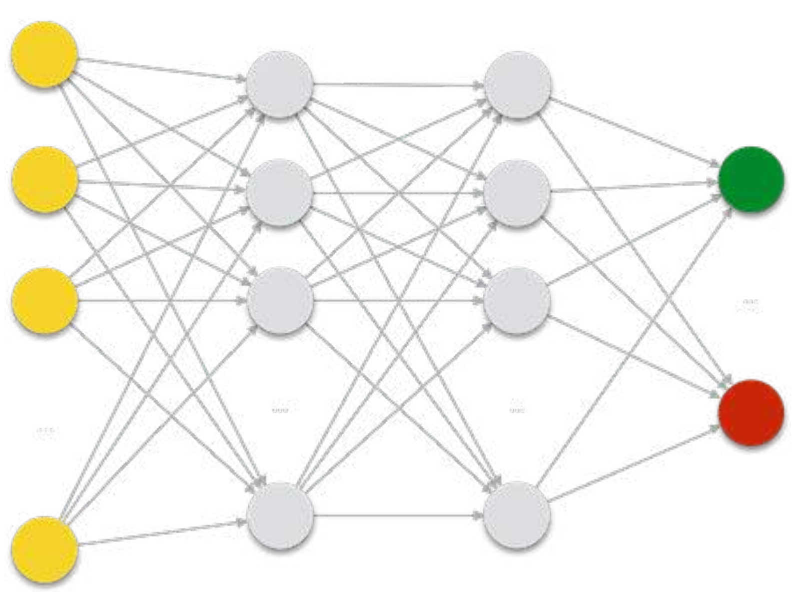
Static Analysis of Neural Networks

MPRI 2-6: Abstract Interpretation, Application to Verification and Static Analysis

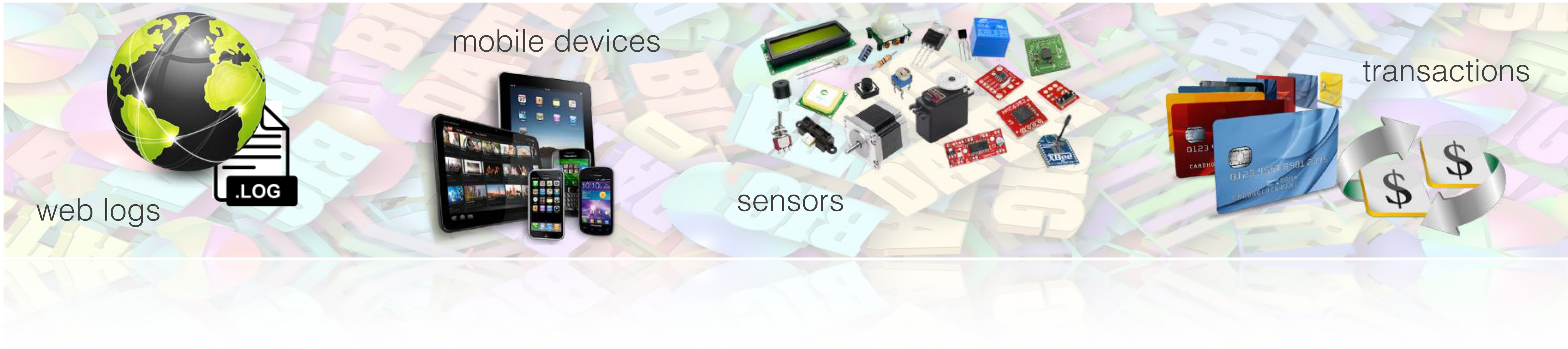


Caterina Urban

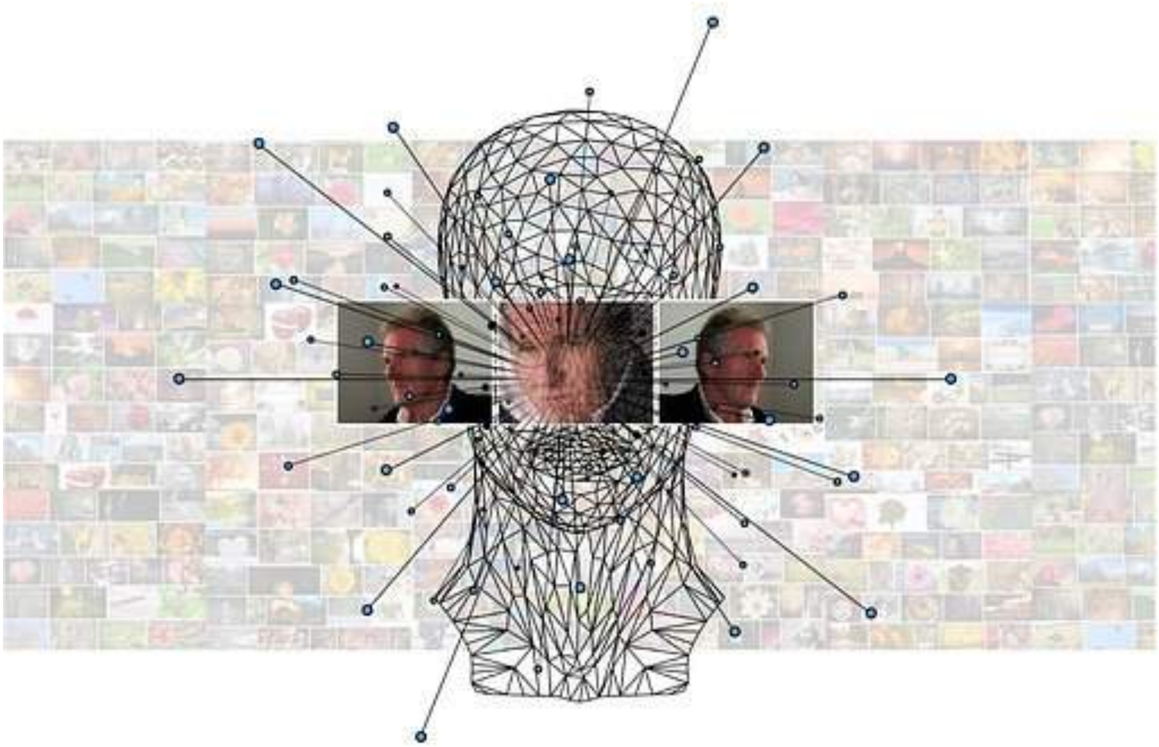
18 December 2020

Year 2020-2021

Availability of vast amounts of **Data**

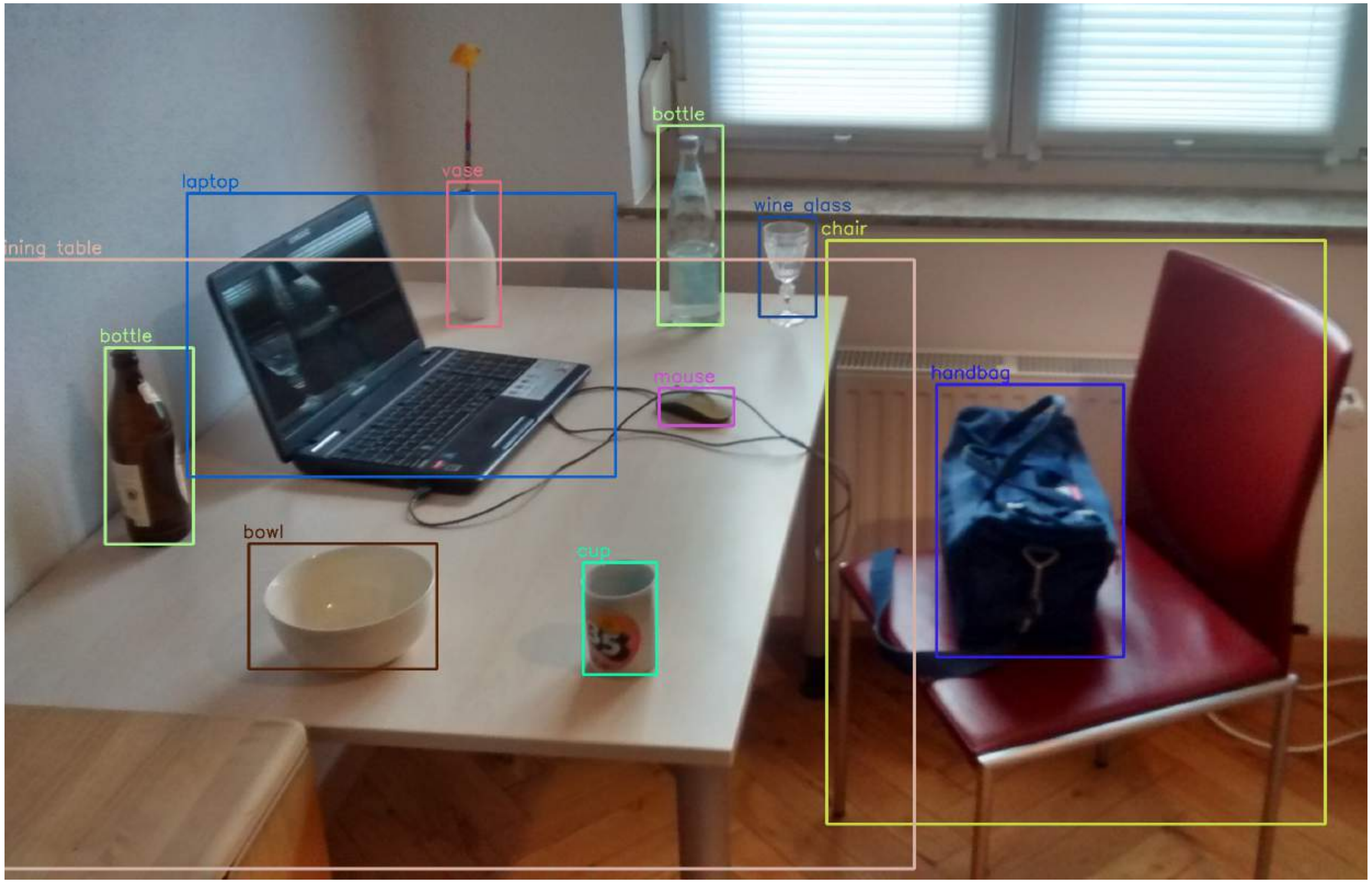


Recent advances in **Machine Learning**



Machine Learning Revolution

Computer software able to efficiently and **autonomously perform tasks** that are difficult or even *impossible* to design using explicit programming



Examples: object recognition, image classification, speech recognition, etc.

ML in Safety-Critical Applications

Enables **new functions that could not be envisioned before**

Self-Driving Cars

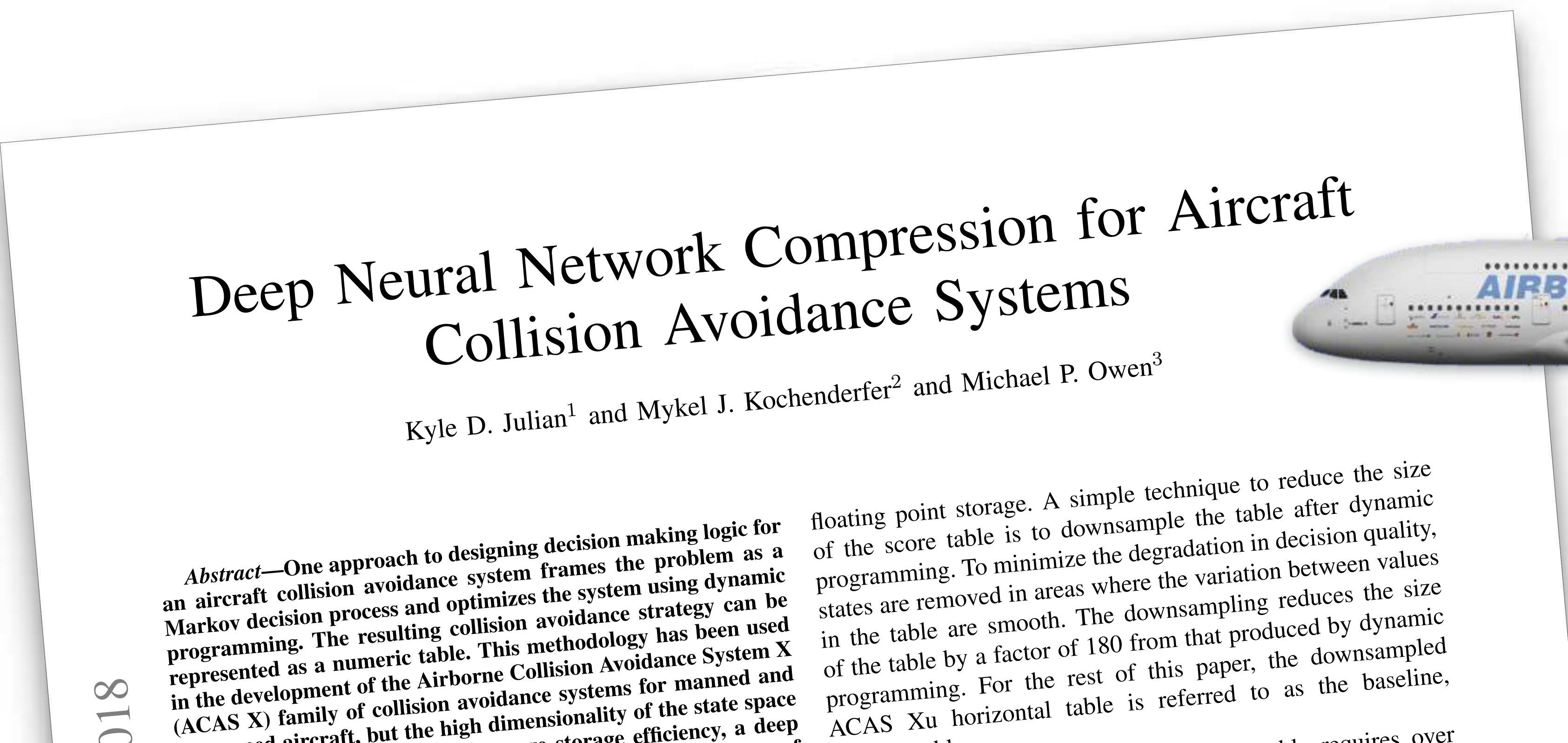
Image-Based Taxiing, Takeoff, Landing

Aircraft Voice Control

ML in Safety-Critical Applications

Approximates complex systems and **automates decision-making**

Diagnosis and Drug Discovery



Deep Neural Network Compression for Aircraft Collision Avoidance Systems

Kyle D. Julian[1] and Mykel J. Kochenderfer[2] and Michael P. Owen[3]

Abstract—One approach to designing decision making logic for an aircraft collision avoidance system frames the problem as a Markov decision process and optimizes the system using dynamic programming. The resulting collision avoidance strategy can be represented as a numeric table. This methodology has been used in the development of the Airborne Collision Avoidance System X (ACAS X) family of collision avoidance systems for manned and unmanned aircraft, but the high dimensionality of the state space

floating point storage. A simple technique to reduce the size of the score table is to downsample the table after dynamic programming. To minimize the degradation in decision quality, states are removed in areas where the variation between values in the table are smooth. The downsampling reduces the size of the table by a factor of 180 from that produced by dynamic programming. For the rest of this paper, the downsampled ACAS Xu horizontal table is referred to as the baseline, original table.

Aircraft Collision Avoidance

ML in Safety-Critical Applications

[1] STAT+ [2]

IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show

By Casey Ross[3] @caseymross[4] *and* Ike Swetlitz

July 25, 2018

A self-driving Uber ran a red light last December, contrary to company claims

Internal documents reveal that the car was at fault

By Andrew Liptak | @AndrewLiptak | Feb 25, 2017, 11:08am EST

Feds Say Self-Driving Uber SUV Did Not Recognize Jaywalking Pedestrian In Fatal Crash

Richard Gonzales November 7, 201910:57 PM ET

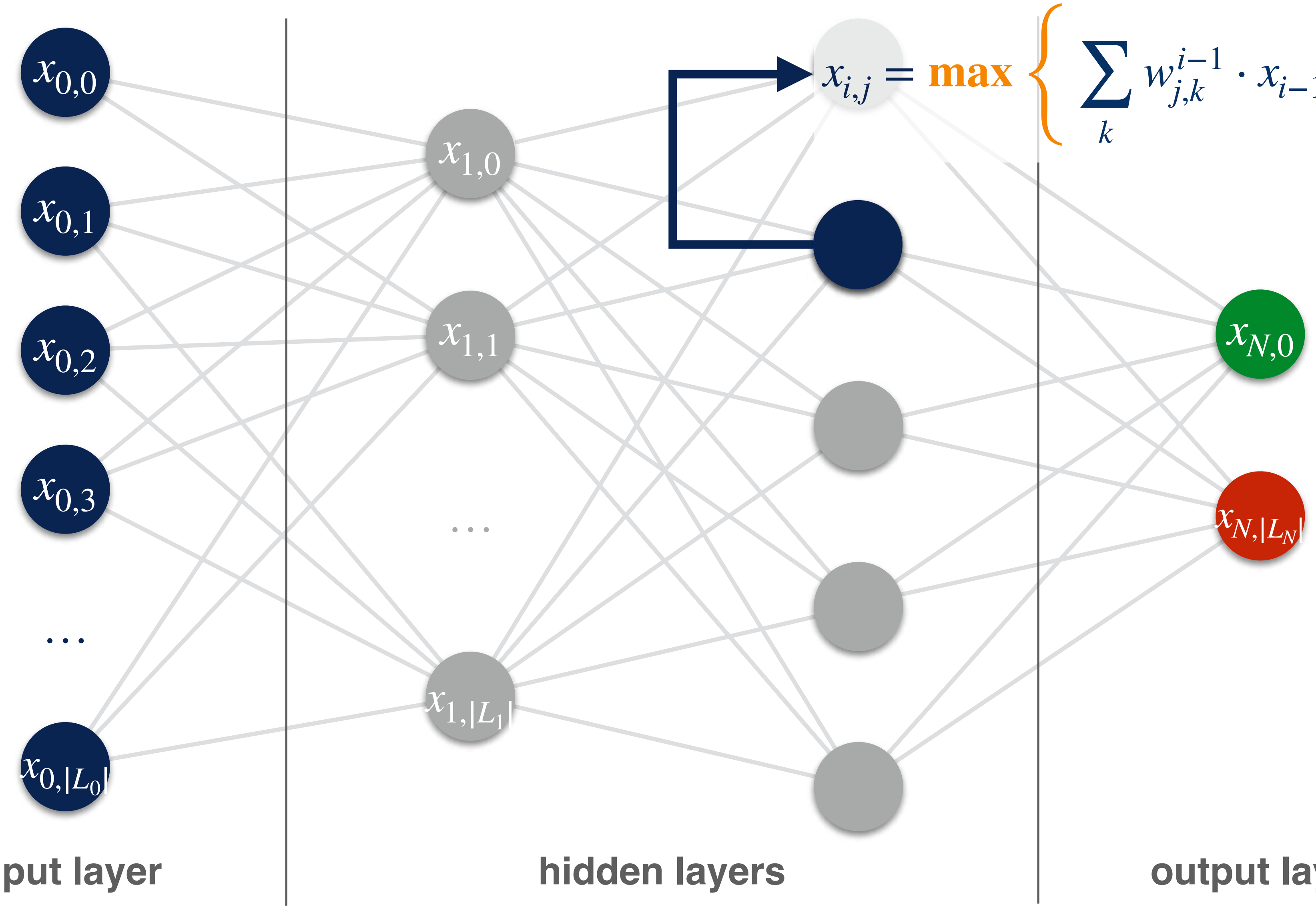
Neural Networks

Neural Networks

Feed-Forward **Fully-Connected** Neural Networks with **ReLU Activation Functions**

