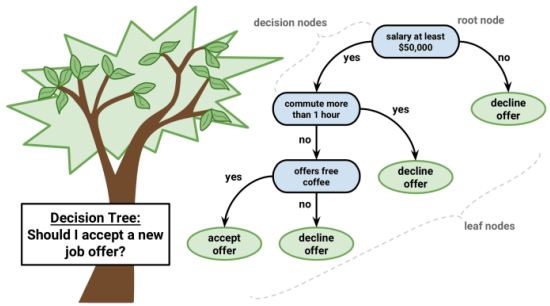


Two kinds of representation









Inductive logic programming

- Given background knowledge

```
parent(ann,mary).        female(ann).
parent(ann,tom).         female(mary).
parent(tom,eve).         female(eve).
```

- Given positive and negative information

```
+ daughter(mary,ann).    - daughter(tom,ann).
+ daughter(eve,tom).     - daughter(tom,eve).
```

- Induce relations

```
daughter(X,Y) :- parent(Y,X), female(X).
```

Inductive logic programming

- Given background knowledge

```
parent(ann,mary).        female(ann).
parent(ann,tom).         female(mary).
parent(tom,eve).         female(eve).
```

- Given positive and negative information

```
+ daughter(mary,ann).    - daughter(tom,ann).
+ daughter(eve,tom).     - daughter(tom,eve).
```

- Induce relations

```
daughter(X,Y) :- parent(Y,X), female(X).
```

Inductive logic programming

- Given background knowledge

```
parent(ann,mary).        female(ann).
parent(ann,tom).         female(mary).
parent(tom,eve).         female(eve).
```

- Given positive and negative information

```
+ daughter(mary,ann).    - daughter(tom,ann).
+ daughter(eve,tom).     - daughter(tom,eve).
```

- Induce relations

```
daughter(X,Y) :- parent(Y,X), female(X).
```

Inductive logic programming

- Given background knowledge

```
parent(ann,mary).        female(ann).
parent(ann,tom).         female(mary).
parent(tom,eve).         female(eve).
```

- Given positive and negative information

```
+ daughter(mary,ann).    - daughter(tom,ann).
+ daughter(eve,tom).     - daughter(tom,eve).
```

- Induce relations

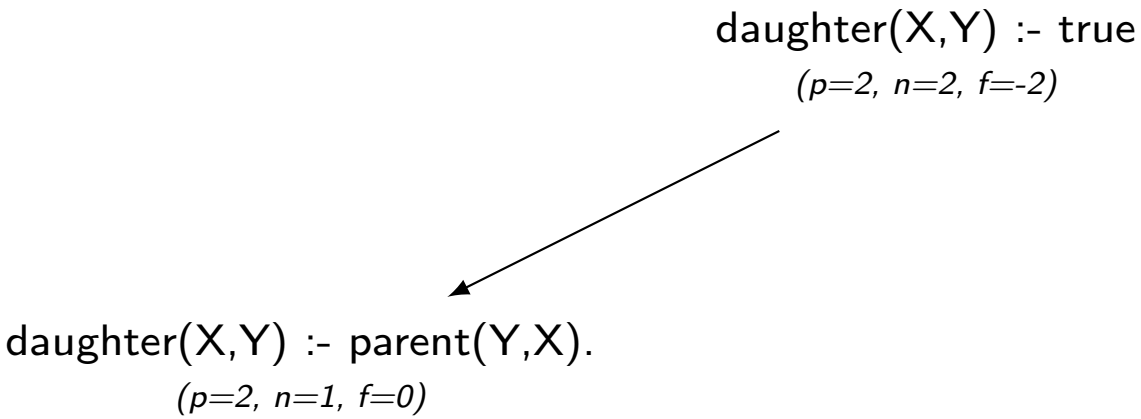
```
daughter(X,Y) :- parent(Y,X), female(X).
```

