22nd International Symposium on Graph Drawing
24-26 September 2014, Würzburg, Germany

# Anchored Drawings of Planar Graphs

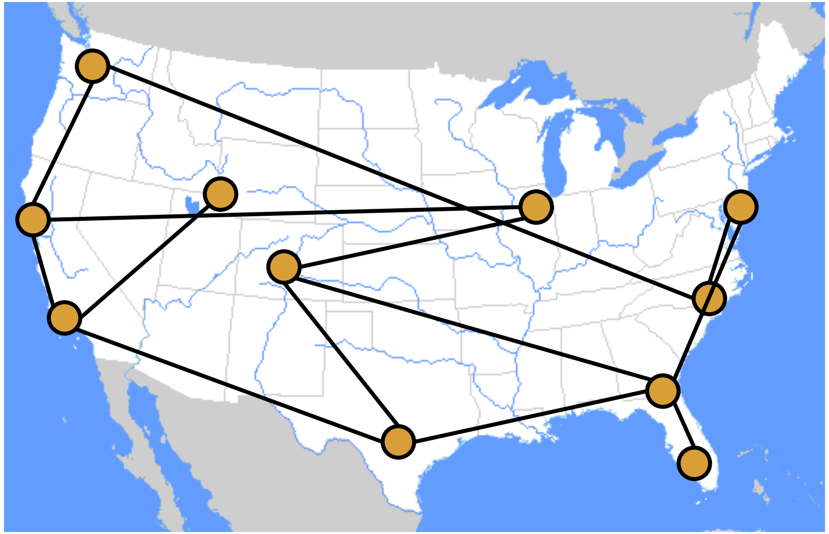
Angelini, Da Lozzo, Di Bartolomeo,
Di Battista, Hong, Patrignani, Roselli

ROMA TRE UNIVERSITÀ DEGLI STUDI

THE UNIVERSITY OF SYDNEY

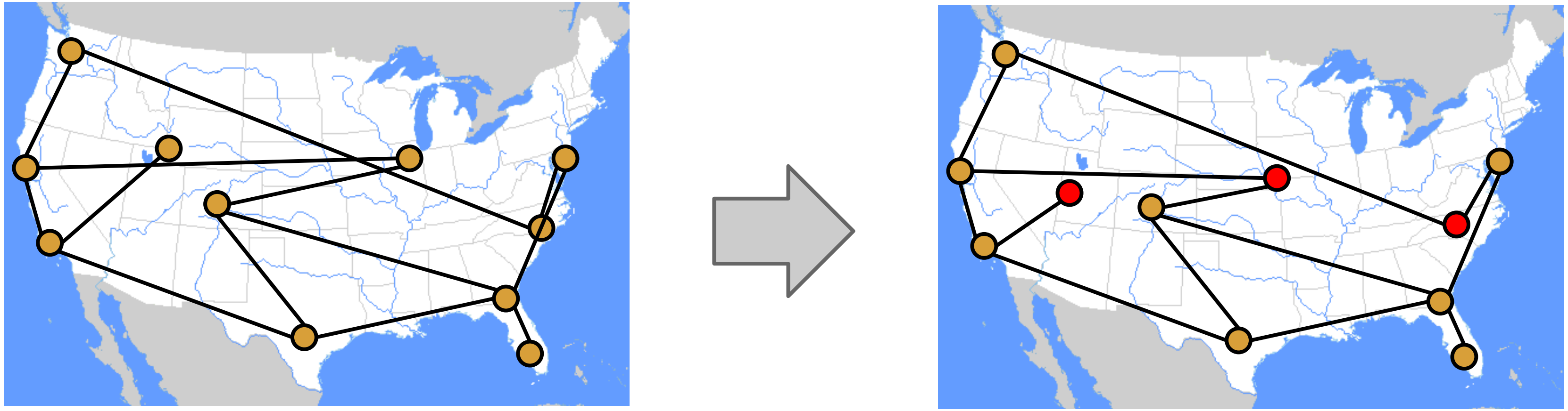
# Applicative Context

- Drawing a graph on a geographical map
- Vertices have fixed positions



# Drawing Nicely

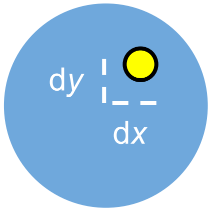
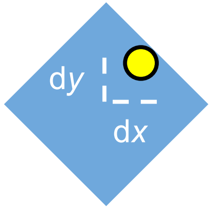
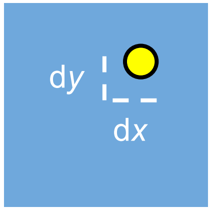
- Our idea:
  - Let vertices move “a bit” around their positions
  - Check if this allows a planar drawing of the graph



# Anchored Graph Drawing Problem

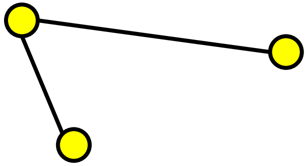
- **Instance**
  - Planar graph $G$
  - Initial vertex positions $\alpha(v)$
  - Maximum distance $\delta$
- **Question**
  - Does G admits a planar drawing
  - ...such that vertices move by distance at most $\delta$
  - ...from their initial positions $\alpha$?

# Considered Settings

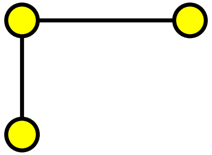
| | | | |
|---|---|---|---|
| **Distance Function** | "Euclidean" $d = (dx^2 + dy^2)^{1/2}$ | "Manhattan" $d = dx + dy$ | "Uniform" $d = \max(dx,dy)$ |
| **Vertex Region** |   |   |   |

**Drawing Style**

Straight-line



Rectilinear

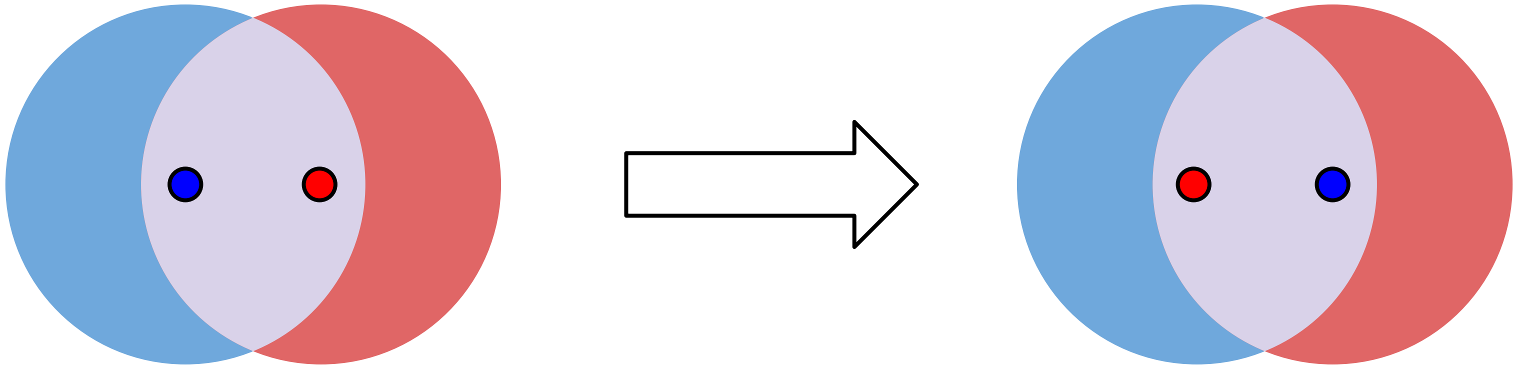


# Previous work

- NP-hard: straight-line and disks of different size
    - Godau. *On the difficulty of embedding planar graphs with inaccuracies*. 1995
- NP-hard: rectilinear and $\delta$ = inf
    - Garg, Tamassia. *On the comp. compl. of upward and rectilinear planarity test*. 2001
- Application of force-directed algorithms
    - Abellanas et. al. *Network drawing with geographical constraints on vertices*. 2005
- Iterative adjustments that preserve mental map
    - Lyons et. al. *Algorithms for cluster busting in anchored graph drawing*. 1998

# Assumption

- No overlap between **vertex regions**
  - Or two vertices may invert their positions
    - Very confusing for a user
  - Relationship with Clustered Planarity with drawn clusters



# Our Results

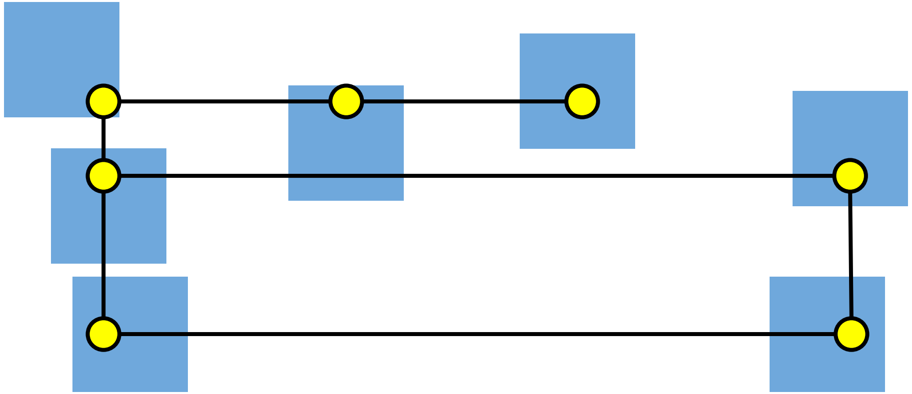
| Metric | Straight-line | Rectilinear |
|---|---|---|
| Manhattan | NP-hard | NP-hard |
| Euclidean | NP-hard | NP-hard |
| Uniform | NP-hard | Polynomial |

# Our Results

| Metric | Straight-line | Rectilinear |
|---|---|---|
| Manhattan | NP-hard | NP-hard |
| Euclidean | NP-hard | NP-hard |
| Uniform | NP-hard | Polynomial |

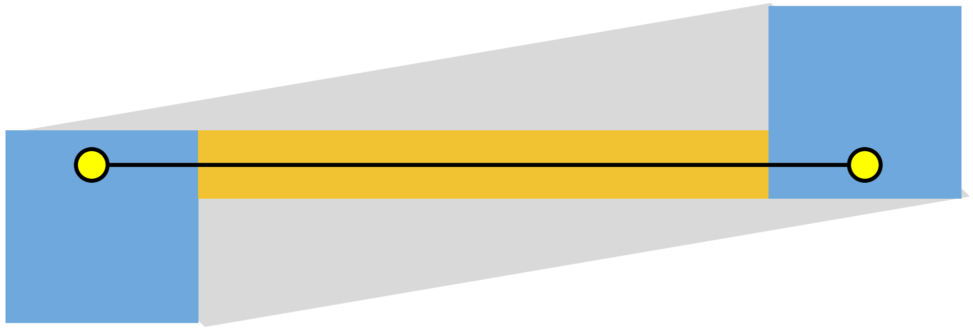
# Polynomial Case

- Connected graph
- Uniform distance (  regions)
- Rectilinear drawing



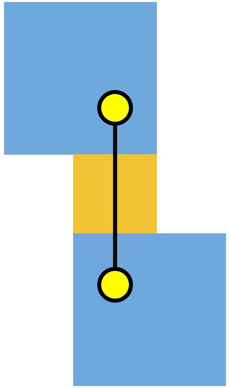
# Edge Pipes

- We call **pipe** the convex hull of two regions
  - Minus the regions
- An edge can be drawn only inside a pipe
- In this setting pipes “get rectilinear” too