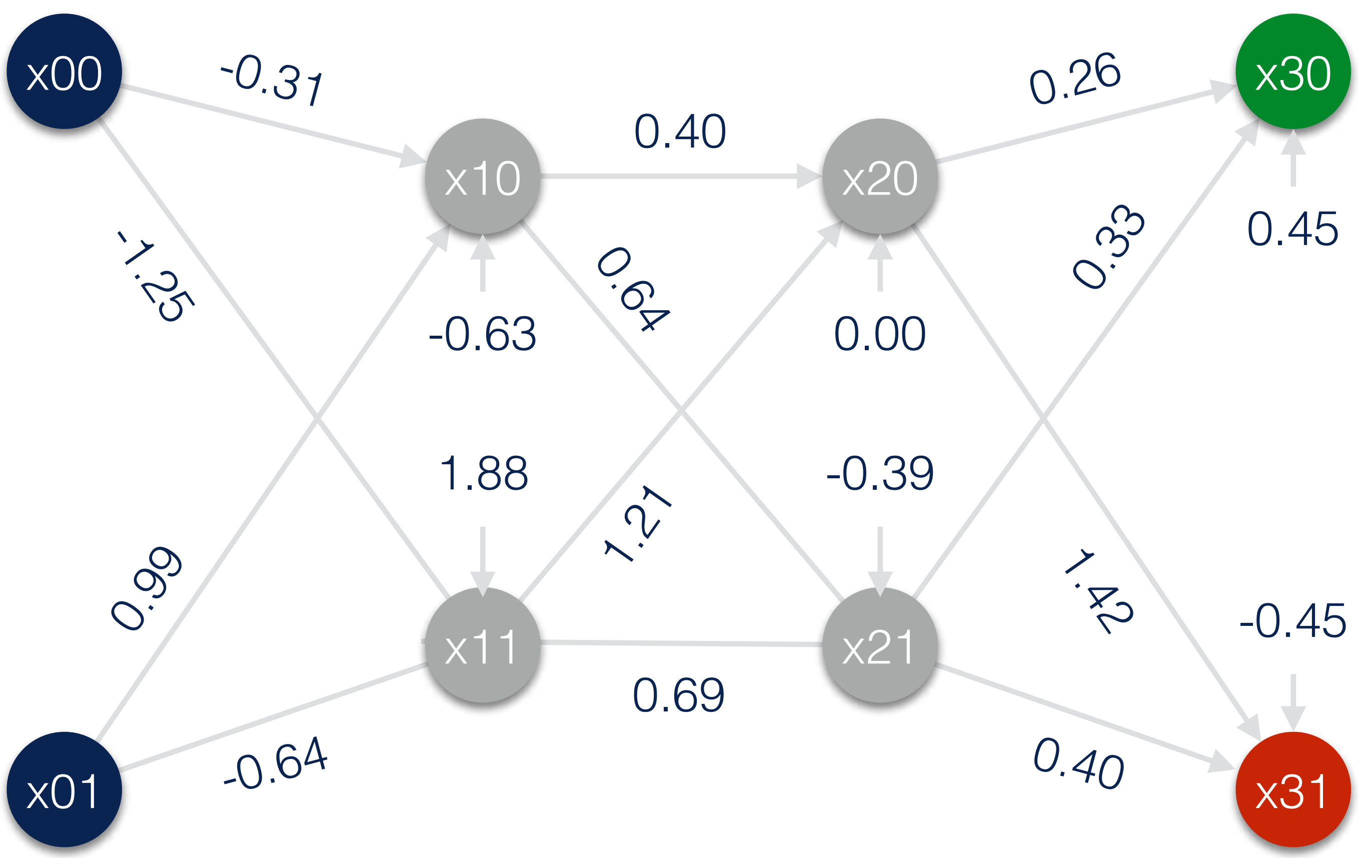
Rectified Linear Unit (ReLU)

$$x_{i,j} = \mathbf{max} \left\{ \sum_k w_{j,k}^{i-1} \cdot x_{i-1,k} + b_{i,j}, \ \mathbf{0} \right\}$$



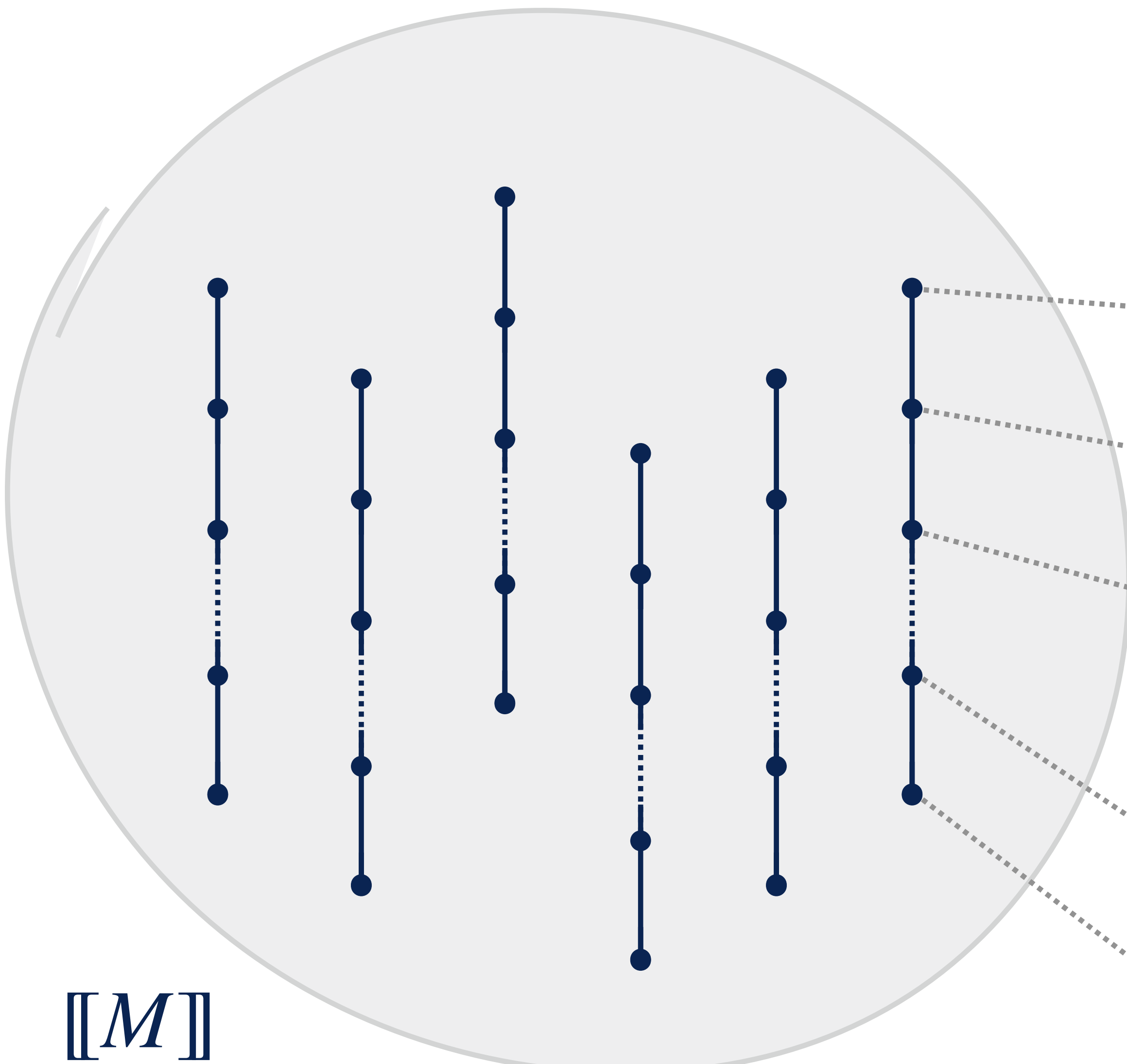
output $\max_j \ x_{N,j}$

Feed-Forward Fully-Connected ReLU Networks as Programs




```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

Maximal Trace Semantics



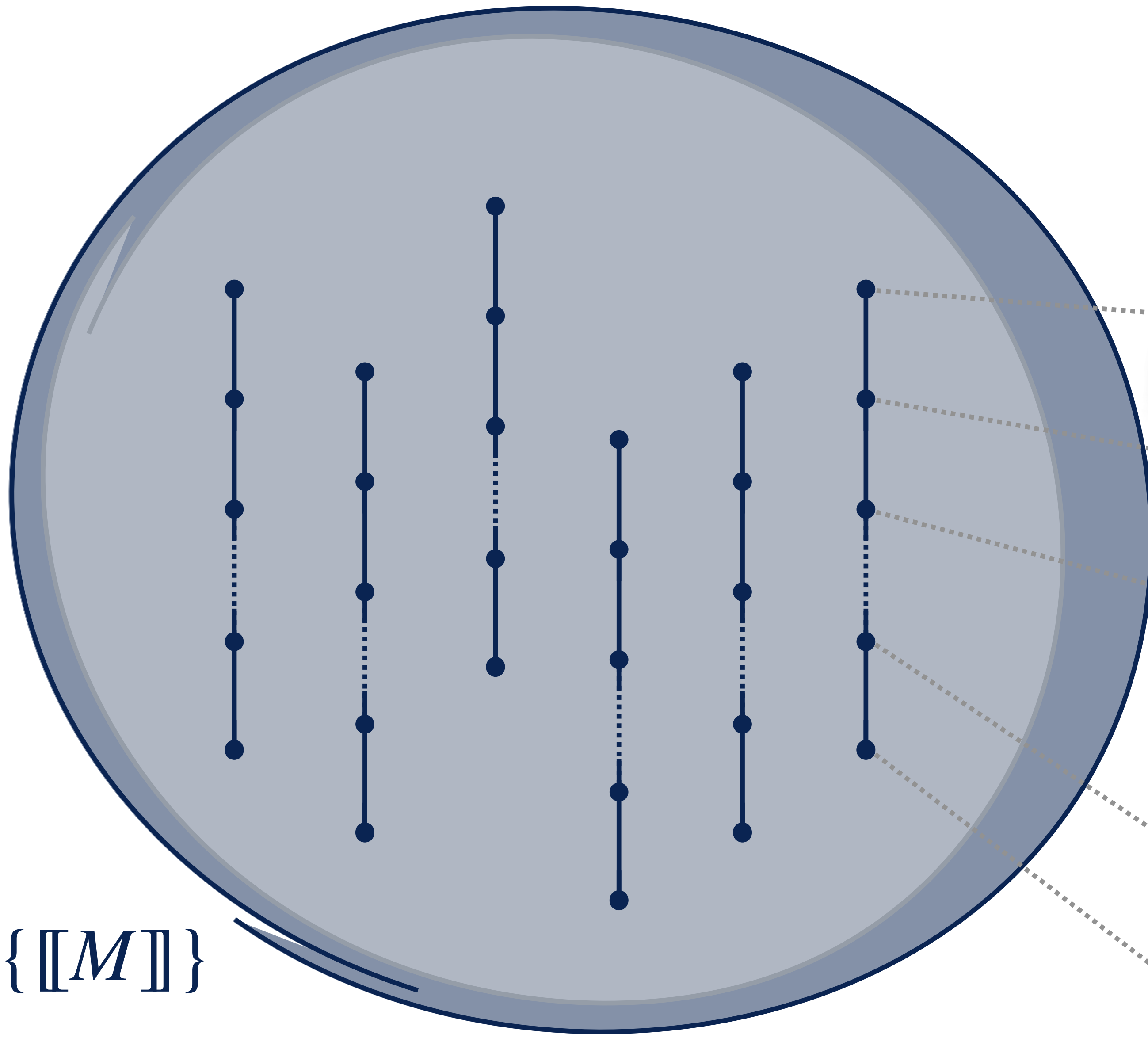
$[\![M]\!]$

M

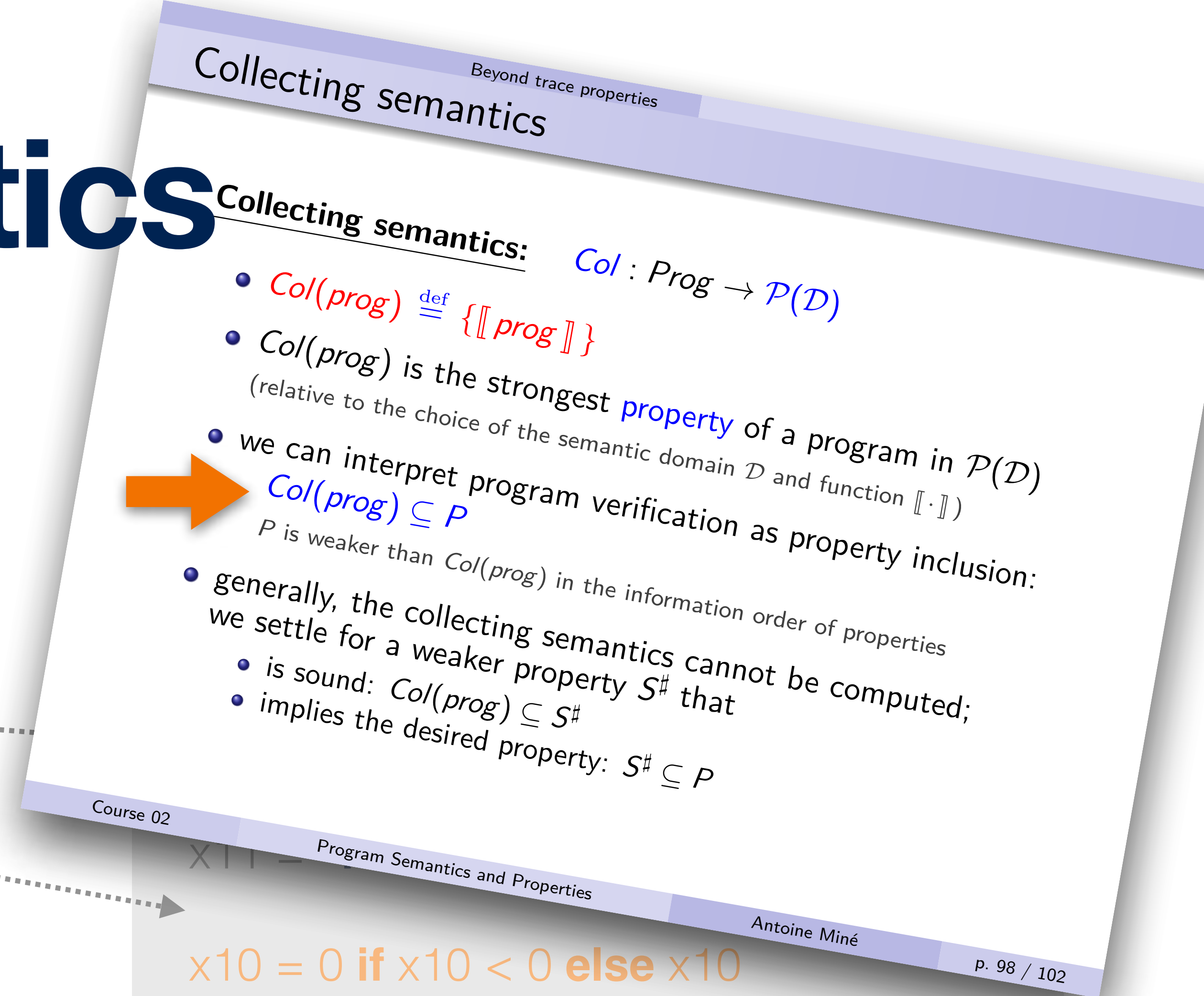
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

Neural Network Verification

Collecting Semantics



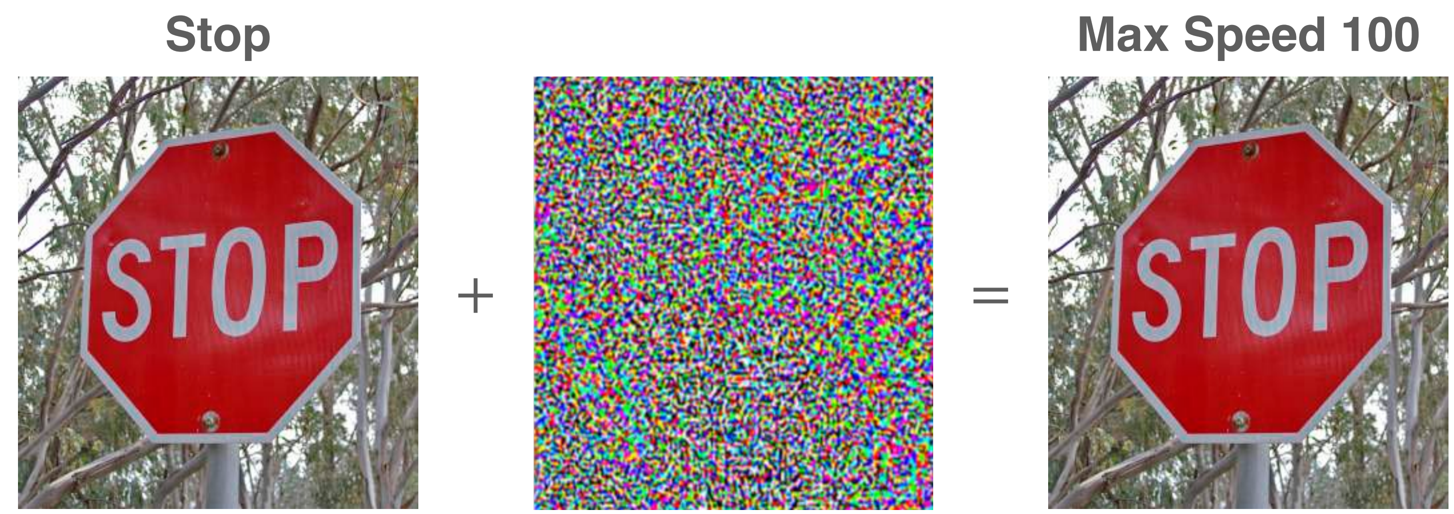
$\{\llbracket M \rrbracket\}$




