# Lightweight Diagramming for Lightweight Formal Methods

## A Grounded Language Design

Siddhartha Prasad
Tim Nelson
Shriram Krishnamurthi


Ben Greenman

Match the domain

Dining Philosophers

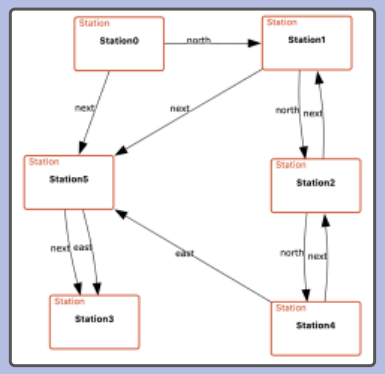
Tic Tac Toe

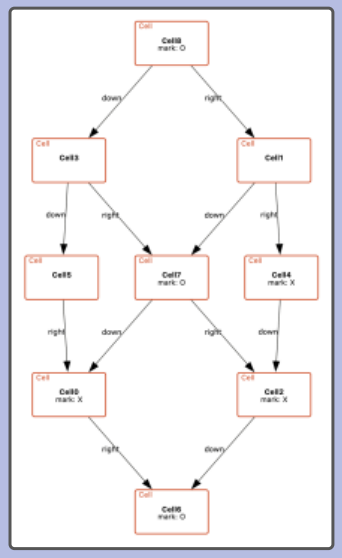
Subway Map

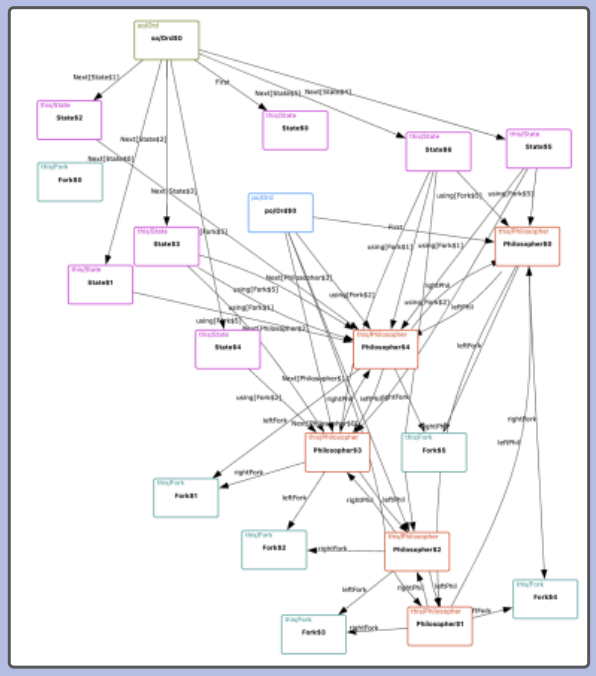
Dining Philosophers

Tic Tac Toe

Subway Map



Dining Philosophers

Tic Tac Toe

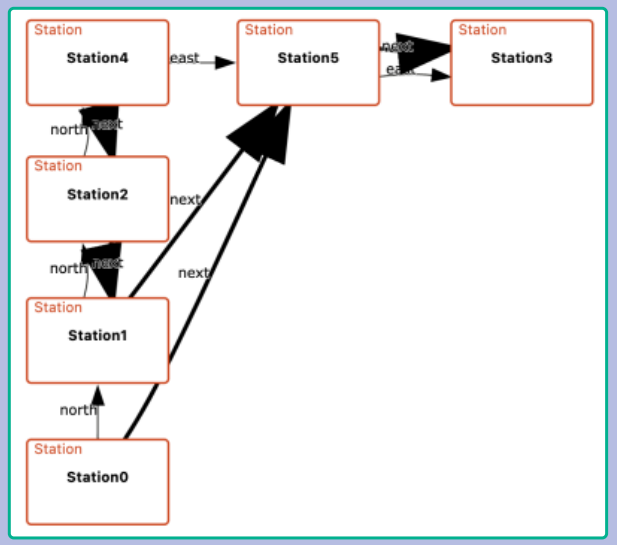
Subway Map

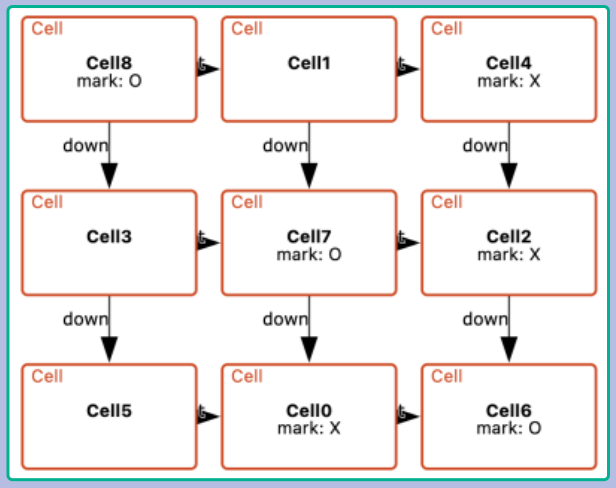
Hmm ... let's try again

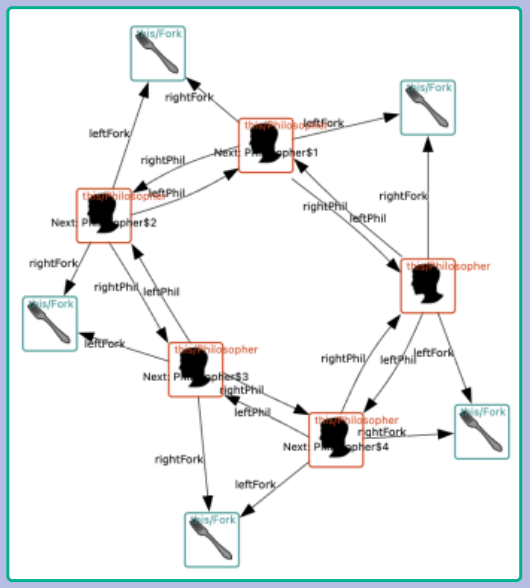
Dining Philosophers

Tic Tac Toe

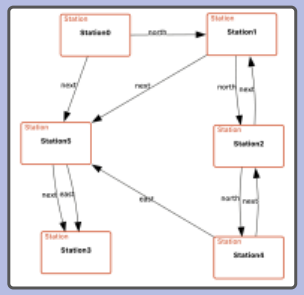
Subway Map

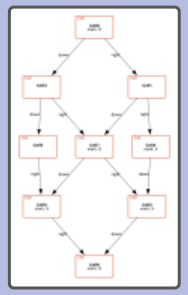


# How did we get better pics?

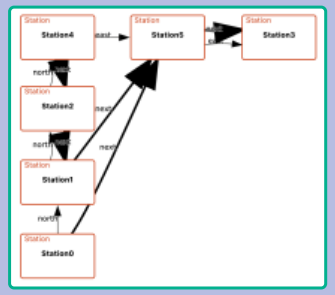


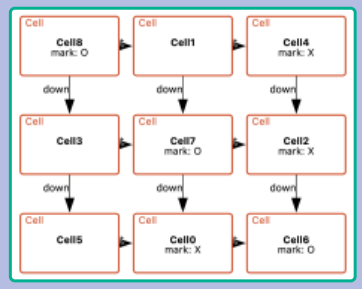



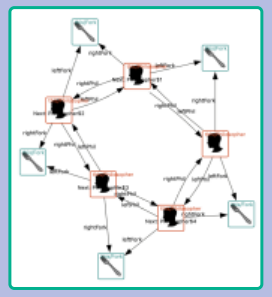
3 LOC

6 LOC

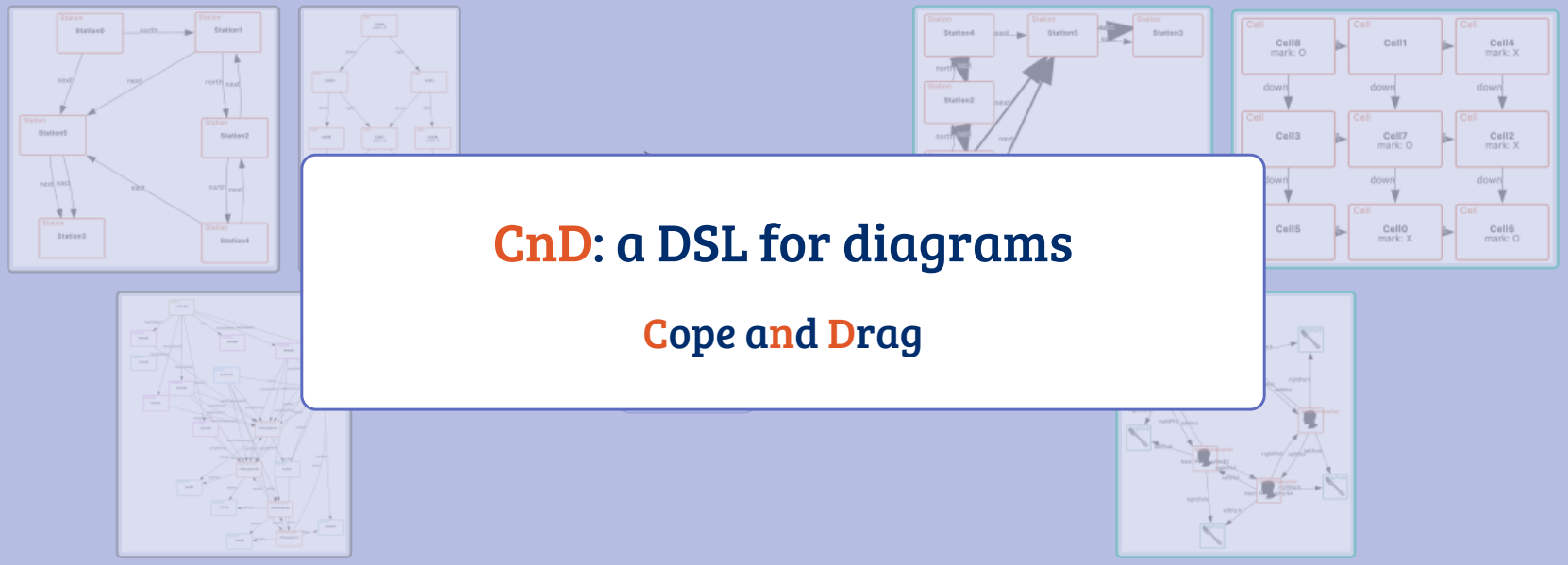
21 LOC



How did we get better pics?