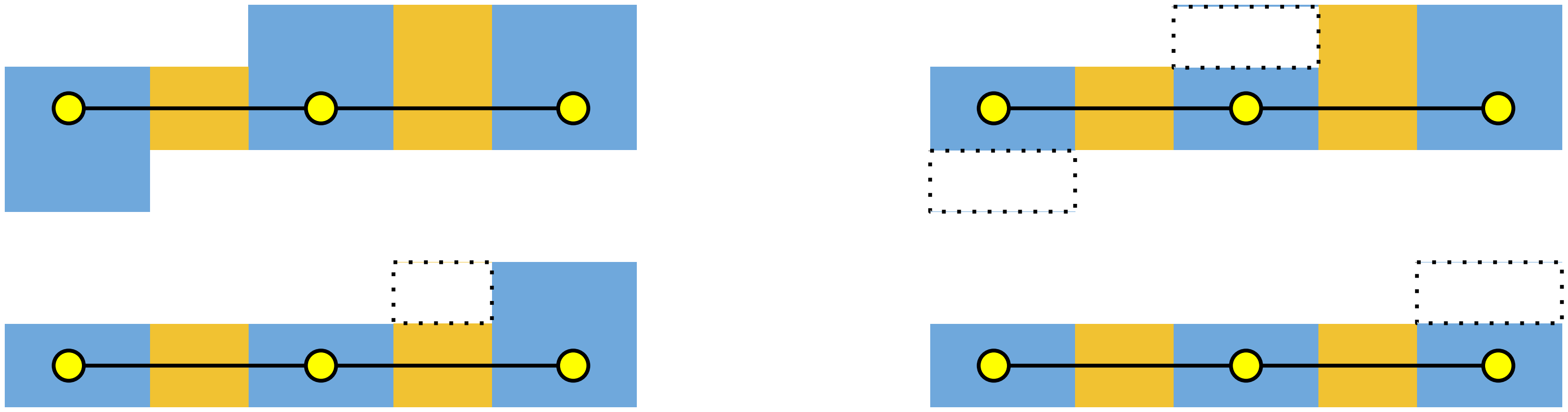
# Rectilinear Edges

- An edge is either **horizontal** or **vertical**
- Can be deduced by the region positions
- **Visibility** is required between two endpoints



# Trimming

- Regions and pipes can trim each other
- A trimmed area cannot be used

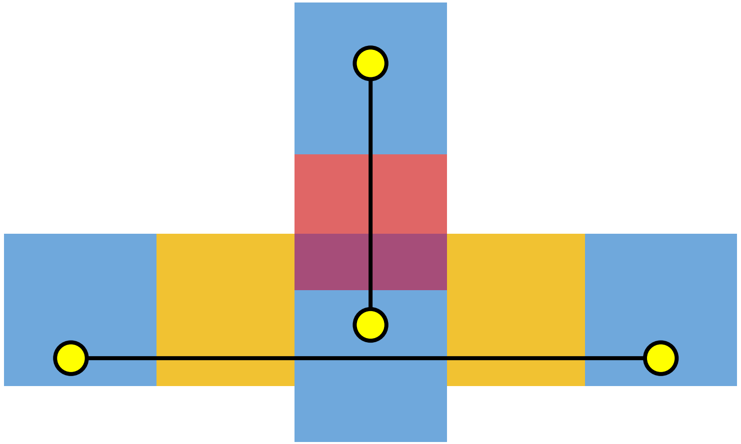


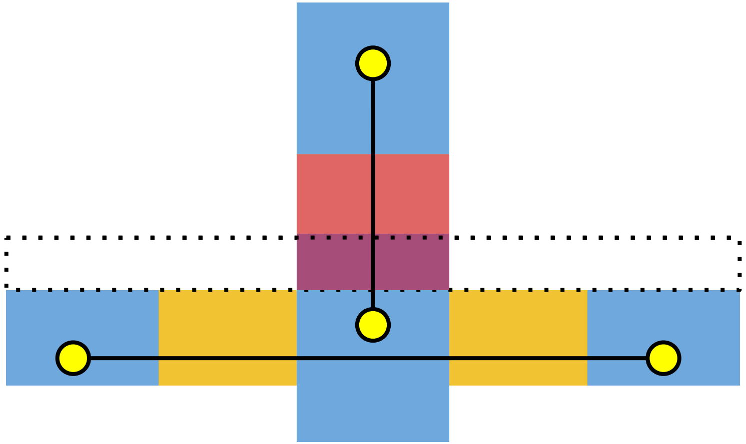
# General Strategy

1. Start from the initial region/pipe configuration
2. While (a trim is possible):
    a. Trim unusable parts of pipes and regions
    b. Check if a negative configuration is obtained
3. Flag the instance as positive
4. Draw edges according to the current pipes