```
x11 =

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

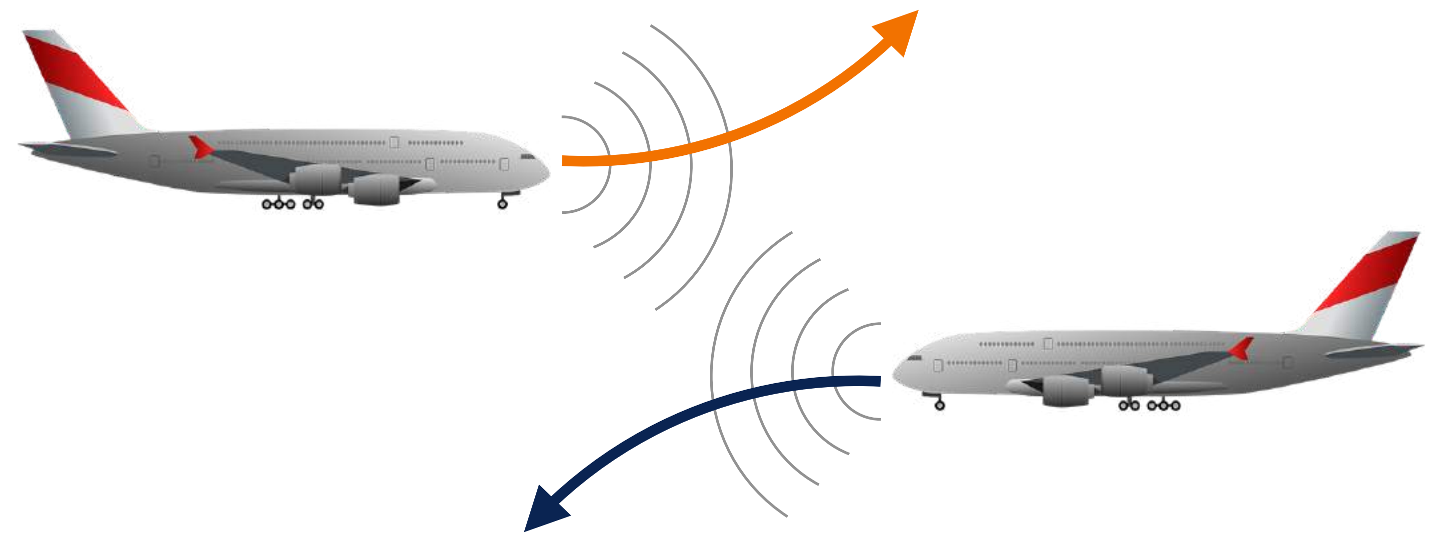
Stability

Goal G3 in [Kurd03]



Safety

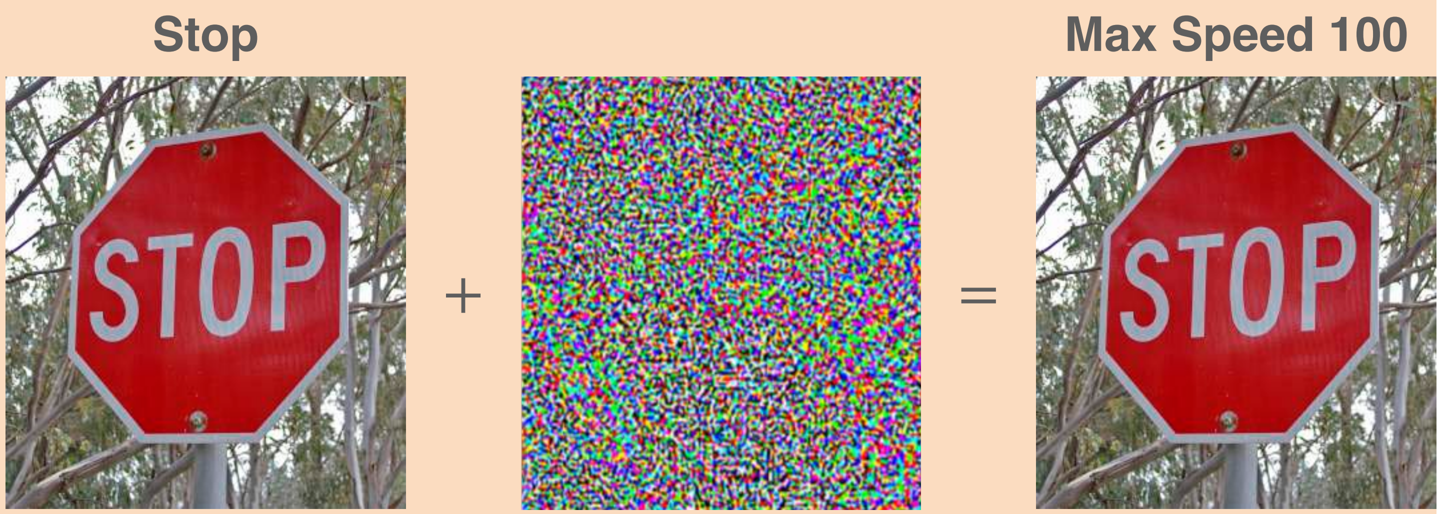
Goal G4 in [Kurd03]



Fairness

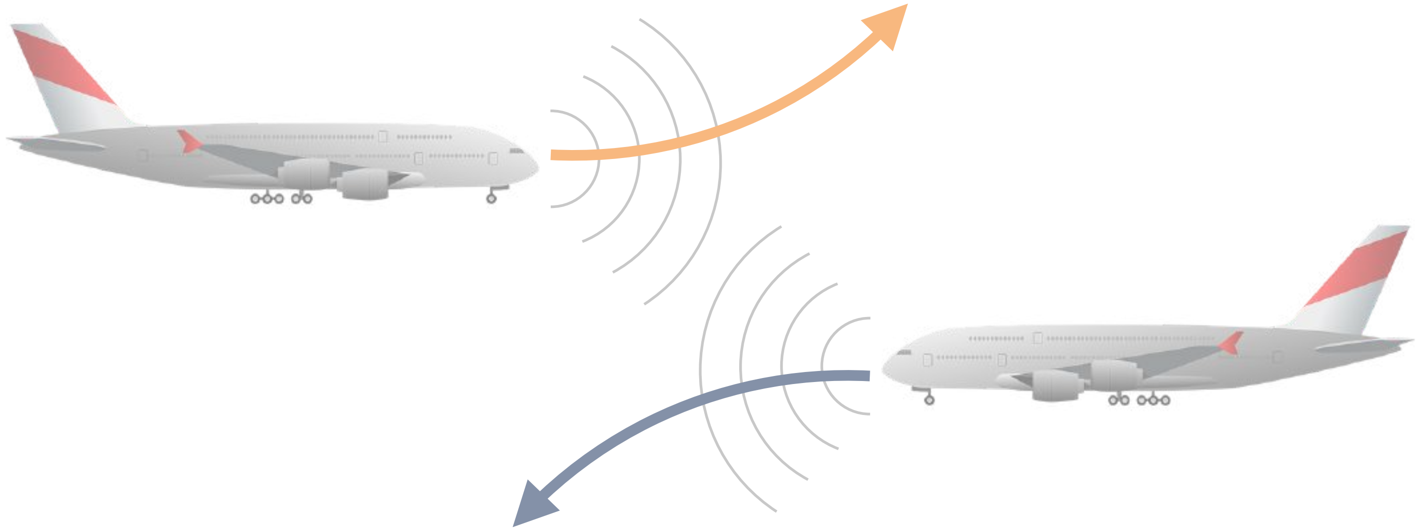
Stability

Goal G3 in [Kurd03]

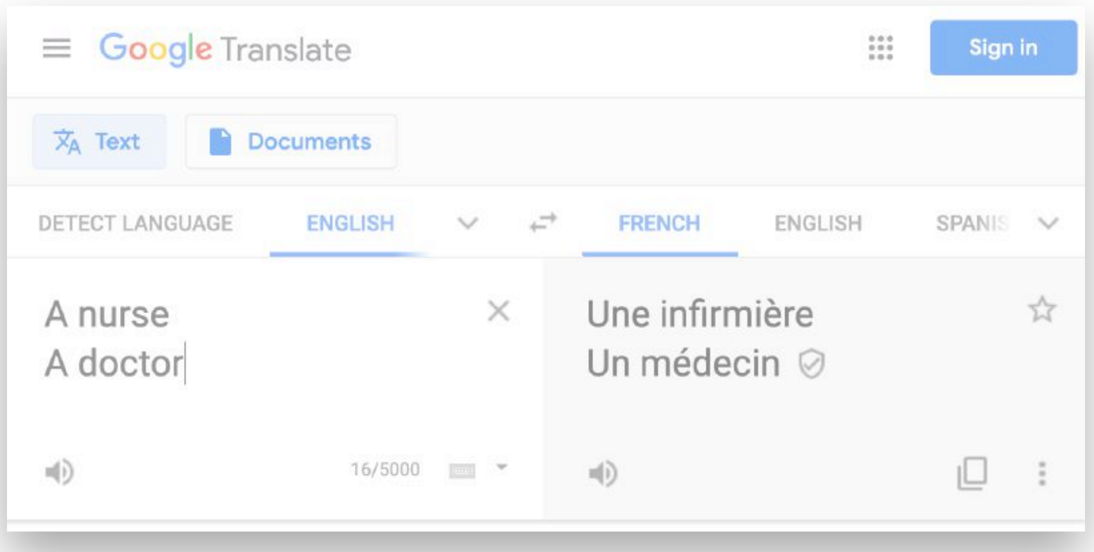


Safety

Goal G4 in [Kurd03]

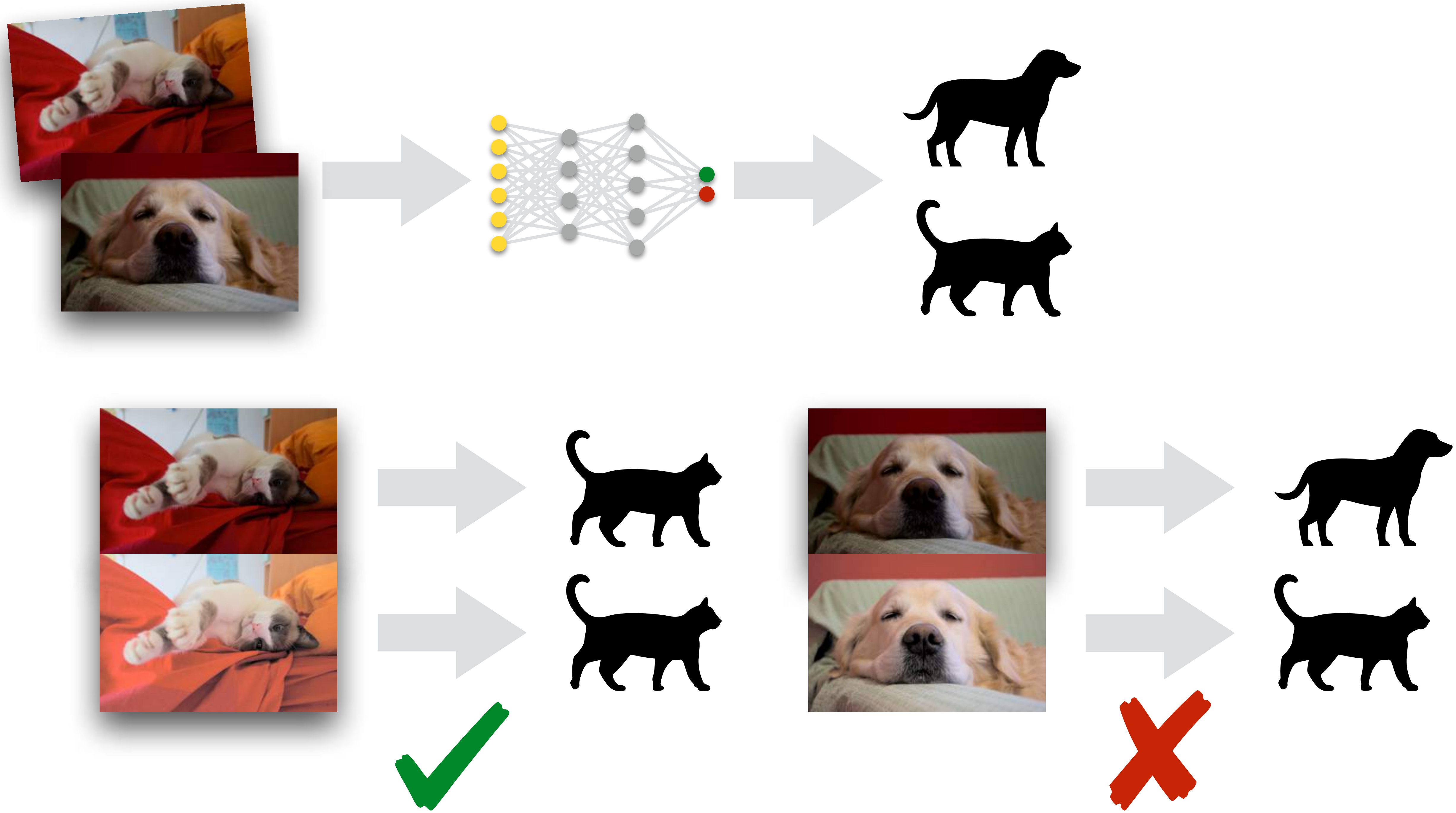


Fairness



Local Stability

The **classification** is **unaffected by small input perturbations**



Local Stability

Distance-Based Perturbations

$P_{\delta,\epsilon}(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{x}' \in \mathcal{R}^{|L_0|} \mid \delta(\mathbf{x}, \mathbf{x}') \leq \epsilon\}$

Example (L_∞ distance): $P_{\infty,\epsilon}(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{x}' \in \mathcal{R}^{|L_0|} \mid \max_i |\mathbf{x}_i - \mathbf{x}'_i| \leq \epsilon\}$

$$\mathcal{R}_{\mathbf{x}}^{\delta,\epsilon} \stackrel{\text{def}}{=} \{[\![M]\!] \in \mathcal{P}(\Sigma^*) \mid \text{STABLE}_{\mathbf{x}}^{\delta,\epsilon}([\![M]\!])\}$$

$\mathcal{R}_{\mathbf{x}}^{\delta,\epsilon}$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that are **stable** in the neighborhood $P_{\delta,\epsilon}(\mathbf{x})$ of a given input $\mathbf{x}$

$$\begin{aligned}\text{STABLE}_{\mathbf{x}}^{\delta,\epsilon}([\![M]\!]) \stackrel{\text{def}}{=} \forall t \in [\![M]\!] \colon & (\exists t' \in [\![M]\!] \colon \forall 0 \leq i \leq |L_0| \colon t'_0(x_{0,i}) = \mathbf{x}_i) \\ & \wedge (\exists \mathbf{x}' \in P_{\delta,\epsilon}(\mathbf{x}) \colon \forall 0 \leq i \leq |L_0| \colon t_0(x_{0,i}) = \mathbf{x}'_i) \\ & \Rightarrow \max_j t_\omega(x_{N,j}) = \max_j t'_\omega(x_{N,j})\end{aligned}$$

Theorem

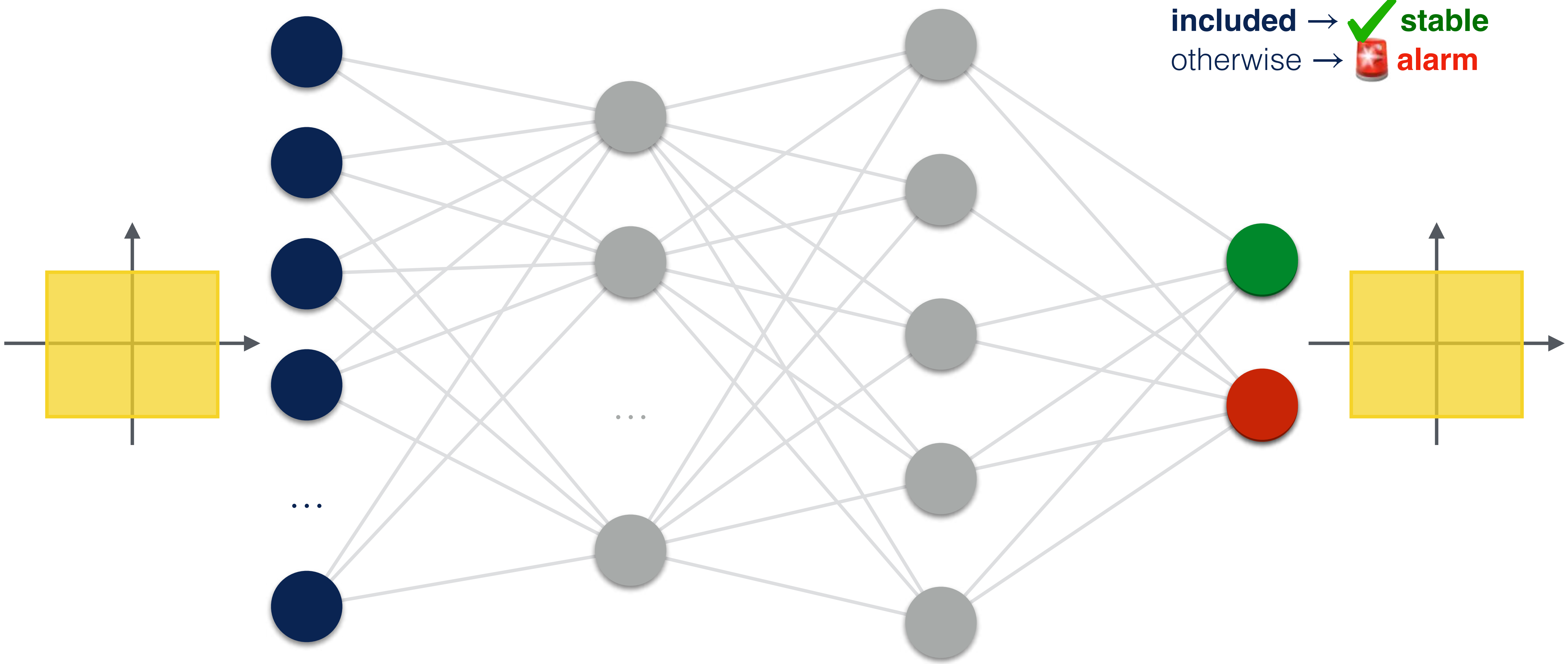
$$M \models \mathcal{R}_{\mathbf{x}}^{\delta,\epsilon} \Leftrightarrow \{[\![M]\!]\} \subseteq \mathcal{R}_{\mathbf{x}}^{\delta,\epsilon}$$