Basic strategy

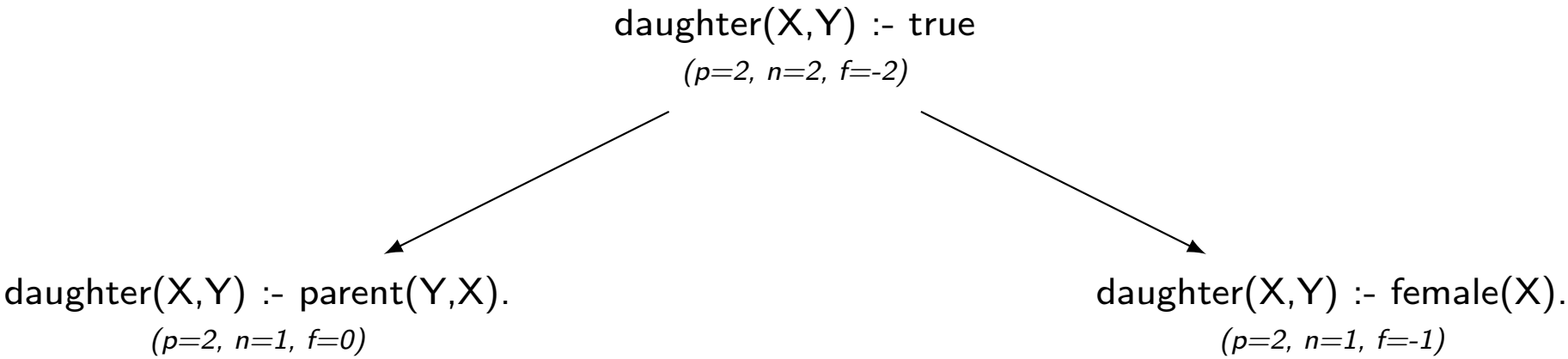
daughter(X,Y) :- true

(p=2, n=2, f=-2)

