ICNP 2013 October 9th

Using Routers to Build Logic Gates: How Powerful is BGP?

Marco Chiesa (Roma Tre University)

joint work with

Luca Cittadini

Giuseppe Di Battista

Laurent Vanbever

Stefano Vissicchio

goal and motivations

goal

- build a very expensive computer

goal and motivations

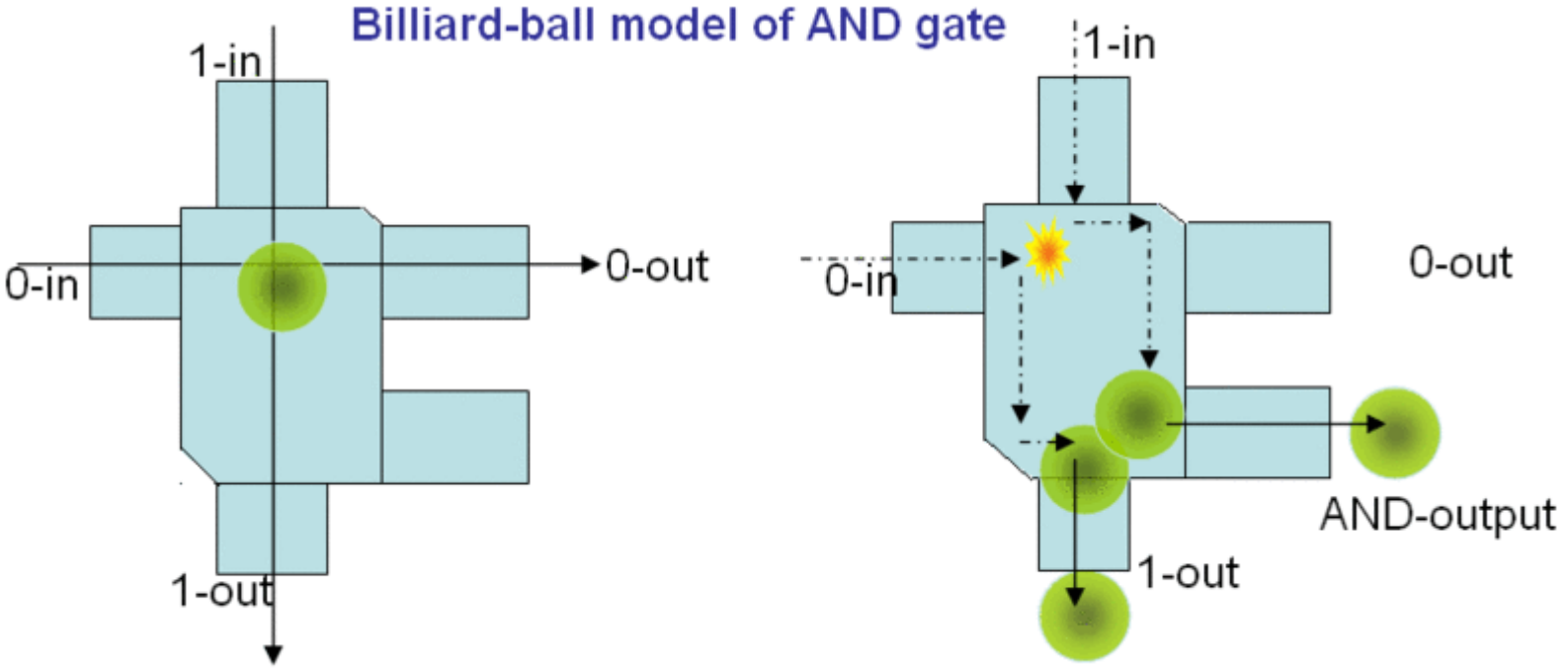
goal

- build a very expensive computer

how?

- logic gates are cheap
- we want to spend more :)

billiard balls [Fredkin,Toffoli]



crab swarms [Gunji,Nishiyama,Adamatzki]

goal and motivations

goal

- build a very expensive computer

how?

- logic gates are cheap
- we want to spend more :)
- carrier grade routers are good candidates ($$$)

goal and motivations

goal

- build a very expensive computer

how?

- logic gates are cheap
- we want to spend more :)
- carrier grade routers are good candidates ($$$)
- we use the simplest BGP (interdomain routing protocol) constructs (`local-preference` and filtering based on adjacent neighbors)

the Border Gateway Protocol (BGP)

- “de-facto” interdomain routing protocol
- policy based
- risk of routing instability

main result

analyzing BGP dynamics is
as hard as
analyzing a computer program

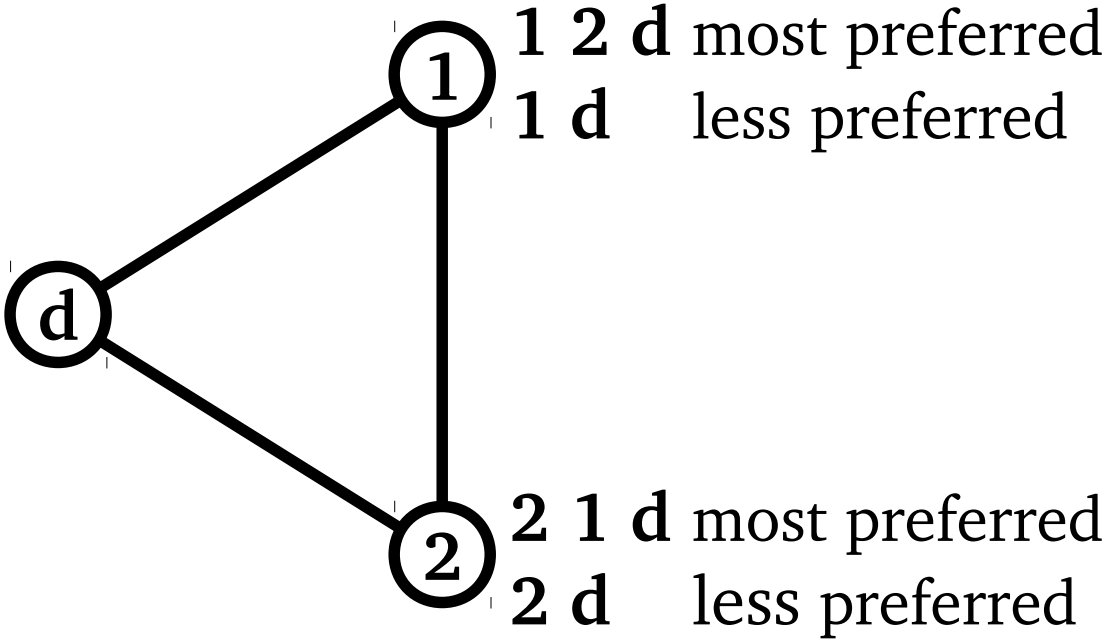
a popular BGP gadget:

DISAGREE [GriffinWilfong1999]