Corollary

$$M \models \mathcal{R}_{\mathbf{x}}^{\delta,\epsilon} \Leftrightarrow [\![M]\!] \subseteq \bigcup \mathcal{R}_{\mathbf{x}}^{\delta,\epsilon}$$

Numerical Abstractions

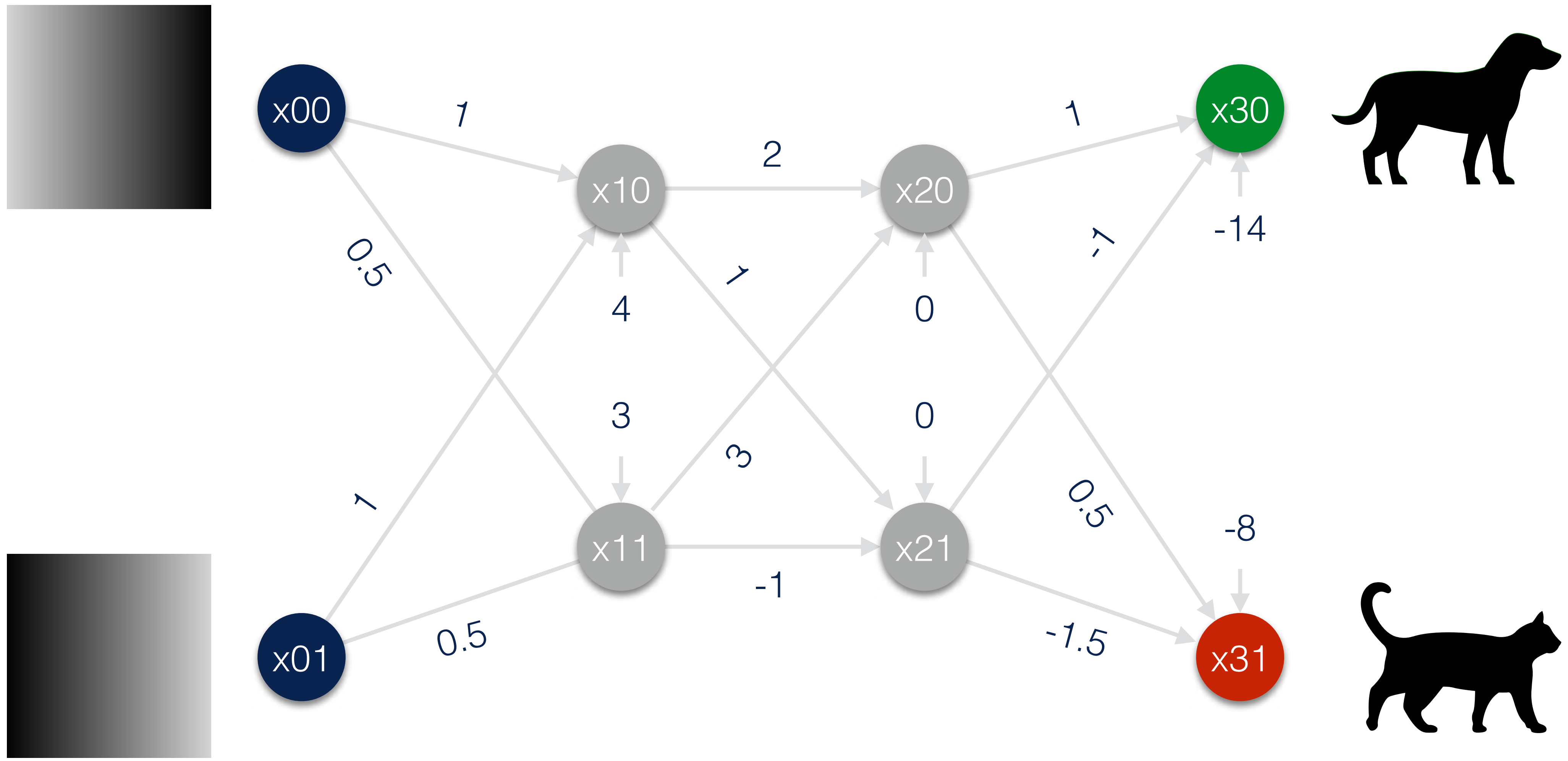
Forward Analysis



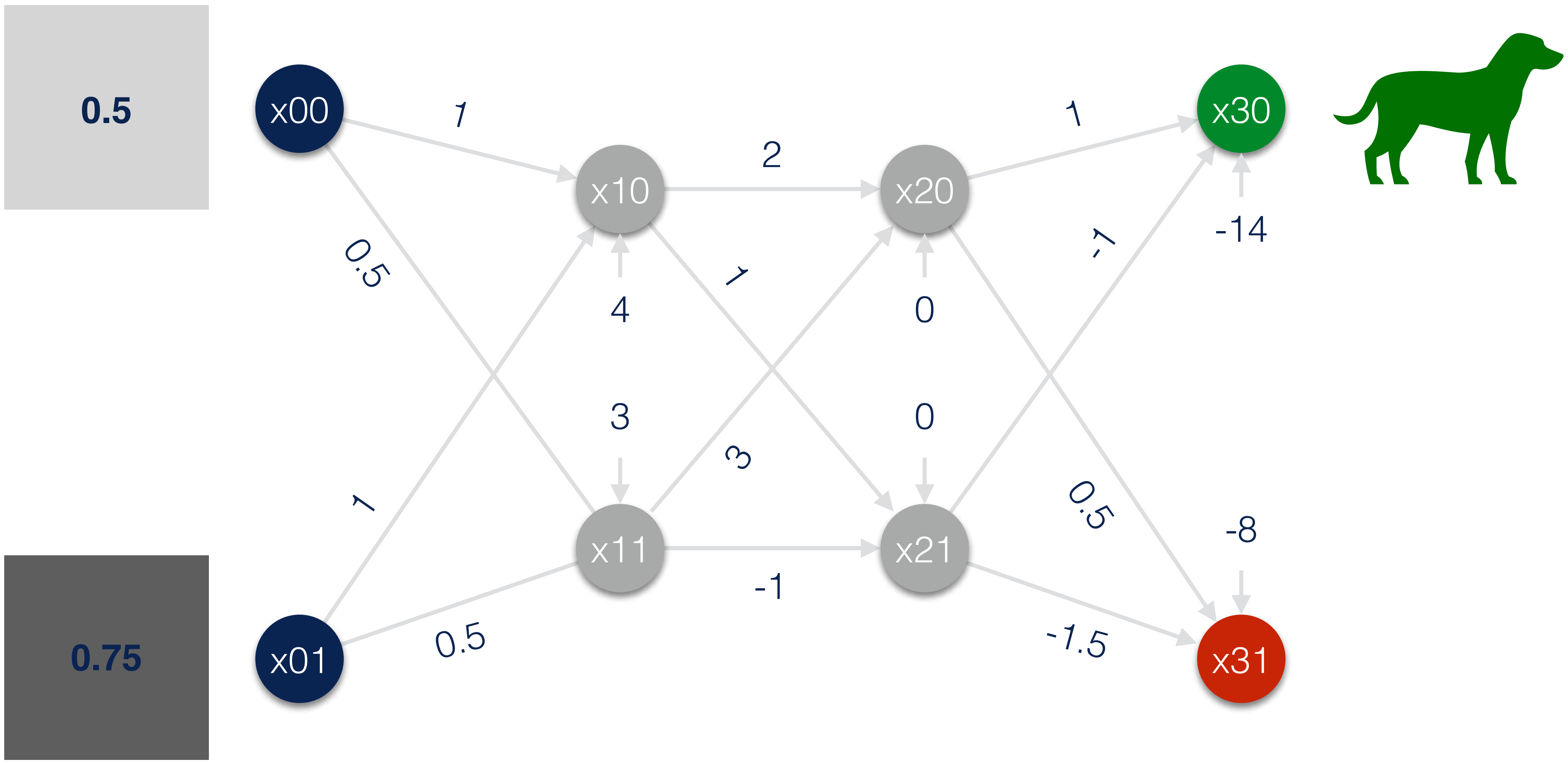
② check output for **inclusion** in **expected output**:
included → ✔ **stable**
otherwise → 🚨 **alarm**

① proceed **forwards from an abstraction** of all possible perturbations

Example



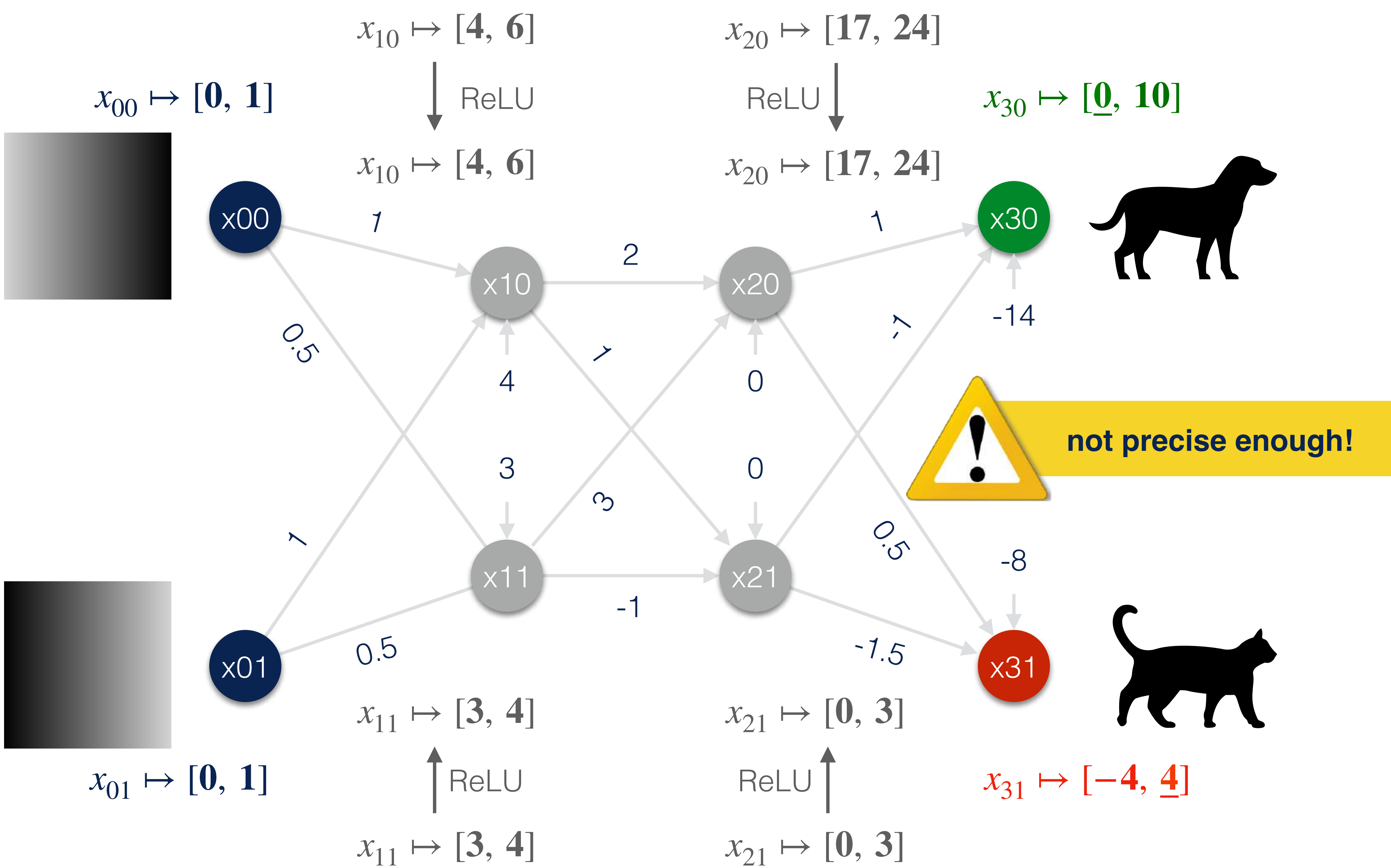
Example



$$P(\langle 0.5, 0.75 \rangle) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathscr{R} \times \mathscr{R} \mid 0 \leq \mathbf{x}_0 \leq 1 \wedge 0 \leq \mathbf{x}_1 \leq 1\}$$

Interval Domain

$$x_{i,j} \mapsto [a, b]$$
$$a, b \in \mathscr{R}$$



Interval Domain

with Symbolic Constant Propagation [Li19]

$$x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} & \mathbf{c}_k, \mathbf{c} \in \mathscr{R}^{|L_k|} \\ [a,\ b] & a, b \in \mathscr{R} \end{cases}$$

$$x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} \\ [a,\ b] \end{cases} \xrightarrow{\text{ReLU}} x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} \\ [a,\ b] \end{cases} \qquad 0 \leq a$$

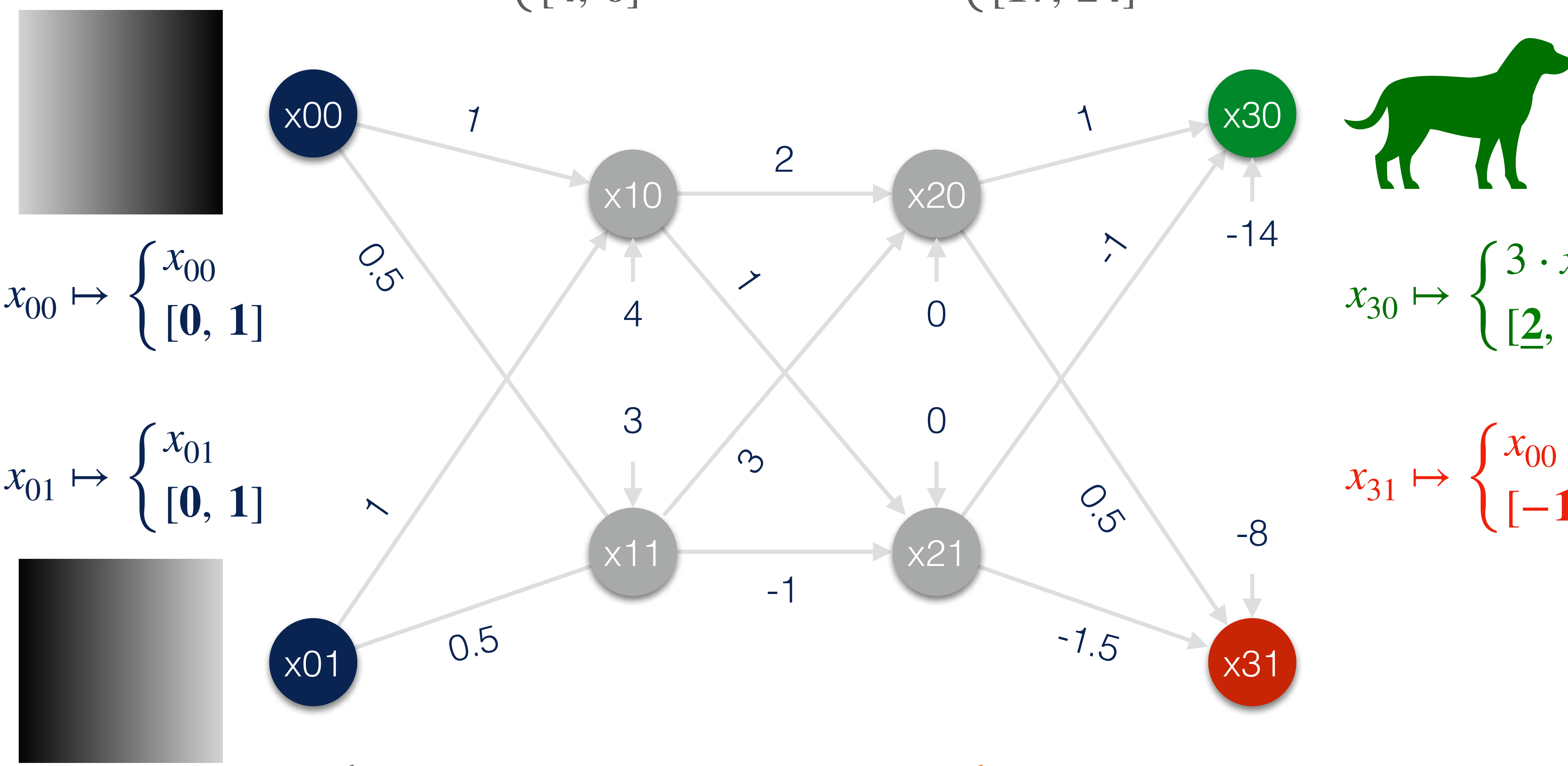
$$x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} \\ [a,\ b] \end{cases} \xrightarrow{\text{ReLU}} x_{i,j} \mapsto \begin{cases} x_{i,j} \\ [0,\ b] \end{cases} \qquad a < 0 \wedge 0 < b$$

$$x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} \\ [a,\ b] \end{cases} \xrightarrow{\text{ReLU}} x_{i,j} \mapsto \begin{cases} 0 \\ [0,\ 0] \end{cases} \qquad b \leq 0$$

Interval Domain

with Symbolic Constant Propagation [Li19]

$$x_{10} \mapsto \begin{cases} x_{00} + x_{01} + 4 \\ [\mathbf{4}, \mathbf{6}] \end{cases} \qquad x_{20} \mapsto \begin{cases} 2 \cdot (x_{00} + x_{01} + 4) + 3 \cdot (0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3) \\ [\mathbf{17}, \mathbf{24}] \end{cases}$$



$$x_{00} \mapsto \begin{cases} x_{00} \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} x_{01} \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