# Trimming Pipes

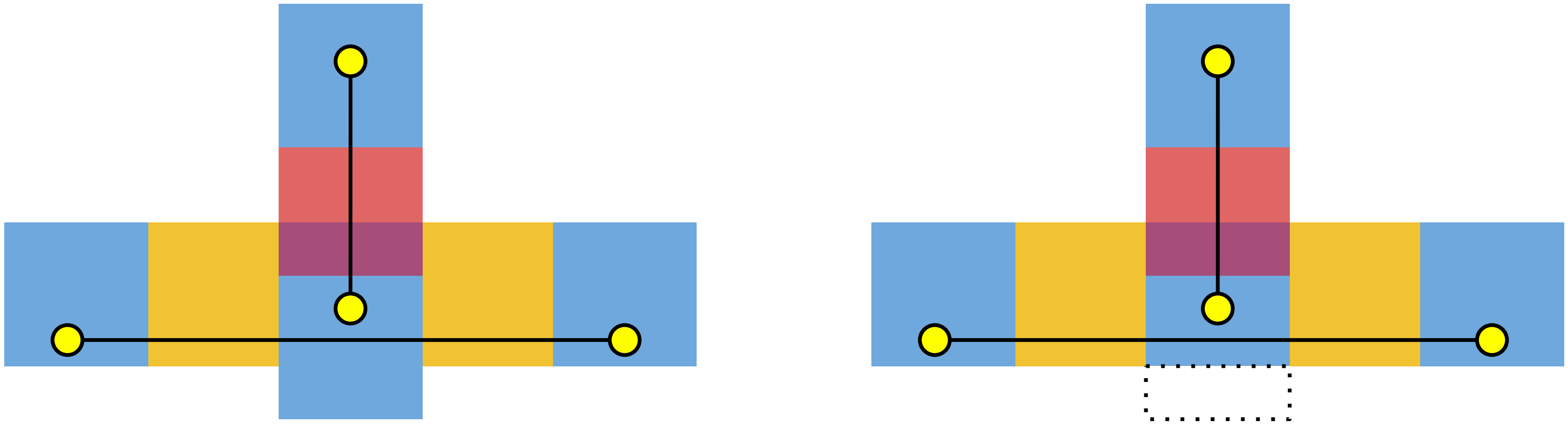
- *VP-overlaps* can trim a pipe





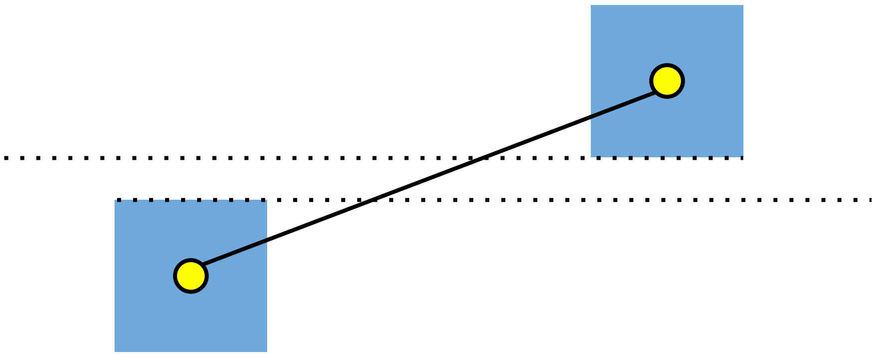
# Trimming Regions

- *VP-overlaps* can trim a region

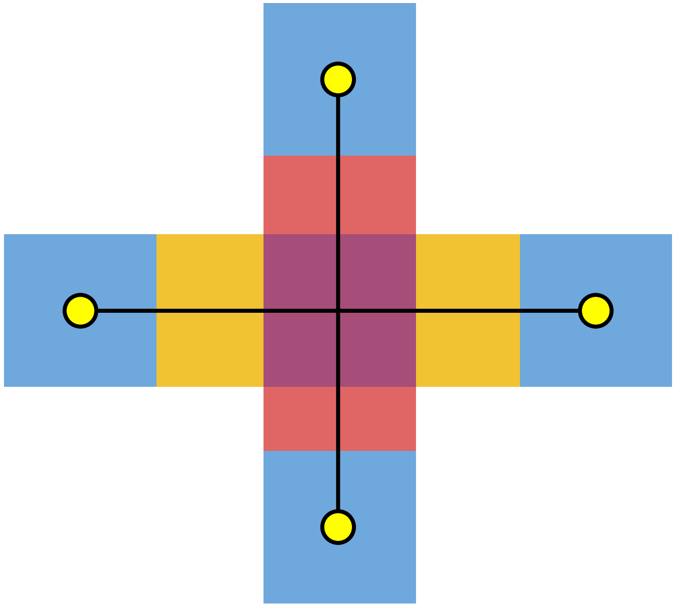


# Negative Instances

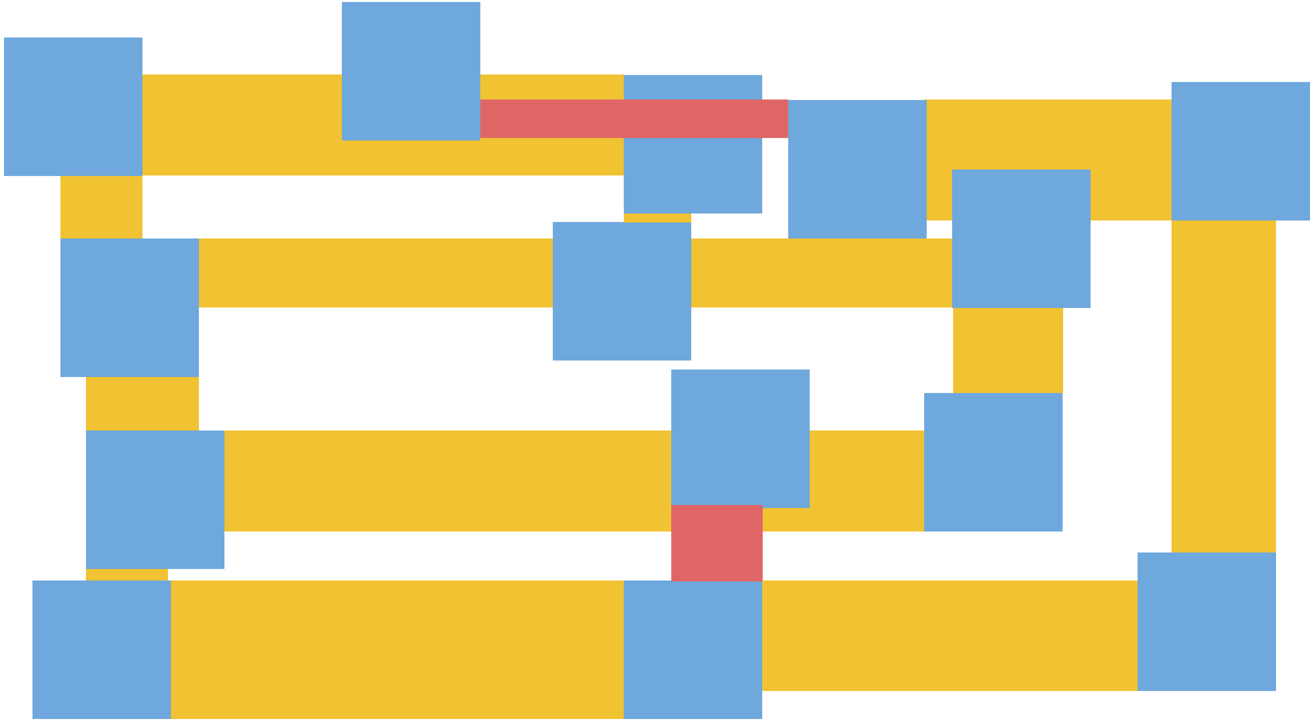
No visibility



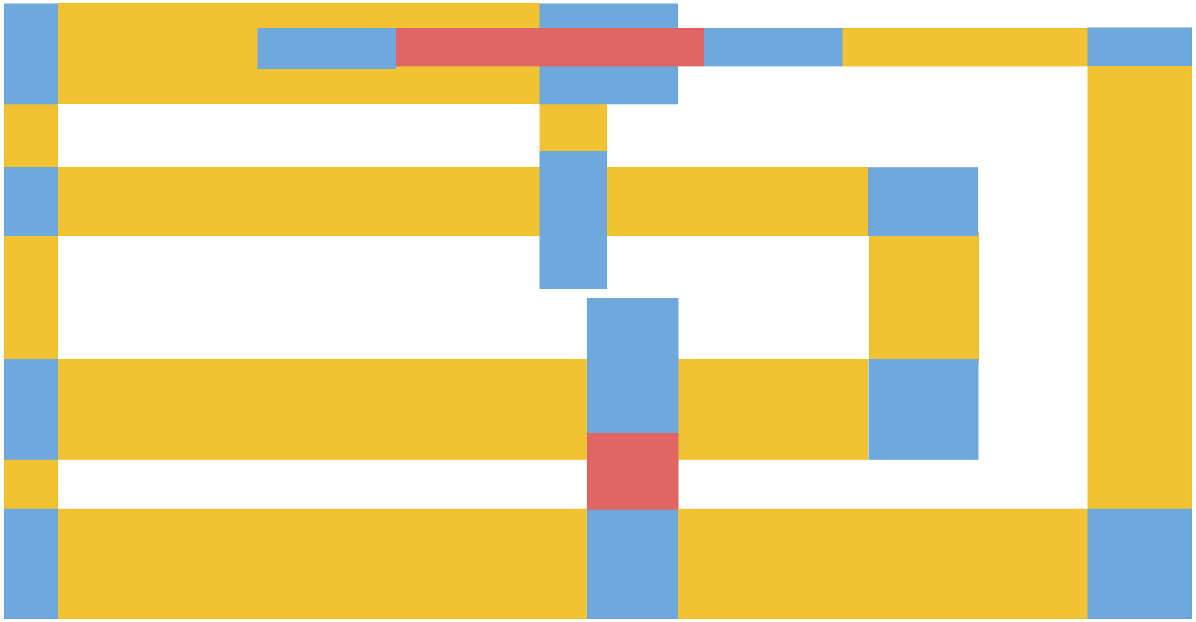
PP-overlap
(Unavoidable crossing)



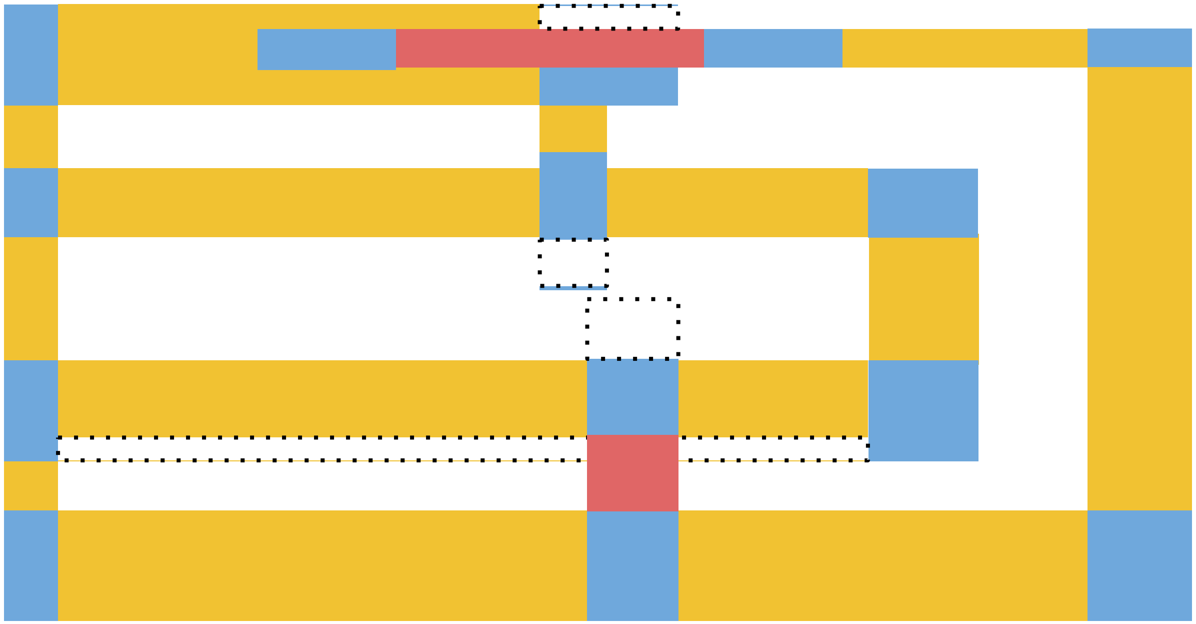
# An Example of Execution



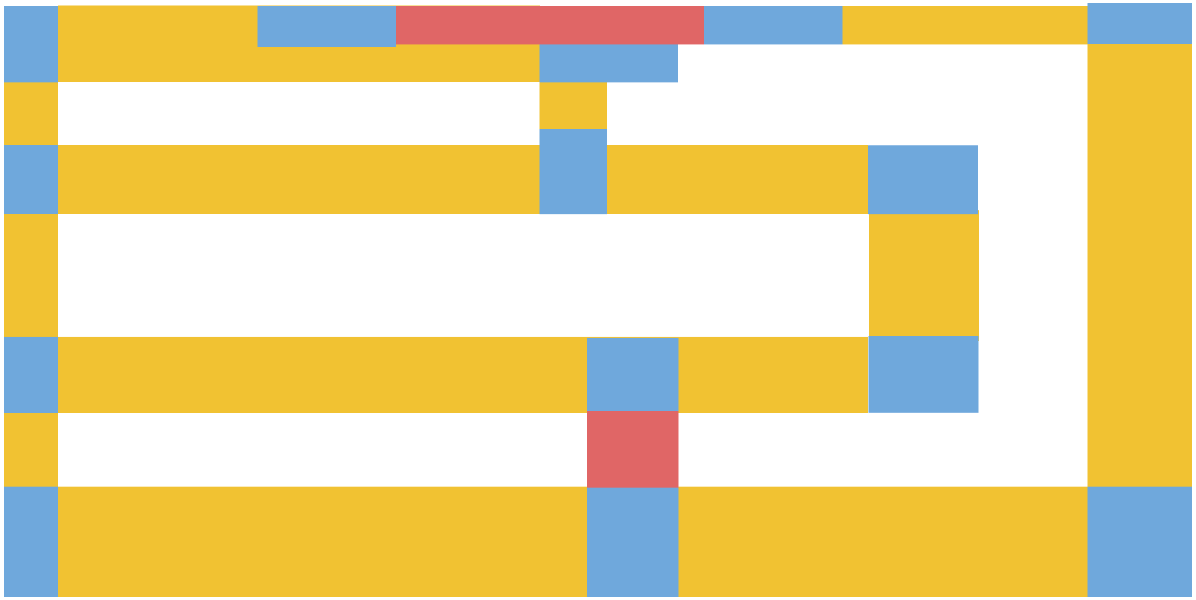
# An Example of Execution



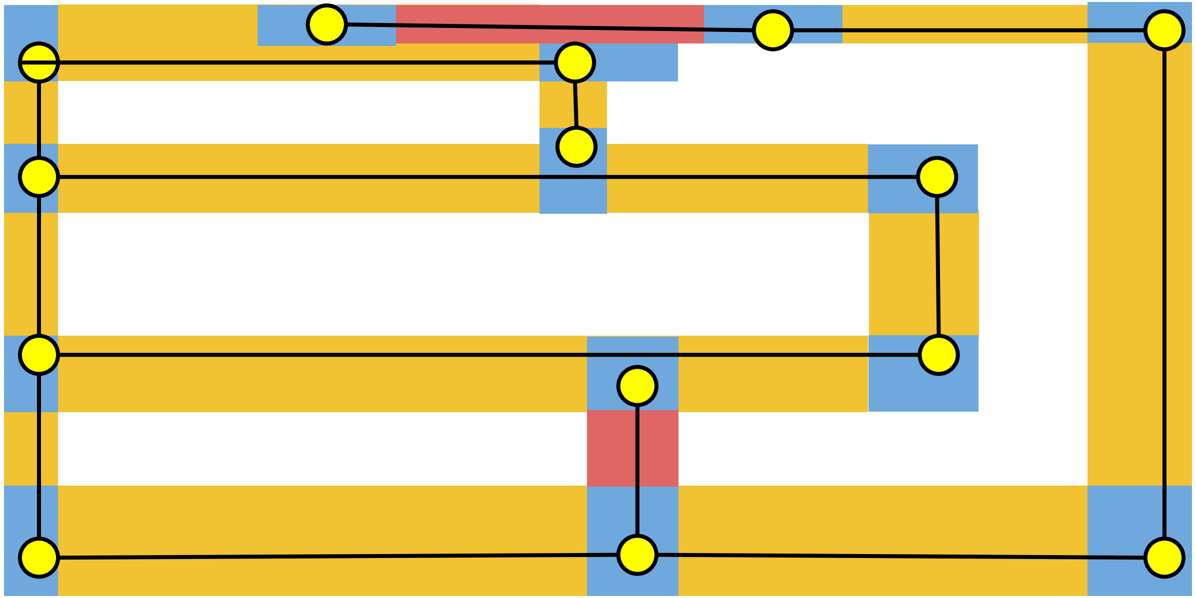
# An Example of Execution



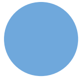
# An Example of Execution

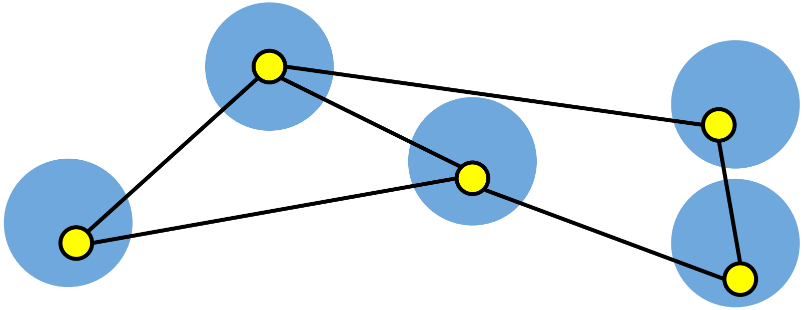


# An Example of Execution



# NP-hard Case

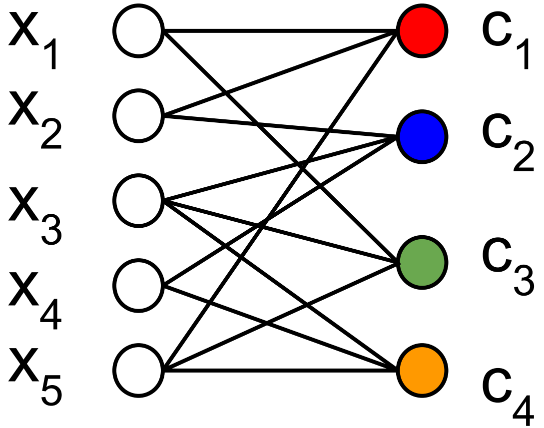
- Euclidean distance (  regions)
- Straight-line drawing
- Reduction from *Planar 3-SAT*