two stable states

+

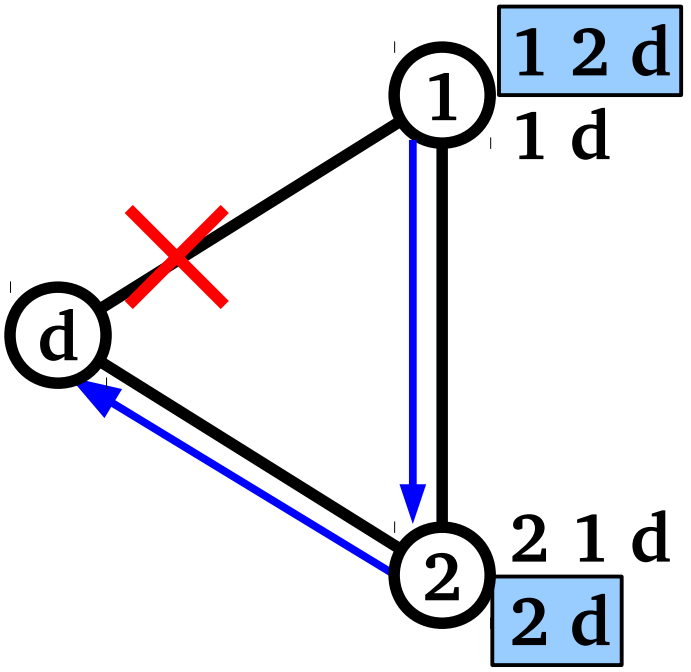
risk of oscillation

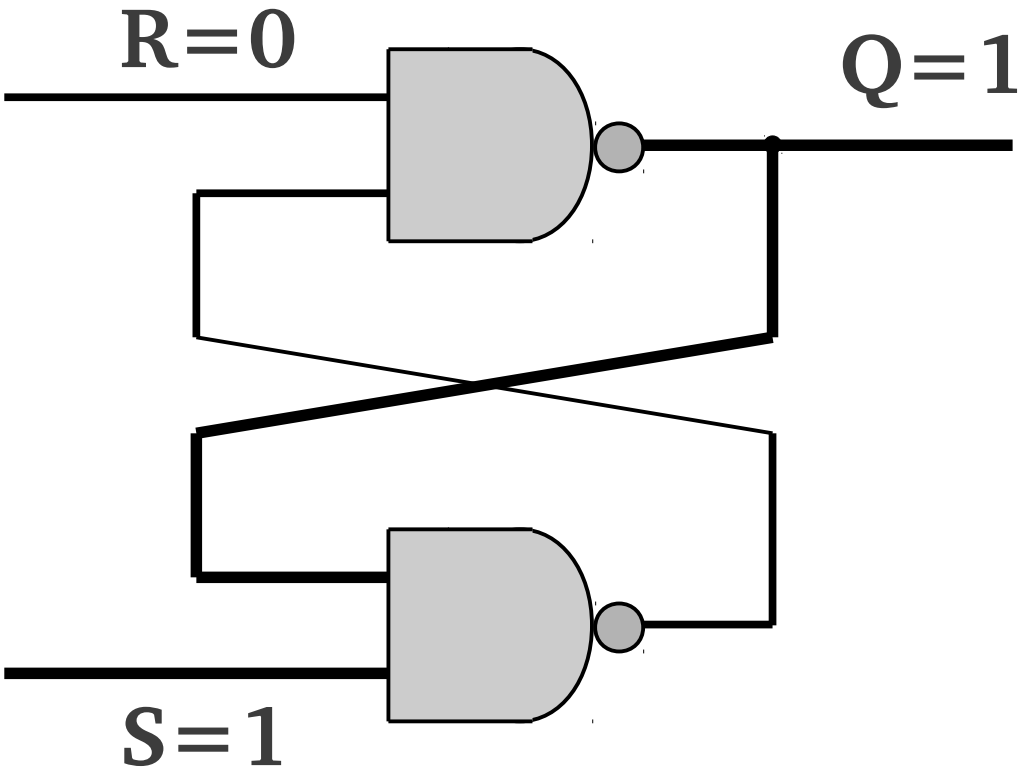


selected next-hop

route available

a mapping between a DISAGREE and a SR flip-flop





(1 d) available at 1 ↔ R=1

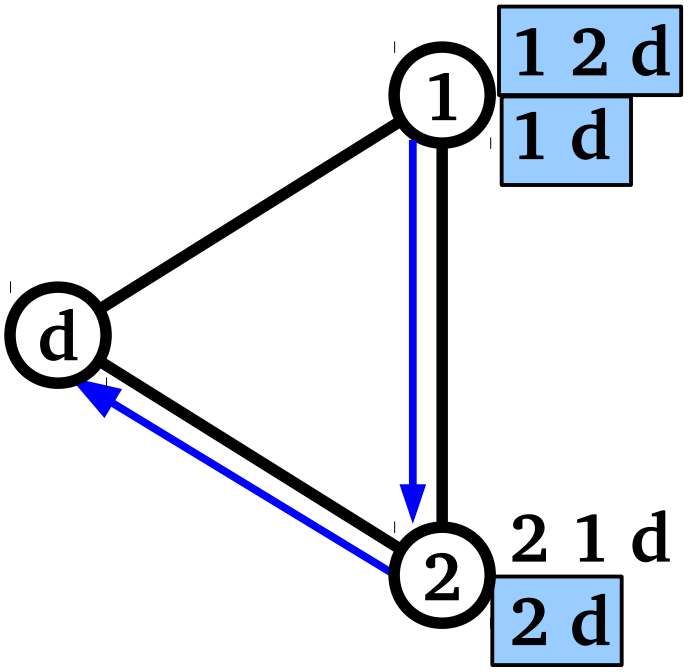
(2 d) available at 2 ↔ S=1

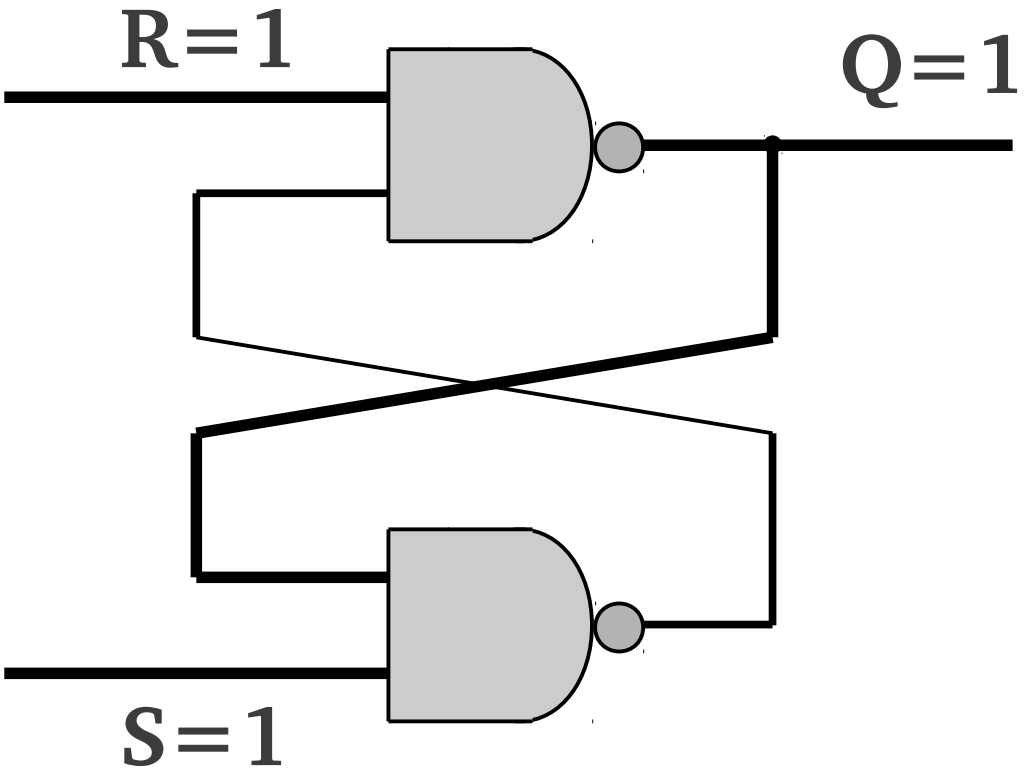
1 selects (1 2 d) ↔ Q=1

route available

selected next-hop

a mapping between a DISAGREE and a SR flip-flop





(1 d) available at 1 ↔ R=1

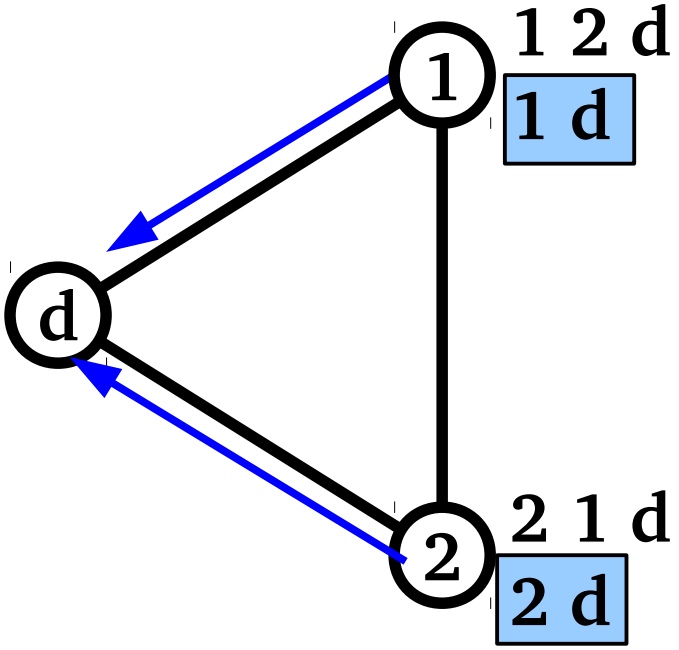
(2 d) available at 2 ↔ S=1

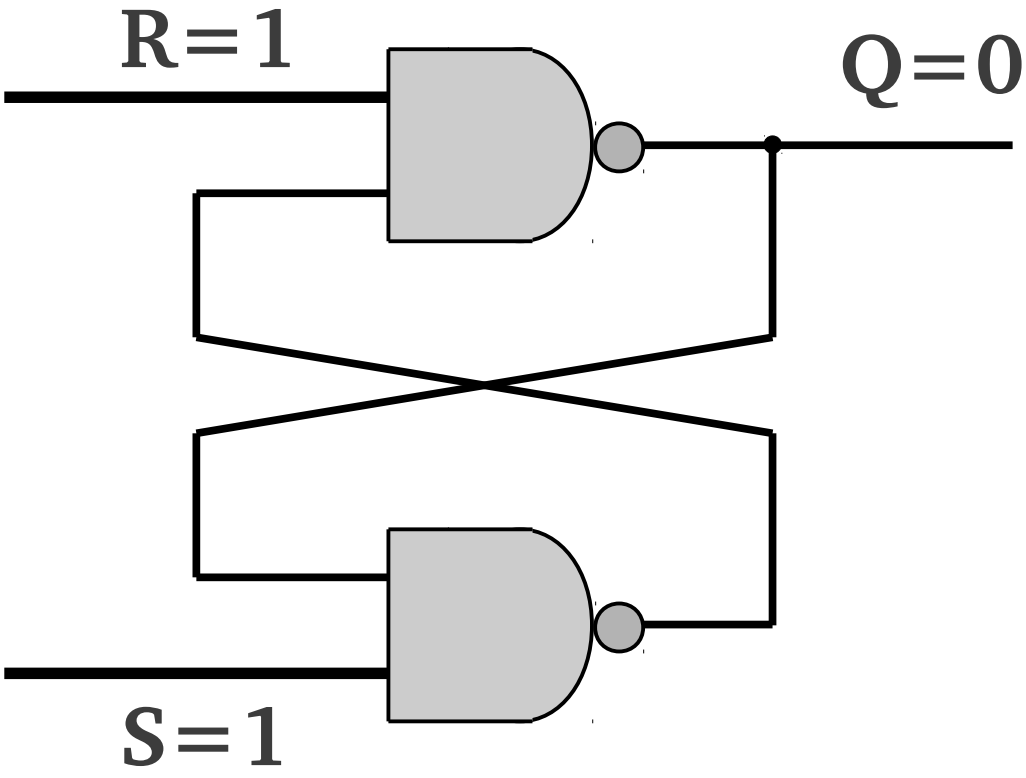
1 selects (1 2 d) ↔ Q=1

route available

selected next-hop

a mapping between a DISAGREE and a SR flip-flop





(1 d) available at 1 ↔ R=1