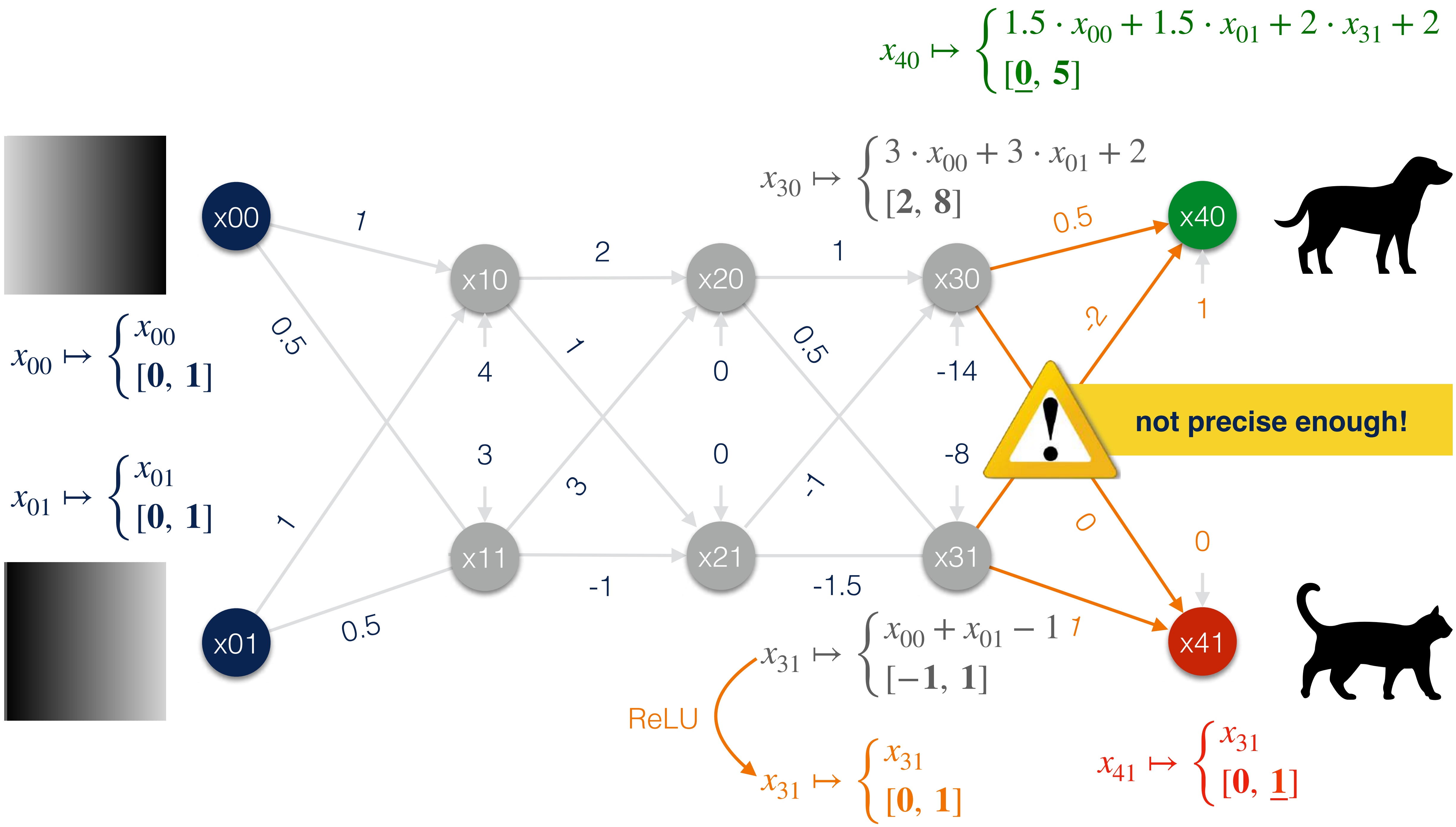
$$x_{30} \mapsto \begin{cases} 3 \cdot x_{00} + 3 \cdot x_{01} + 2 \\ [\underline{\mathbf{2}}, \mathbf{8}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} x_{00} + x_{01} - 1 \\ [\mathbf{-1}, \underline{\mathbf{1}}] \end{cases}$$

$$x_{11} \mapsto \begin{cases} 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3 \\ [\mathbf{3}, \mathbf{4}] \end{cases} \qquad x_{21} \mapsto \begin{cases} (x_{00} + x_{01} + 4) - 1 \cdot (0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3) \\ [\mathbf{1}, \mathbf{2}] \end{cases}$$

Interval Domain

with Symbolic Constant Propagation [Li19]



DeepPoly Domain [Singh19]

$$x_{i+1,j} \mapsto \begin{cases} [\sum_k c_{i,k} \cdot x_{i,k} + c, \sum_k d_{i,k} \cdot x_{i,k} + d] & c_{i,k}, c, d_{i,k}, d \in \mathscr{R} \\ [a,\ b] & a, b \in \mathscr{R} \end{cases}$$

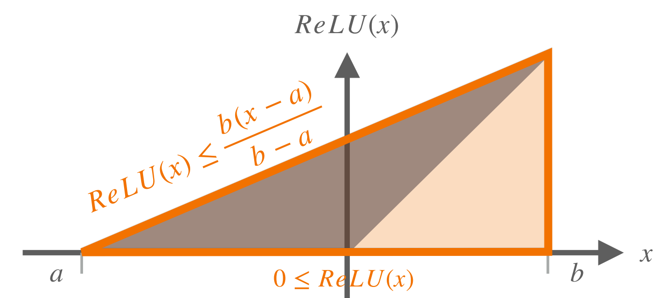
$$x_{i,j} \mapsto \begin{cases} [A,\ B] \\ [a,\ b] \end{cases}$$

ReLU, $0 \leq a$:

$$x_{i,j} \mapsto \begin{cases} [A,\ B] \\ [a,\ b] \end{cases}$$

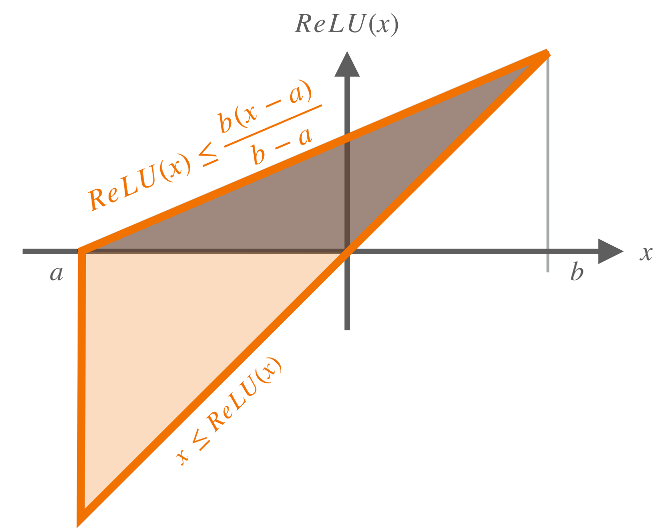
ReLU, $a < 0 \wedge 0 < b$, $b \leq -a$:

$$x_{i,j} \mapsto \begin{cases} \left[0,\ \dfrac{b(x_{i,j} - a)}{b - a}\right] \\ [0,\ b] \end{cases}$$



ReLU, $a < 0 \wedge 0 < b$, $-a < b$:

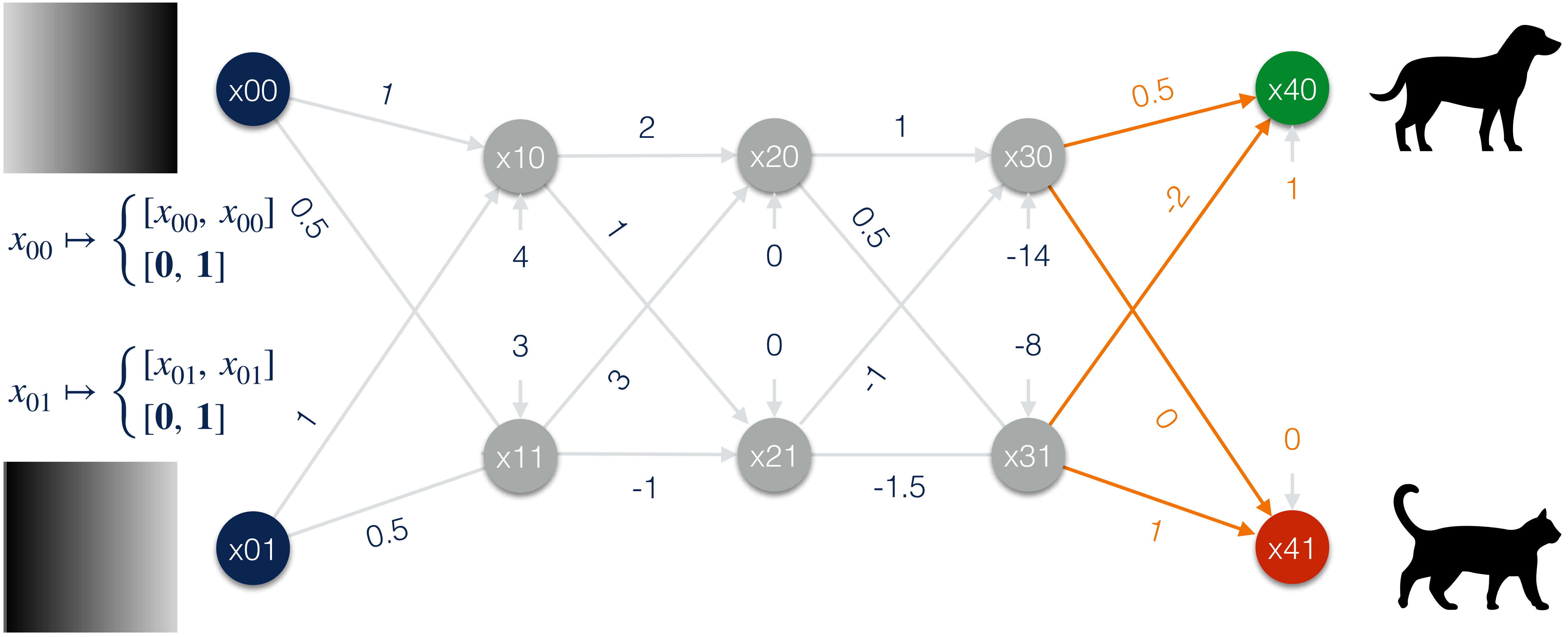
$$x_{i,j} \mapsto \begin{cases} \left[x_{i,j},\ \dfrac{b(x_{i,j} - a)}{b - a}\right] \\ [a,\ b] \end{cases}$$



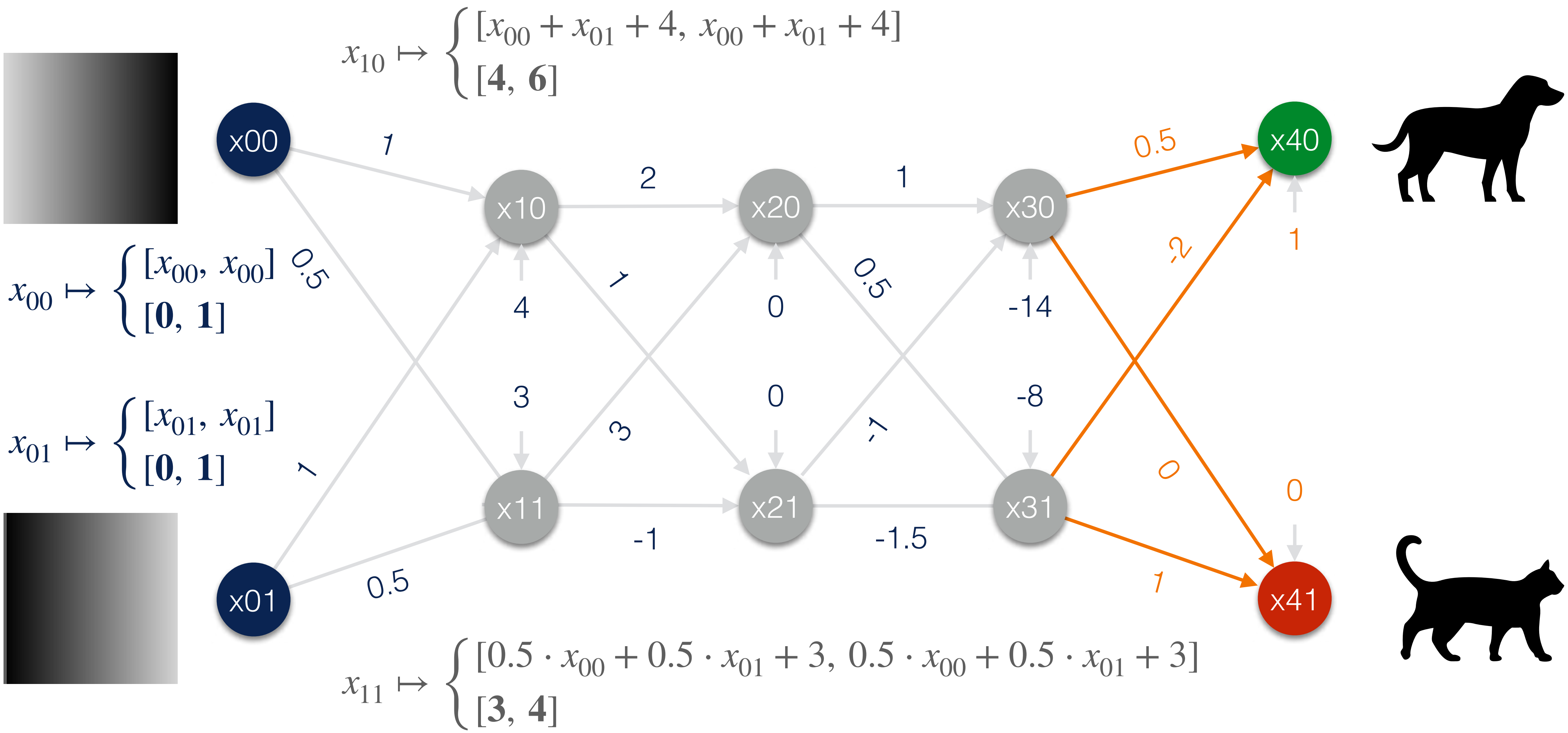
ReLU, $b \leq 0$:

$$x_{i,j} \mapsto \begin{cases} [0,\ 0] \\ [0,\ 0] \end{cases}$$

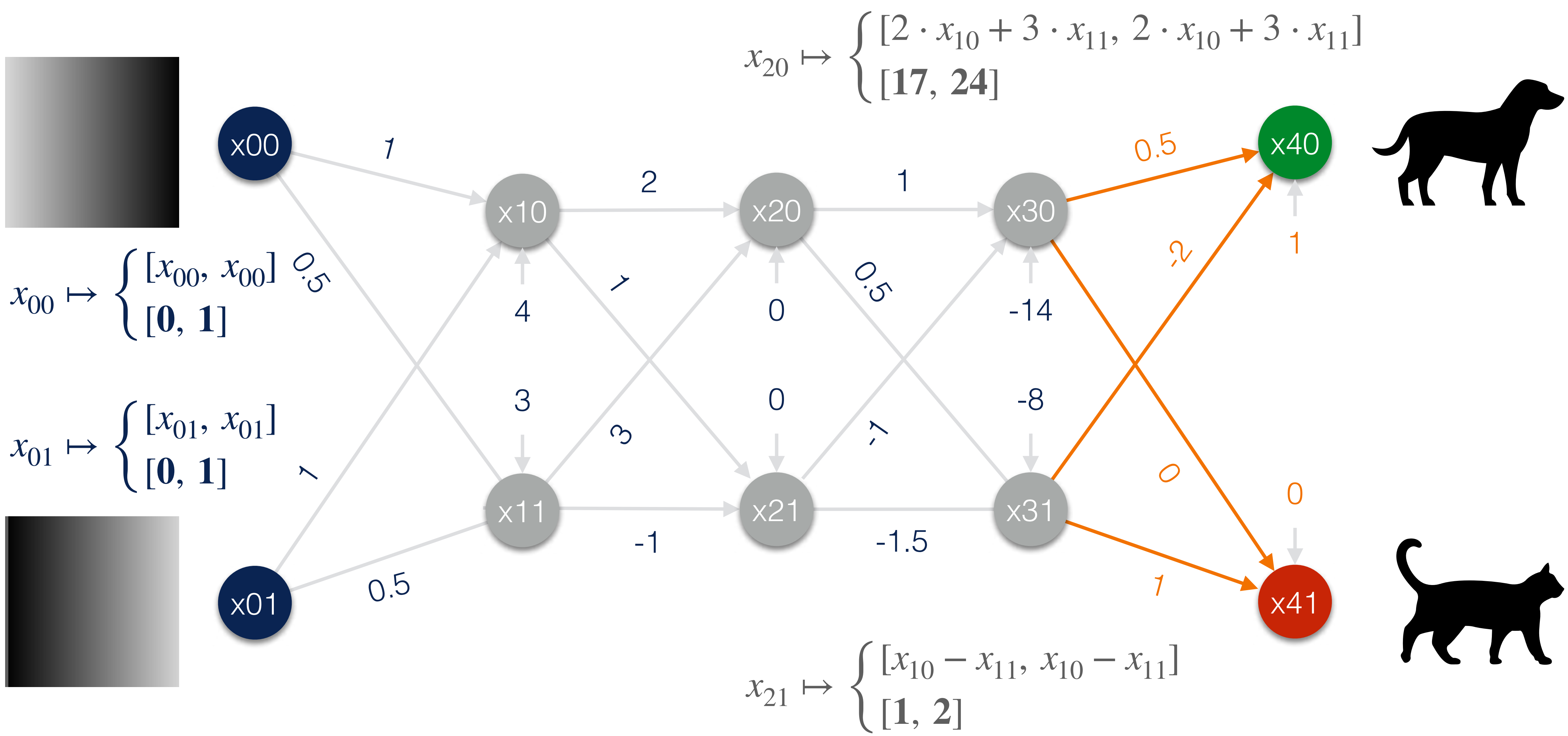
DeepPoly Domain [Singh19]