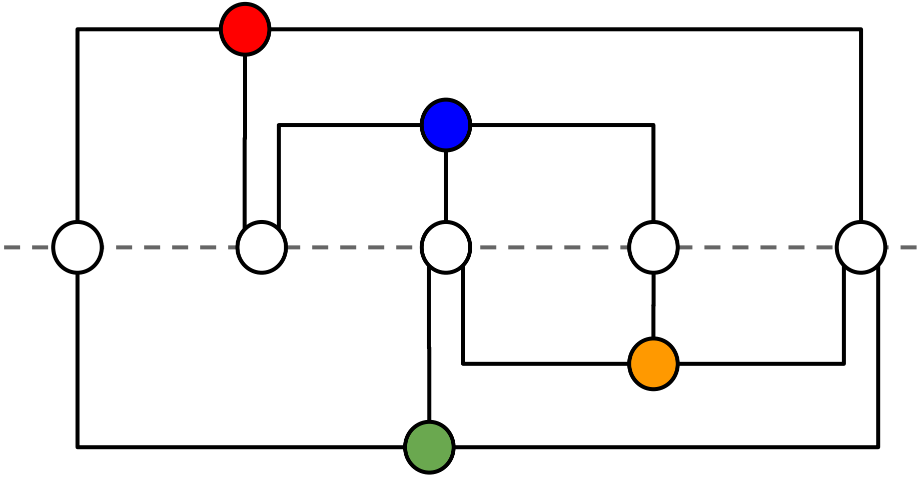


# Planar 3-SAT

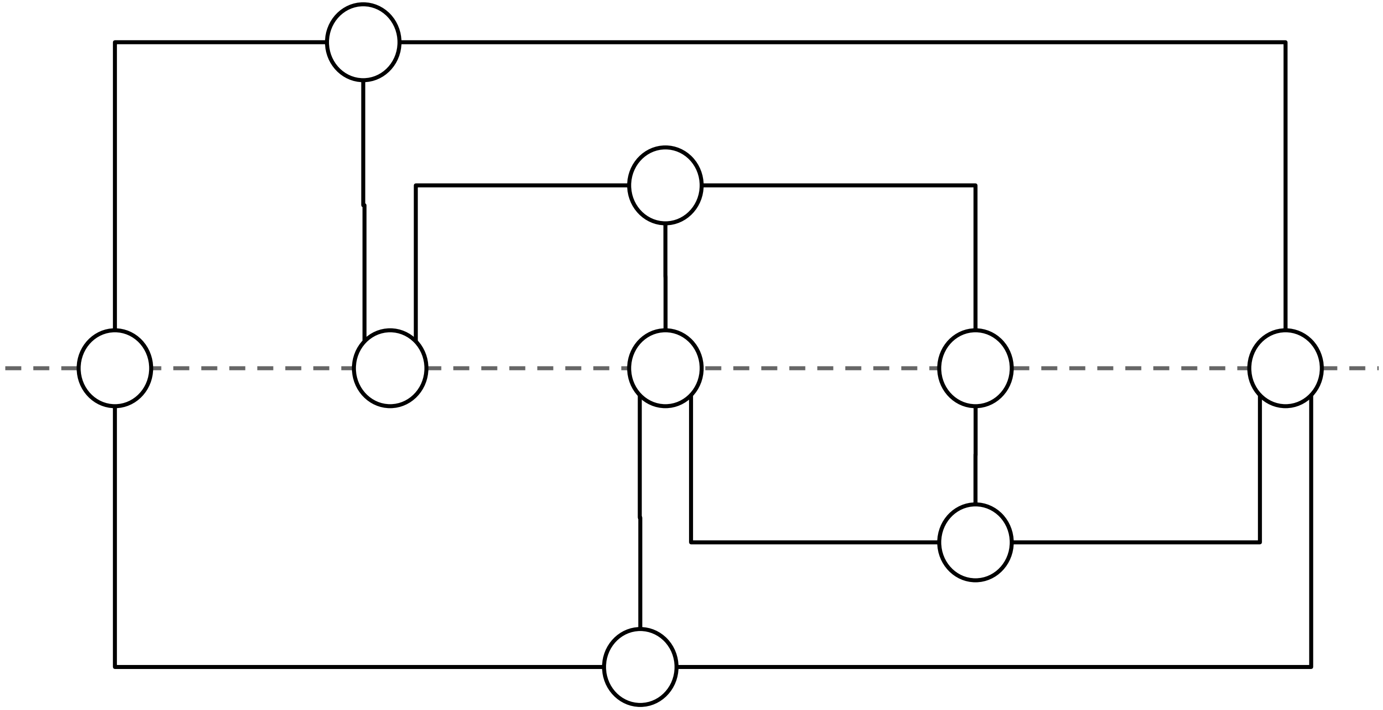
$(x_1 \vee \neg x_2 \vee x_5) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_3 \vee x_5) \wedge (x_3 \vee x_4 \vee x_5)$