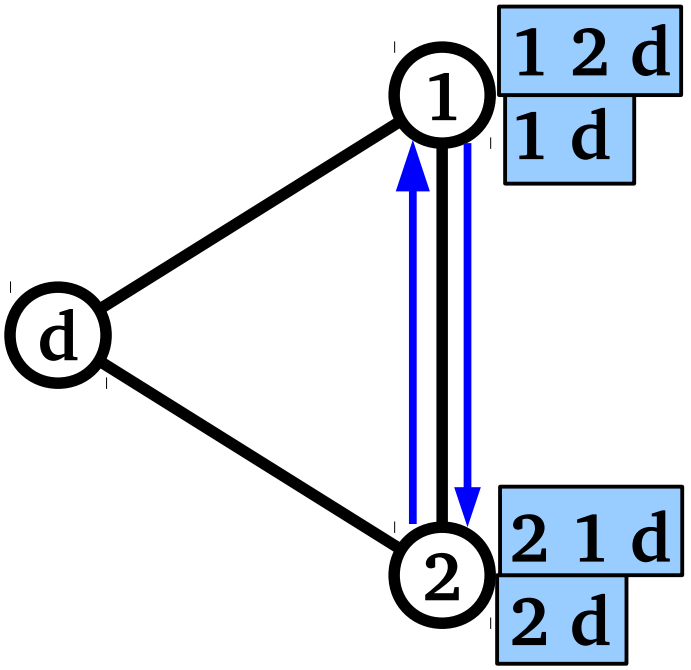
(2 d) available at 2 ↔ S=1

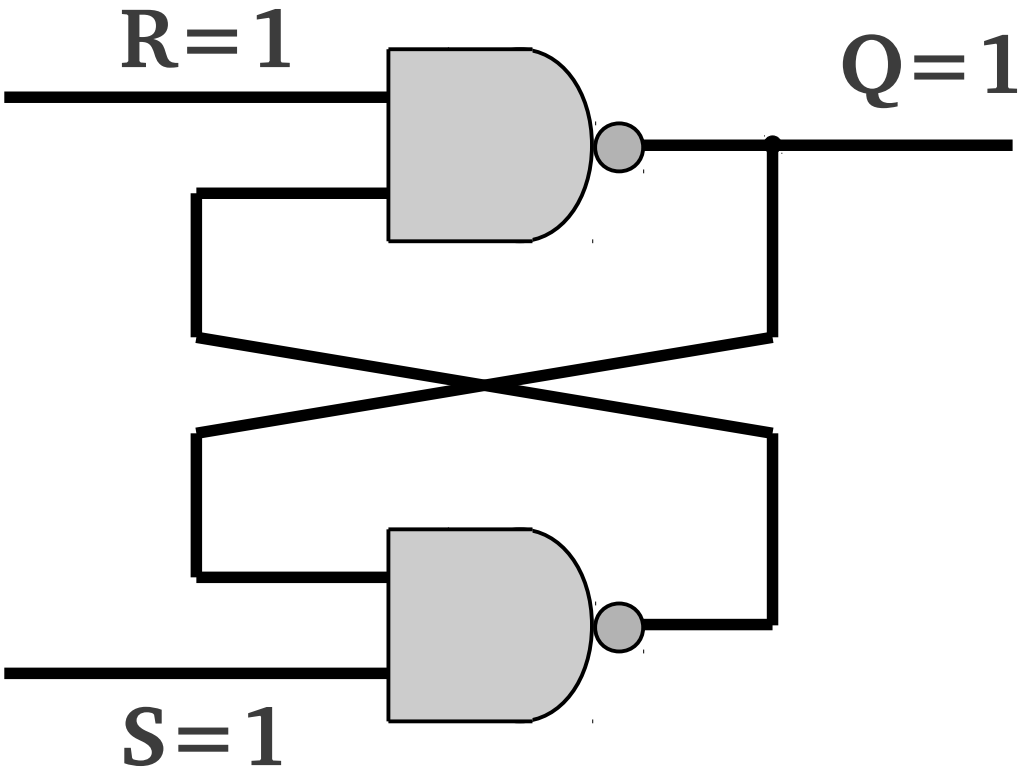
1 selects (1 2 d) ↔ Q=1

route available

selected next-hop

a mapping between a DISAGREE and a SR flip-flop





(1 d) available at 1 ↔ R=1

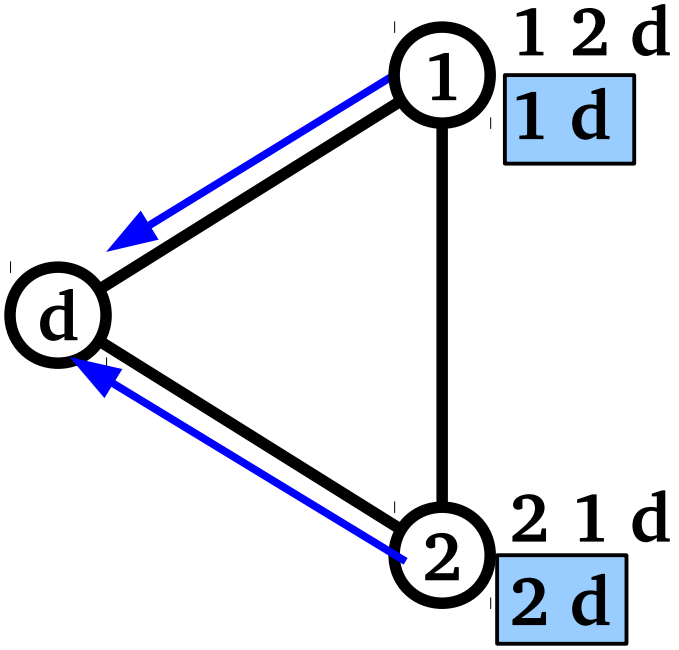
(2 d) available at 2 ↔ S=1

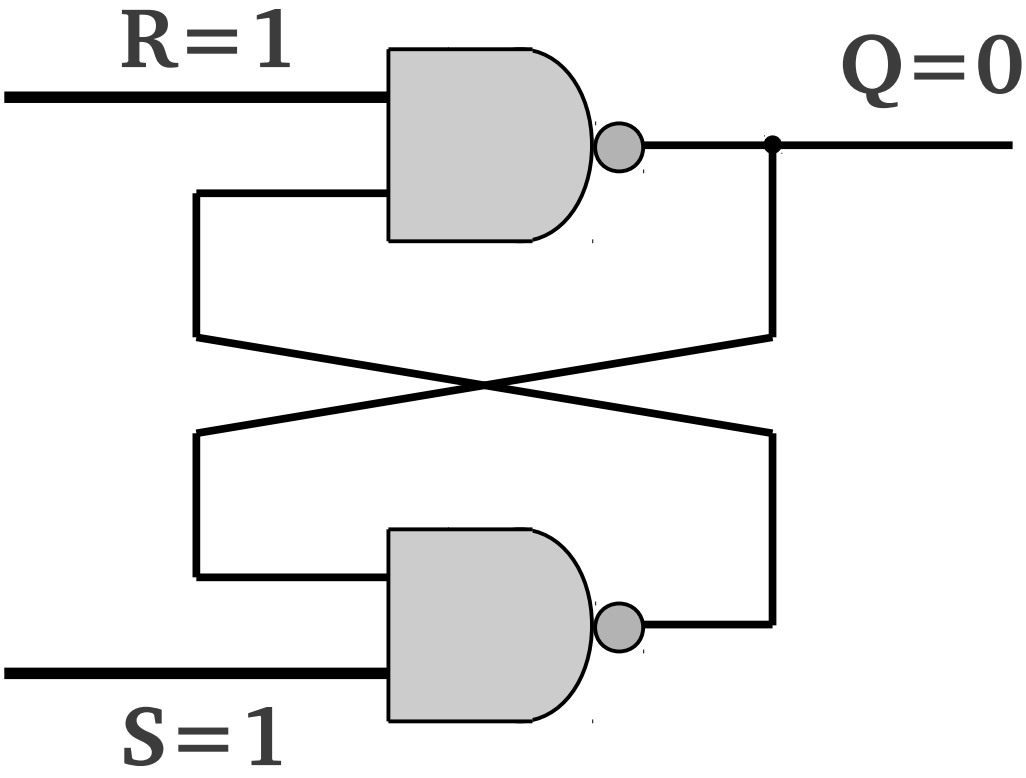
1 selects (1 2 d) ↔ Q=1

route available

selected next-hop

a mapping between a DISAGREE and a SR flip-flop





(1 d) available at 1 ↔ R=1

(2 d) available at 2 ↔ S=1

1 selects (1 2 d) ↔ Q=1

route available

selected next-hop

a simple mapping between BGP router states and logic values

absence of any available route to d = logic 0

availability of a route to d = logic 1

it is possible to construct:

- OR
- NOT
- AND

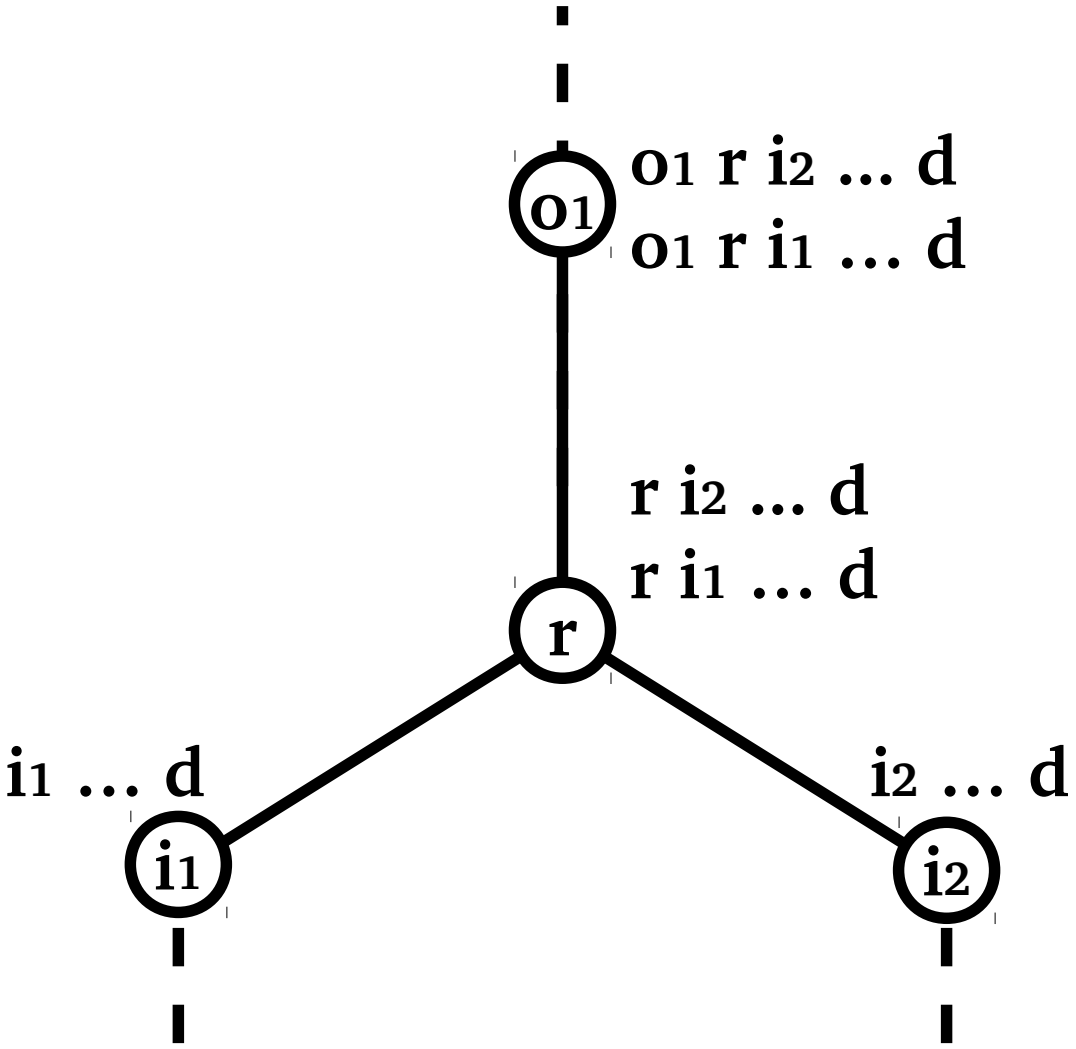
gates using BGP routers

BGP and logic gates
And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.



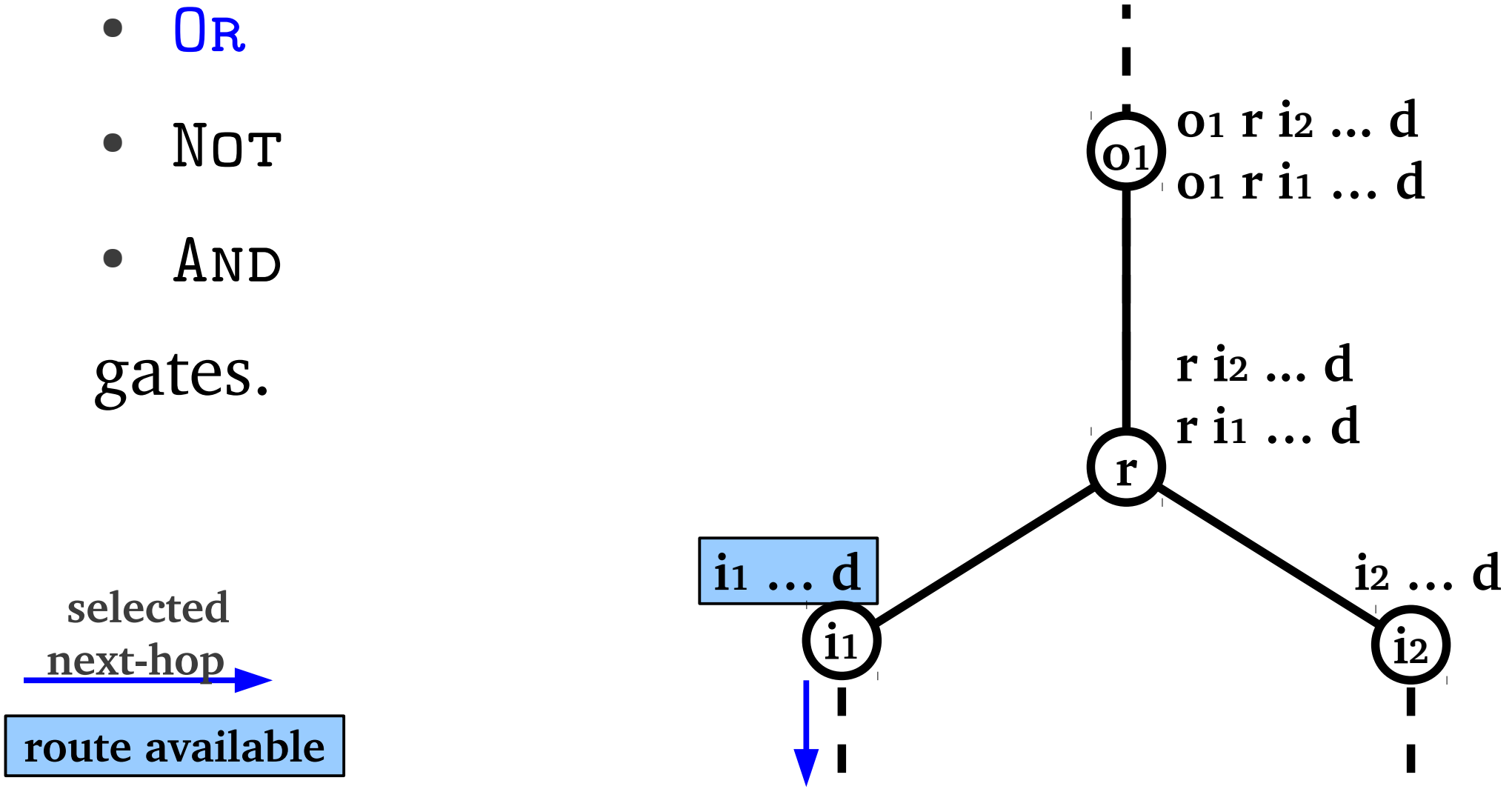
BGP and logic gates

And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.