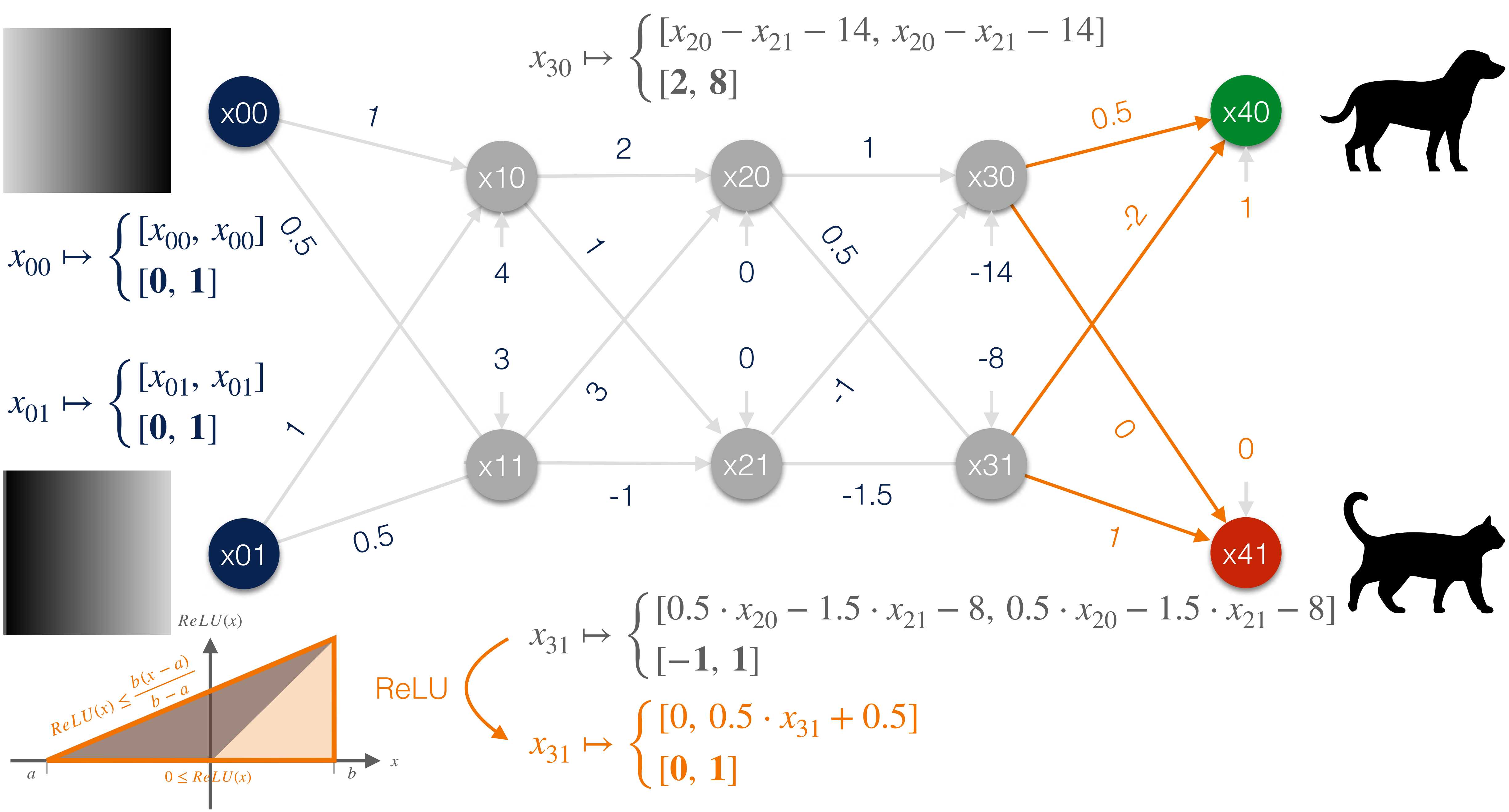
DeepPoly Domain [Singh19]



DeepPoly Domain [Singh19]



DeepPoly Domain [Singh19]

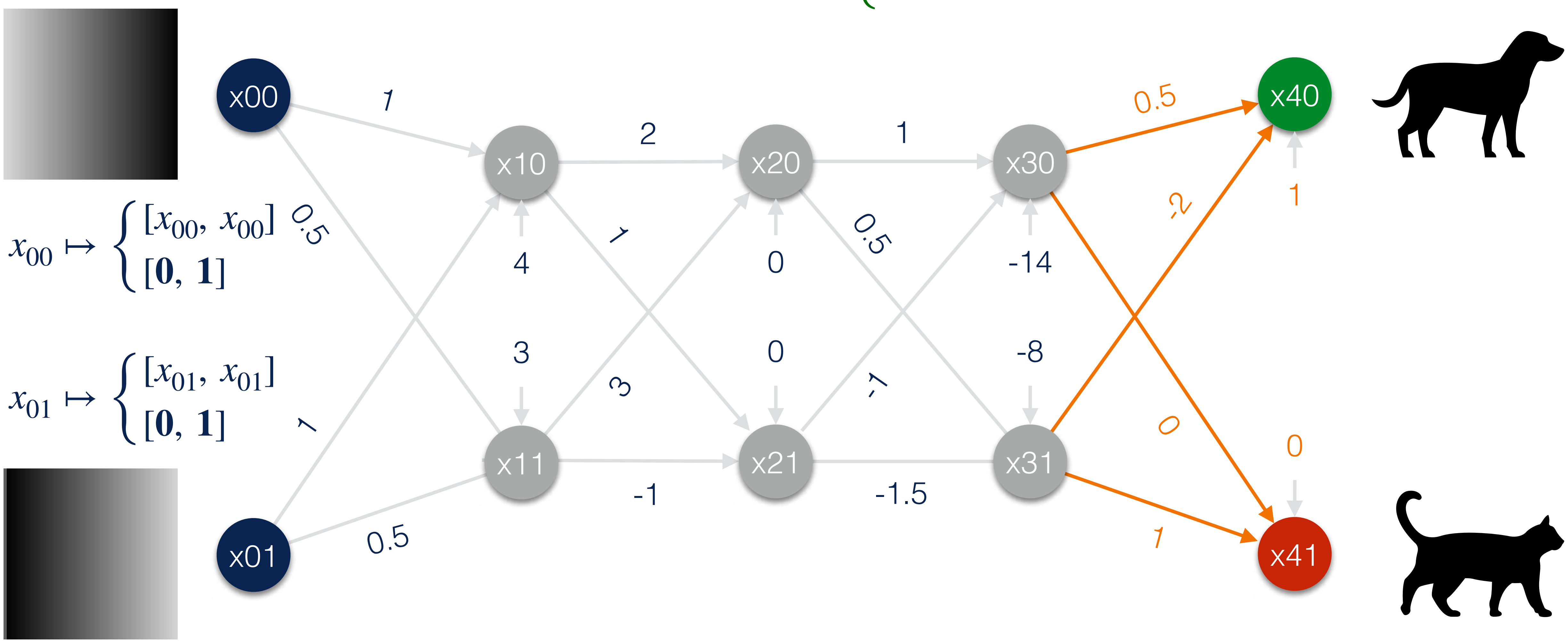


DeepPoly Domain [Singh19]

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$



DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00}, x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01}, x_{01}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4, x_{00} + x_{01} + 4] \\ [\mathbf{4}, \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3, 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3}, \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11}, 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17}, \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11}, x_{10} - x_{11}] \\ [\mathbf{1}, \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14, x_{20} - x_{21} - 14] \\ [\mathbf{2}, \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0, 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1, 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{10} - x_{11} + 1,\ 0.5 \cdot x_{10} + 2 \cdot x_{11} - 6] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} + x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{10} - x_{11} + 1,\ 0.5 \cdot x_{10} + 2 \cdot x_{11} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 2,\ 1.5 \cdot x_{00} + 1.5 \cdot x_{11} + 2] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

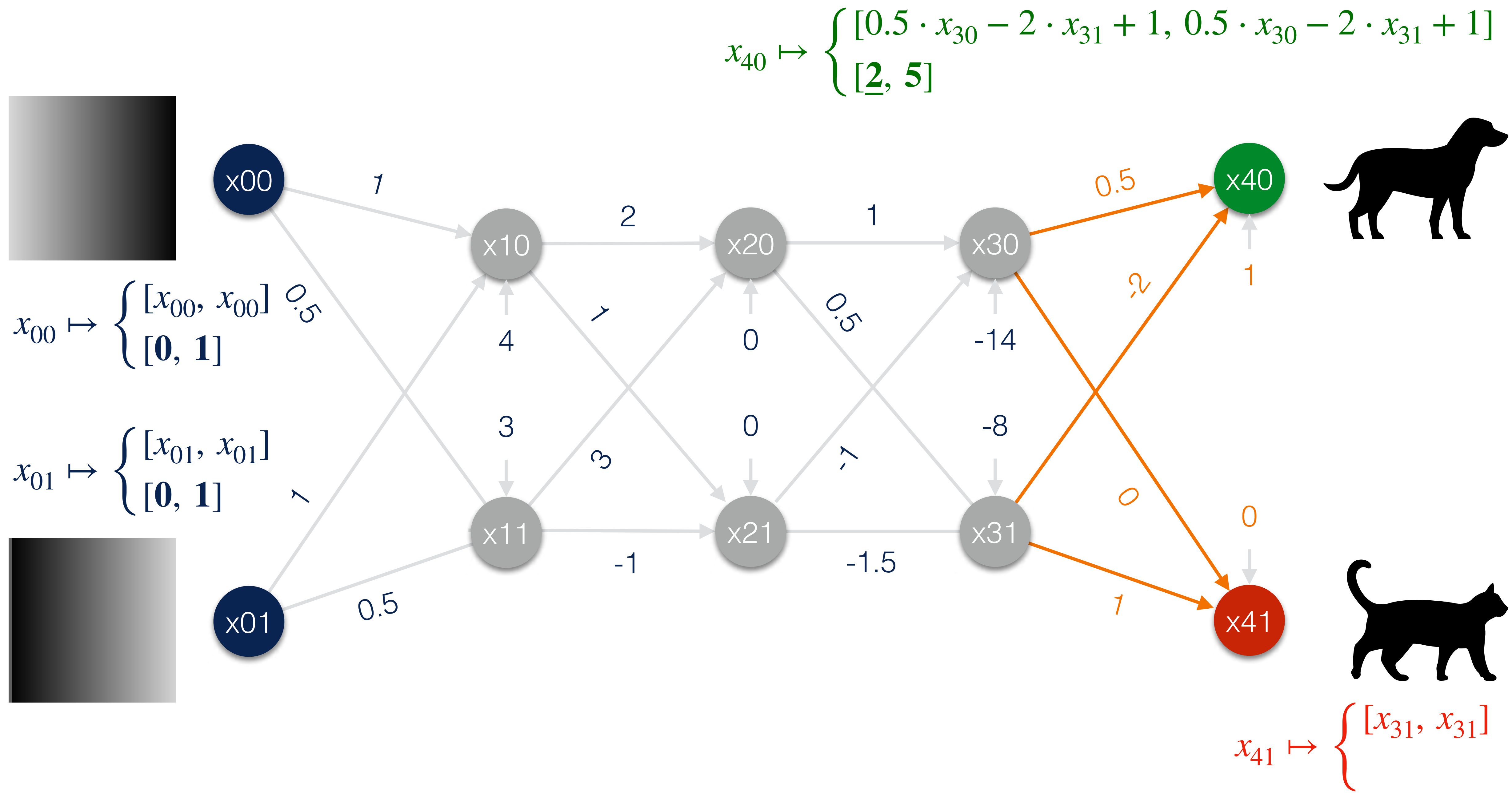
$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{10} - x_{11} + 1,\ 0.5 \cdot x_{10} + 2 \cdot x_{11} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 2,\ 1.5 \cdot x_{00} + 1.5 \cdot x_{11} + 2] \\ [\underline{\mathbf{2}},\ \mathbf{5}] \end{cases}$$

DeepPoly Domain [Singh19]



DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases}$$

$$x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases}$$

$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.25 \cdot x_{20} - 0.75 \cdot x_{21} - 3.5] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.25 \cdot x_{20} - 0.75 \cdot x_{21} - 3.5] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ -0.25 \cdot x_{10} + 1.5 \cdot x_{11} - 3.5] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.25 \cdot x_{20} - 0.75 \cdot x_{21} - 3.5] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ -0.25 \cdot x_{10} + 1.5 \cdot x_{11} - 3.5] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01}] \\ \end{cases}$$

DeepPoly Domain [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases} \qquad x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4},\ \mathbf{6}] \end{cases} \qquad x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3},\ \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17},\ \mathbf{24}] \end{cases} \qquad x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases} \qquad x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

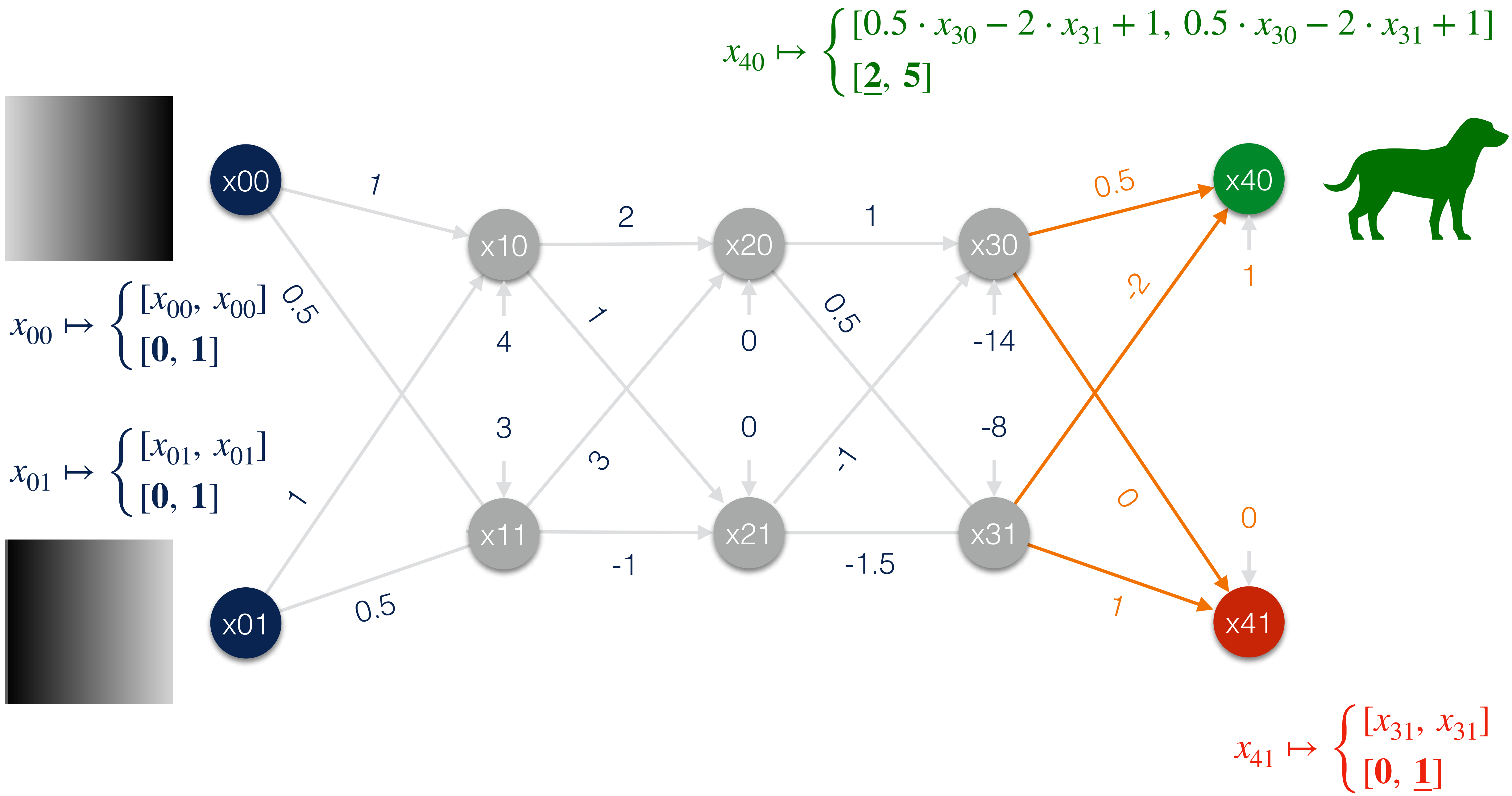
$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.25 \cdot x_{20} - 0.75 \cdot x_{21} - 3.5] \\ \end{cases}$$

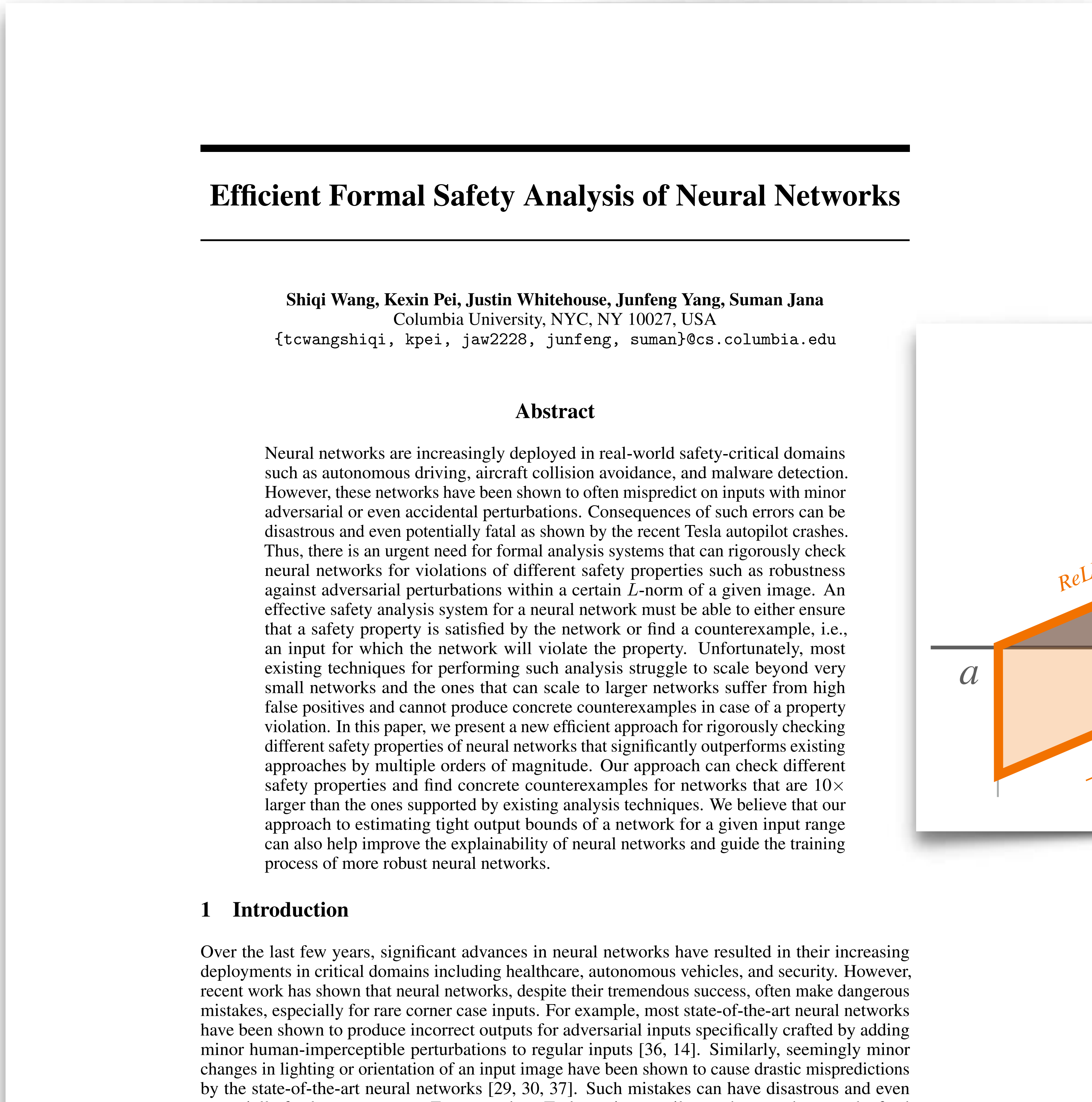
$$\mapsto \begin{cases} [0,\ -0.25 \cdot x_{10} + 1.5 \cdot x_{11} - 3.5] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01}] \\ [\mathbf{0},\ \underline{\mathbf{1}}] \end{cases}$$