$C_1$ $C_2$ $C_3$ $C_4$

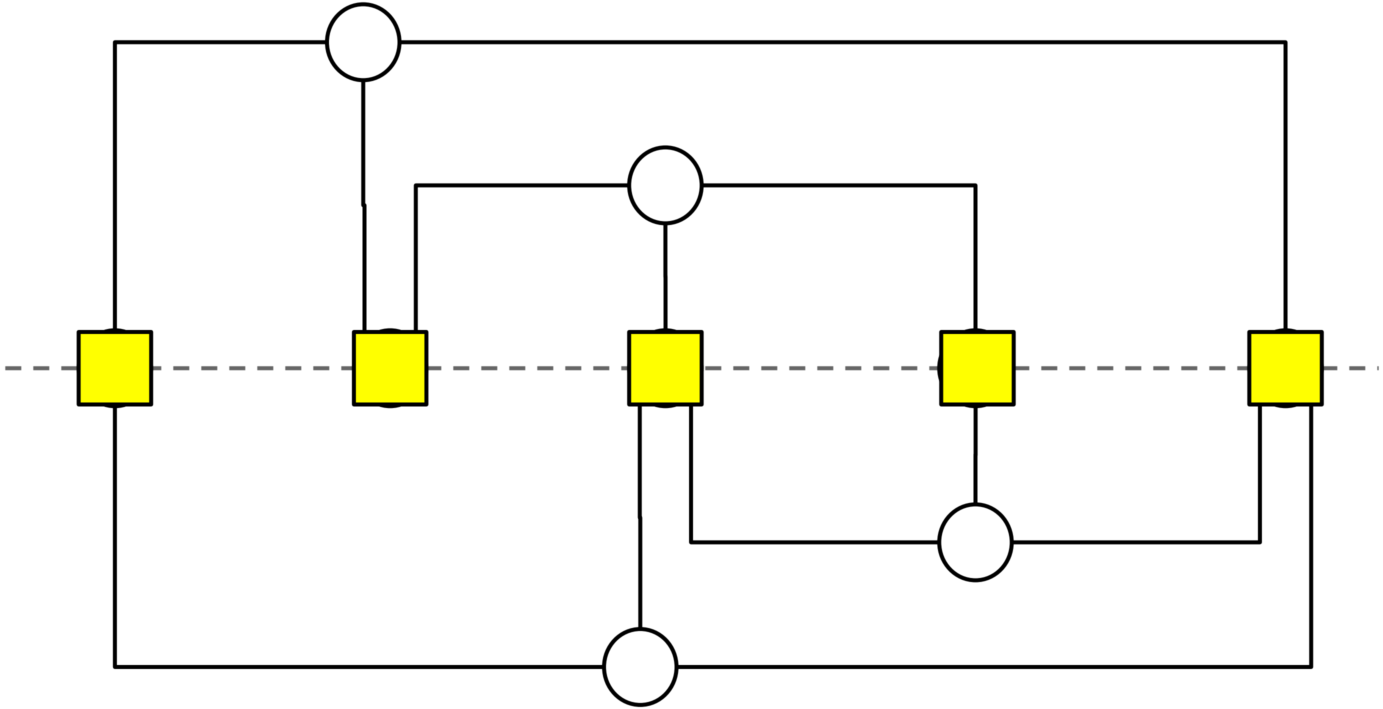


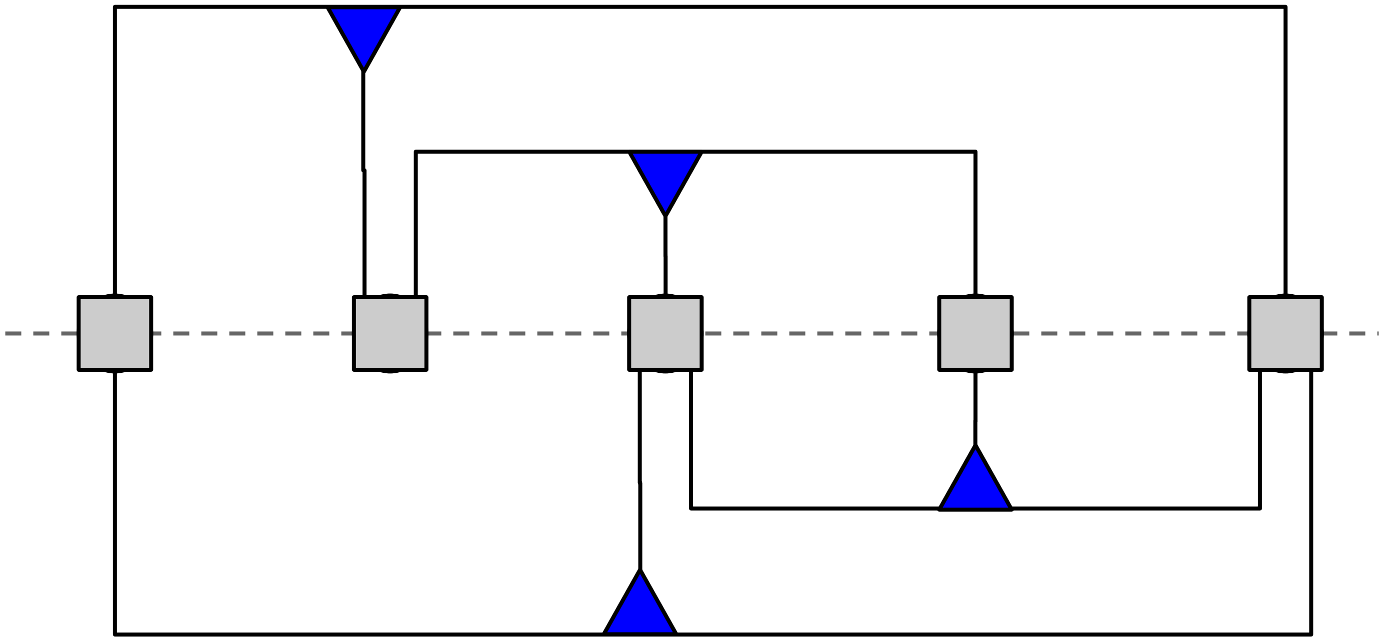
# Planar 3-SAT - Gadgets



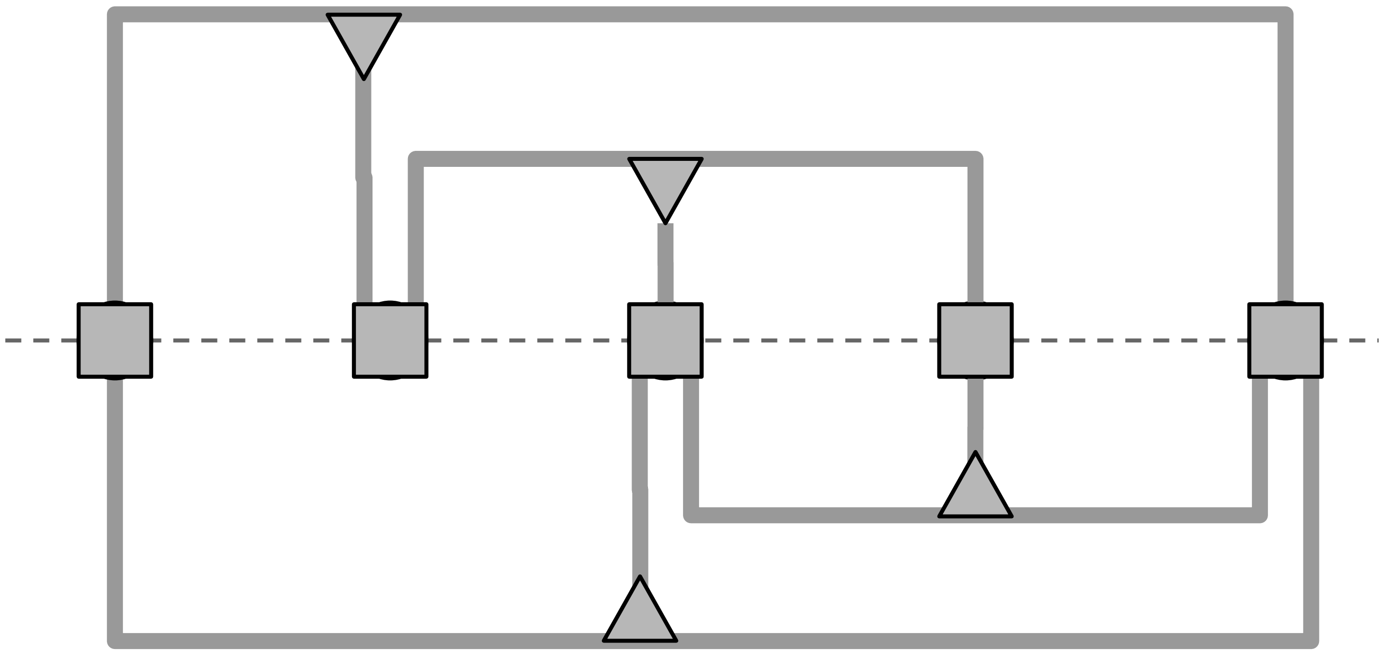
# Planar 3-SAT - *Variable* Gadget



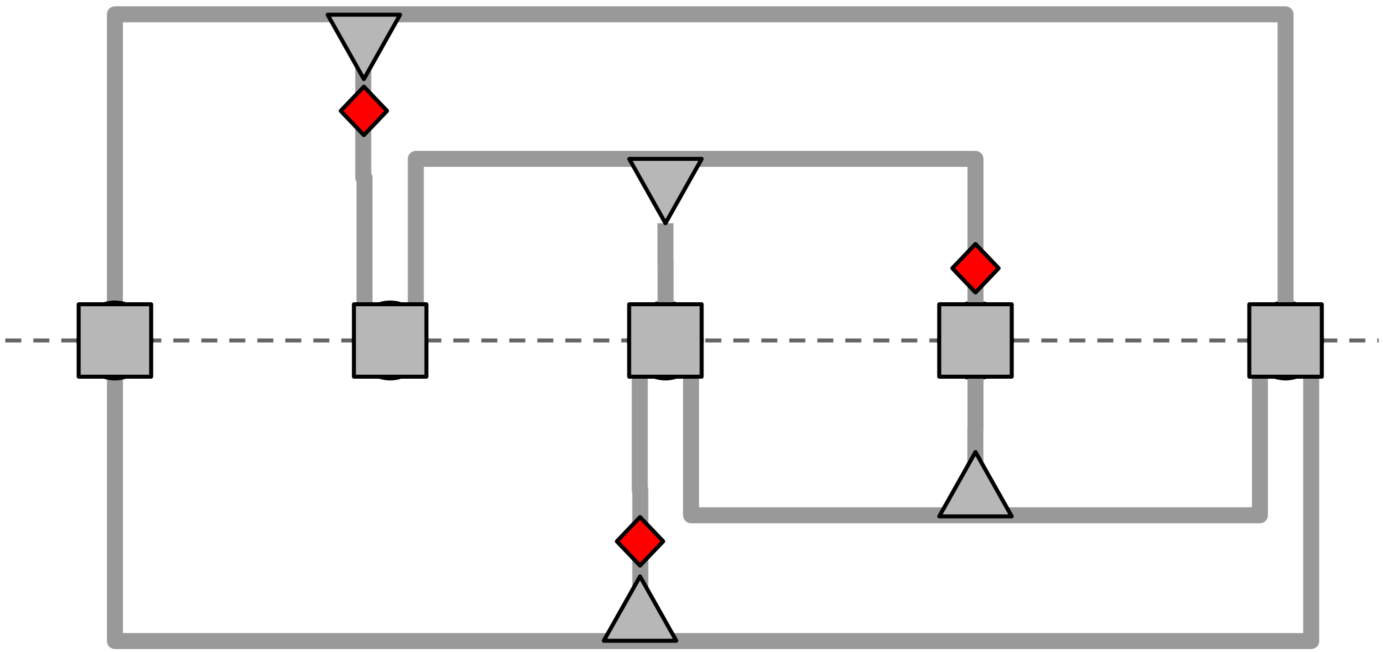
# Planar 3-SAT - *Clause* Gadget



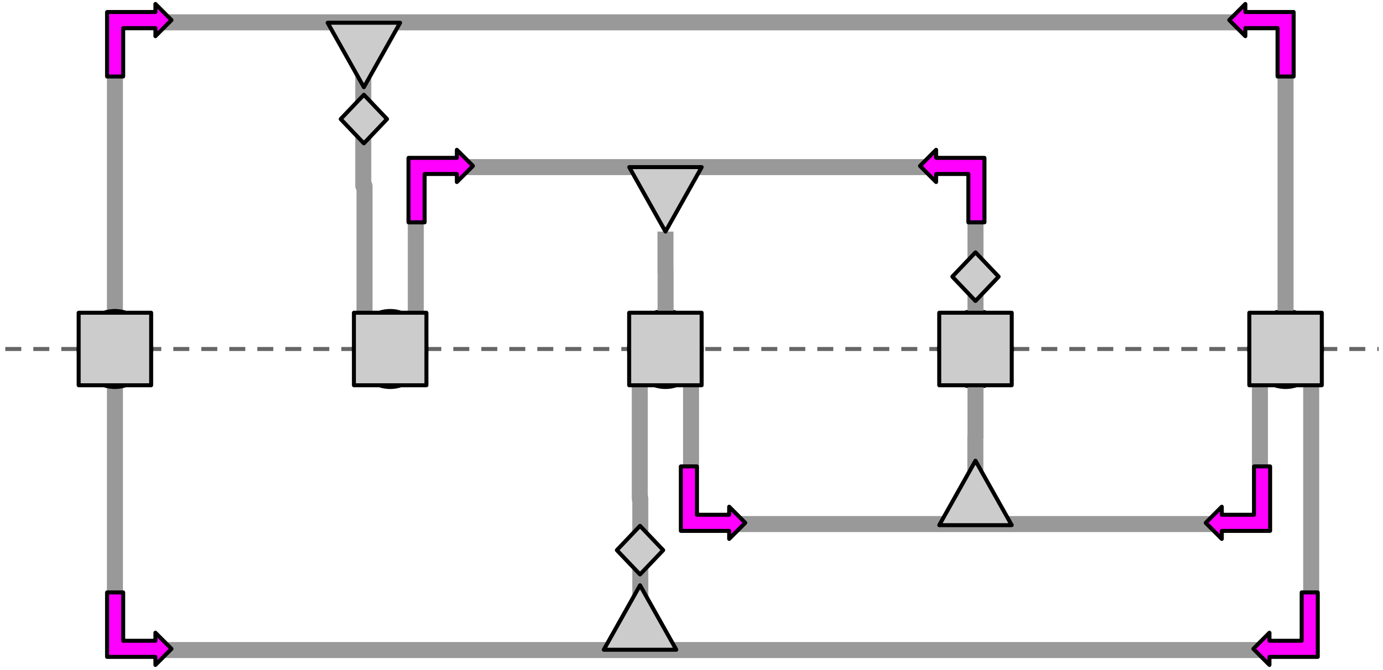
# Planar 3-SAT - Truth Propagation



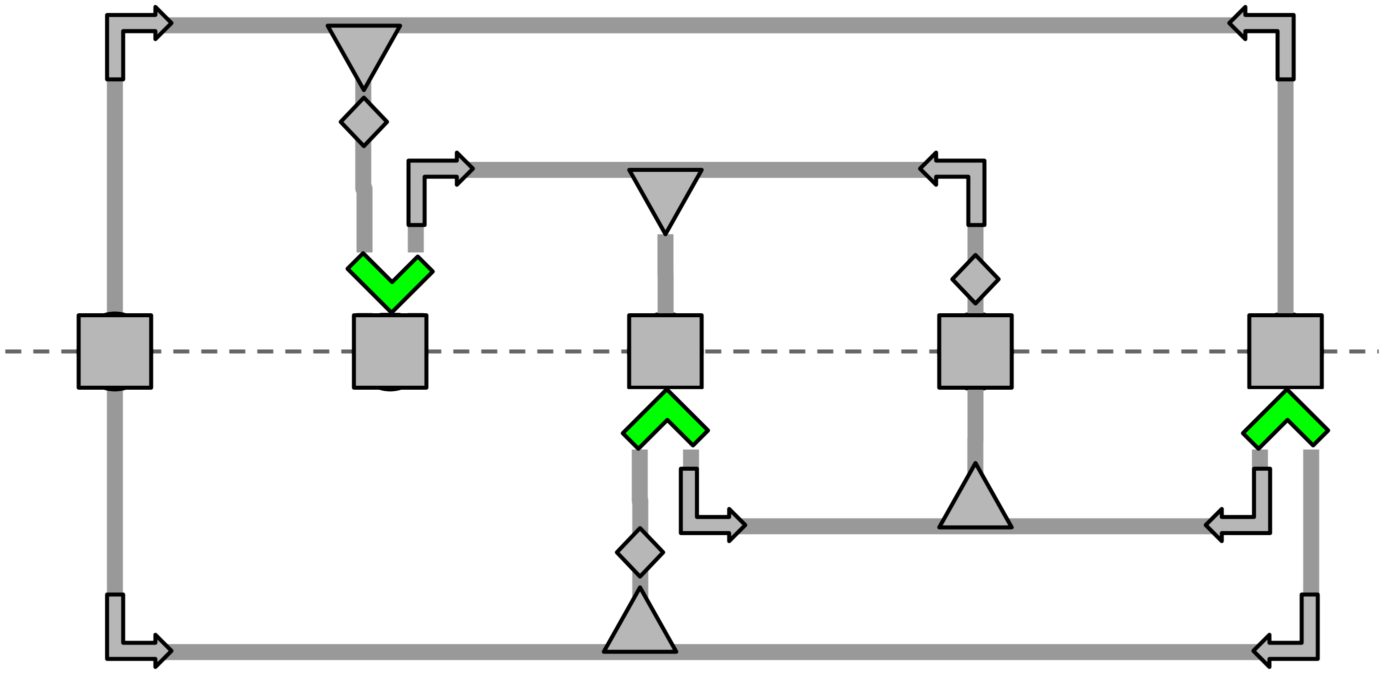
# Planar 3-SAT - *Not* Gadget



# Planar 3-SAT - *Turn* Gadget

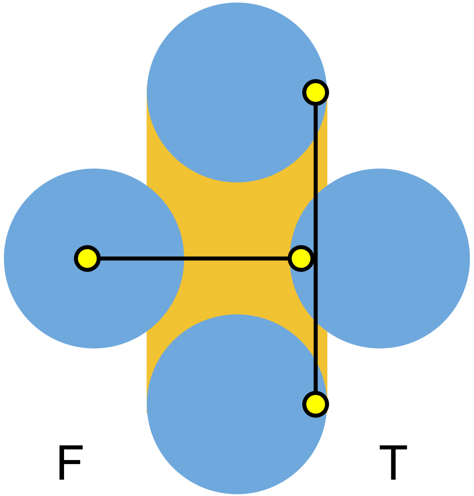