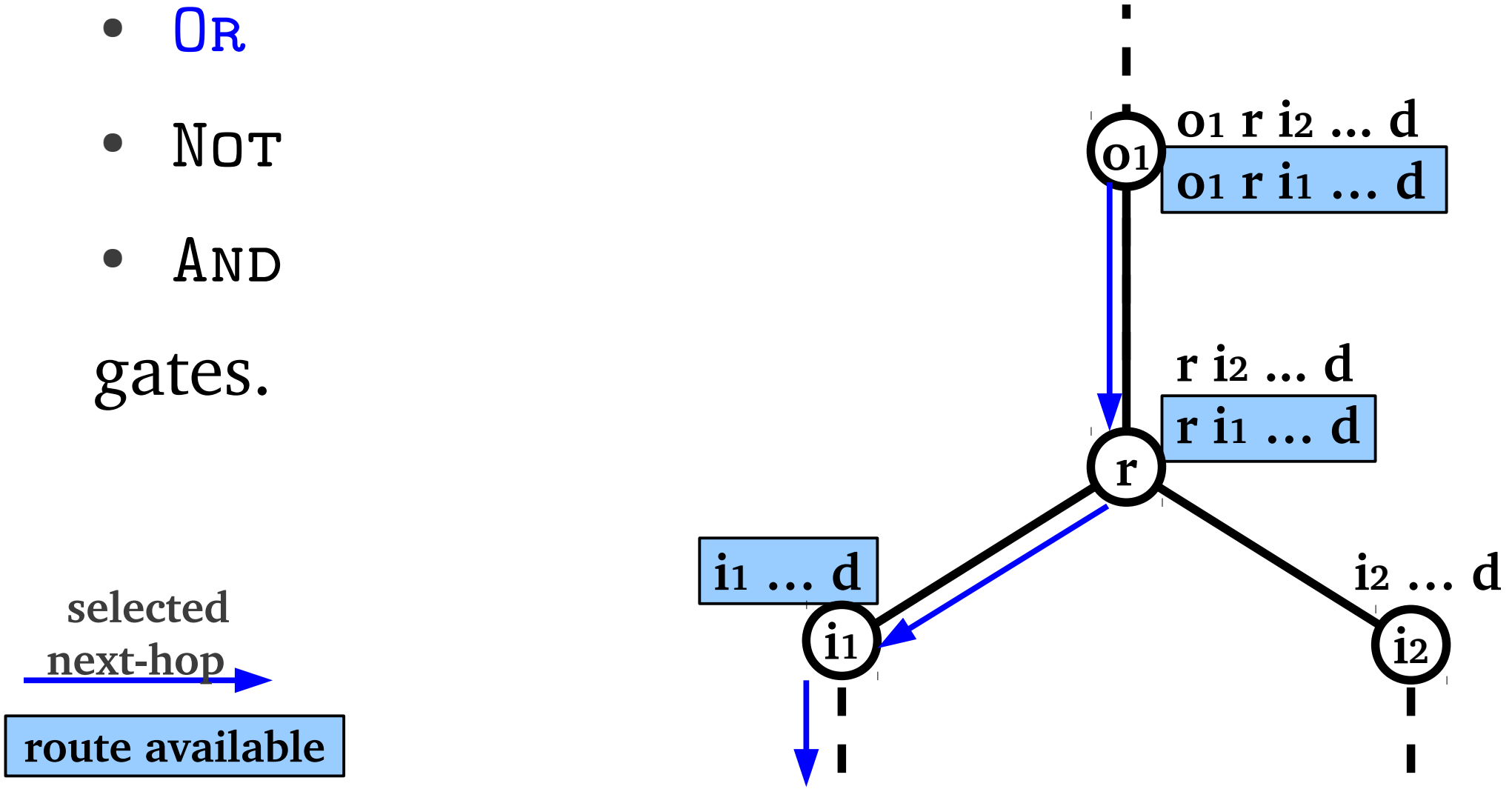


BGP and logic gates
And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.

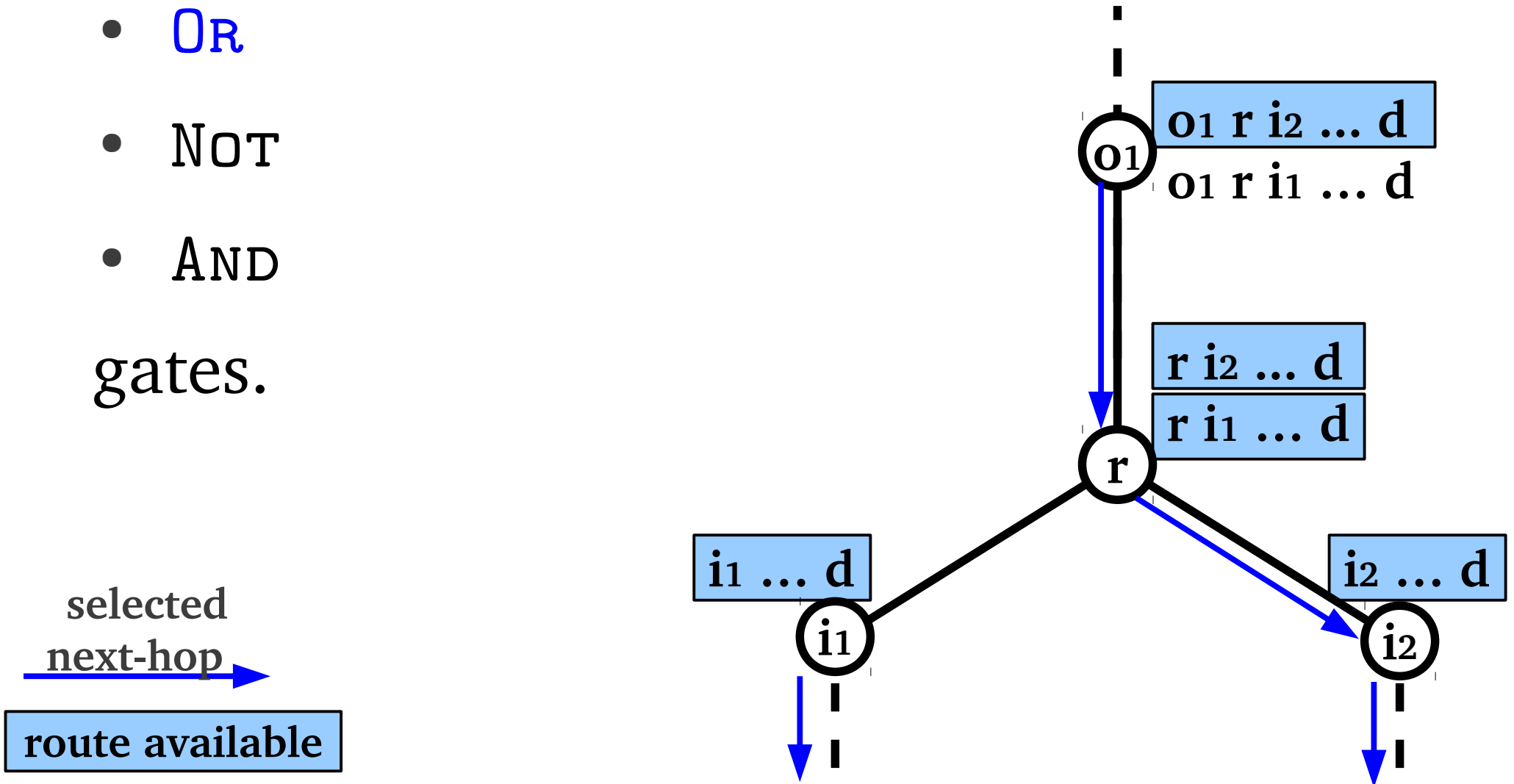


BGP and logic gates
And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.



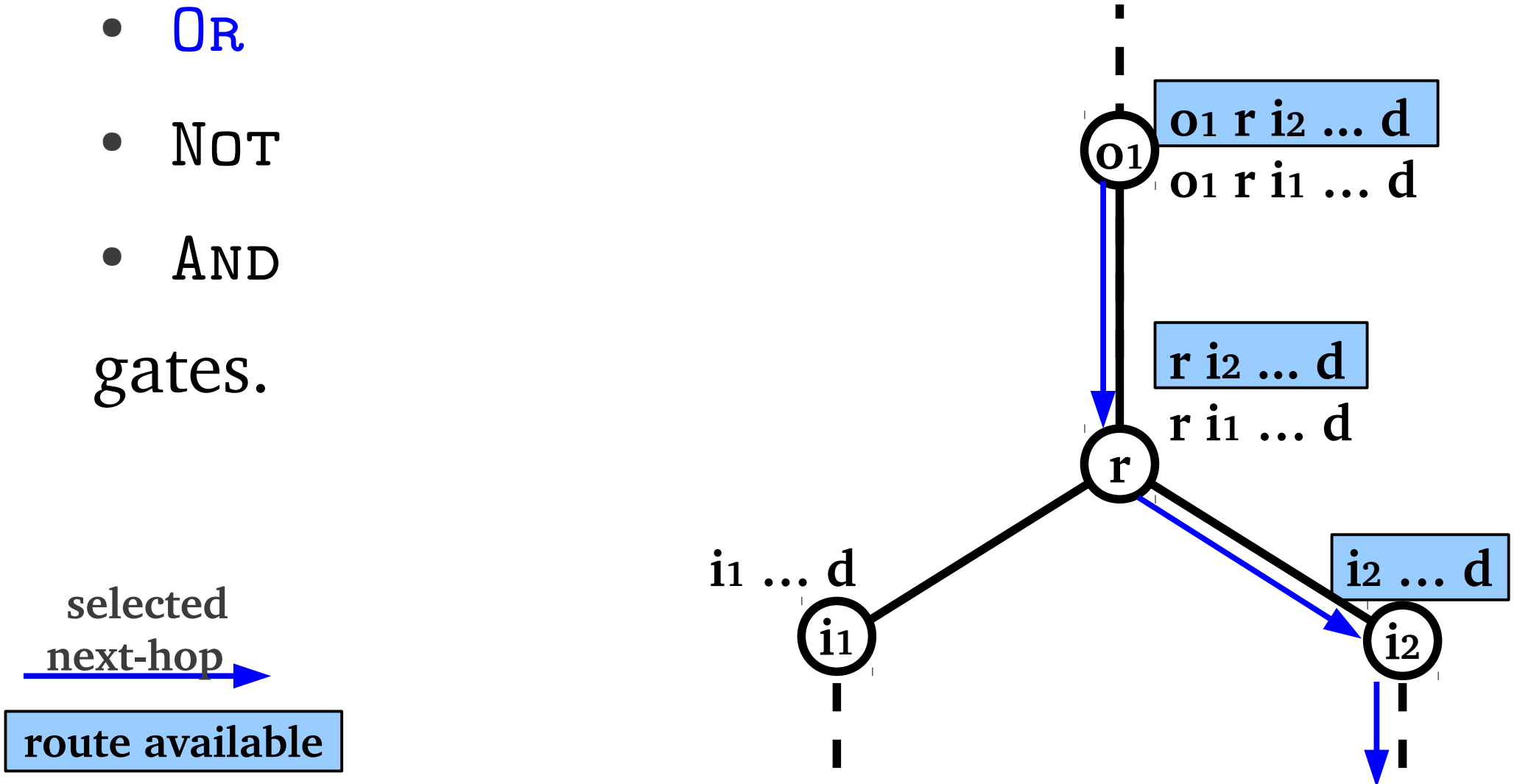
BGP and logic gates

And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.