# CnD: a DSL for diagrams

Cope and Drag

CnD is for Forge/Alloy specifications

Lightweight Formal Methods

# CnD is for Forge/Alloy specifications

Lightweight Formal Methods

```
#lang forge

sig Node {
  key : one Int,
  left : lone Node,
  right : lone Node
}

pred binaryTree {
  all disj n1, n2 : Node |
    ....
```


- binaryTree #1
- binaryTree #2
- ...

# CnD is for Forge/Alloy specifications

Lightweight Formal Methods

```
#lang forge

sig Node {
  key : one Int,
}

pred binaryTree {
  all disj n1, n2 : Node |
    ....
```

**Specification**

Objects + Constraints


**Instances**

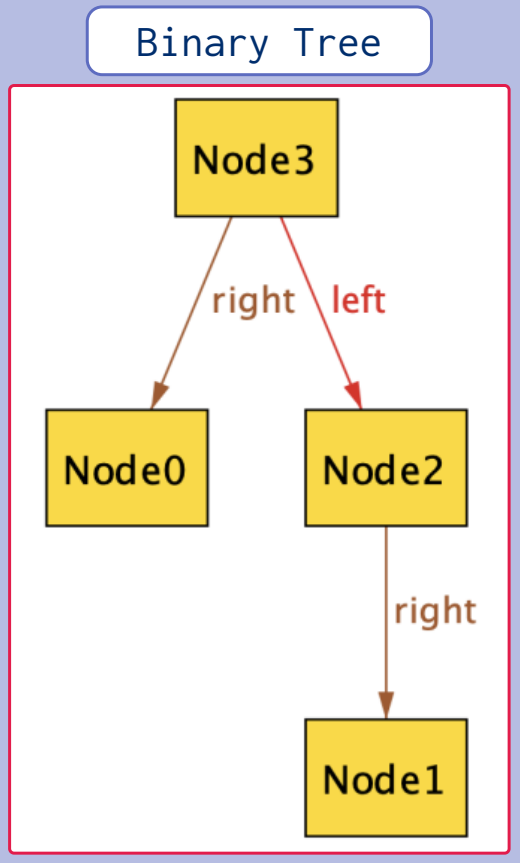
Directed Graphs

...

# Why CnD?

What's so hard about drawing directed graphs?

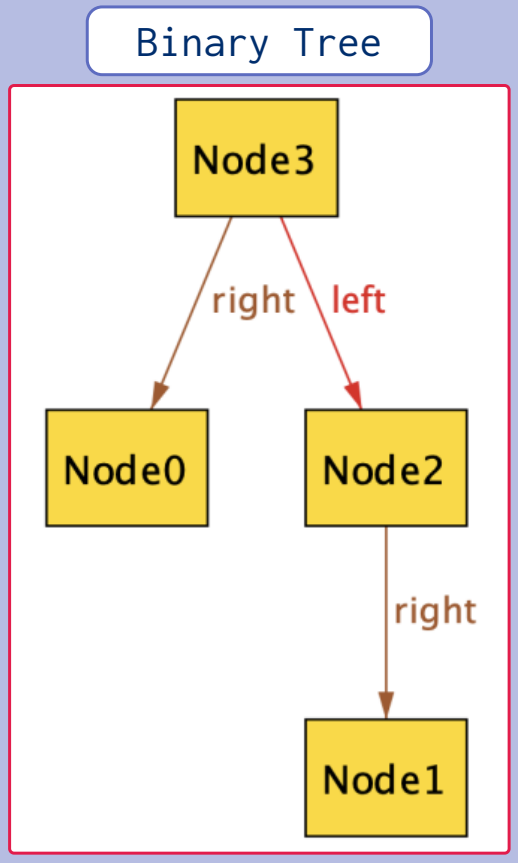
Q. Anything wrong with this diagram?



Q. Anything wrong with this diagram?

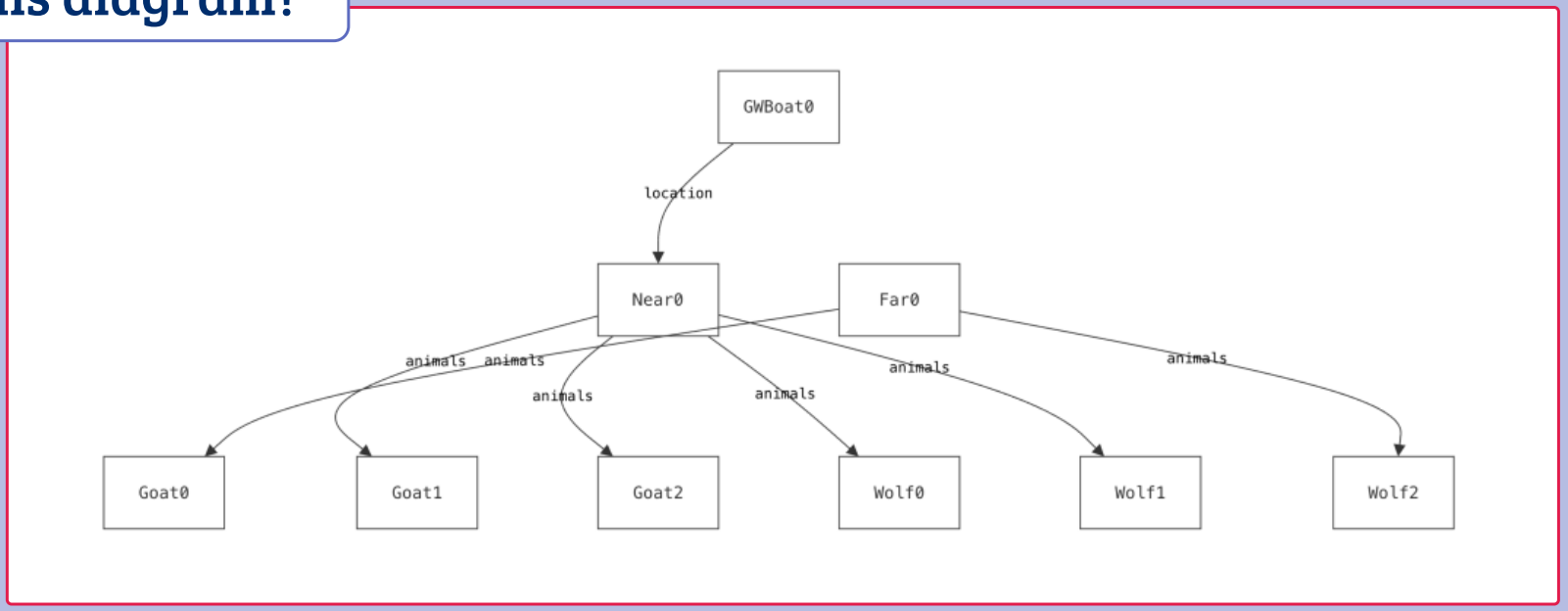
A. Bad Orientation

(a Stroop effect)



**Q. Anything wrong with this diagram?**

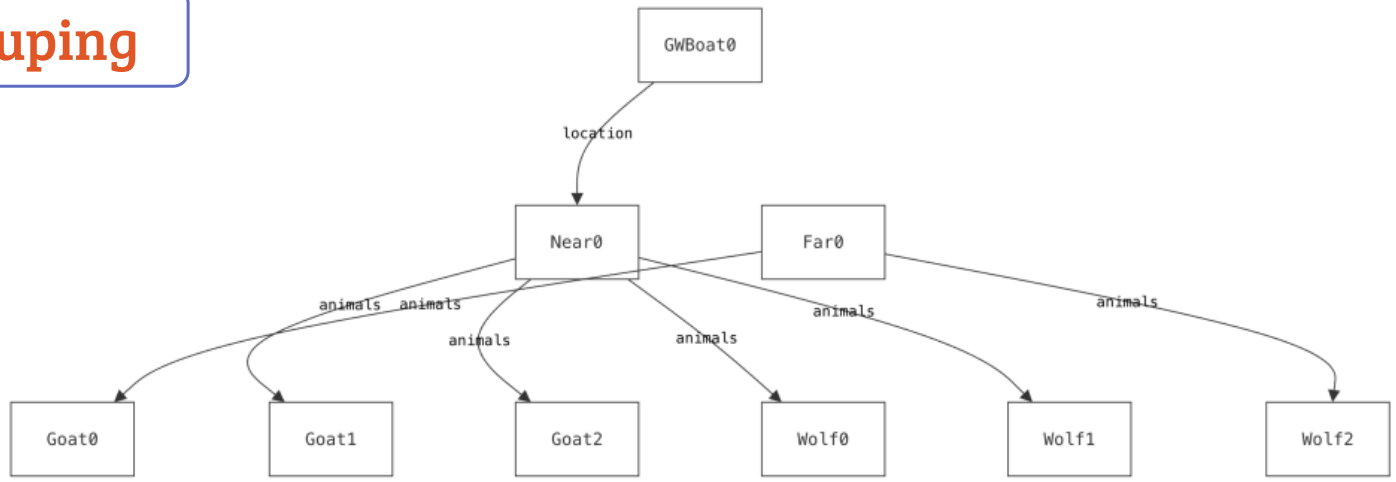
River Crossing



Q. Anything wrong with this diagram?

A. Bad Grouping

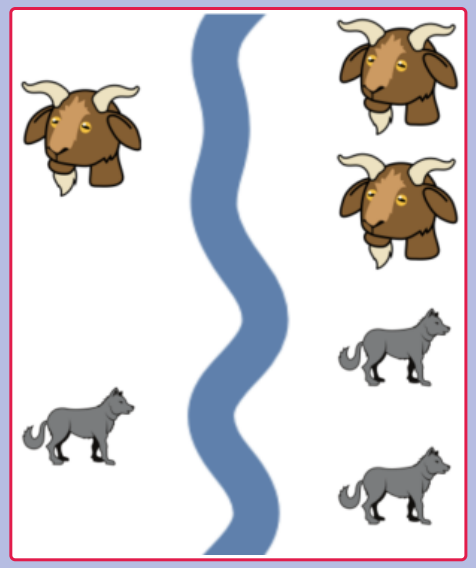
River Crossing



# Pretty diagram, by Sterling

https://sterling-js.github.io

River Crossing



River Crossing

**Problems:**

River Crossing

**Problems:**

1. **Programming required**

   **+ order of magnitude**

River Crossing

**Problems:**

1. **Programming required**
   + order of magnitude
2. **Distracts from modeling**