# Planar 3-SAT - *Split* Gadget

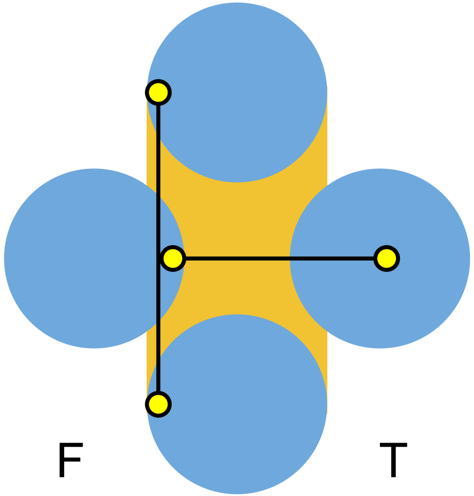


# *Variable* Gadget

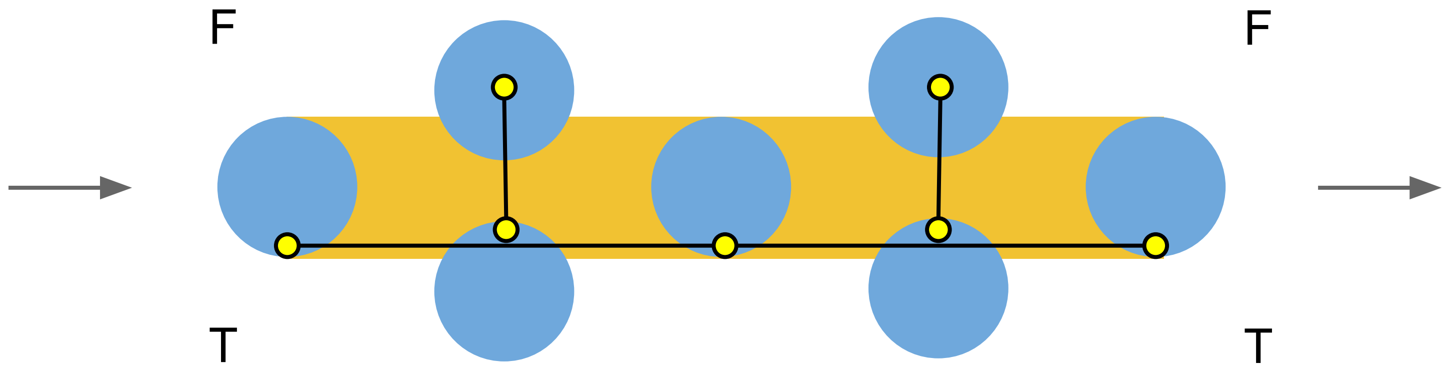
*True* configuration



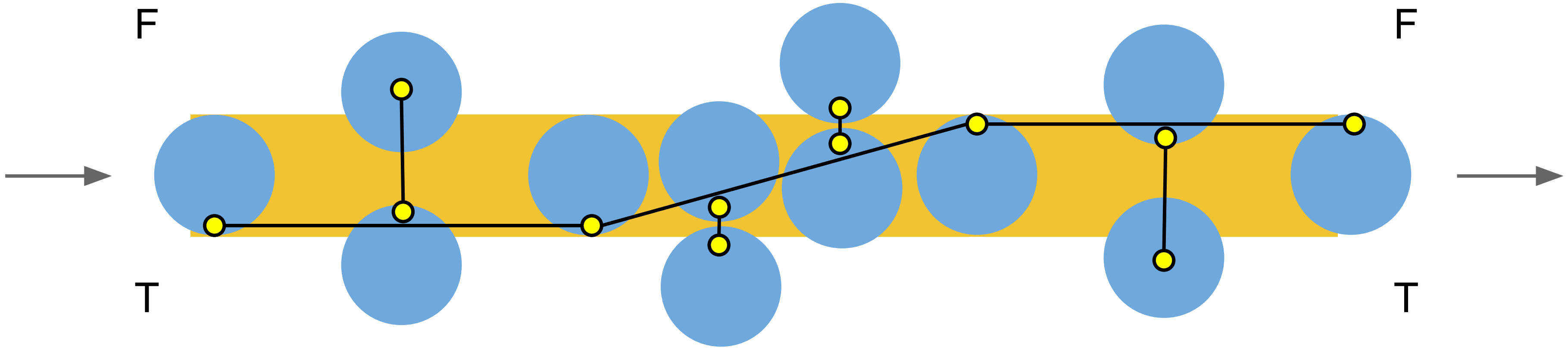

*False* configuration



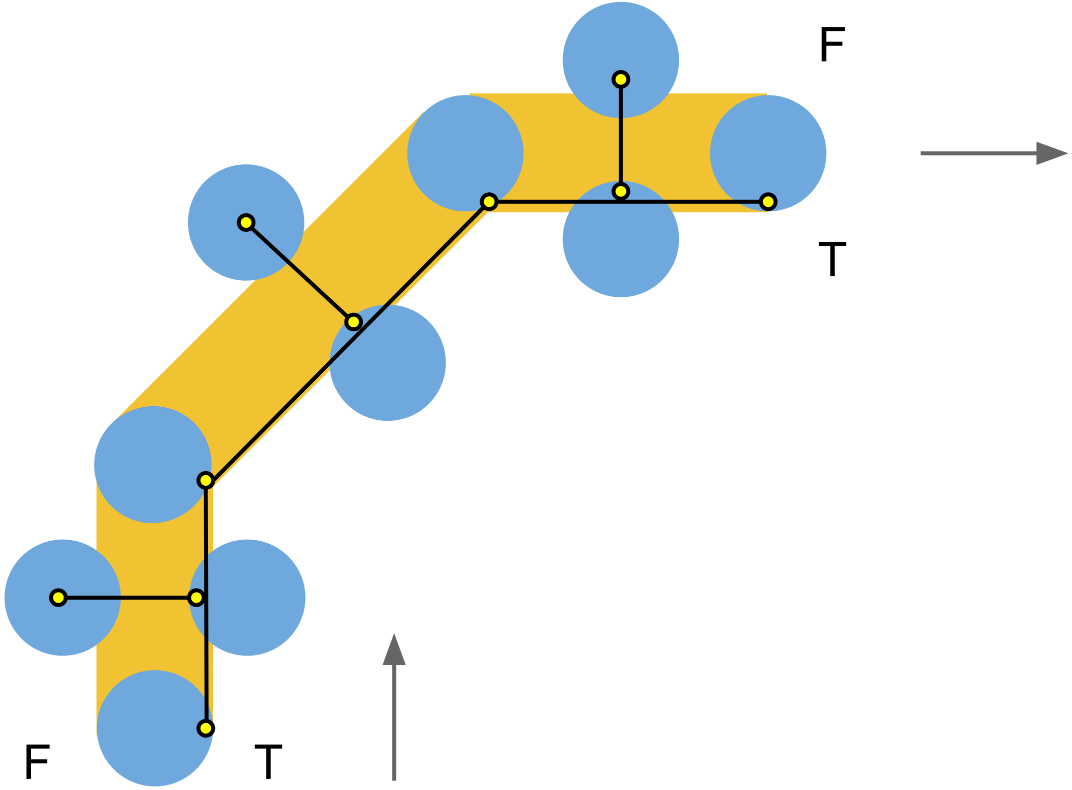
# Truth Propagation



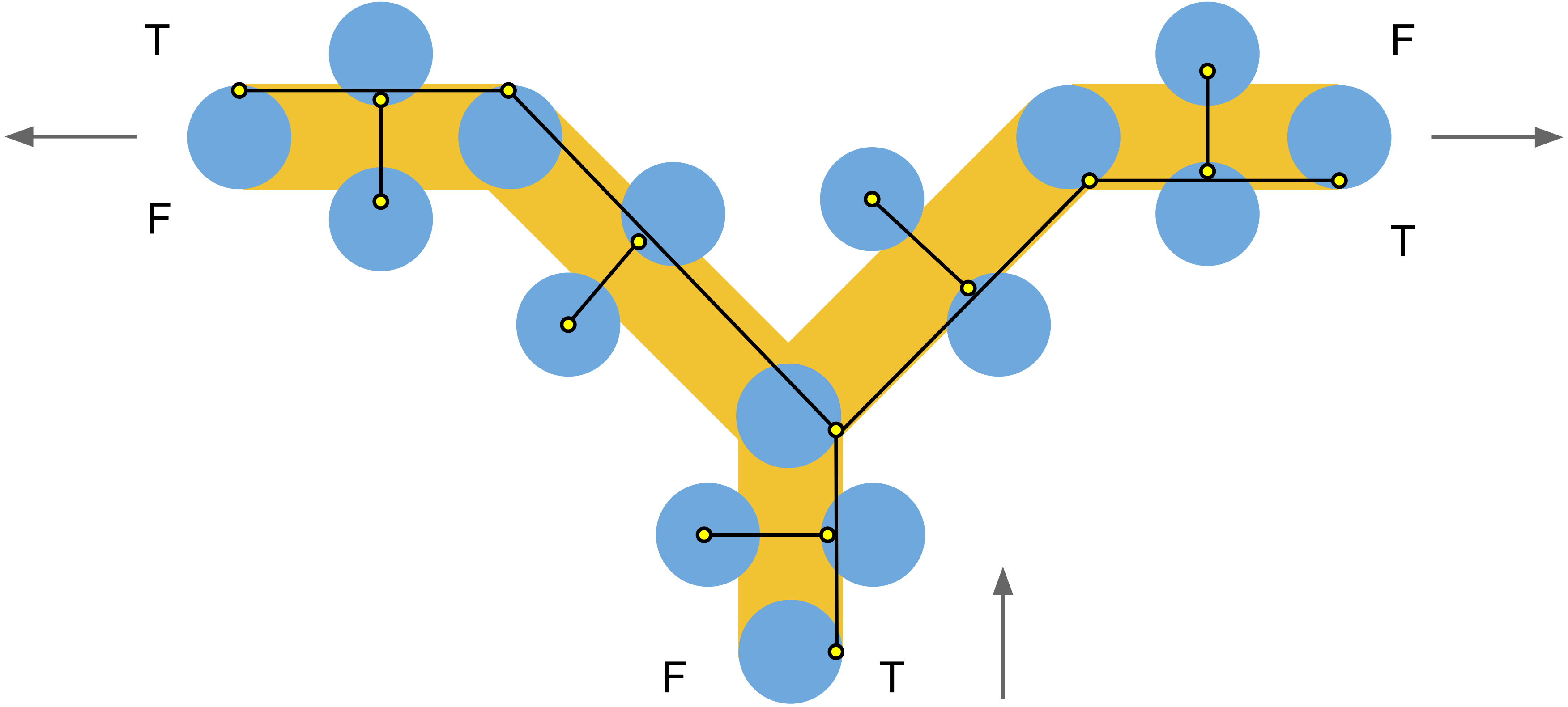
# *Not* Gadget



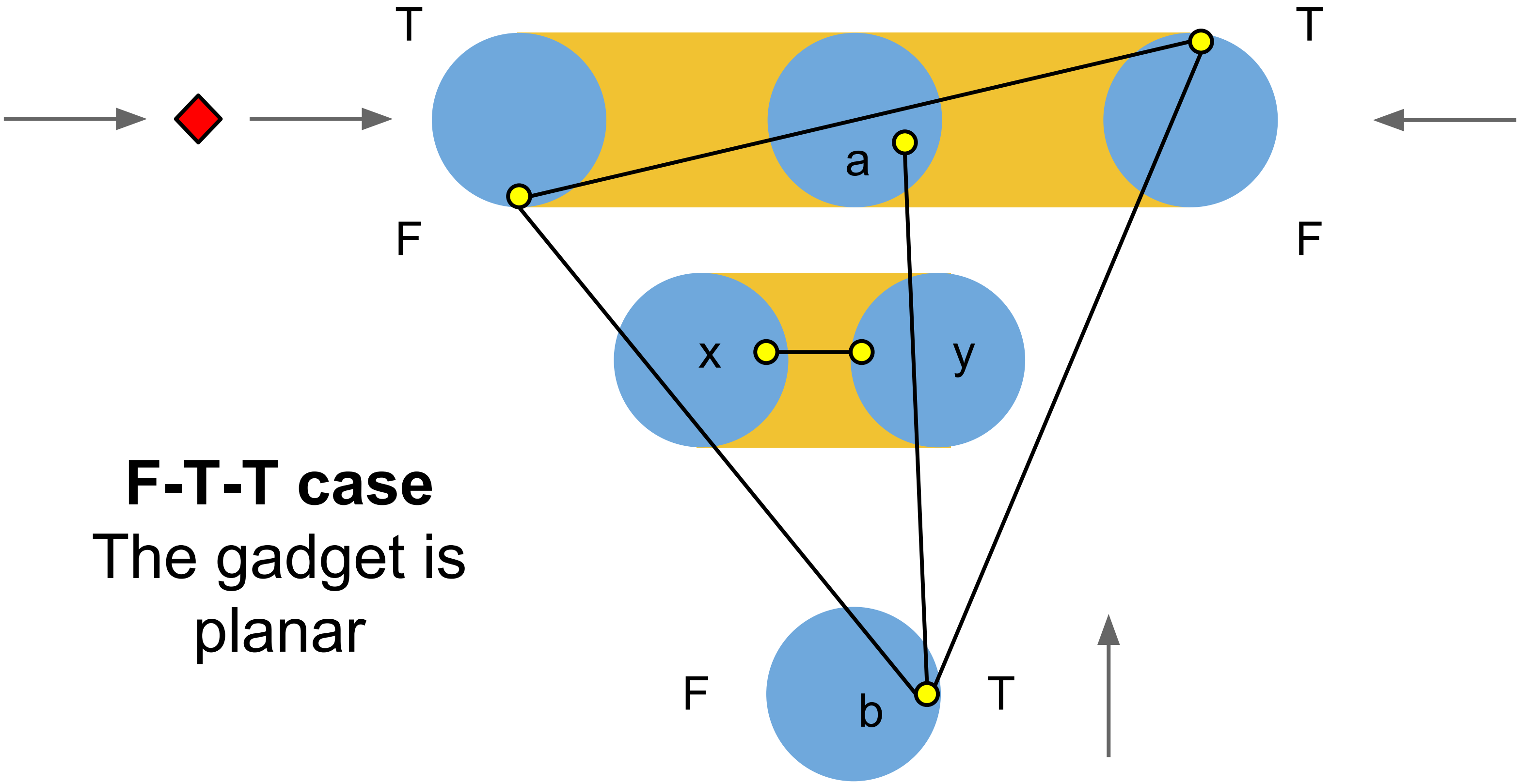
# Turn Gadget



# Split Gadget



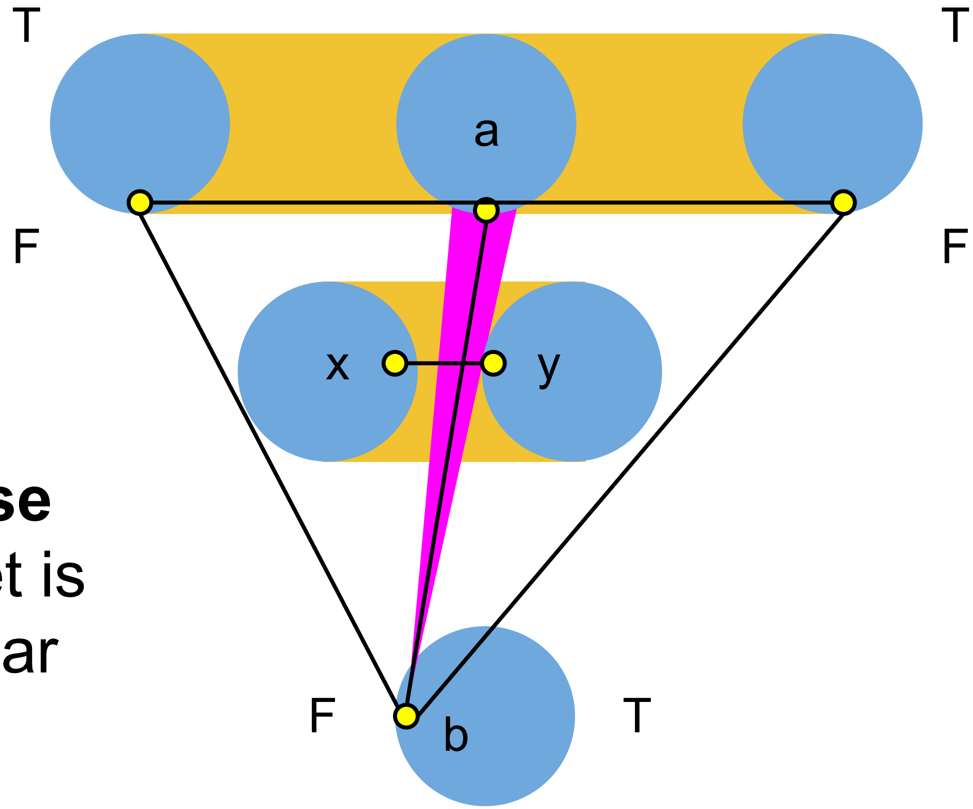