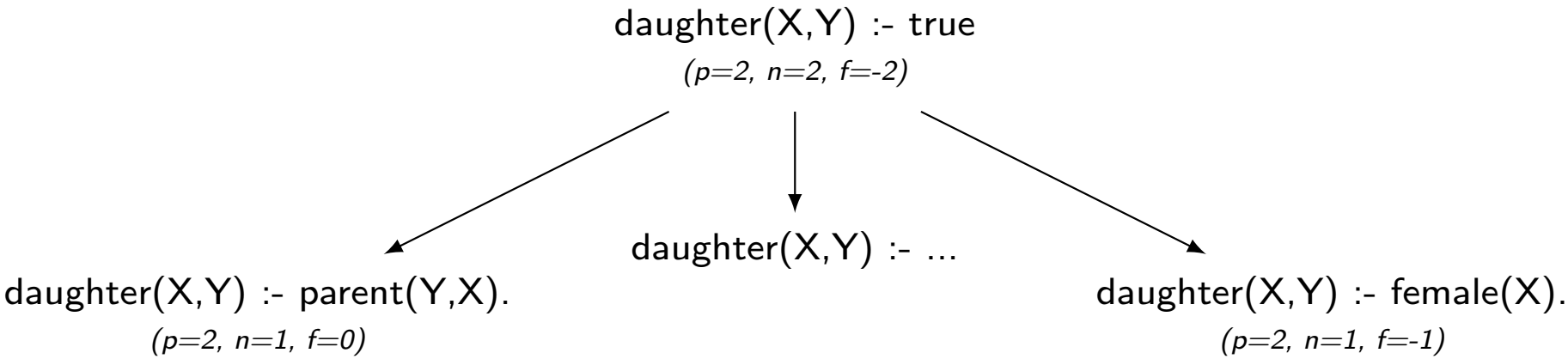
Basic strategy



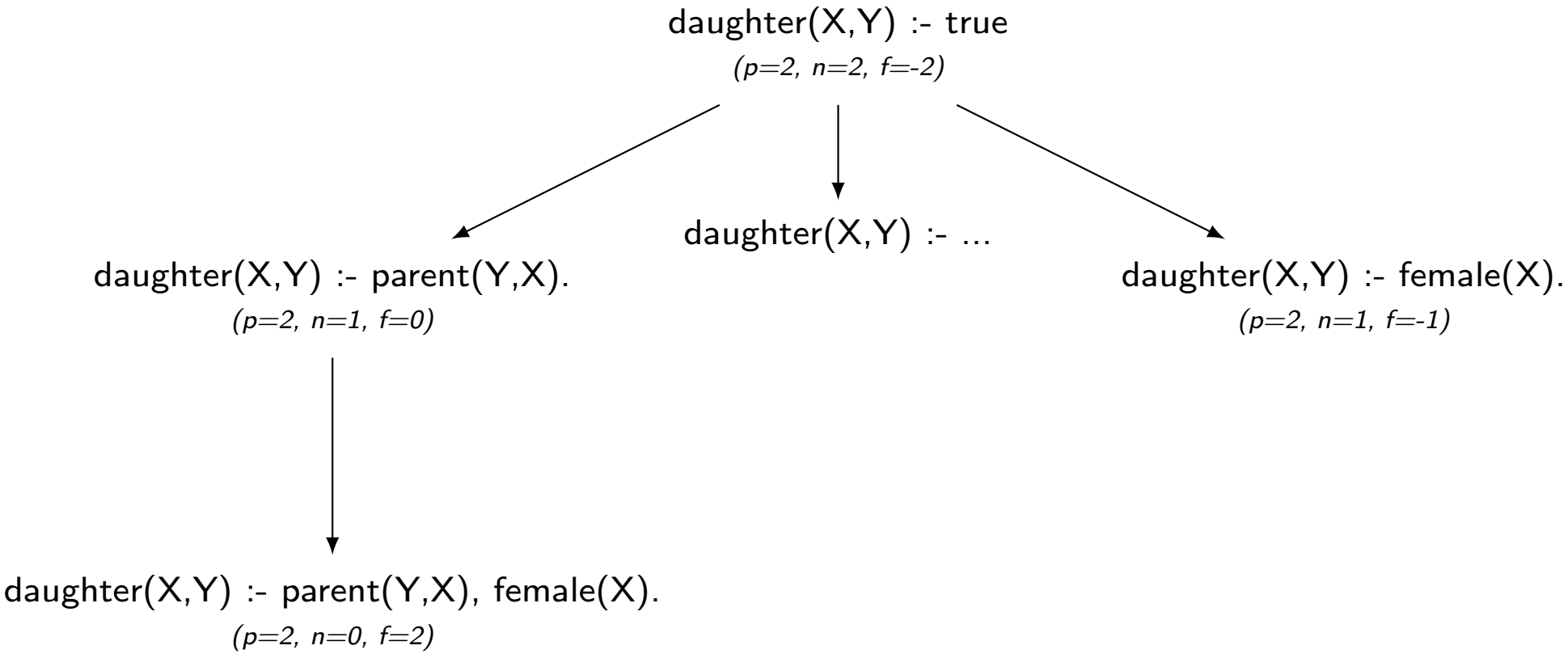
Basic strategy



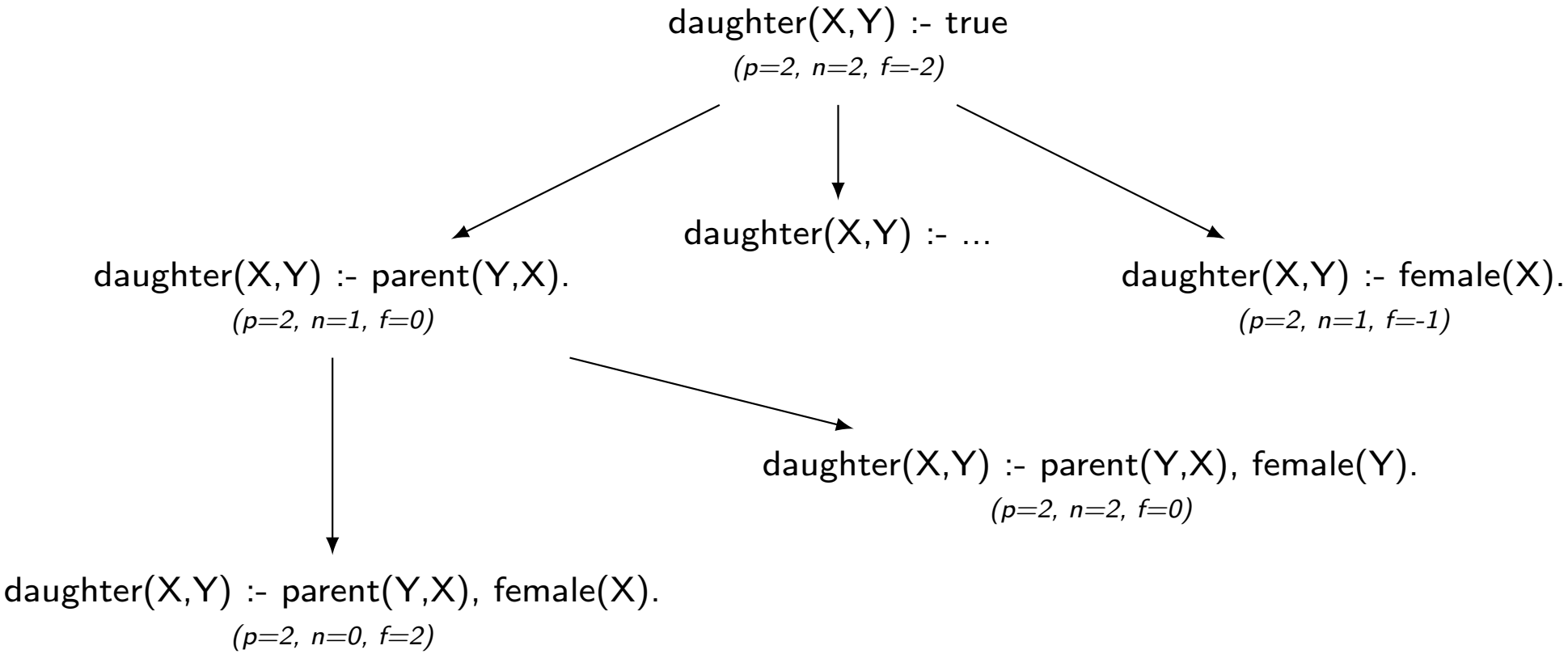