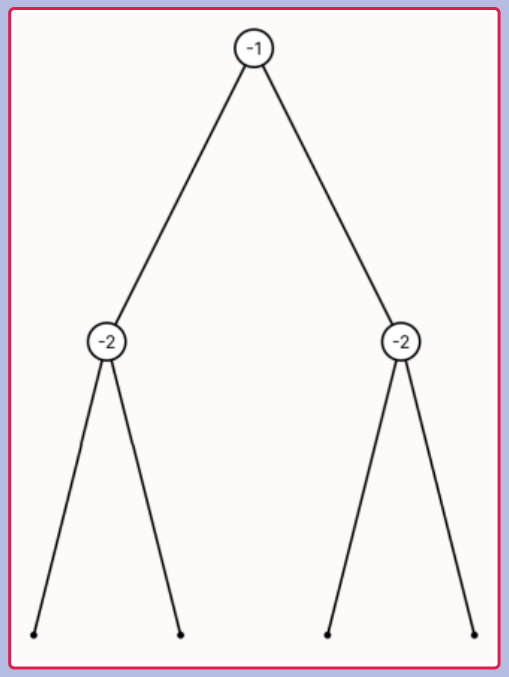
River Crossing

**Problems:**

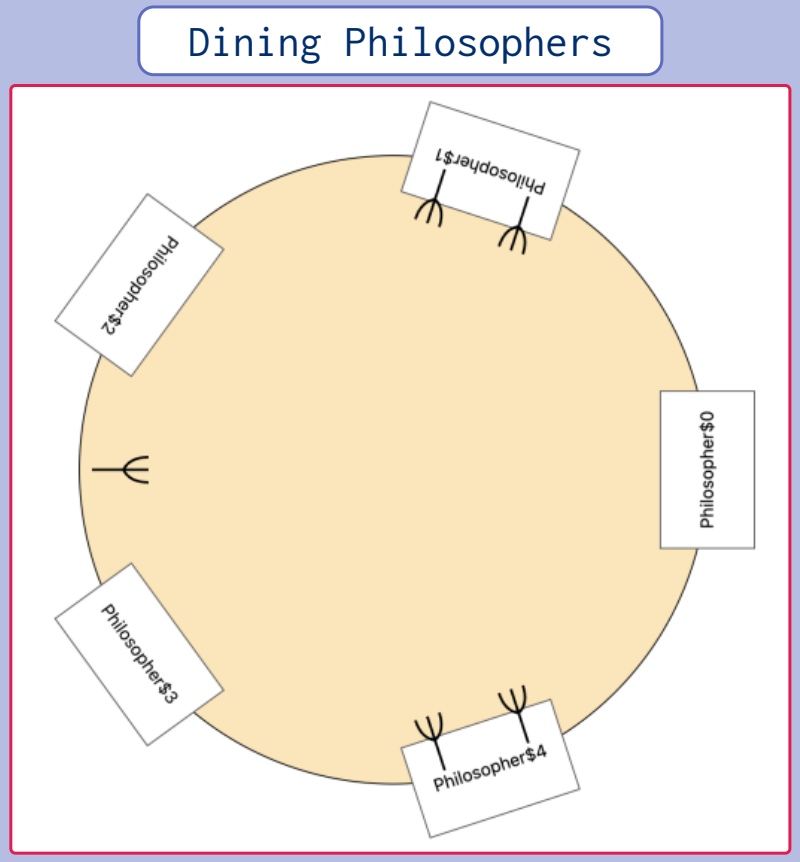
1. **Programming required**
   + order of magnitude
2. **Distracts from modeling**
3. **Useless in development**

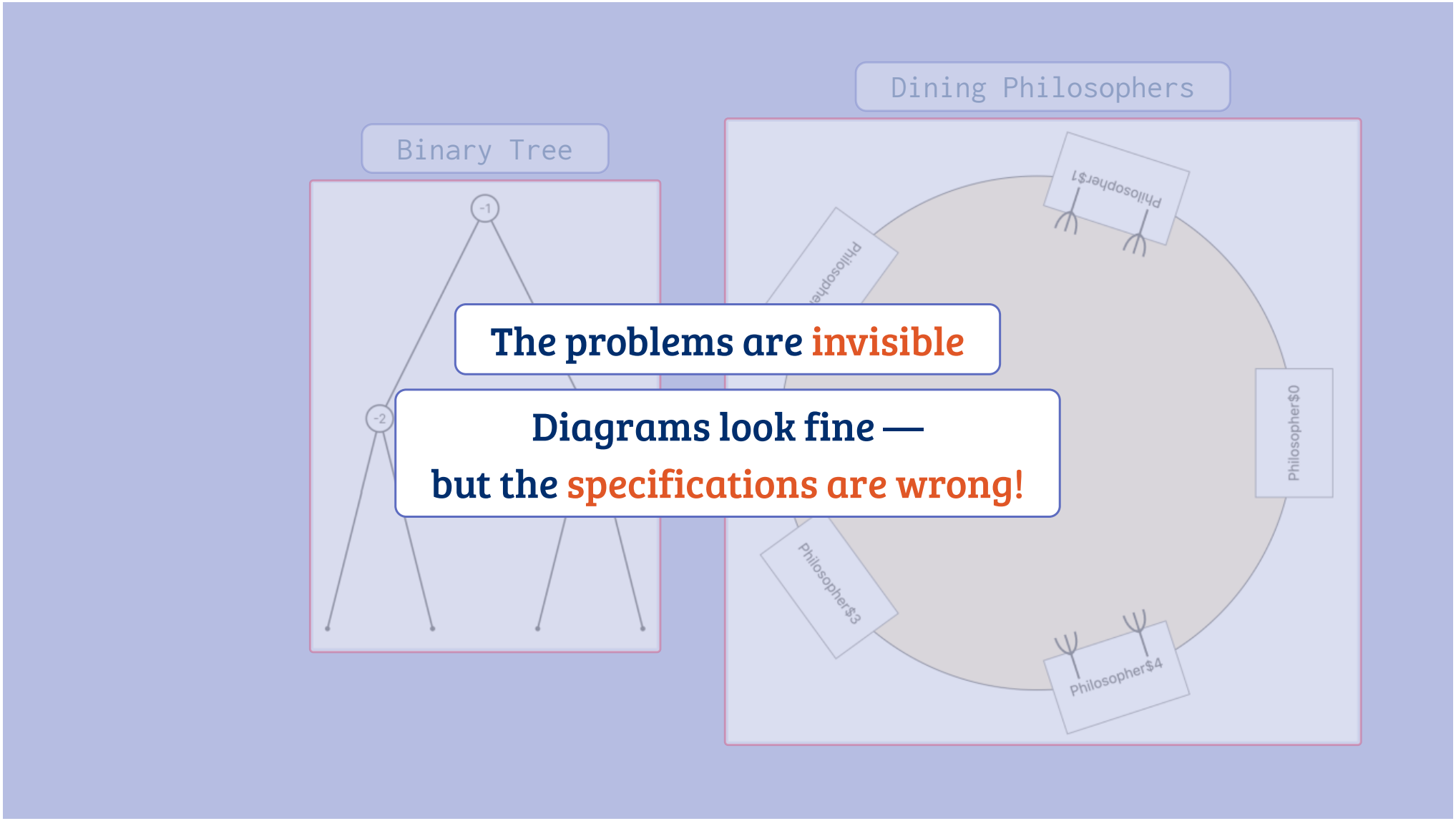
Q. Anything wrong with this diagram?

Binary Tree



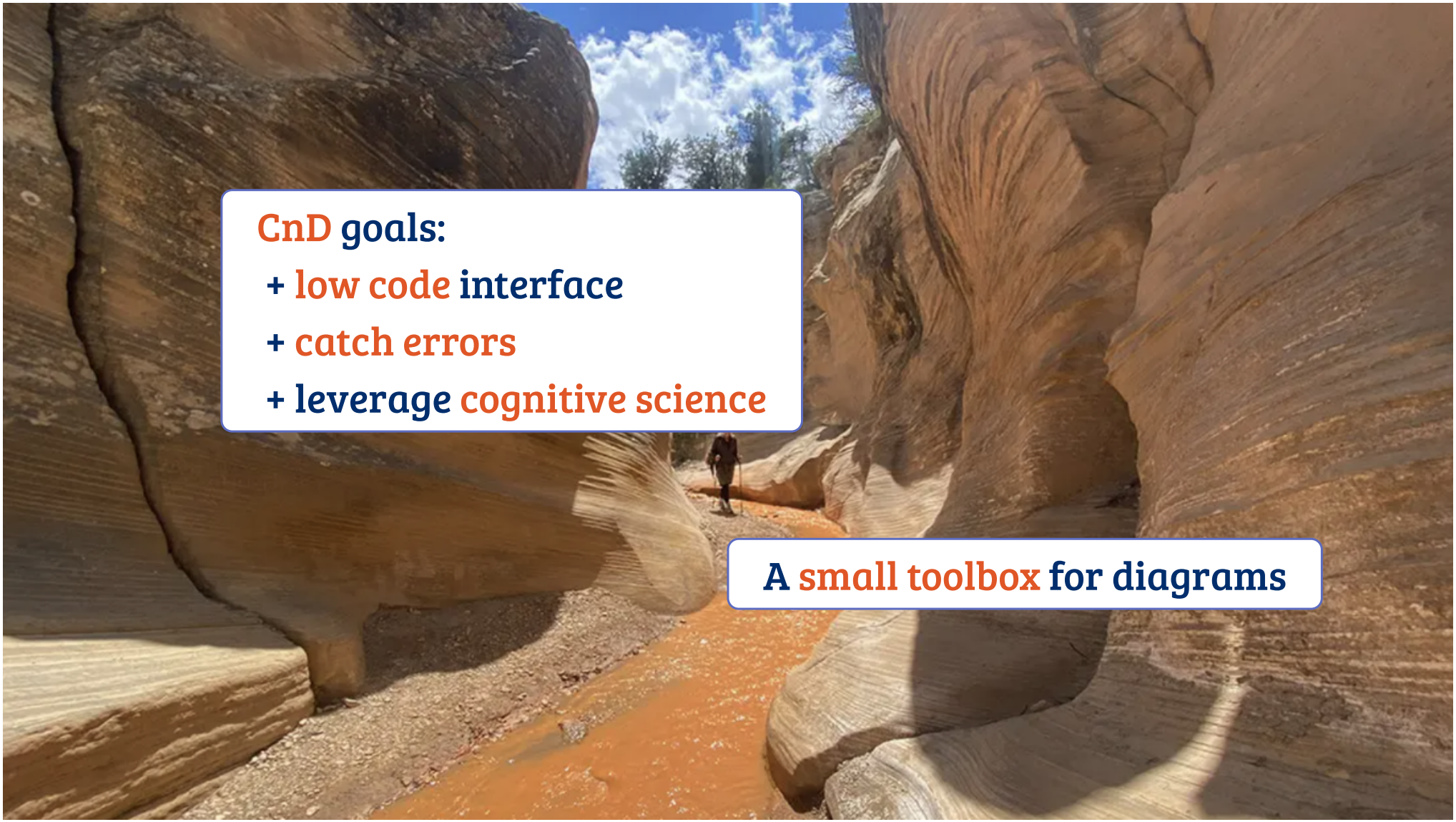
**Q. Anything wrong with this diagram?**