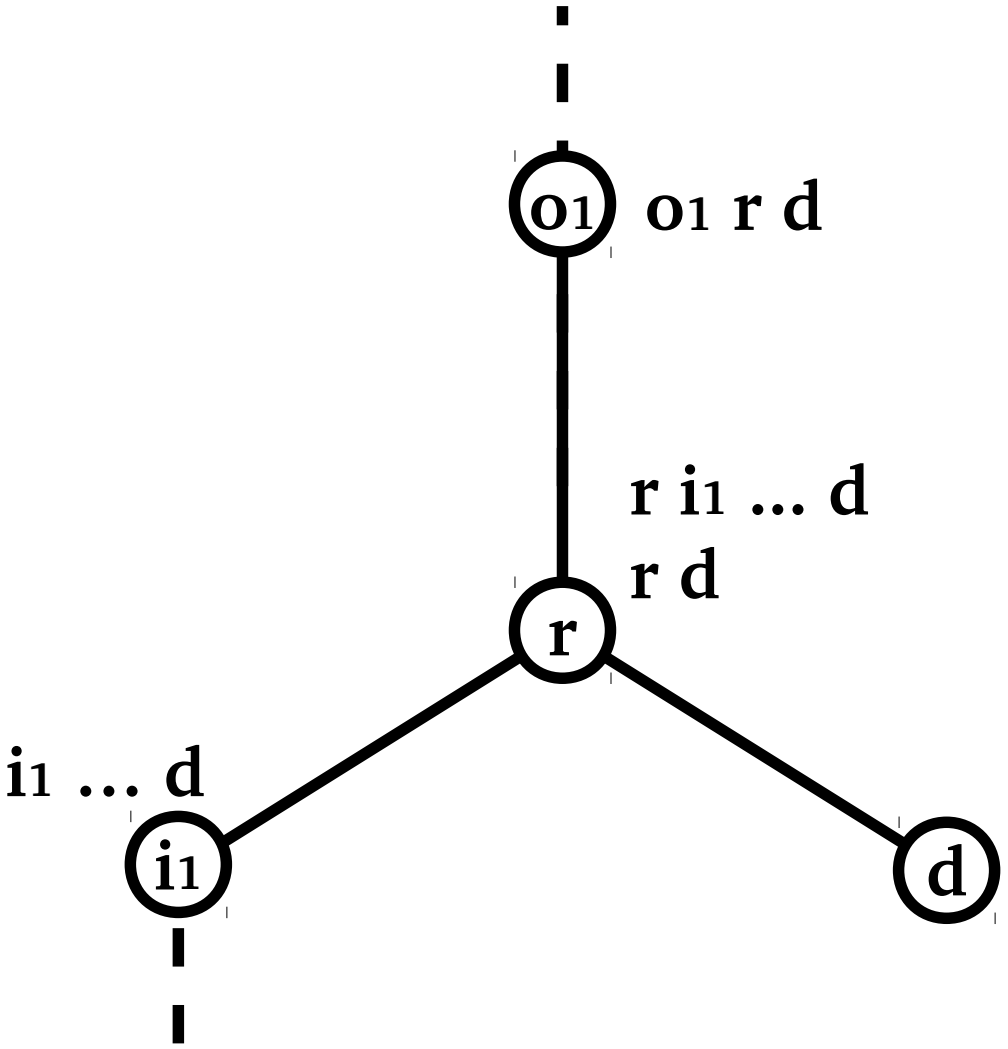


BGP and logic gates
AND, OR, NOT gates

it is possible to construct

- OR
- NOT
- AND

gates.

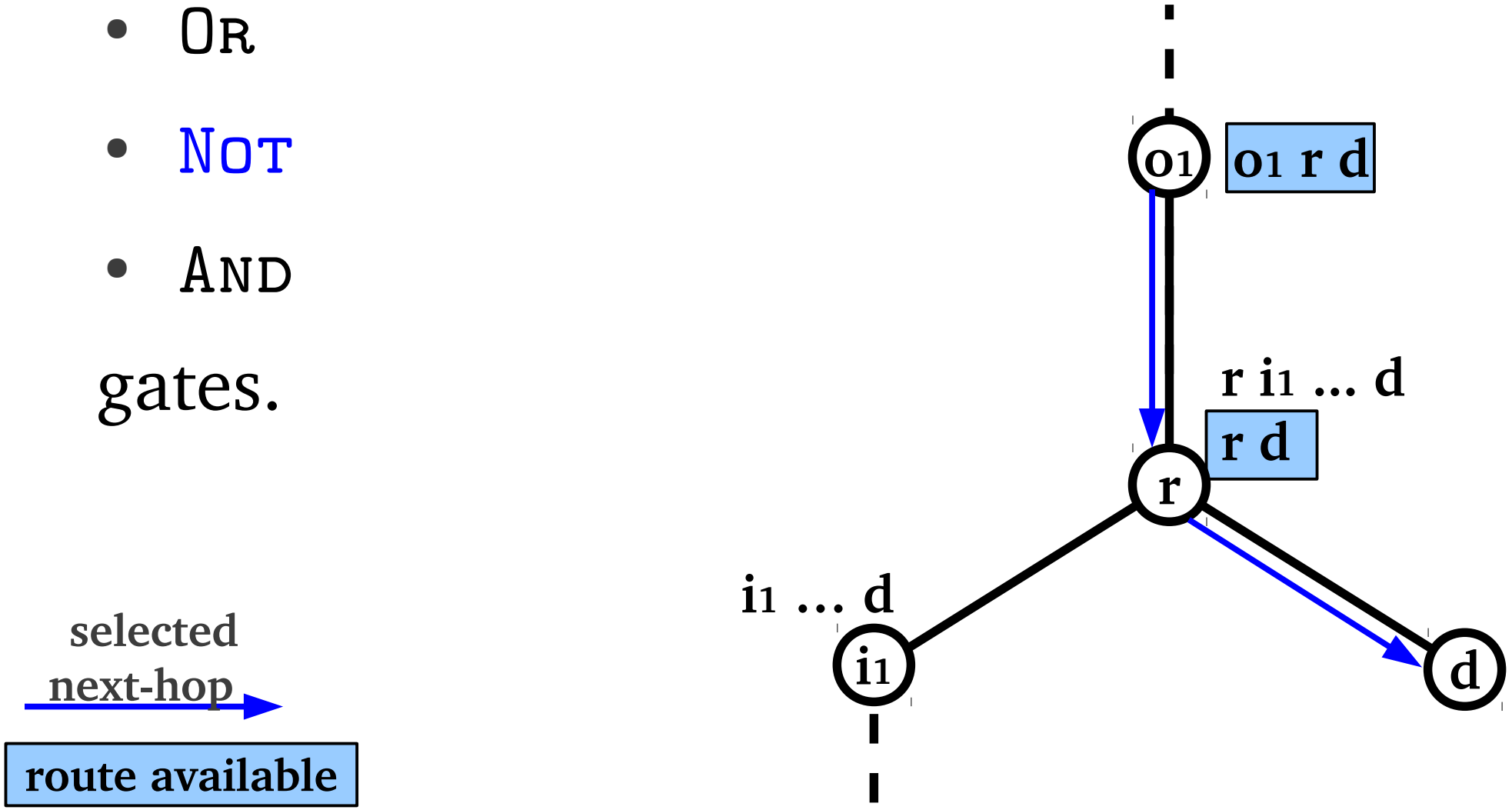


BGP and logic gates
And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.