Basic strategy



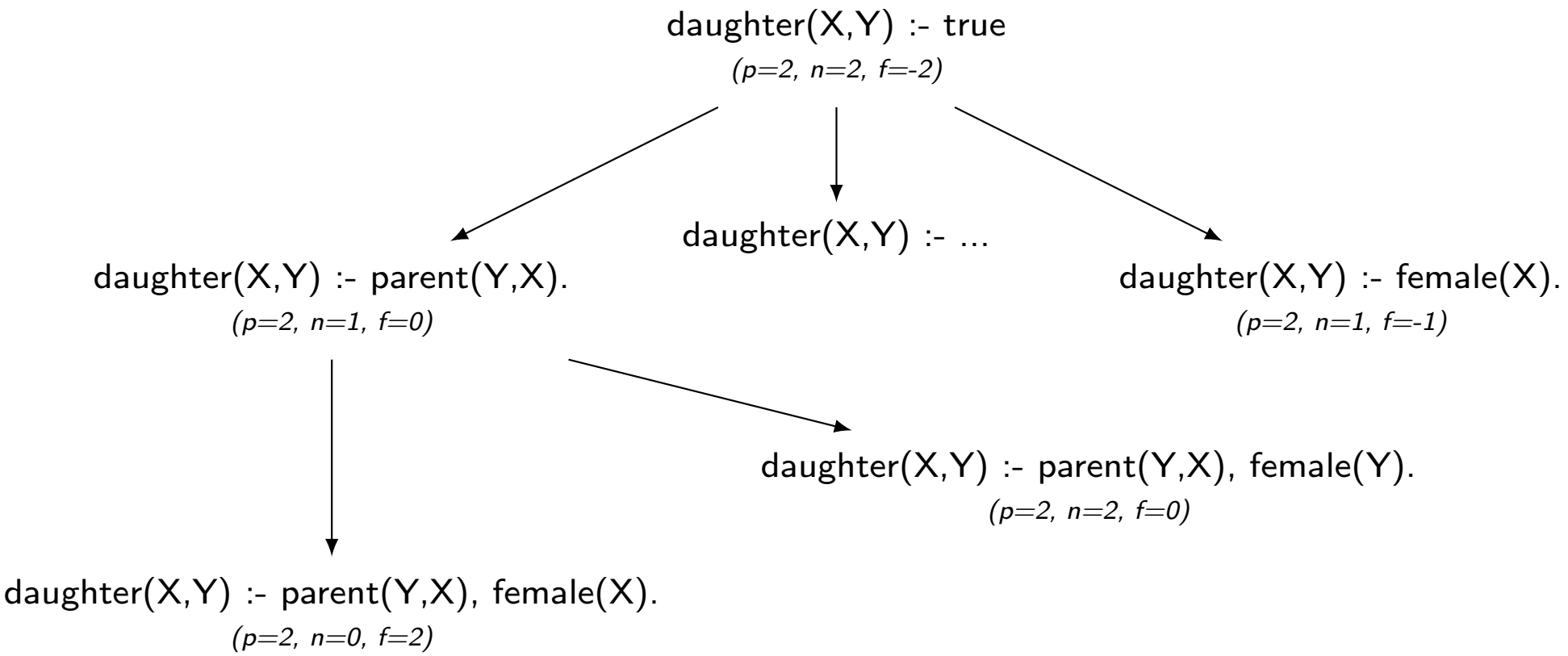
Basic strategy



Basic strategy



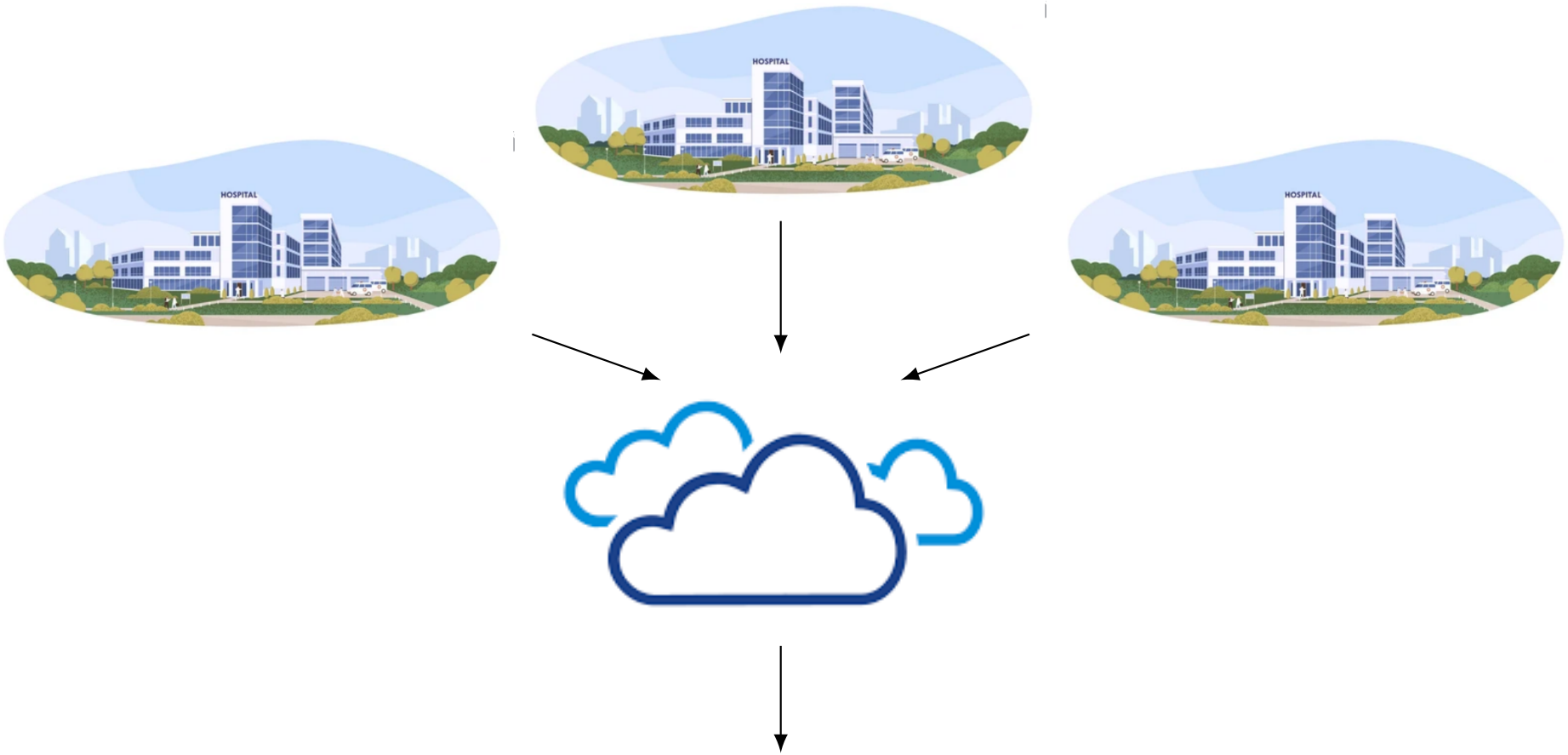
Basic strategy



Key features