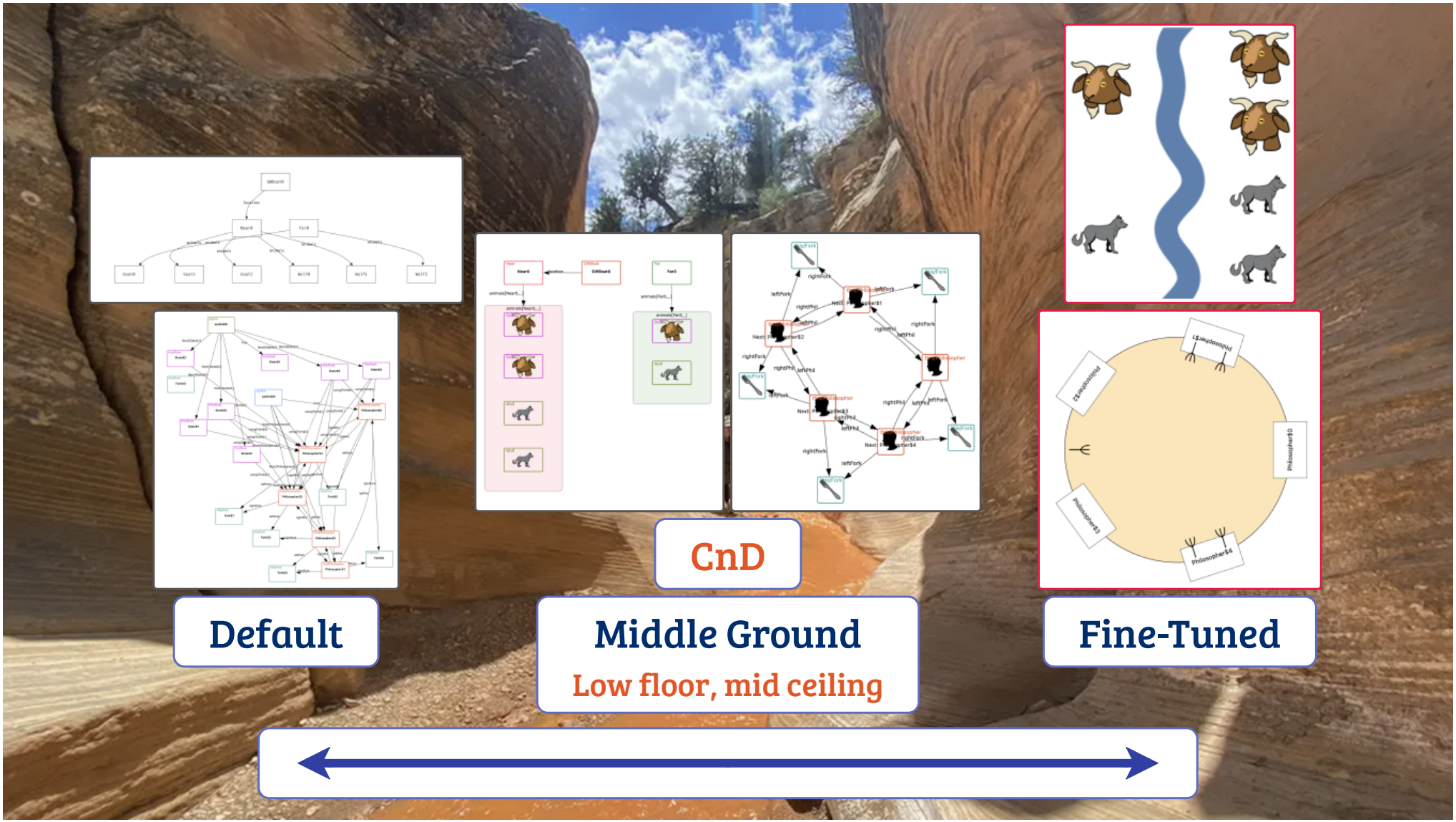



Binary Tree
-1
-2
Dining Philosophers
Philosopher$1
Philosopher$0
Philosopher$3
Philosopher$4
The problems are invisible
Diagrams look fine —
but the specifications are wrong!

**Problems:**

1. Programming required
   + order of magnitude
2. Distracts from modeling
3. Useless in development
4. Can fail silently


CnD goals:
+ low code interface
+ catch errors
+ leverage cognitive science
A small toolbox for diagrams

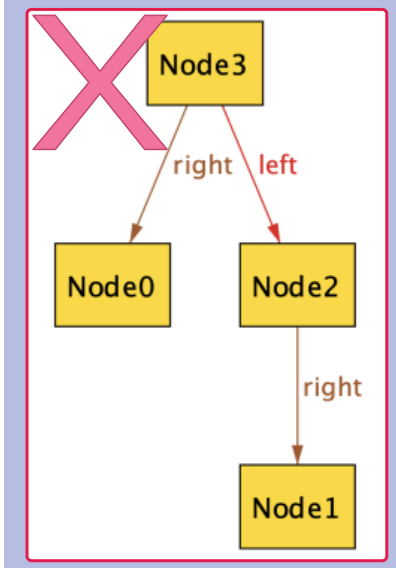

CnD
Default
Middle Ground
Low floor, mid ceiling
Fine-Tuned

# CnD, the language

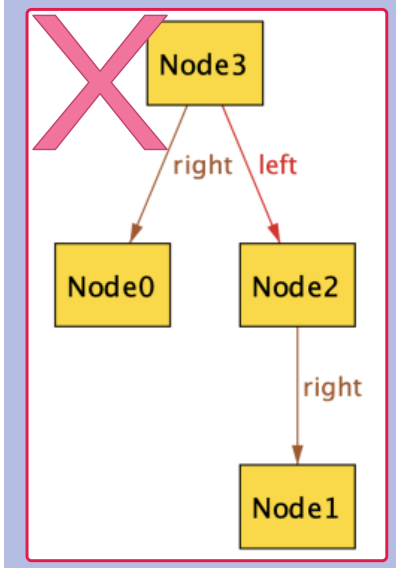
```
constraints:
  - constraint 1
  - constraint 2
  - ...

directives:
  - directive 1
  - directive 2
  - ...
```

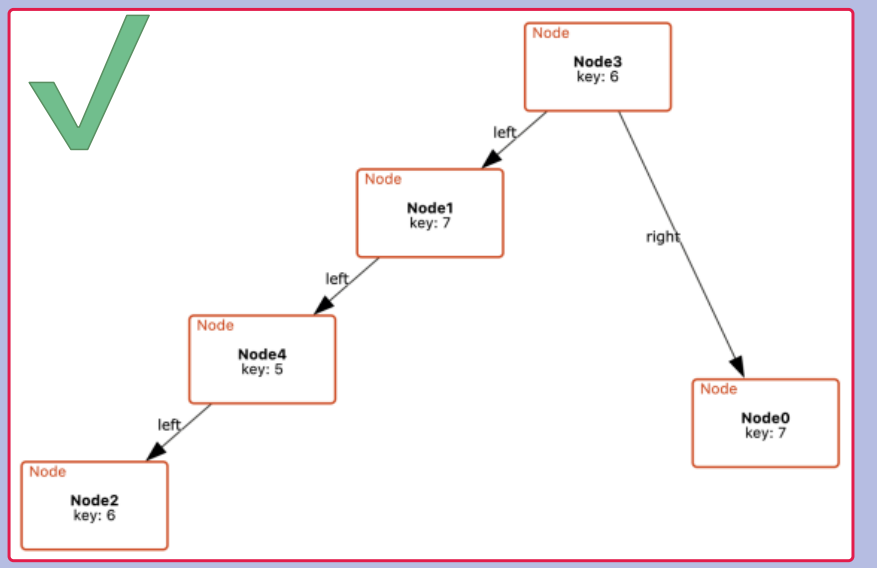
**A program is made of constraints and directives.**