DeepPoly Domain [Singh19]



Reading Assignment



Efficient Formal Safety Analysis of Neural Networks

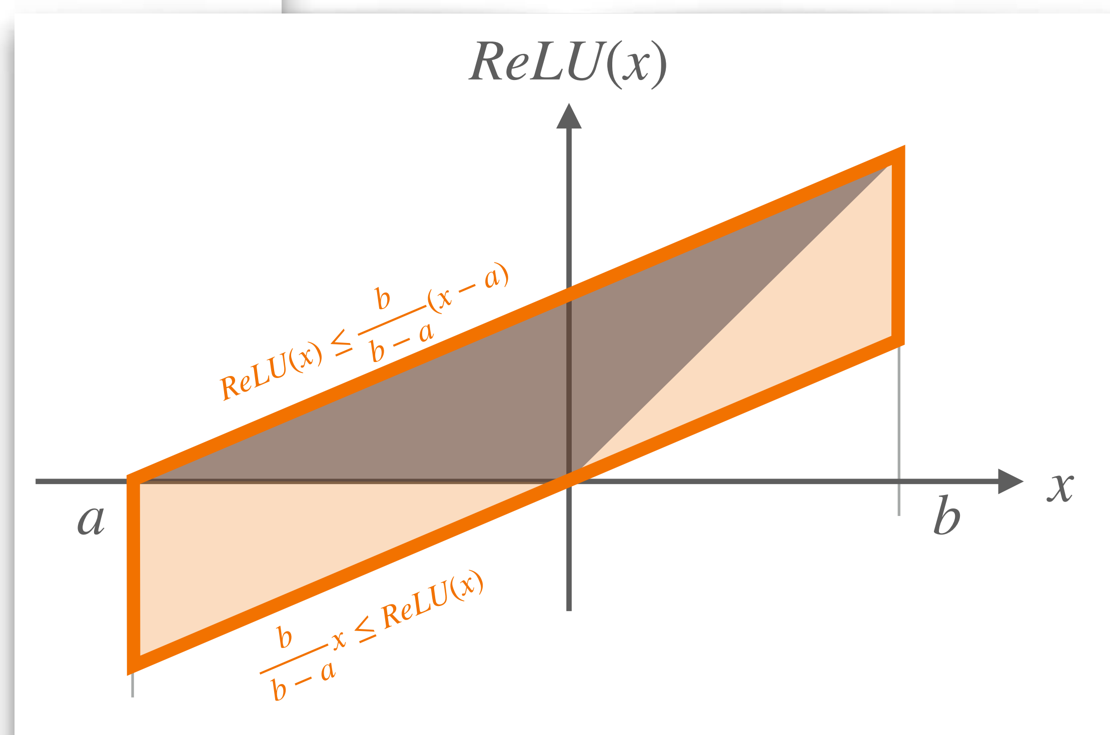
Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, Suman Jana
Columbia University, NYC, NY 10027, USA
{tcwangshiqi, kpei, jaw2228, junfeng, suman}@cs.columbia.edu

Abstract

Neural networks are increasingly deployed in real-world safety-critical domains such as autonomous driving, aircraft collision avoidance, and malware detection. However, these networks have been shown to often mispredict on inputs with minor adversarial or even accidental perturbations. Consequences of such errors can be disastrous and even potentially fatal as shown by the recent Tesla autopilot crashes. Thus, there is an urgent need for formal analysis systems that can rigorously check neural networks for violations of different safety properties such as robustness against adversarial perturbations within a certain L-norm of a given image. An effective safety analysis system for a neural network must be able to either ensure that a safety property is satisfied by the network or find a counterexample, i.e., an input for which the network will violate the property. Unfortunately, most existing techniques for performing such analysis struggle to scale beyond very small networks and the ones that can scale to larger networks suffer from high false positives and cannot produce concrete counterexamples in case of a property violation. In this paper, we present a new efficient approach for rigorously checking different safety properties of neural networks that significantly outperforms existing approaches by multiple orders of magnitude. Our approach can check different safety properties and find concrete counterexamples for networks that are 10× larger than the ones supported by existing analysis techniques. We believe that our approach to estimating tight output bounds of a network for a given input range can also help improve the explainability of neural networks and guide the training process of more robust neural networks.

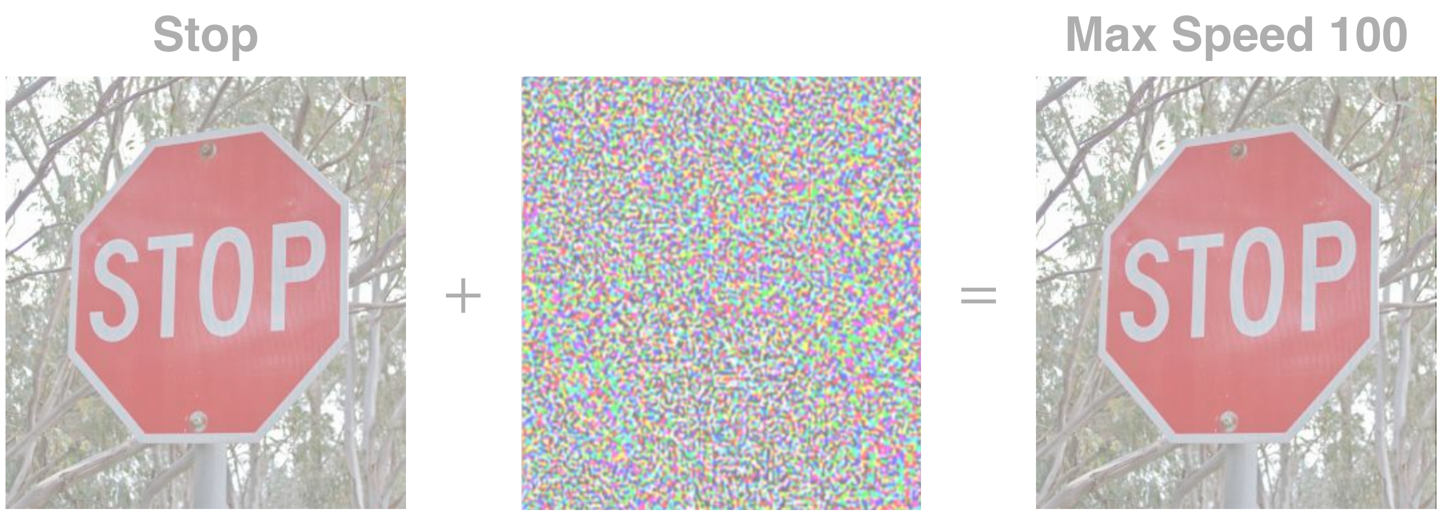
1 Introduction

Over the last few years, significant advances in neural networks have resulted in their increasing deployments in critical domains including healthcare, autonomous vehicles, and security. However, recent work has shown that neural networks, despite their tremendous success, often make dangerous mistakes, especially for rare corner case inputs. For example, most state-of-the-art neural networks have been shown to produce incorrect outputs for adversarial inputs specifically crafted by adding minor human-imperceptible perturbations to regular inputs [36, 14]. Similarly, seemingly minor changes in lighting or orientation of an input image have been shown to cause drastic mispredictions by the state-of-the-art neural networks [29, 30, 37]. Such mistakes can have disastrous and even potentially fatal consequences. For example, a Tesla car in autopilot mode recently caused a fatal



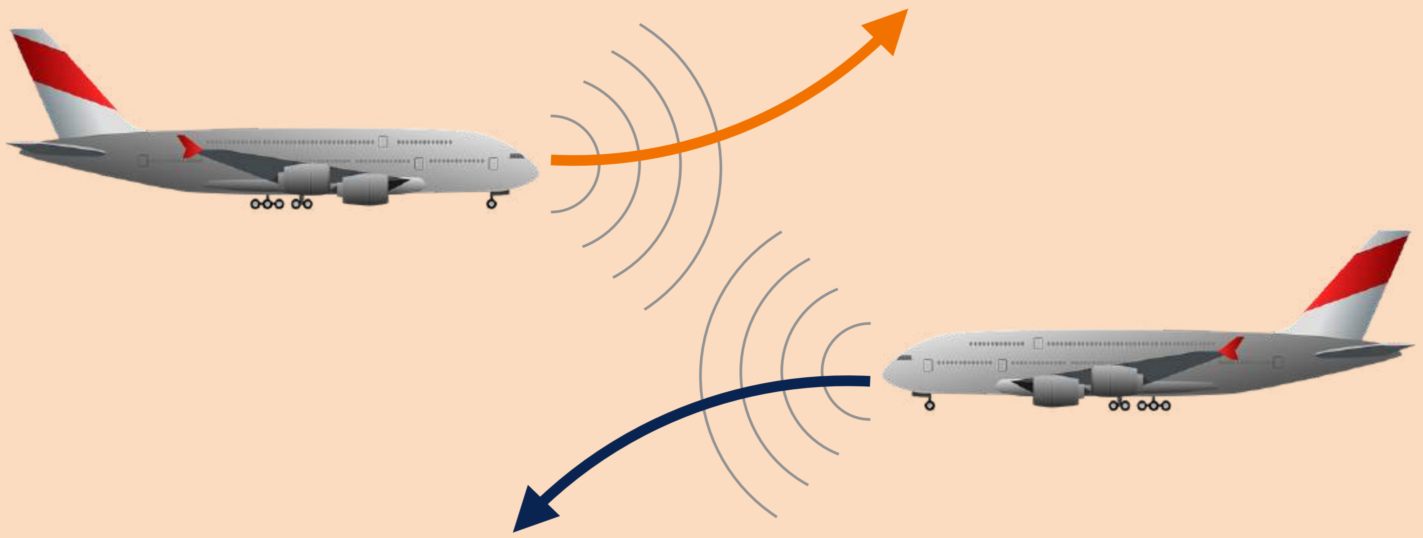
Stability

Goal G3 in [Kurd03]

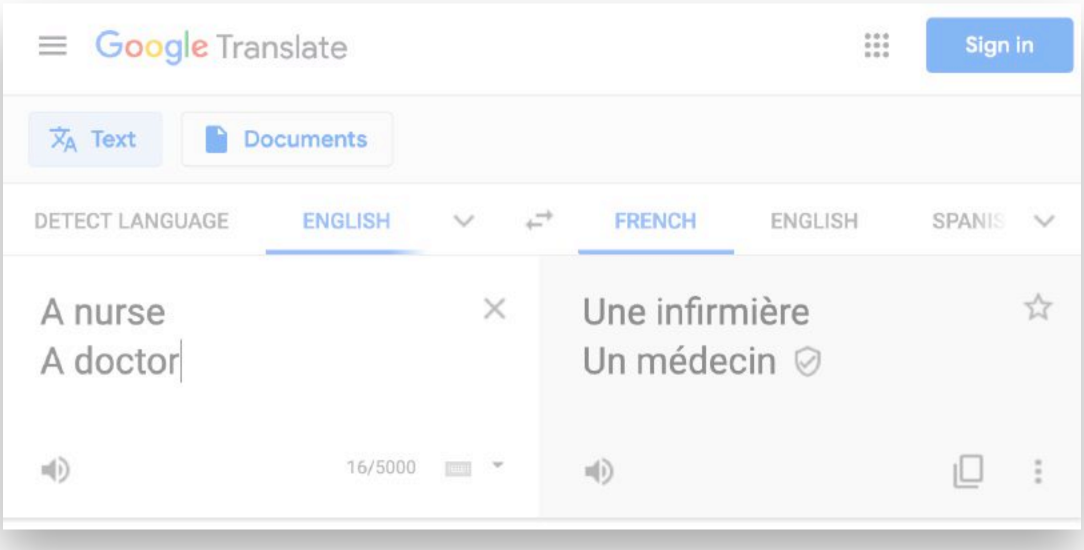


Safety

Goal G4 in [Kurd03]



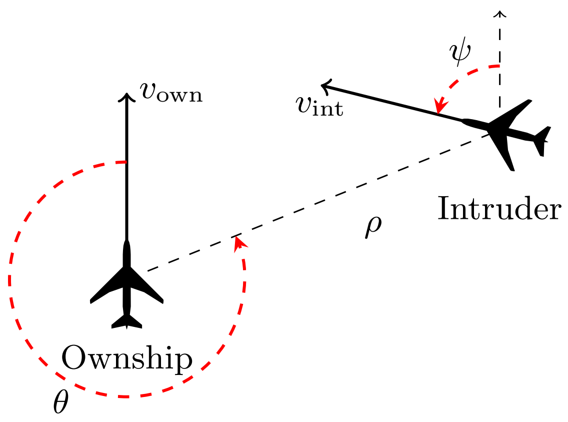
Fairness

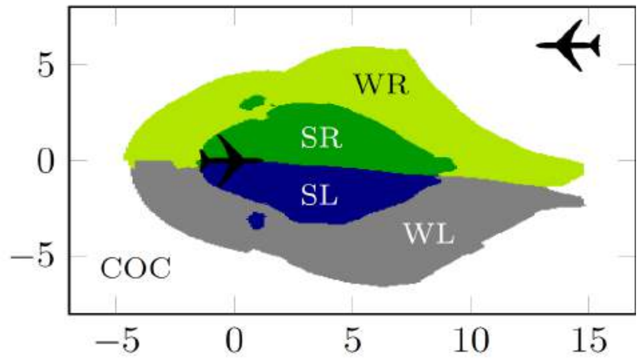


ACAS Xu [Julian16][Katz17]

Airborne Collision Avoidance System for Unmanned Aircraft

implemented using **45 feed-forward fully-connected ReLU networks**





5 input **sensor measurements**

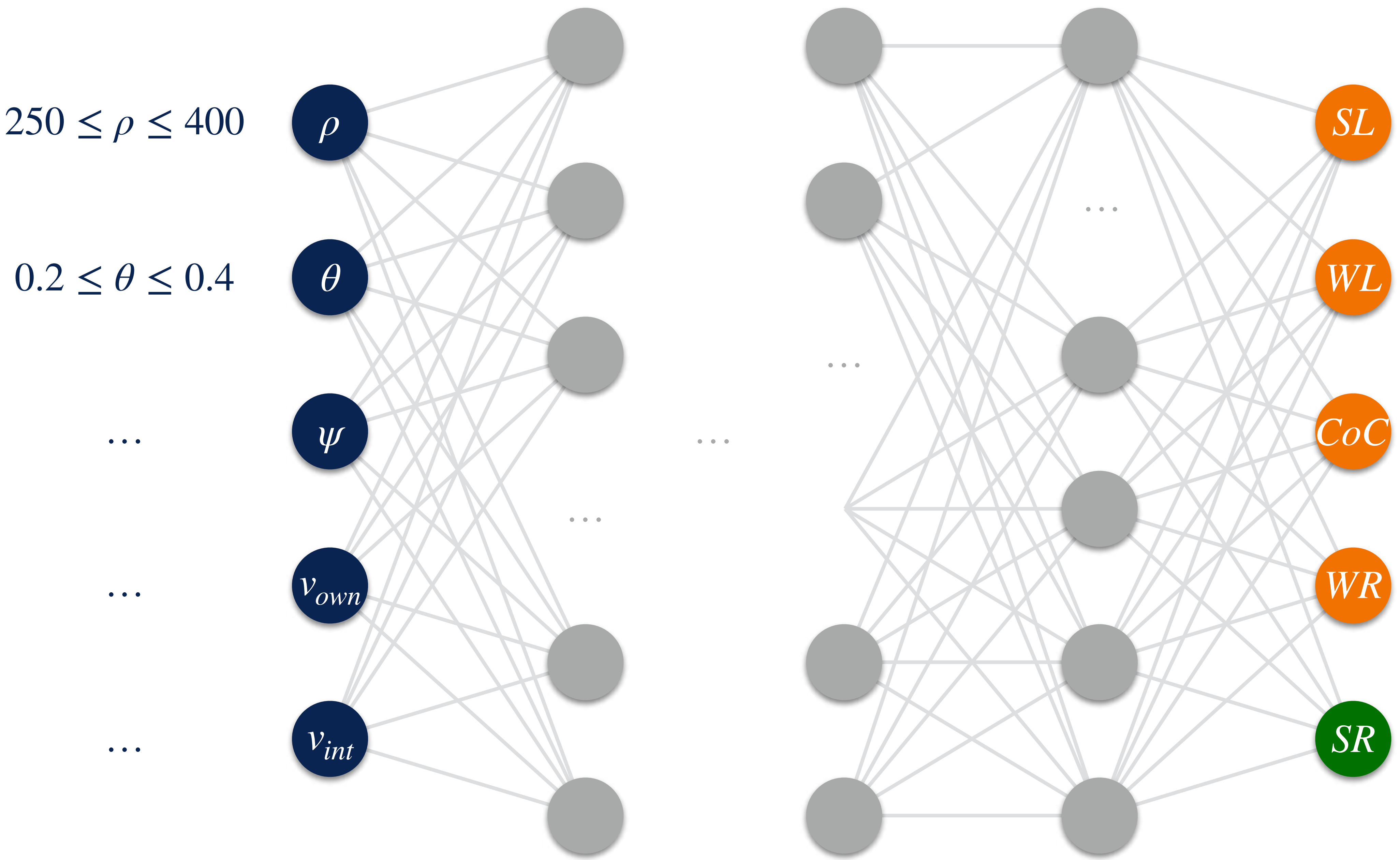
- ρ: distance from ownship to intruder
- θ: angle to intruder relative to ownship heading direction
- ψ: heading angle to intruder relative to ownship heading direction
- v_{own}: speed of ownship
- v_{int}: speed of intruder

5 output **horizontal advisories**

- Strong Left
- Weak Left
- Clear of Conflict
- Weak Right
- Strong Right

ACAS Xu Properties [Katz17]

Example: "if intruder is **near** and approaching **from the left**, go **Strong Right**"



Safety

Input-Output Properties

$\mathbf{I}$: input specification

$\mathbf{O}$: output specification

$$\mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \stackrel{\text{def}}{=} \{ [\![M]\!] \in \mathcal{P}(\Sigma^*) \mid \text{SAFE}^{\mathbf{I}}_{\mathbf{O}}([\![M]\!]) \}$$

$\mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **satisfy** the input and output specification $\mathbf{I}$ and $\mathbf{O}$

$$\text{SAFE}^{\mathbf{I}}_{\mathbf{O}}([\![M]\!]) \stackrel{\text{def}}{=} \forall t \in [\![M]\!] : t_0 \models \mathbf{I} \Rightarrow t_\omega \models \mathbf{O}$$