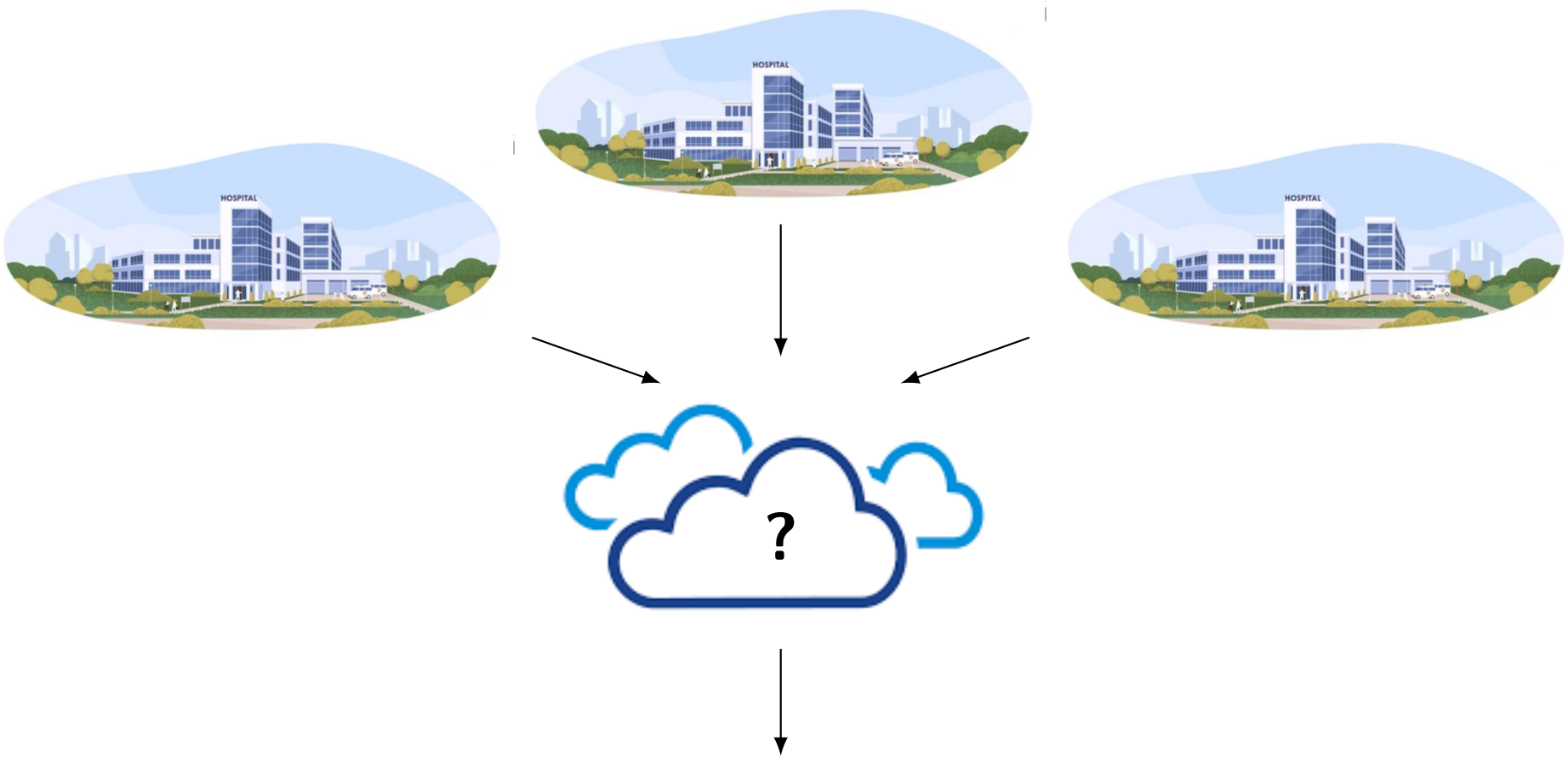
Incremental & theory-based

Federated learning

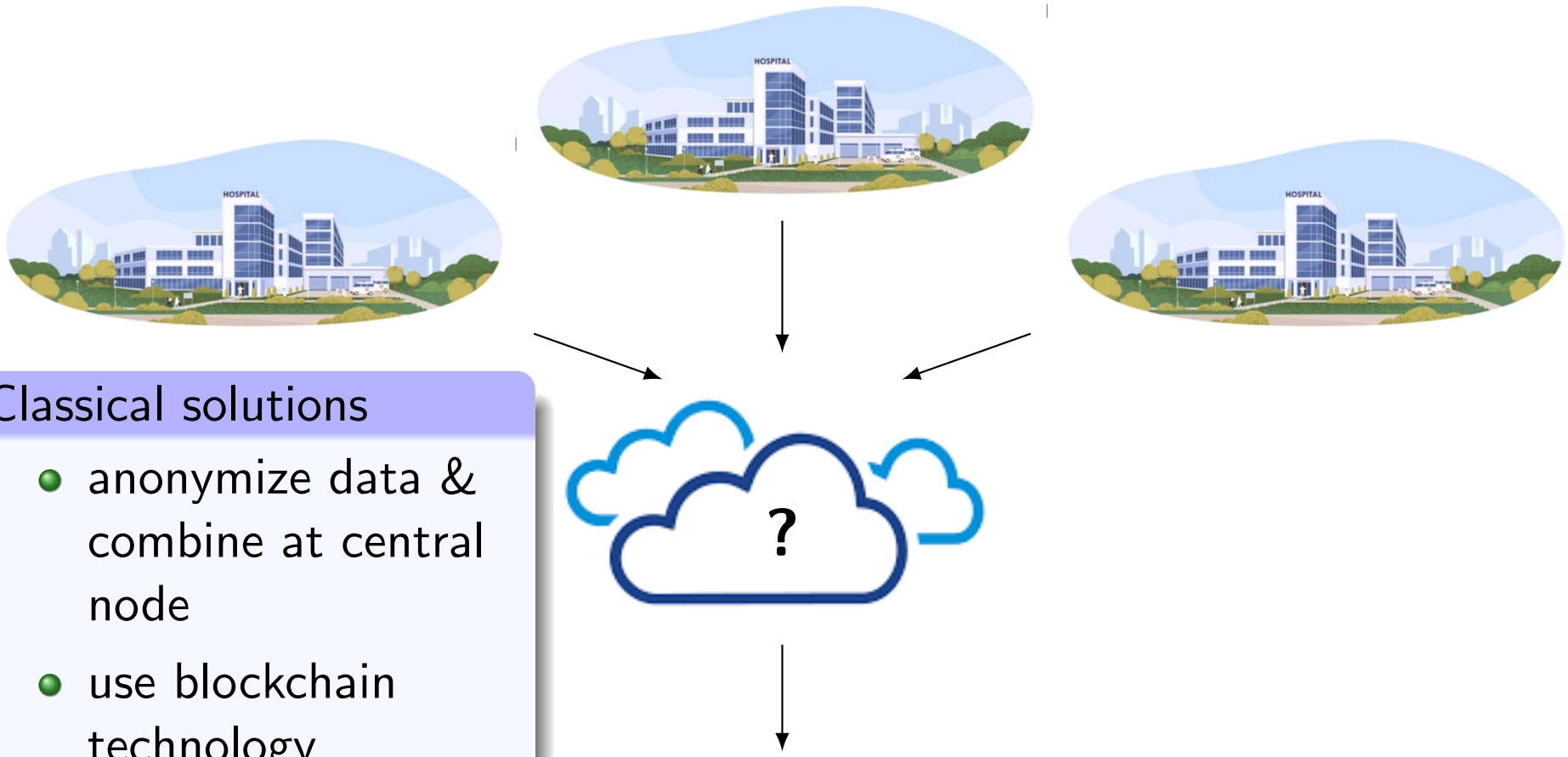



```
parent(abraham, isaac). %p1
parent(isaac, jacob).   %p2
parent(sarah, isaac).   %p3
parent(jacob, joseph).  %p4
parent(jacob, dan).     %p5
parent(jacob, dinah).   %p6

male(abraham).          %m1
male(isaac).            %m2
male(jacob).            %m3
male(joseph).           %m4
male(dan).              %m5

son(X,Y) :- parent(Y,X), %s1
            male(X).
```