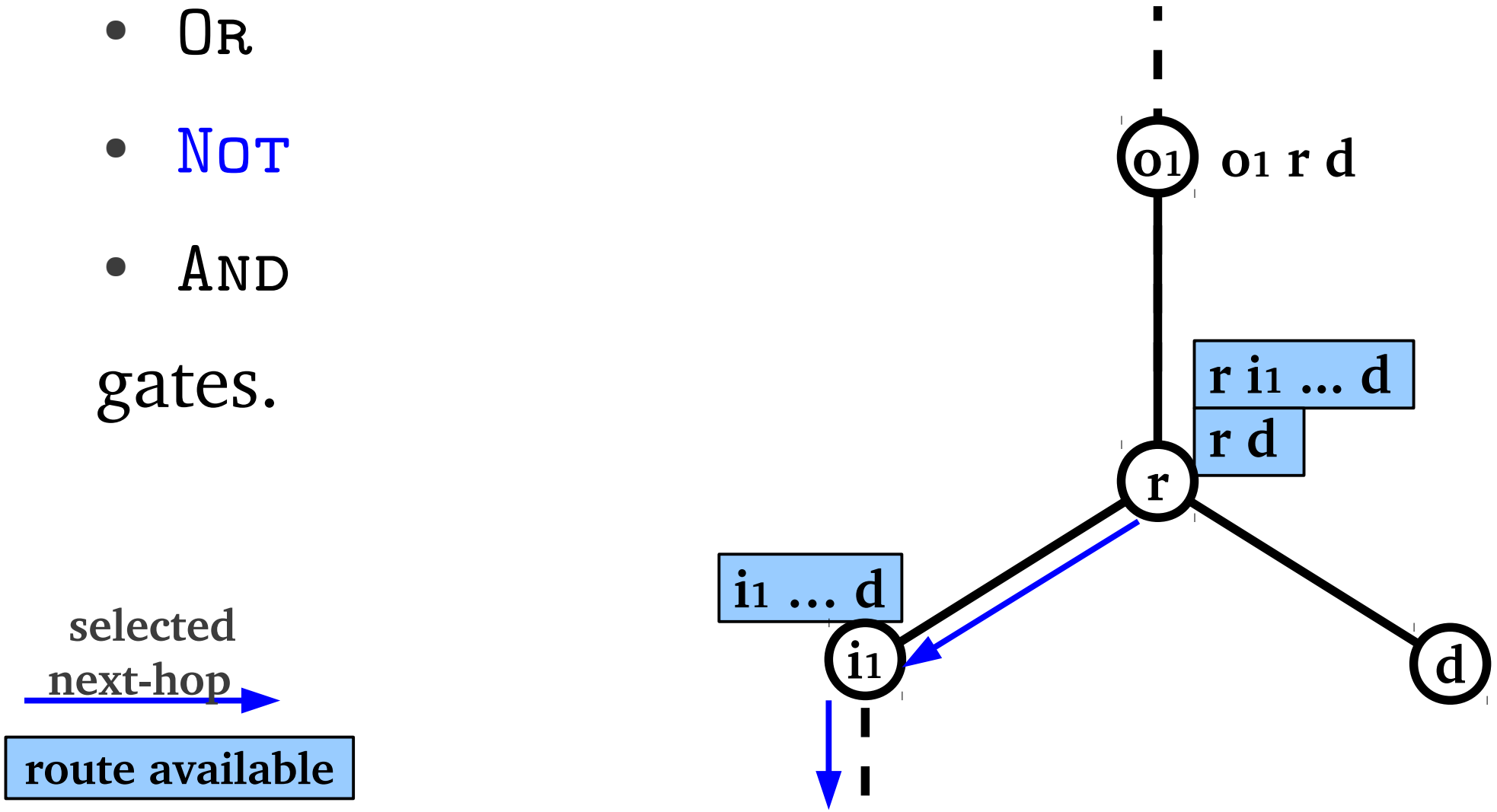


BGP and logic gates
And, Or, Not gates

it is possible to construct

- Or
- Not
- And

gates.

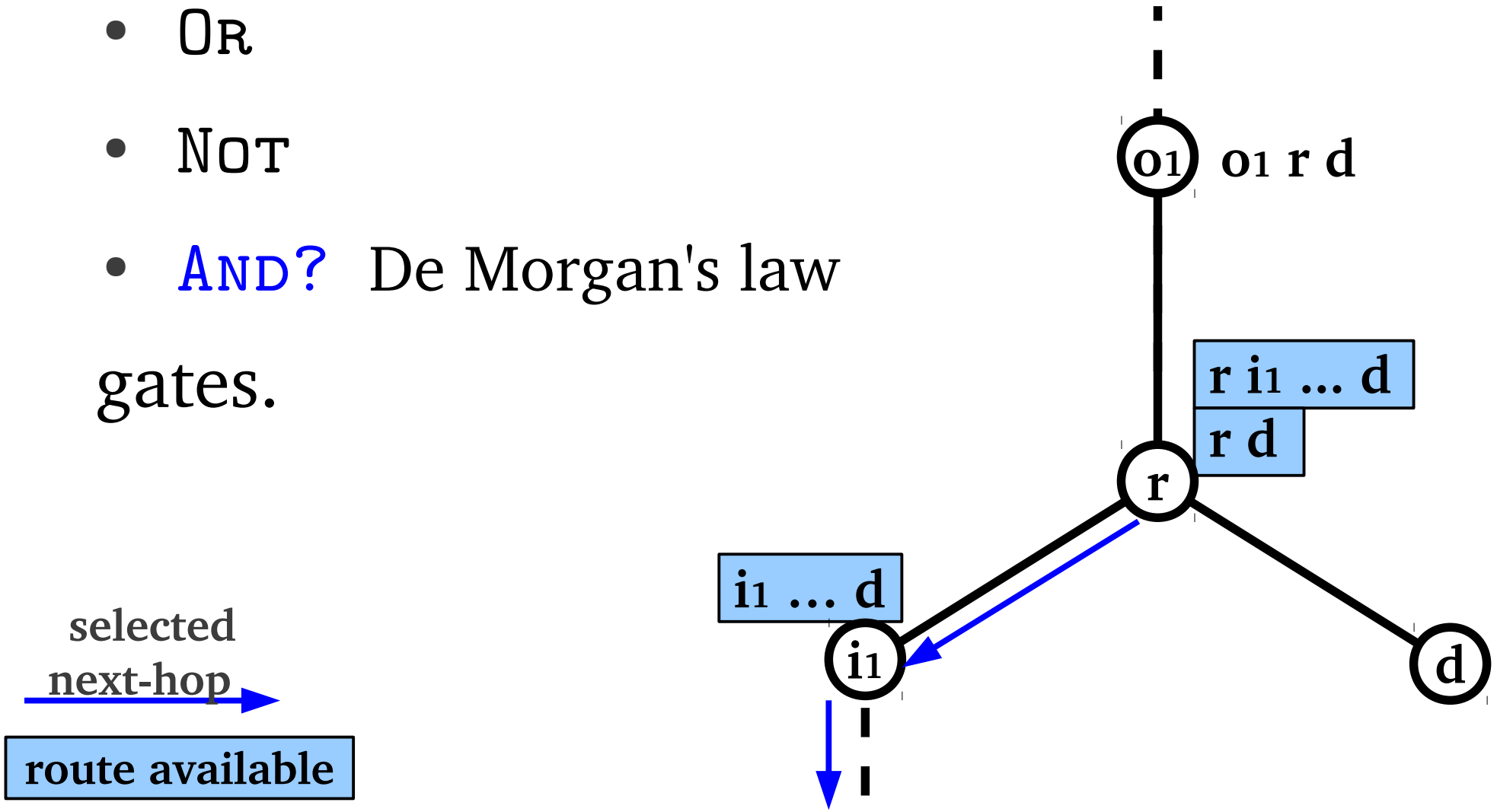


BGP and logic gates
AND, OR, NOT gates

it is possible to construct

- OR
- NOT
- AND? De Morgan's law

gates.



can we already derive something from this mapping?

yes, we can build any “combinational” logic circuit → many BGP stability problems are computational untractable (NP-hard):

- Reachability
- Solvability
- Trapped
- Unique
- Multiple
- … and many others

can we build sequential circuits?

combinational circuit → no loops

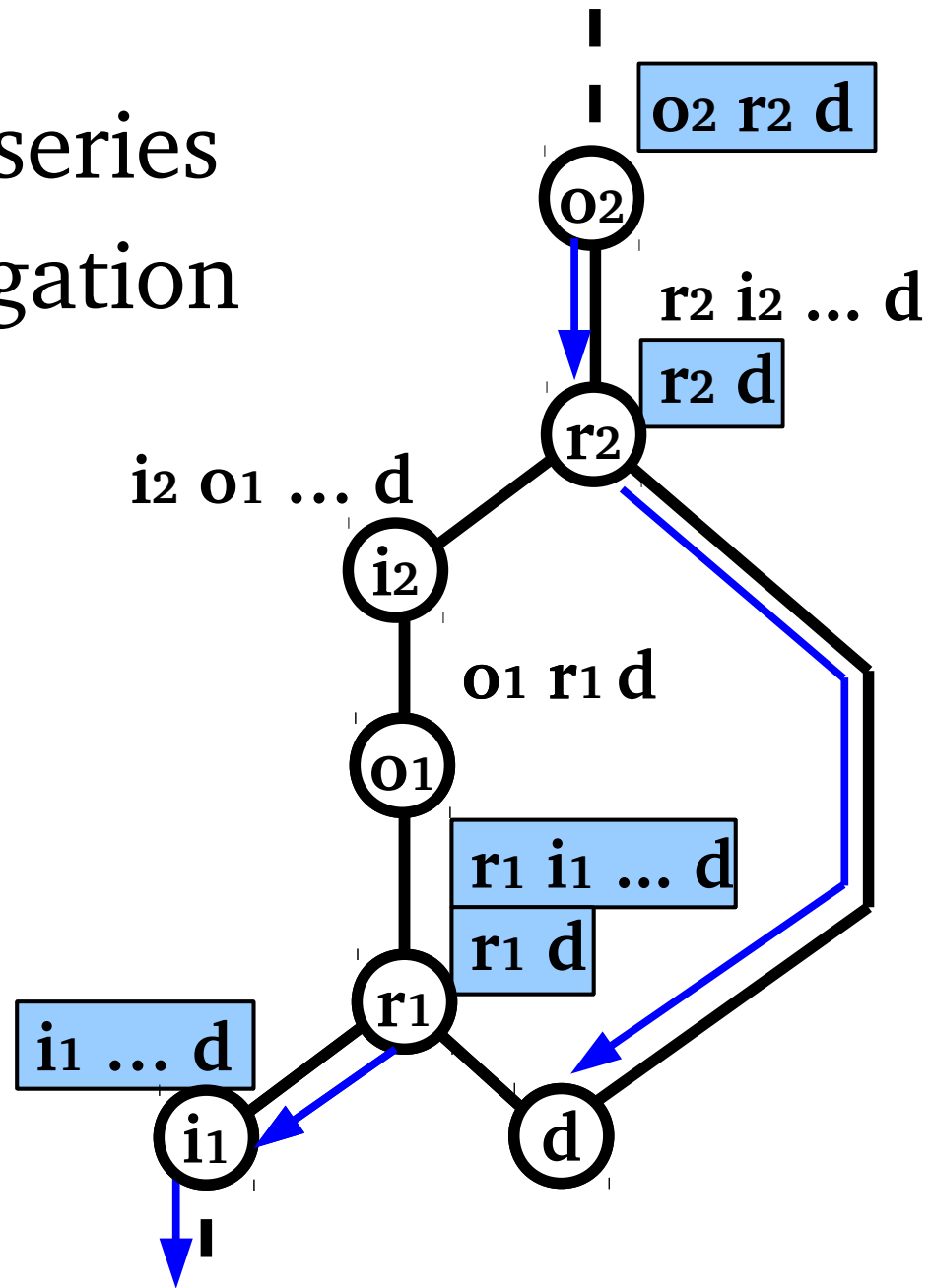
sequential circuit → feedback loops

two main issues:

- loop-avoidance mechanism of BGP
  - solution: “clean” routes using `NOT` gadgets
- backward propagation
  - solution: use filters

feedback loops: HUB gadget

- two NOT gadgets in series
- no backward propagation

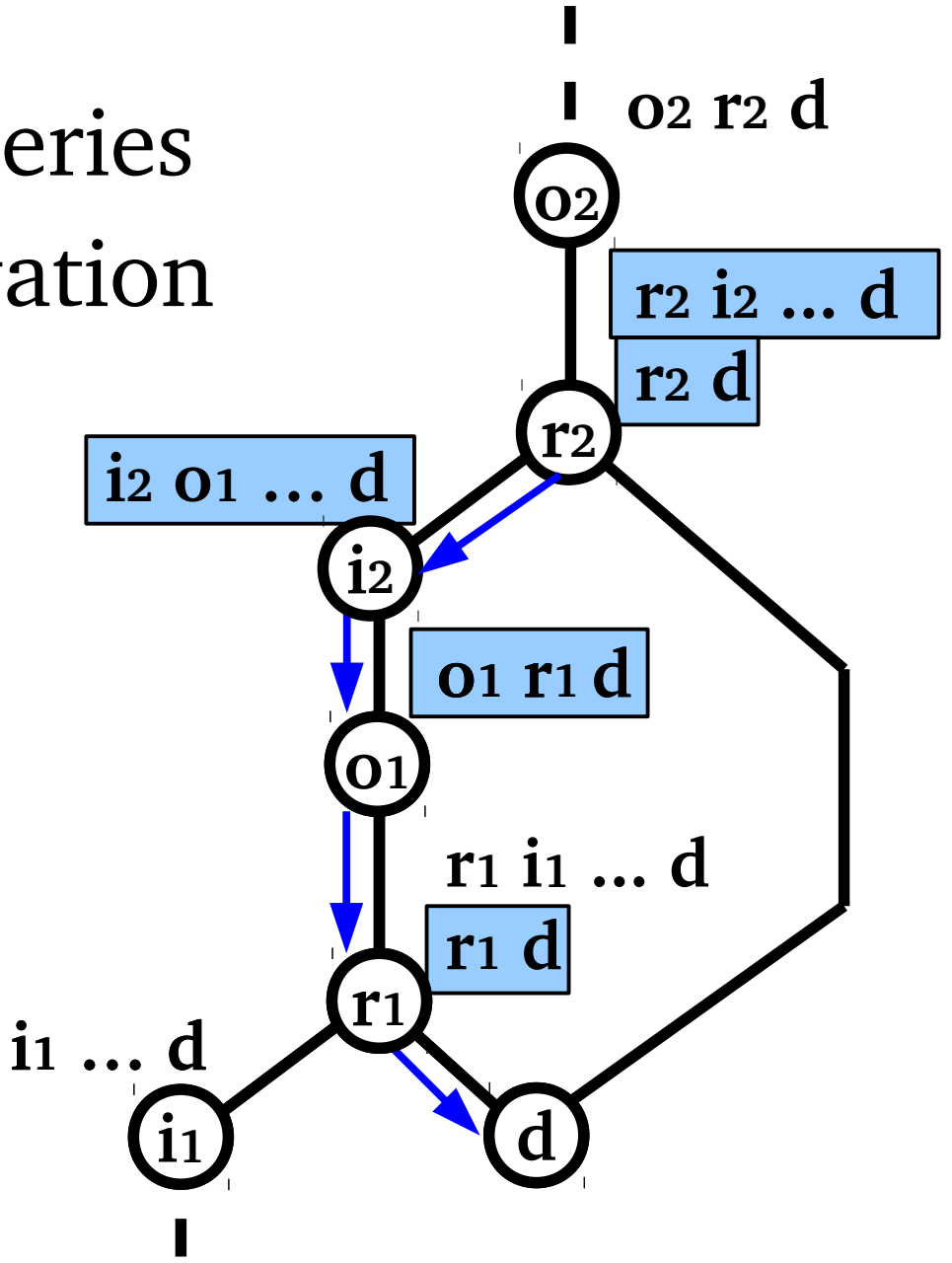


selected next-hop

route available

feedback loops: HUB gadget

- two NOT gadgets in series
- no backward propagation



selected next-hop

route available

BGP and Turing Machines

Turing Machines can be constructed using logic circuits with feedback loops
→ Turing Machines can be constructed using BGP routers! (clock signal implemented by a BAD-GADGET!)

Theorem. Determining whether an *infinite* BGP network does not oscillate is Turing-complete

Theorem. Determining whether a finite BGP network does not oscillate is PSPACE-hard

implications

- analyzing a simple control plane protocol is as hard as analyzing any computer program

- oscillation patterns of exponential length
→ extremely difficult to find them
    - are there any in the Internet?

- BGP dynamics problems cannot be expressed as SAT formulas → no SAT solvers

BGP restrictions where logic gates emulation is possible

- **Local-Transit policies:** ranking and filtering policies only based on neighbors

- **internal BGP (iBGP):** very limited ranking and filtering

- **2 out of 3 Gao-Rexford conditions:** "typical" customer-provider relationships that guarantee stability

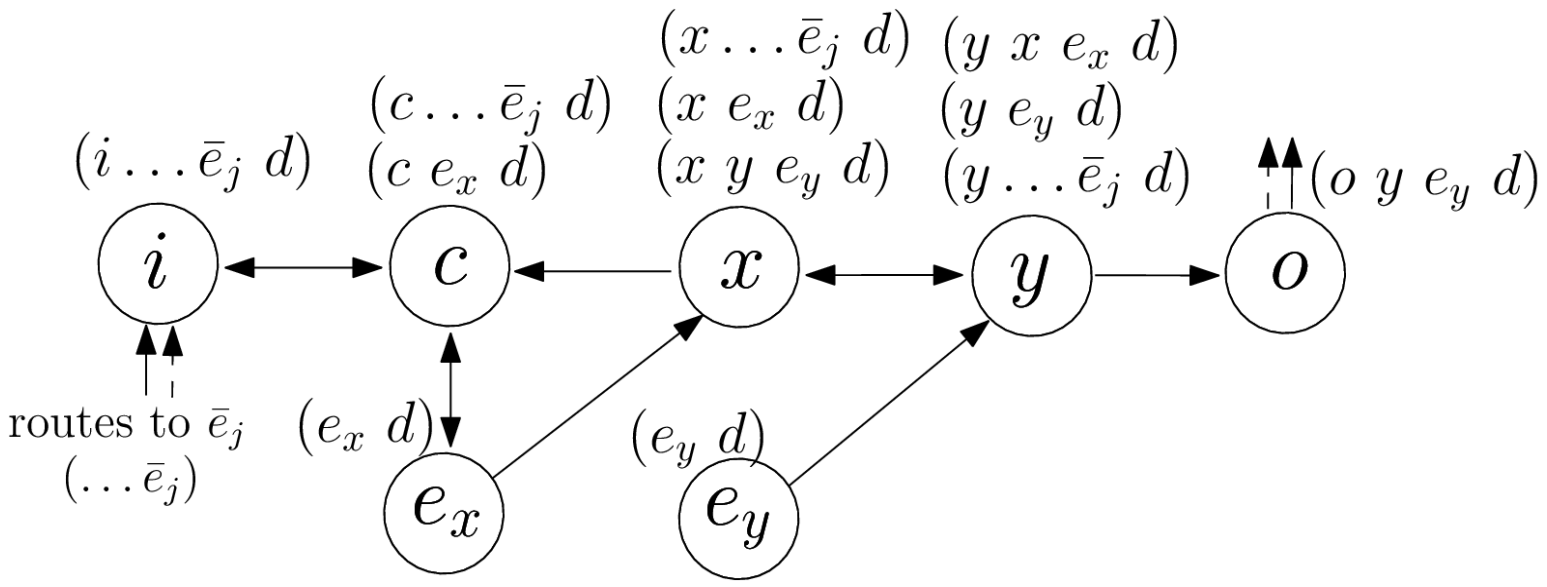
HUB gadget in iBGP

an observation from iBGP:

- the lower the expressiveness of the filtering expressiveness

→

the harder the construction of a HUB gadget



specific open problems

is filtering necessary?

- Local-Transit policies

is filtering sufficient?

- Local-Transit with a restricted export-all/filter-all filter

there is a still unclear tradeoff between ranking and filtering expressiveness

Beyond BGP: routing protocols based on several metrics

consider a routing protocol (called Metric-DV):

- distance-vector
- each route has at least three metrics
  - e.g. path-length, bandwidth
- routers rank routes based on their preferred metric
- no filtering

is it still still possible to emulate logic circuits?

- yes, but a more complex mapping is needed

specific open problems

is filtering necessary?

- Local-Transit policies
- **Metric-DV with less than 3 metrics**

is filtering enough?

- Local-Transit with an export-all/filter-all filter

there is a still unclear tradeoff between ranking and filtering expressiveness

conclusion and final remarks

- we shown a **simple** and strong connection between logic circuits and BGP dynamics

- we leveraged hardness results from logic circuits to prove hardness result in BGP
  - an effective tool for proving computational hardness

- can we learn something more from other fields? (e.g. timed automata theory, π calculus, game theory, asynchronous circuit design)

thank you!