# Orientation Constraint

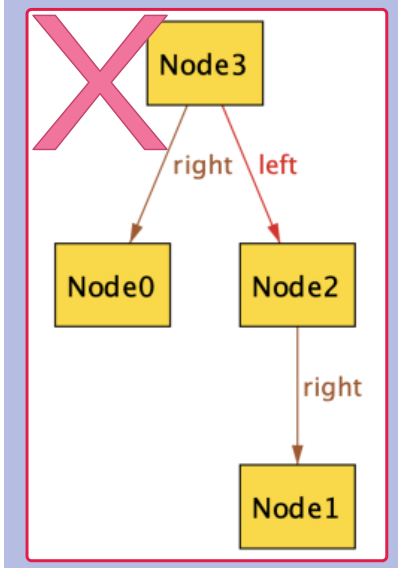


# Orientation Constraint

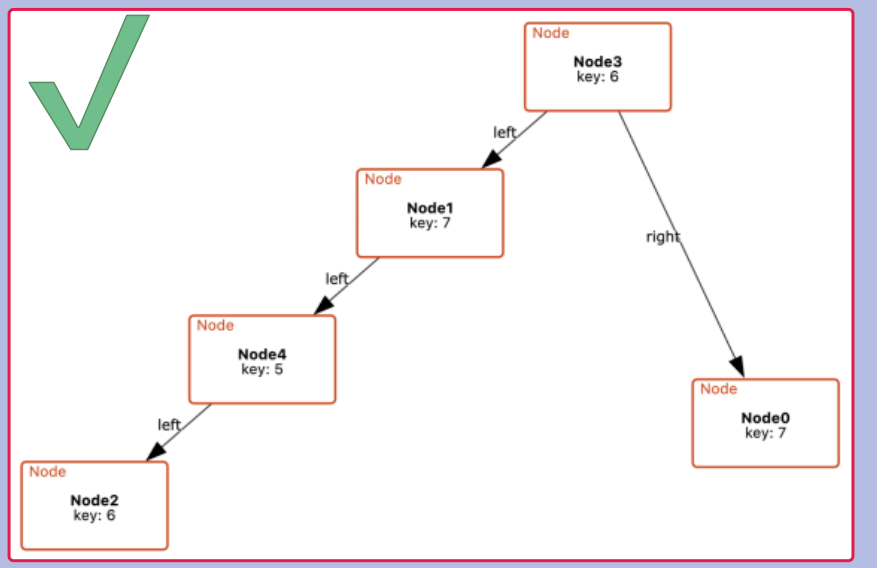


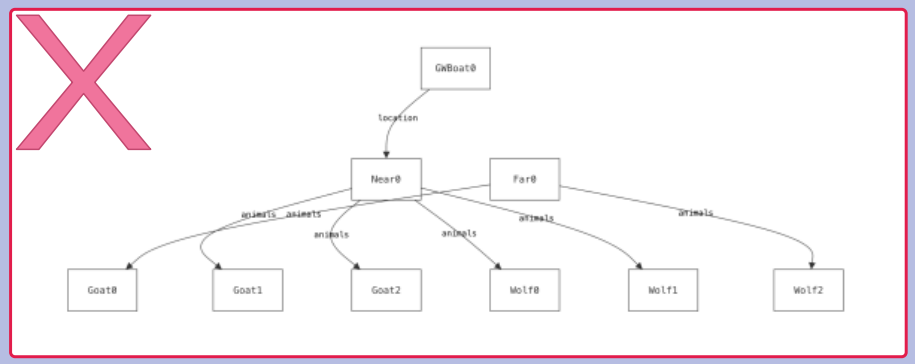
# Orientation Constraint



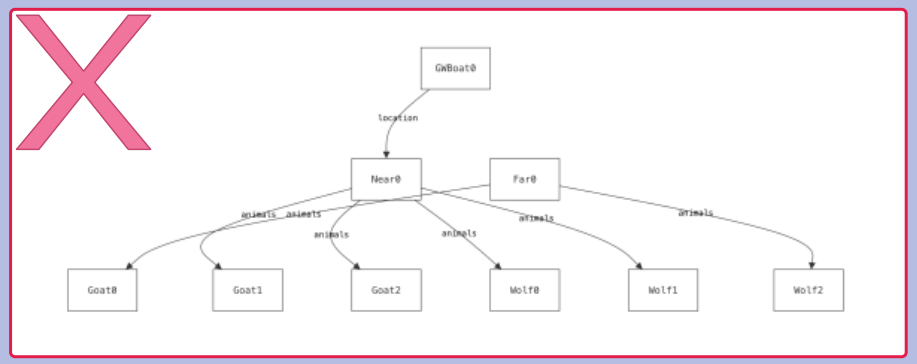

```
constraints:
  - orientation:
      field: left
      directions: [left, below]
  - orientation:
      field: right
      directions: [right, below]
```

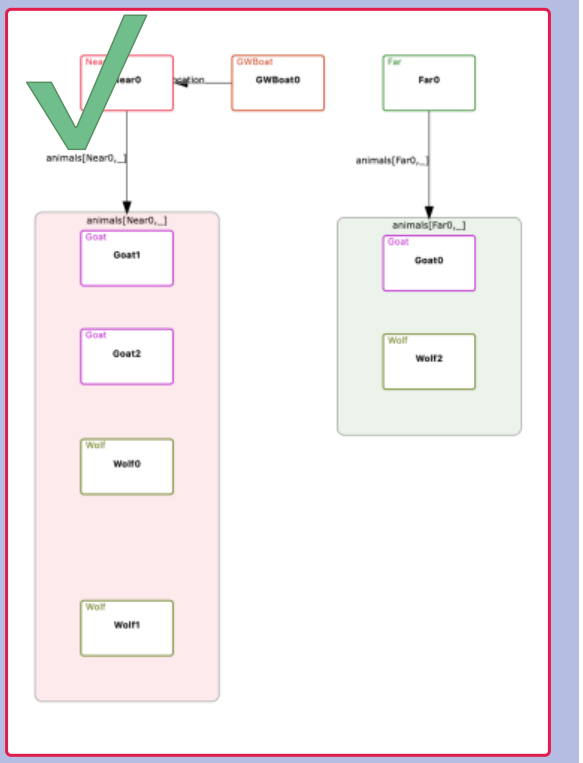


# Grouping Constraint

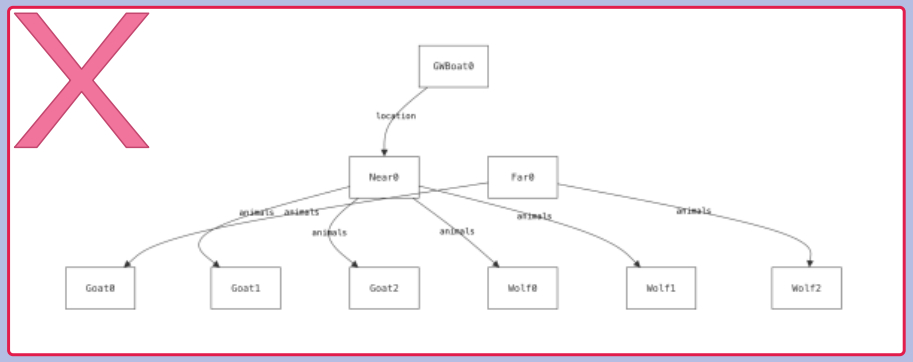


# Grouping Constraint

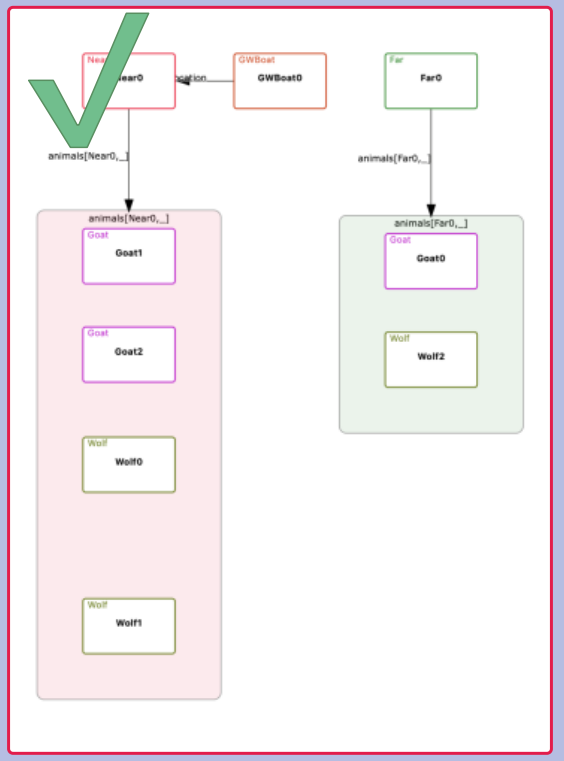


# Grouping Constraint



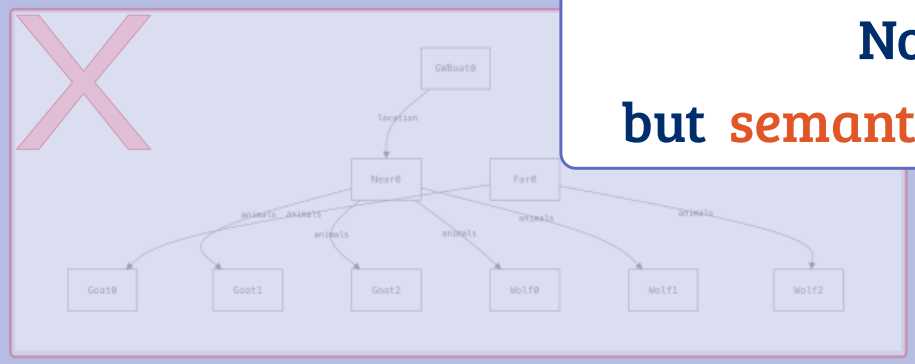

```
constraints:
  - group:
      field: animals
      target: domain
```



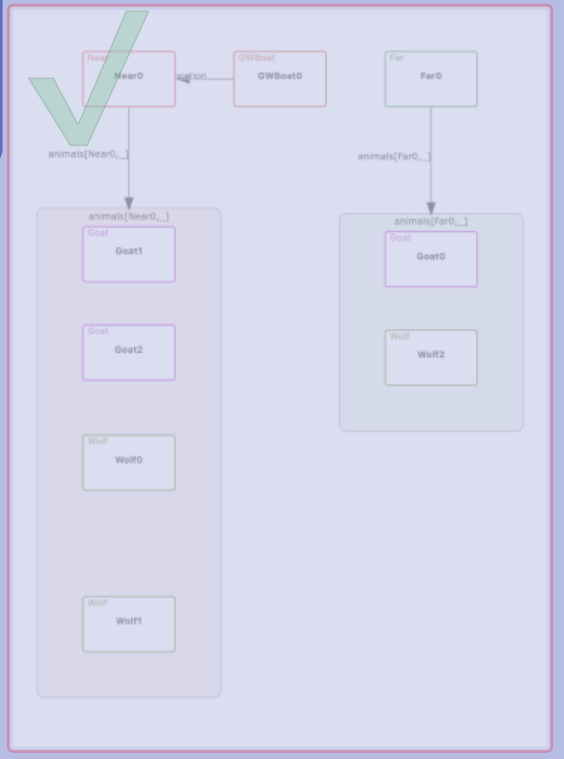
## Grouping Constraint

Not pretty,

but semantically meaningful!