Federated learning



Federated learning

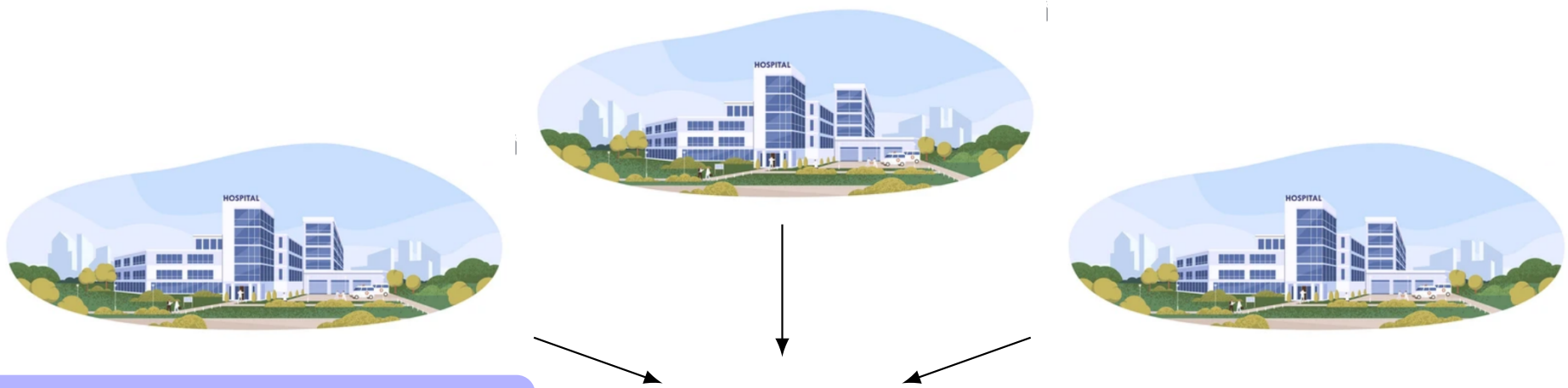


Classical solutions

- anonymize data & combine at central node
- use blockchain technology

```
parent(abraham, isaac). %p1
parent(isaac, jacob).   %p2
parent(sarah, isaac).   %p3
parent(jacob, joseph).  %p4
parent(jacob, dan).     %p5
parent(jacob, dinah).   %p6

male(abraham).          %m1
male(isaac).            %m2
male(jacob).            %m3
male(joseph).           %m4
male(dan).              %m5

son(X,Y) :- parent(Y,X), %s1
            male(X).
```

Federated learning



Classical solutions

- anonymize data & combine at central node
- use blockchain technology


```
parent(abraham, isaac).  %p1
parent(isaac, jacob).    %p2
parent(sarah, isaac).    %p3
parent(jacob, joseph).   %p4
parent(jacob, dan).      %p5
parent(jacob, dinah).    %p6

male(abraham).           %m1
male(isaac).             %m2
male(jacob).             %m3
male(joseph).            %m4
male(dan).               %m5

son(X,Y) :- parent(Y,X), %s1
            male(X).
```

Proposed solution

- learn theories locally
- combine theories at central node