# Clause Gadget




**F-T-T case**

The gadget is planar

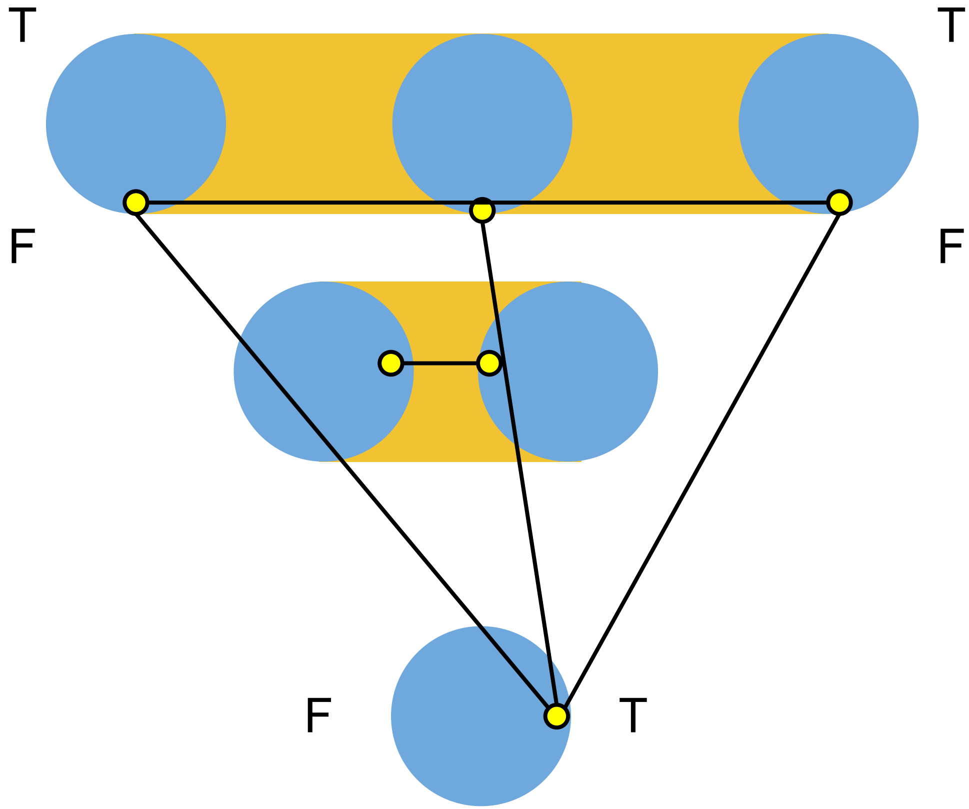
# Clause Gadget



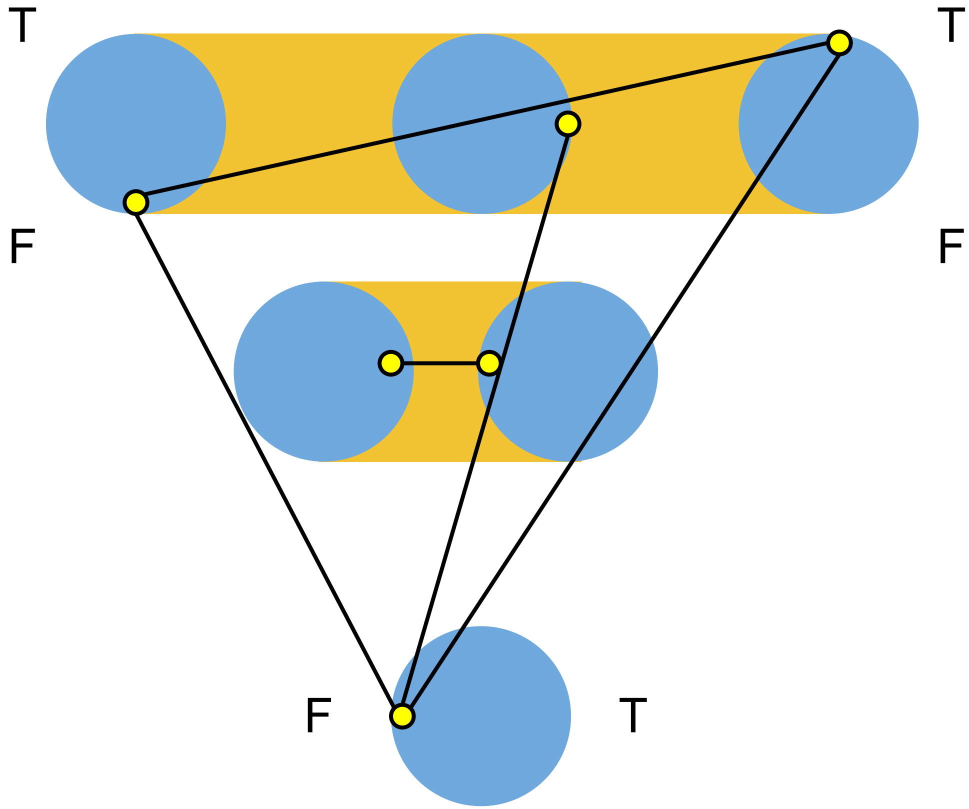

**F-F-F case**

The gadget is NOT planar

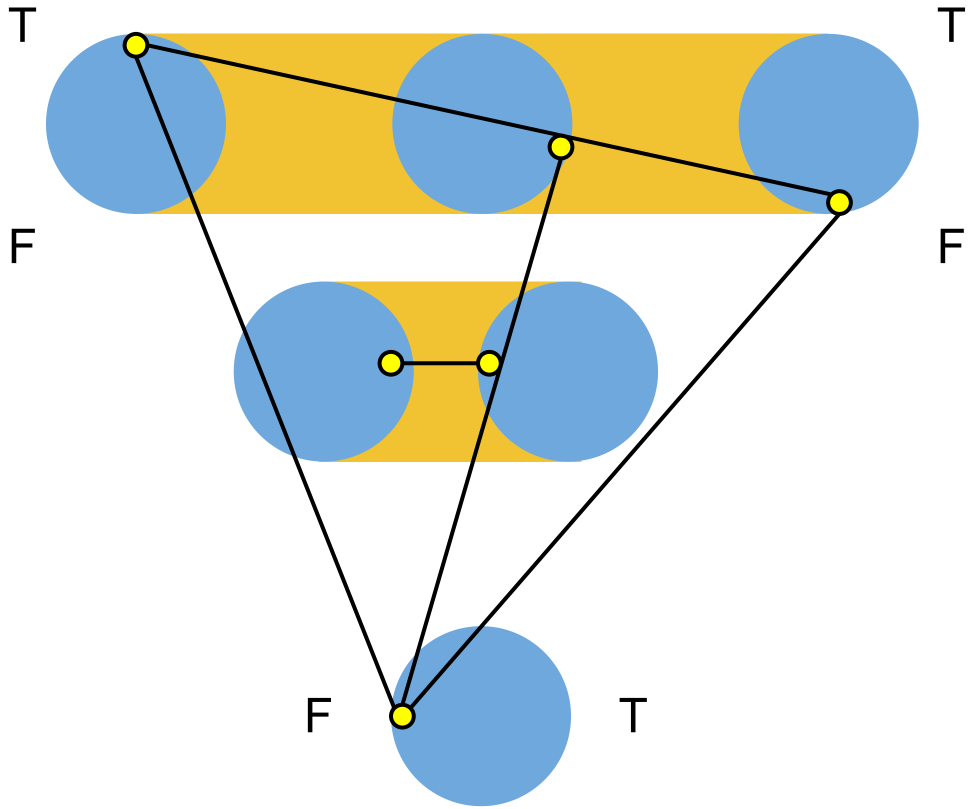
# Clause Gadget



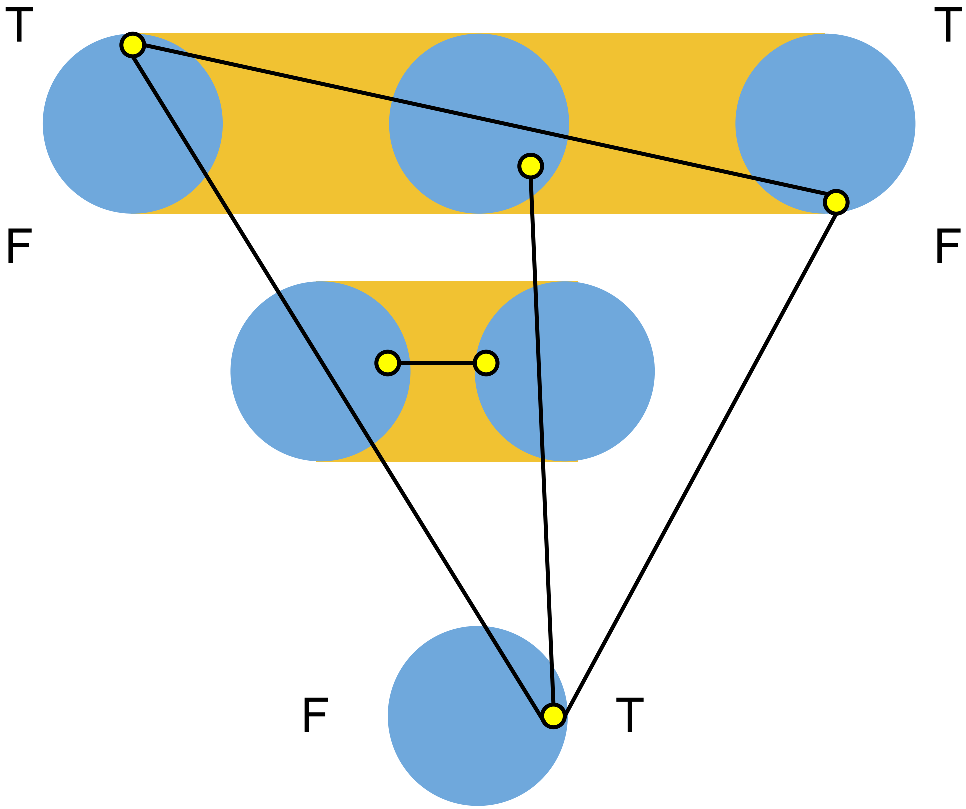
# Clause Gadget



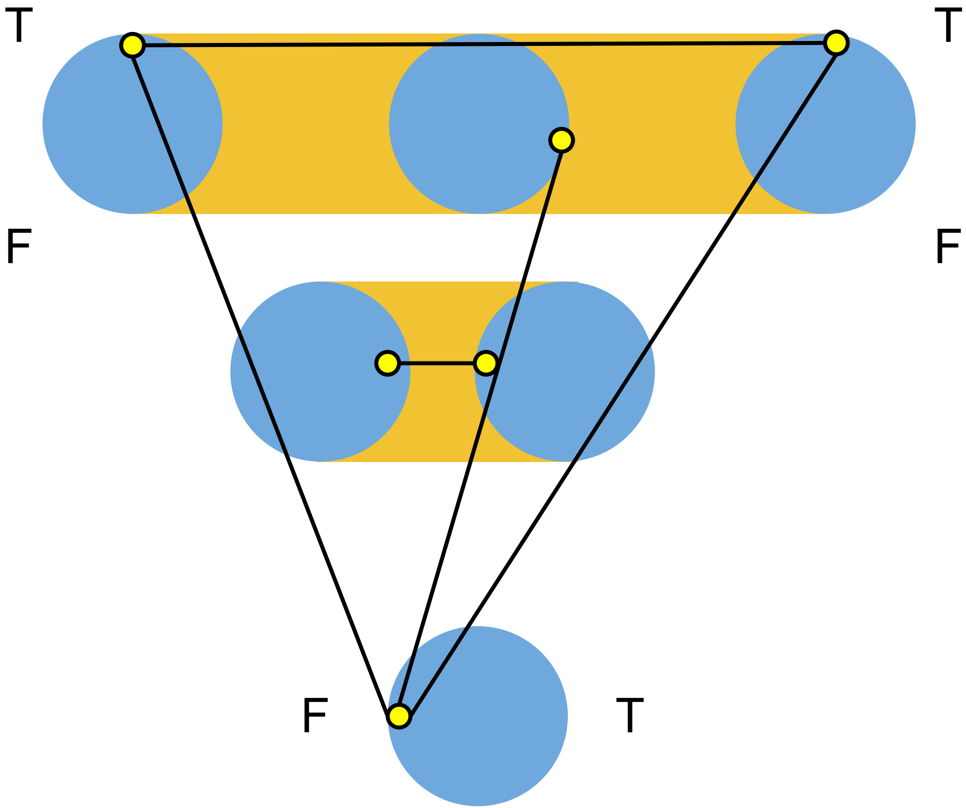
# Clause Gadget



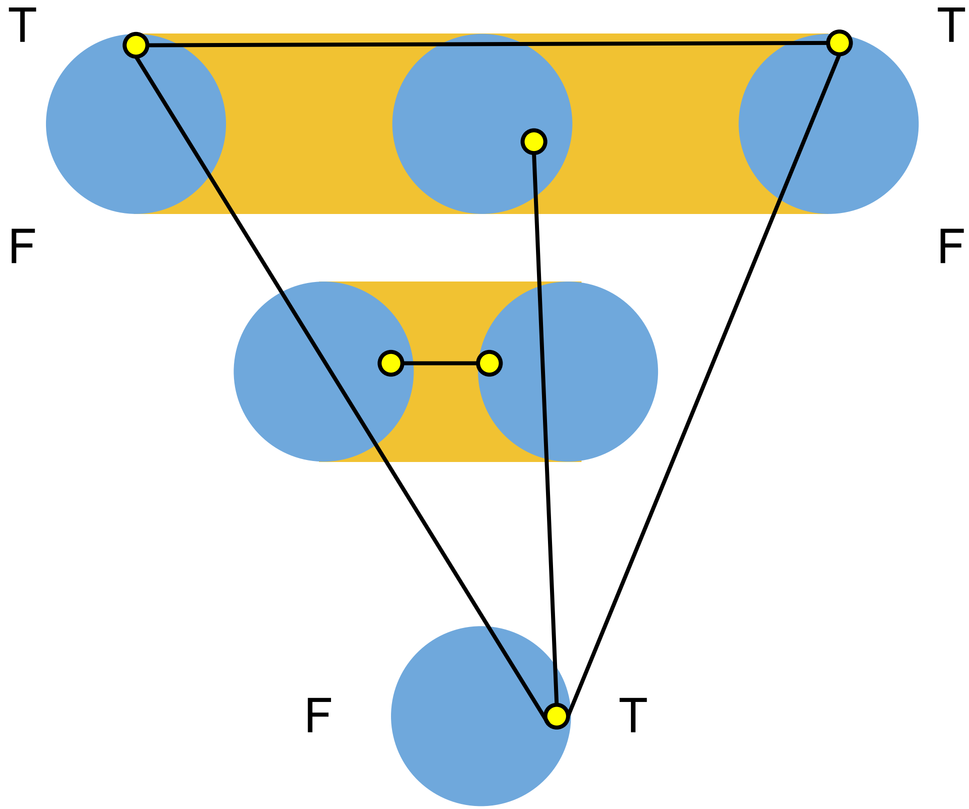
# Clause Gadget



# Clause Gadget



# Clause Gadget



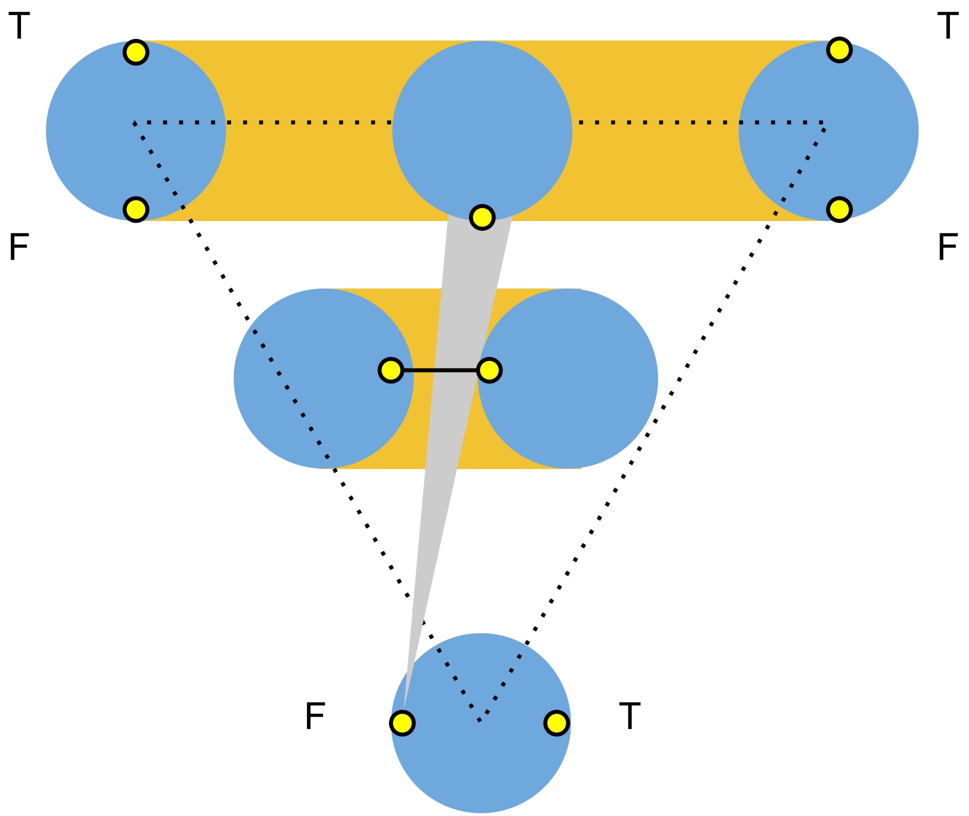
# Open Problems

- Do the hard problems belong to NP?
- Still hard with biconnected gadgets. What if triconnected?
- What if we allow regions to partially overlap?
- What if we allow some crossings?

# Applicative Context

- Drawing a graph on a geographical map
- Vertices have fixed positions

# *Clause* Gadget (master slide)



# Challenges

- Vertex cluttering, edge crossings
- Techniques exist to mitigate cluttering
- However, crossings are still an issue