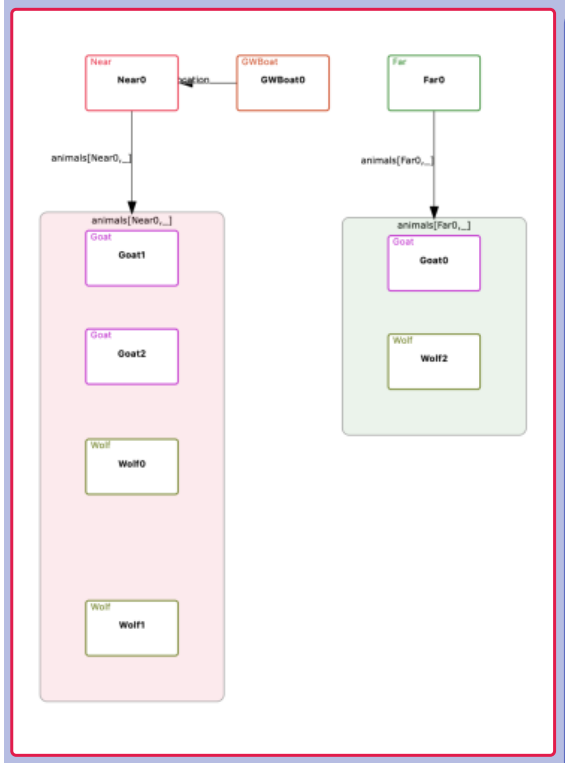



```
target: domain
```

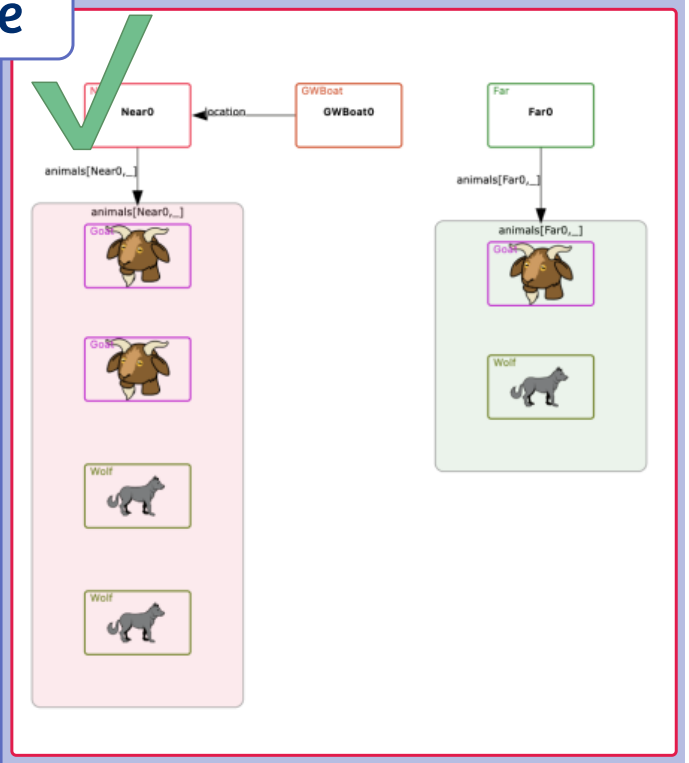


## Grouping Constraint

## + Icon Directive




```
...

directives:
  - icon:
      sig: Wolf
      icon:
        path: '/img/wolf.png'
        height: 70
        width: 70
  - icon:
      sig: Goat
      icon:
        path: '/img/goat.png'
        height: 70
        width: 70
  - flag: hideDisconnectedBuiltIns
```

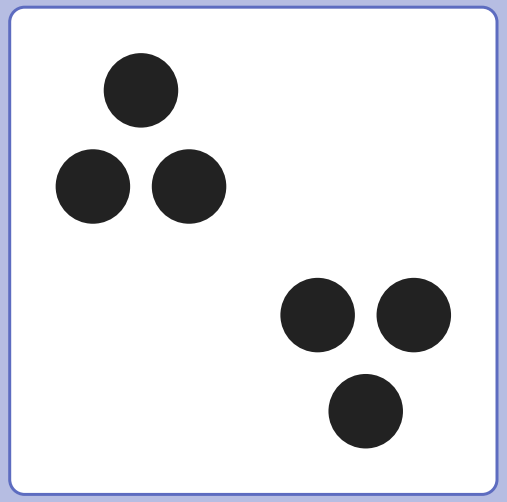


# Design Methods

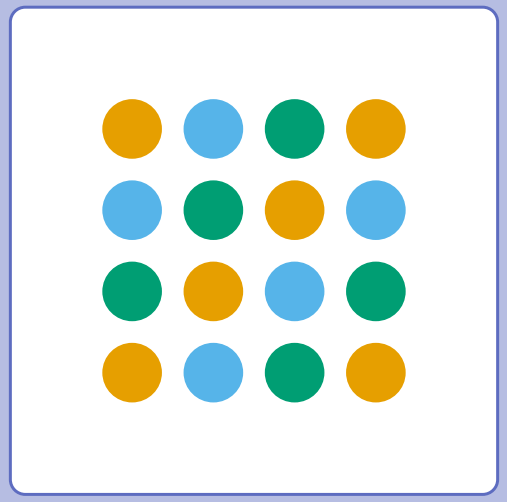
## Top-down and Bottom-up

Top-down I

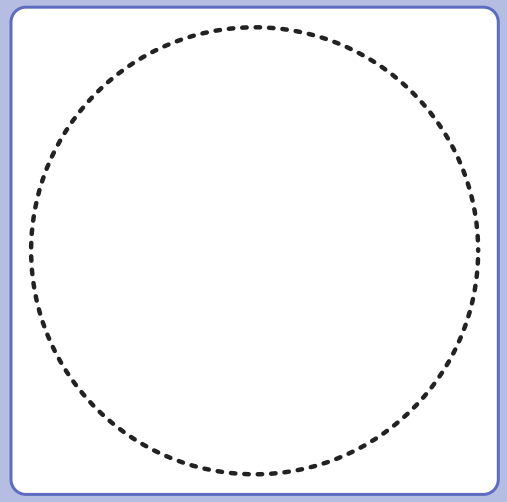
# Visual Principles



Proximity



Similarity



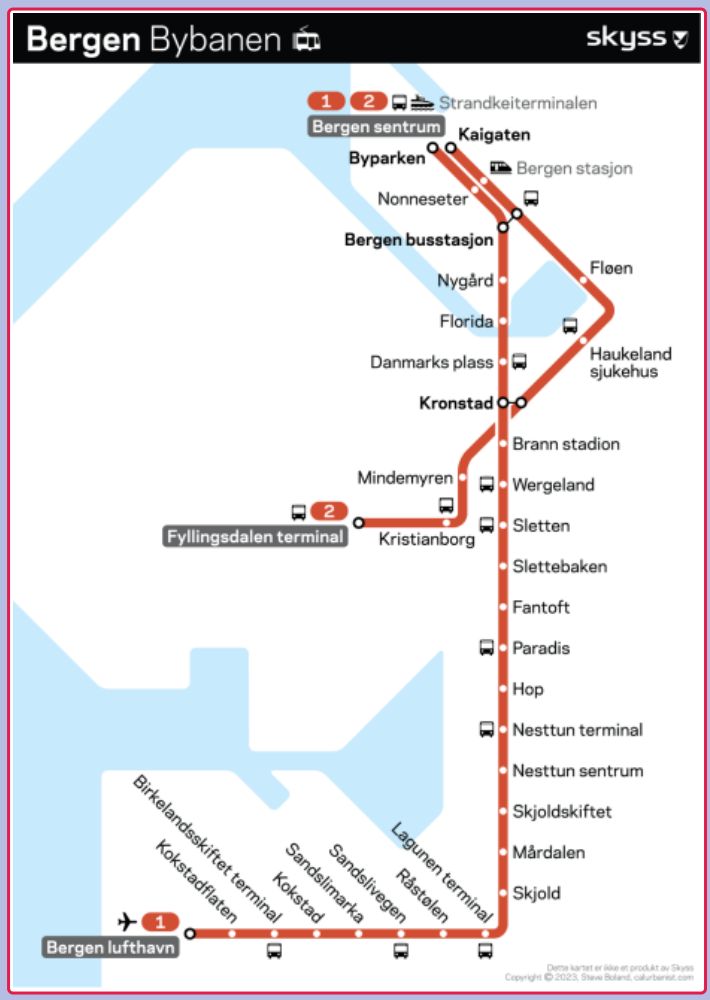
Closure

## Top-down II

# Diagramming Principles

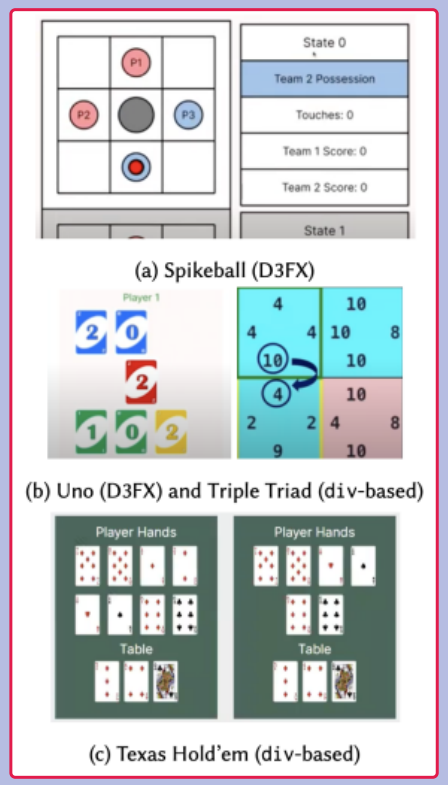
## Address the main task

**Subway = route planning, not geography**



Bottom-up

# Student Projects




(a) Spikeball (D3FX)

