Gadget general model: (for robot motion planning)
- locations (entrances/exits)
- states
- transitions \((l_1, s_1) \rightarrow (l_2, s_2)\)

\[ s_1: \quad \overset{s_2}{\raisebox{3pt}{\text{transition graph}}} \]

\[ \text{visual notation} \]

[Demaine, Grosos, Lynch, Rudoy - Fun 2018]
[Demaine, Hendrickson, Lynch - arXiv 2018]

Examples:

- 2-toggle

\[ \text{states: } 1 \quad 2 \quad \text{[Demaine, Grosos, Lynch - CIAC 2017]} \]

- Closing door (directed, single use)

- Opening door (directed, single use)
Gadget types:
- **k-tunnel** = all transitions are along edges of perfect matching on locations (states can control traversability & directions)
- **DAG** = state-transition graph is acyclic → possible transitions on states (merging all locations)

Motion planning = traversal s→t in graph of:
- gadgets (instances from some allowed set)
- connections between gadget locations

Characterization: motion planning with DAG k-tunnel gadgets is NP-complete iff:
- some gadget has distant opening: traversal that opens another tunnel
OR - some gadget has forced distant closing: traversal that always closes another tunnel
\[ s, l_1 \rightarrow l_2 \quad (s_1, l_1) \rightarrow (\ast, l_2) \]

Eventually static: allow loops in terminal states e.g.: door closing is NP-hard
\[ \overset{\text{≈ Metatheorem 4b}}{\text{≈ Metatheorem 3}} \overset{\text{of [Viglietta-Fun 2012]}}{\text{of [Frošek-Fun 2010]}} \]
& door opening & diode is NP-hard
(also need crossover for now)