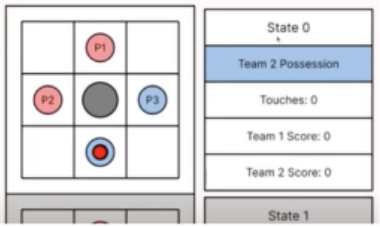
(b) Uno (D3FX) and Triple Triad (div-based)

(c) Texas Hold'em (div-based)

58 projects
2022 - 2024

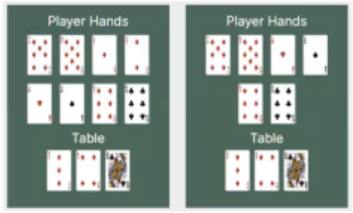
Bottom-up

# Student Projects




(a) Spikeball (D3FX)


(b) Uno (D3FX) and Triple Triad (div-based)




(c) Texas Hold'em (div-based)

58 projects
2022 - 2024

75% Representation Salience
70% Relative Positioning
33% Grouping
7% Directionality


Top-down + Bottom-up
in Sync
They're the same picture

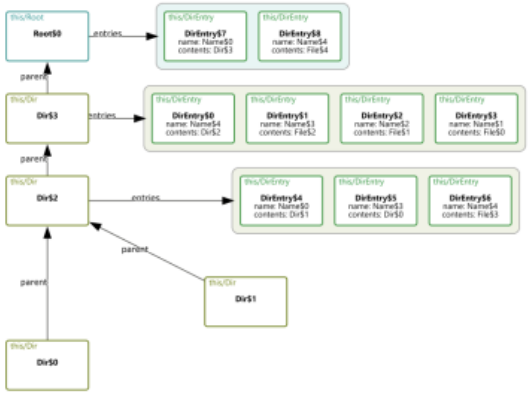
# Evaluation

(3 parts)

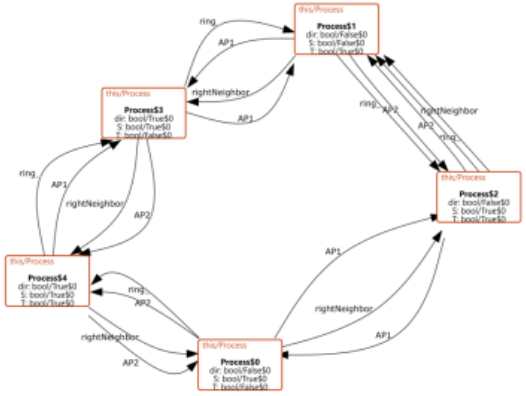
# Evaluation

(3 parts)

## I. Test Domains




(a) filesystem.




(b) ring-orientation.


(c) chord.

## Evaluation (3 parts)

## II. User studies on Prolific

+ Understanding instances
+ Understanding specifications
+ Finding wrong specifications

## Evaluation (3 parts)

## II. User studies on Prolific

+ Understanding instances
+ Understanding specifications
+ Finding wrong specifications

**Almost always, CnD > default**

Stats in paper

# Evaluation

(3 parts)

## III. Performance

## Evaluation (3 parts)

### III. Performance

20 benchmark programs, 50 runs each

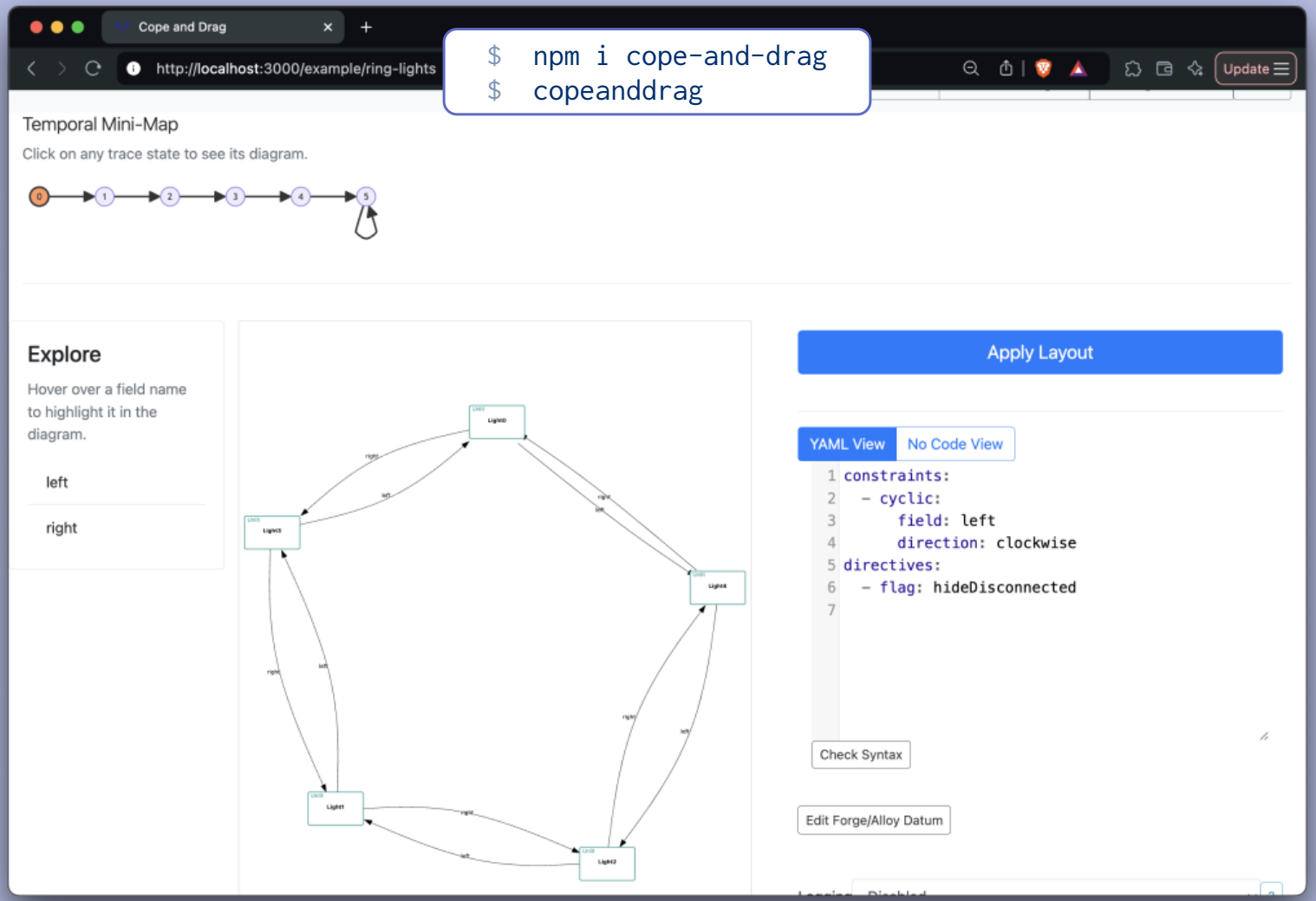
Median:

30ms for constraint solving

432ms for graph layout

Details in appendix

# Conclusion


Cope and Drag
http://localhost:3000/example/ring-lights
$ npm i cope-and-drag
$ copeanddrag
Update
Temporal Mini-Map
Click on any trace state to see its diagram.
0
1
2
3
4
5
Explore
Hover over a field name to highlight it in the diagram.
left
right
Apply Layout
YAML View
No Code View
1 constraints:
2 - cyclic:
3 field: left
4 direction: clockwise
5 directives:
6 - flag: hideDisconnected
7
Check Syntax
Edit Forge/Alloy Datum

CnD

+ domain-specific diagramming
+ grounded in cognitive science

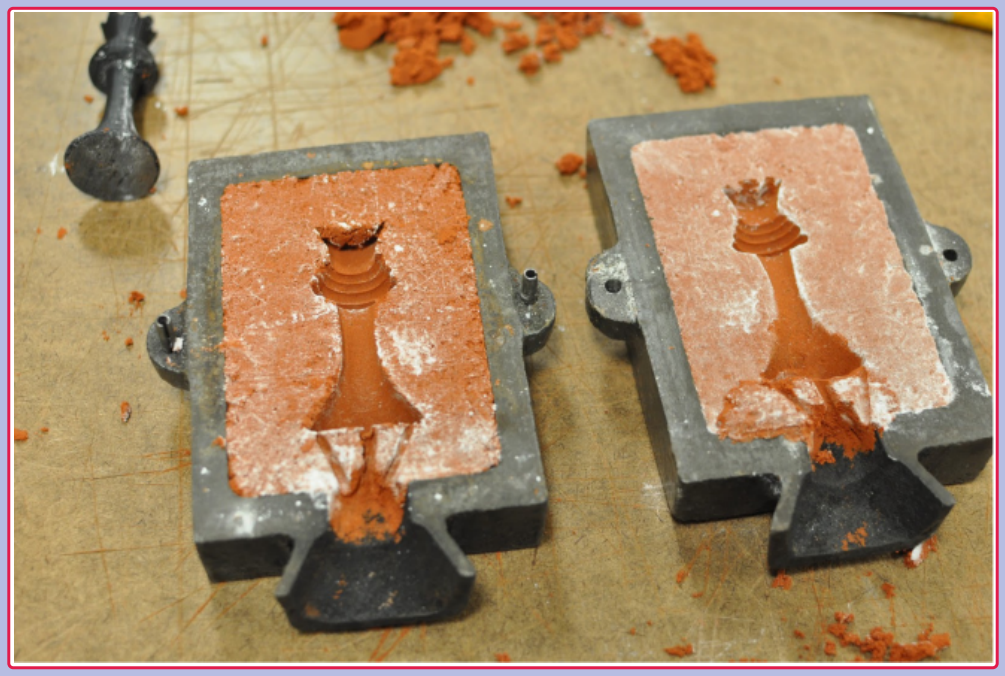
Useful but not pretty

Diagramming by refinement

Low floor, Mid ceiling

Docs: https://tinyurl.com/copeanddrag

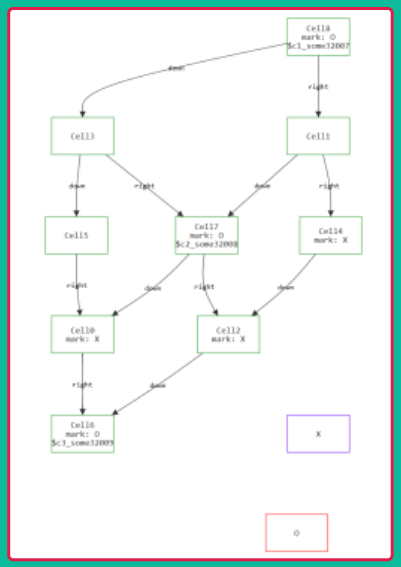
CnD == Cope and Drag

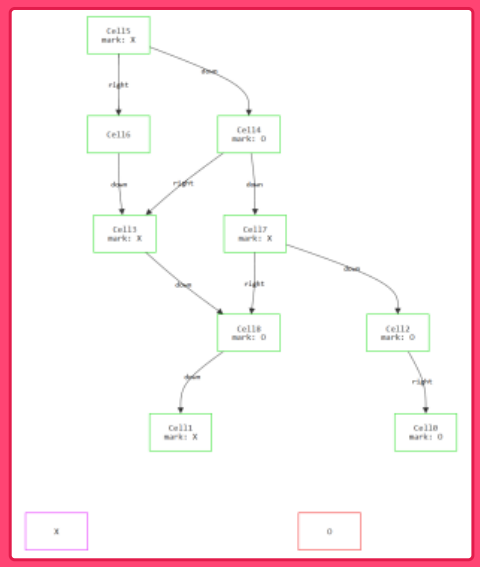


https://frankieflood.blogspot.com/2014/05/readymake-sand-molds.html

+ Finding wrong specifications

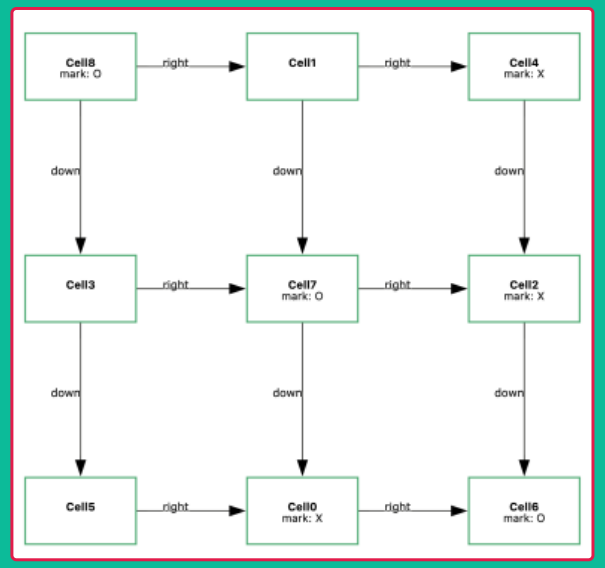
Default

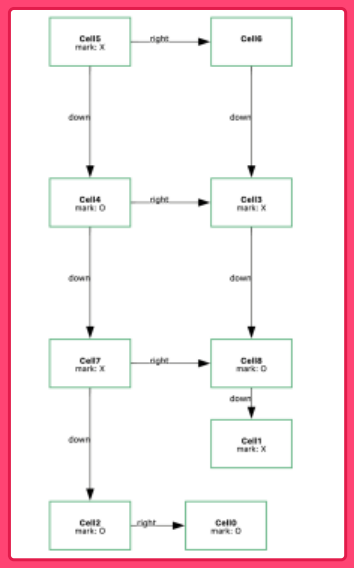


+ Finding wrong specifications

CnD



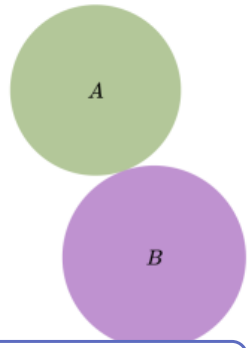
# Penrose, silent failure

https://penrose.cs.cmu.edu/

.substance .style .domain

```
1    Set A, B
2
3    Intersecting(A, B)
4    Disjoint(A, B)
5
6    AutoLabel All
7
```

Diagram Diagram Variations




**Unsatisfiable spec, but diagram appears**

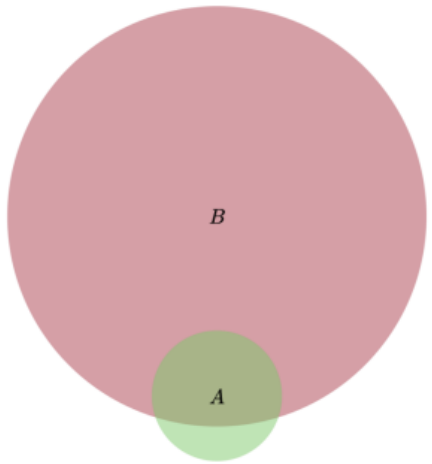
## Penrose = for presentation, not exploration

https://penrose.cs.cmu.edu/

.substance .style .domain

```
1    Set A, B
2    Subset(A, B)
3    Intersecting(A, B)
4
5    AutoLabel All
6
```

Diagram Diagram Variations




Style file assumes intersecting ==> disjoint

# CnD Grammar

**Program** ::= Constraint* Directive*

**Constraint** ::= CyclicConstraint
| OrientationConstraint
| GroupingConstraint

**CyclicConstraint** ::= Field FlowDirection

**OrientationConstraint** ::= Field Direction+
| Sig Direction+

**GroupingConstraint** ::= Field Target

**FlowDirection** ::= clockwise | counterclockwise

**Direction** ::= above | below | left | right
| directlyAbove | directlyBelow
| directlyLeft | directlyRight

**Target** ::= domain | range

**Directive** ::= PictorialDirective
| ThemingDirective

**PictorialDirective** ::= Sig Icon

**ThemingDirective** ::=
Attribute | Color
| Projection | VisibilityFlag

**Icon** ::= path height width

**Attribute** ::= Field

**Color** ::= SigName color

**Projection** ::= Sig

**VisibilityFlag** ::= hideDisconnected
| hideDisconnectedBuiltIns

CnD

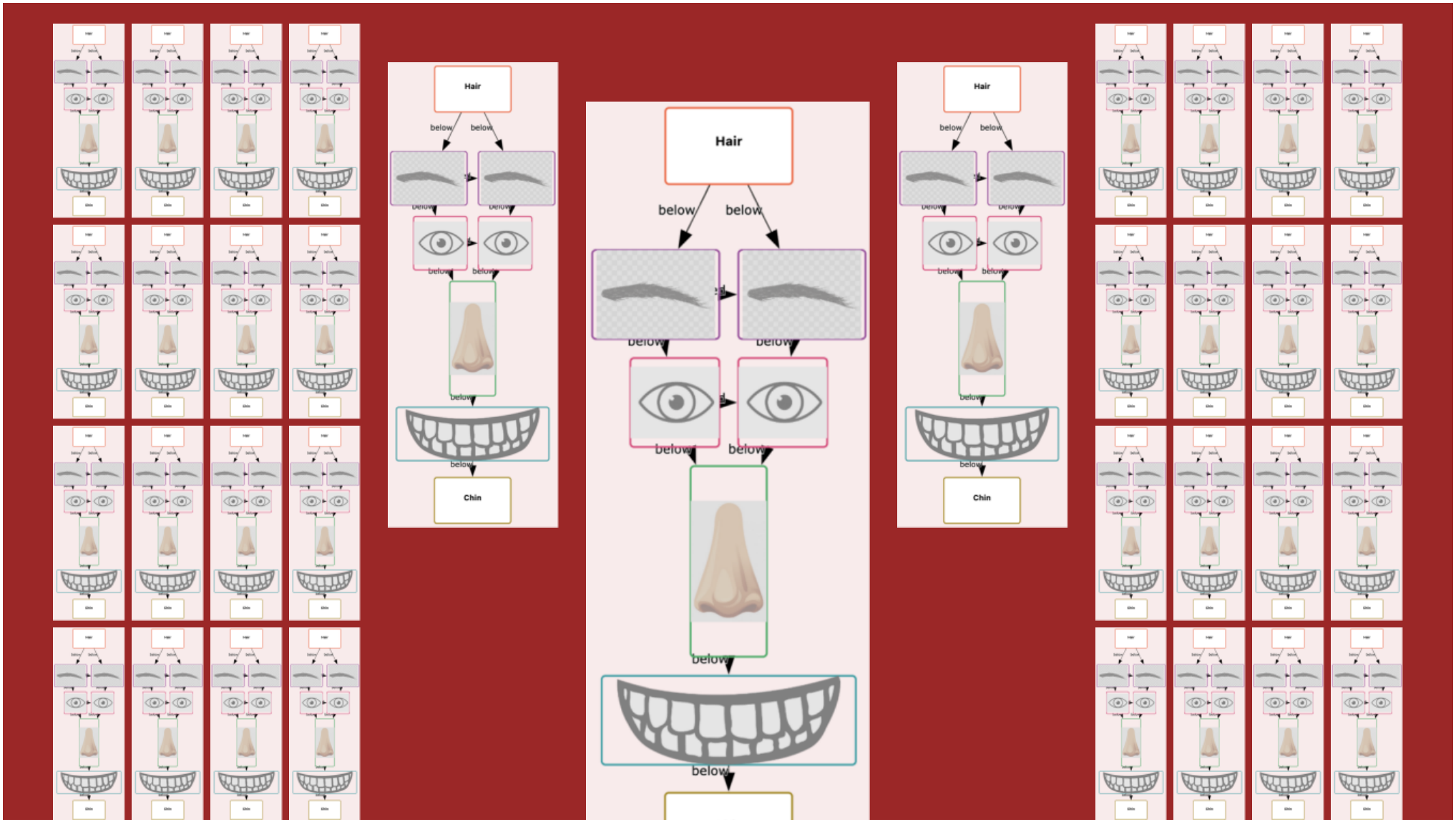
+ domain-specific diagramming
+ grounded in cognitive science

Useful but not pretty

Diagramming by refinement

Low floor, Mid ceiling

Docs: https://sidprasad.github.io/copeanddrag


Hair
below
below
below
below
below
below
below
below
Chin