Theorem

$$M \models \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Leftrightarrow \{ [\![M]\!] \} \subseteq \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

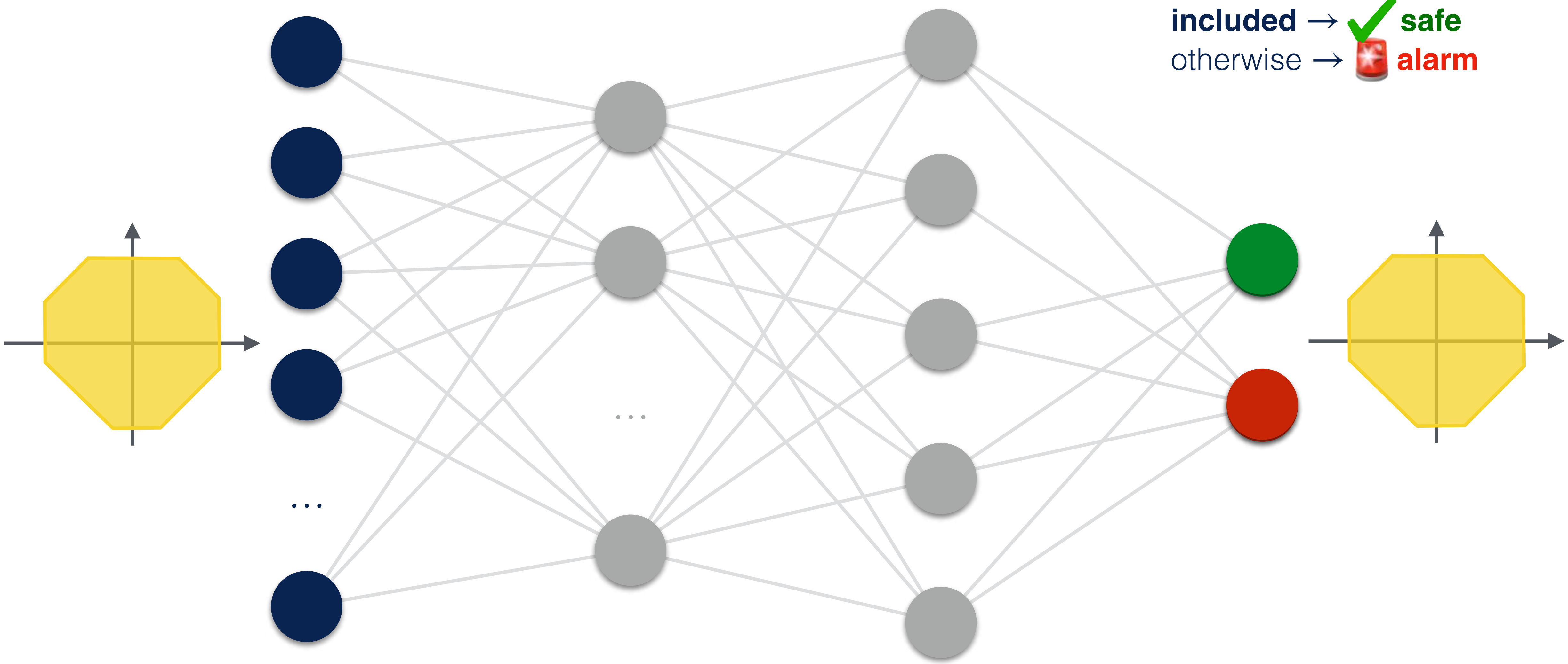
Corollary

$$M \models \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Leftrightarrow [\![M]\!] \subseteq \bigcup \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

Numerical Abstractions

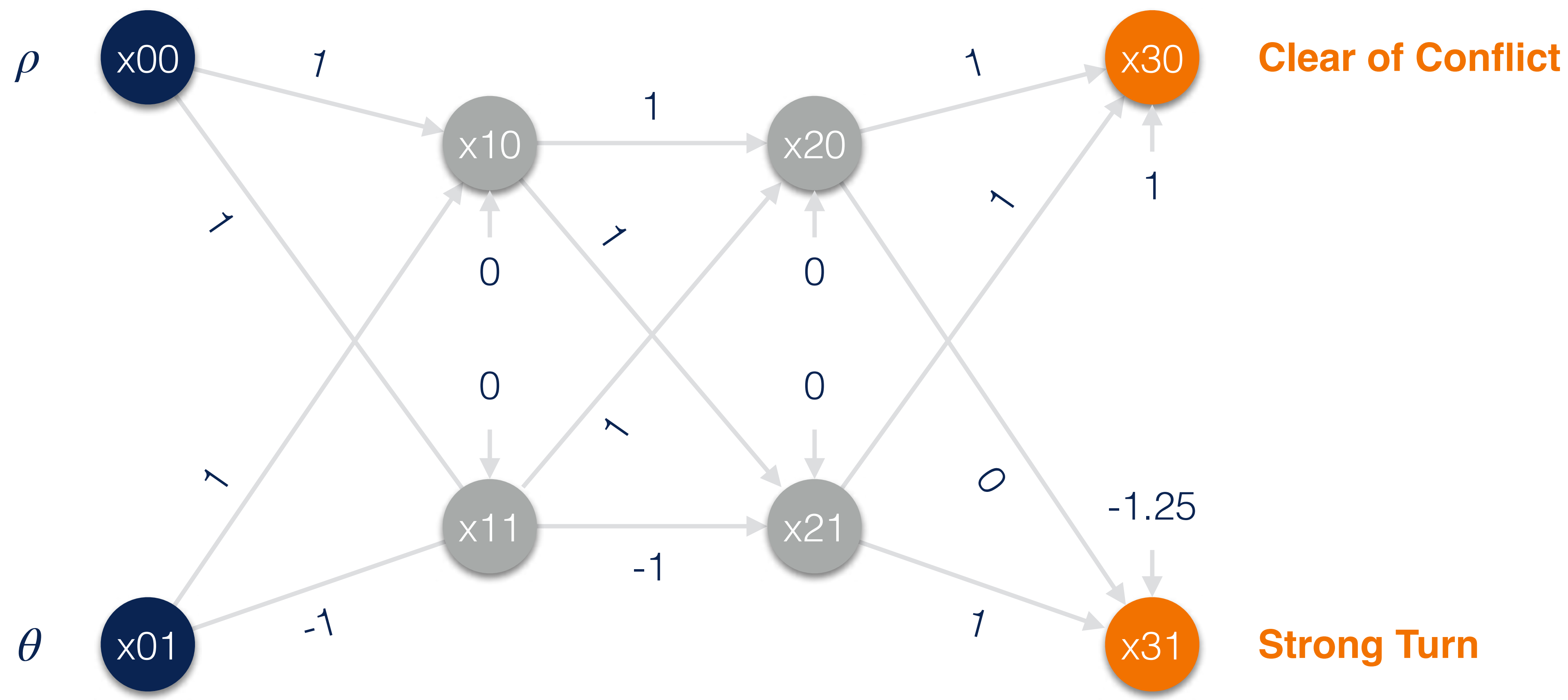
Forward Analysis

② check output for **inclusion** in **output specification $\mathbf{O}$**:
included → ✓ **safe**
otherwise → 🚨 **alarm**



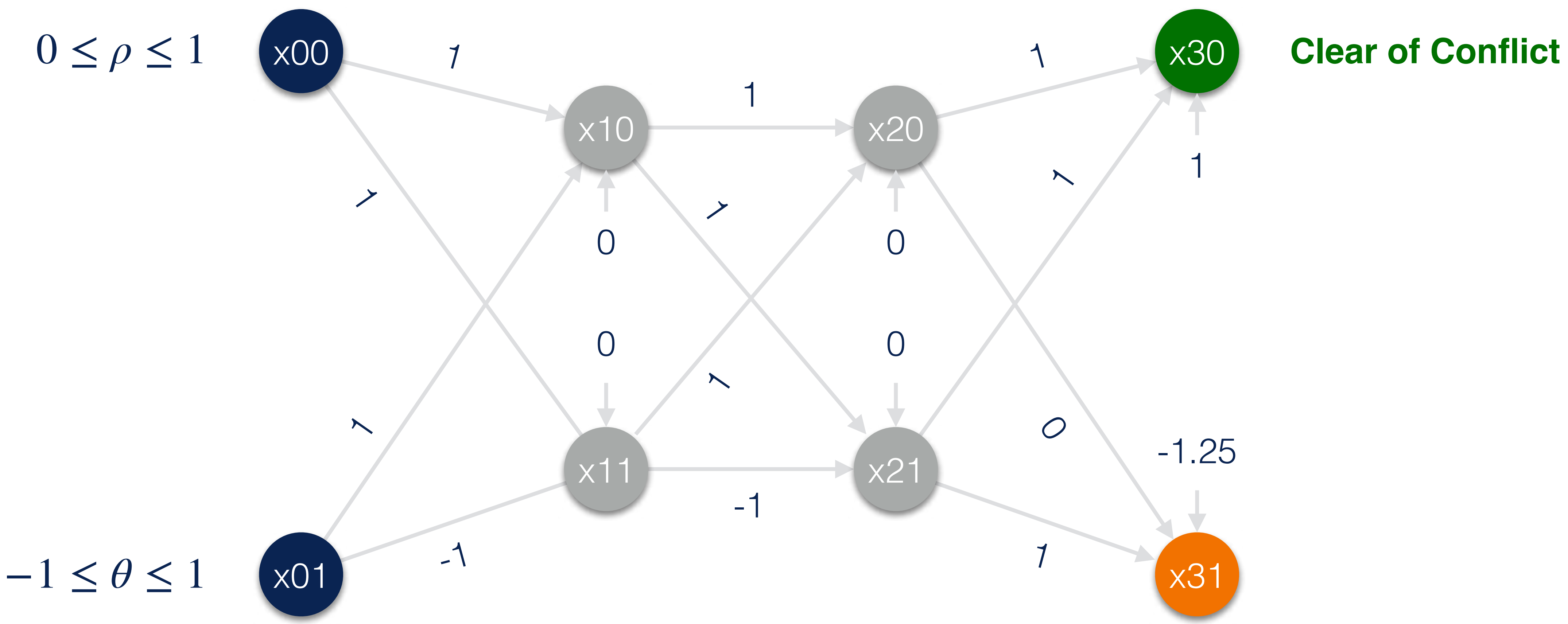
① proceed **forwards from an abstraction** of the input specification $\mathbf{I}$

Example



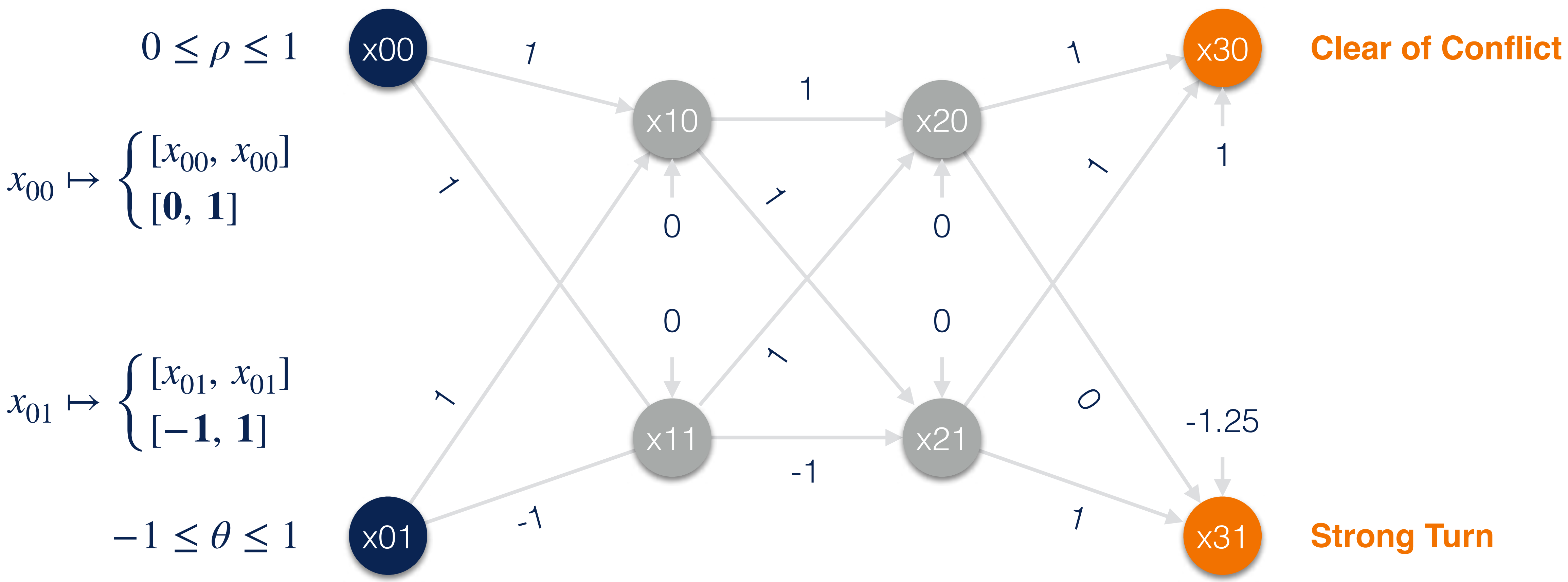
ρ
θ
x00
x01
x10
x11
x20
x21
x30
x31
1
1
1
-1
1
1
1
-1
0
0
0
0
1
1
0
1
1
-1.25
Clear of Conflict
Strong Turn

Example

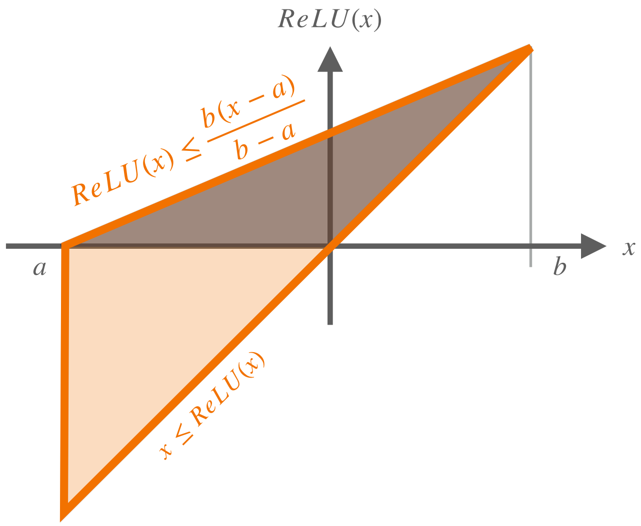


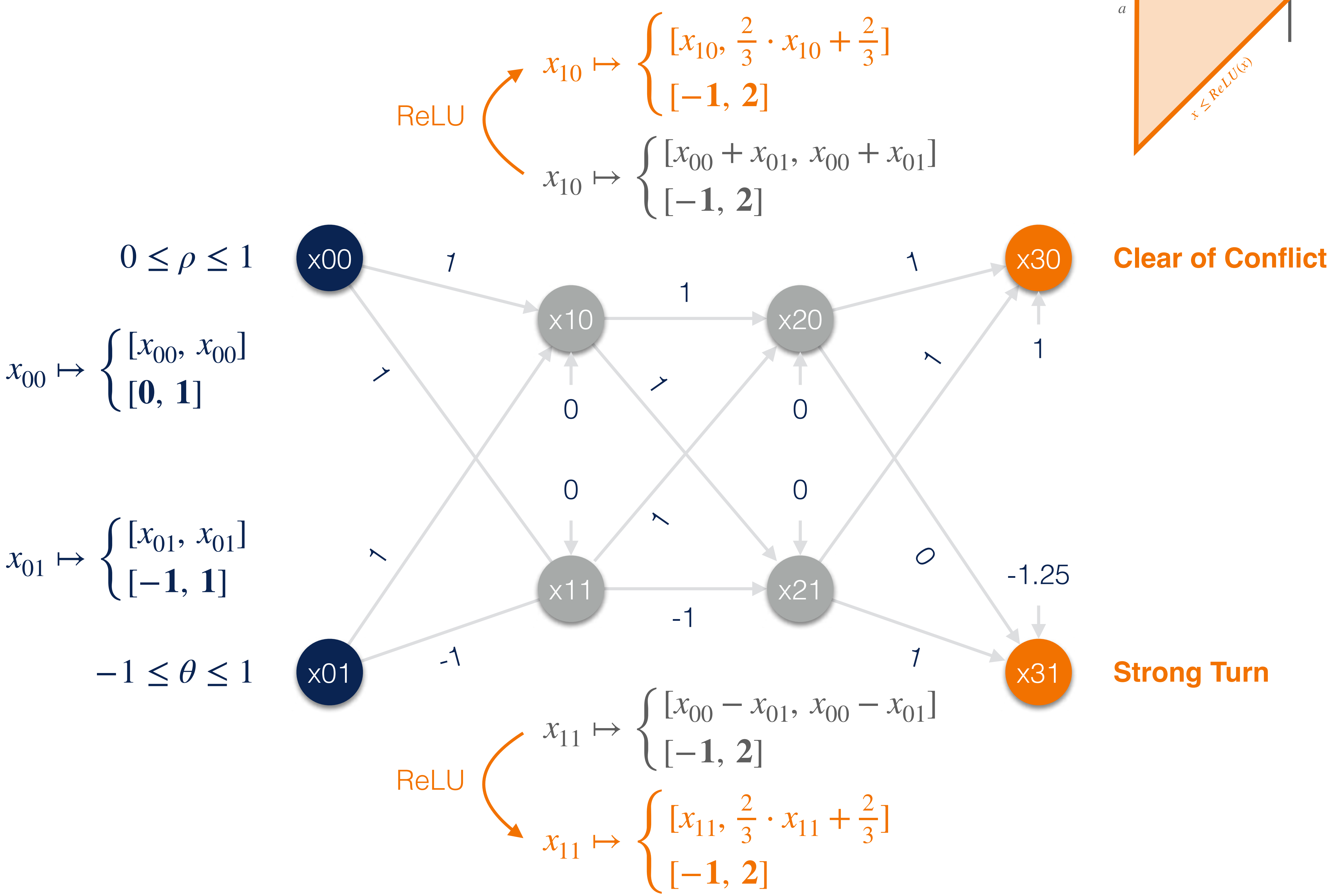
$0 \leq \rho \leq 1$
$-1 \leq \theta \leq 1$
x00
x01
x10
x11
x20
x21
x30
x31
Clear of Conflict
1
1
1
-1
1
0
0
1
1
-1
0
0
1
1
1
0
1
-1.25
1

DeepPoly Domain [Singh19]

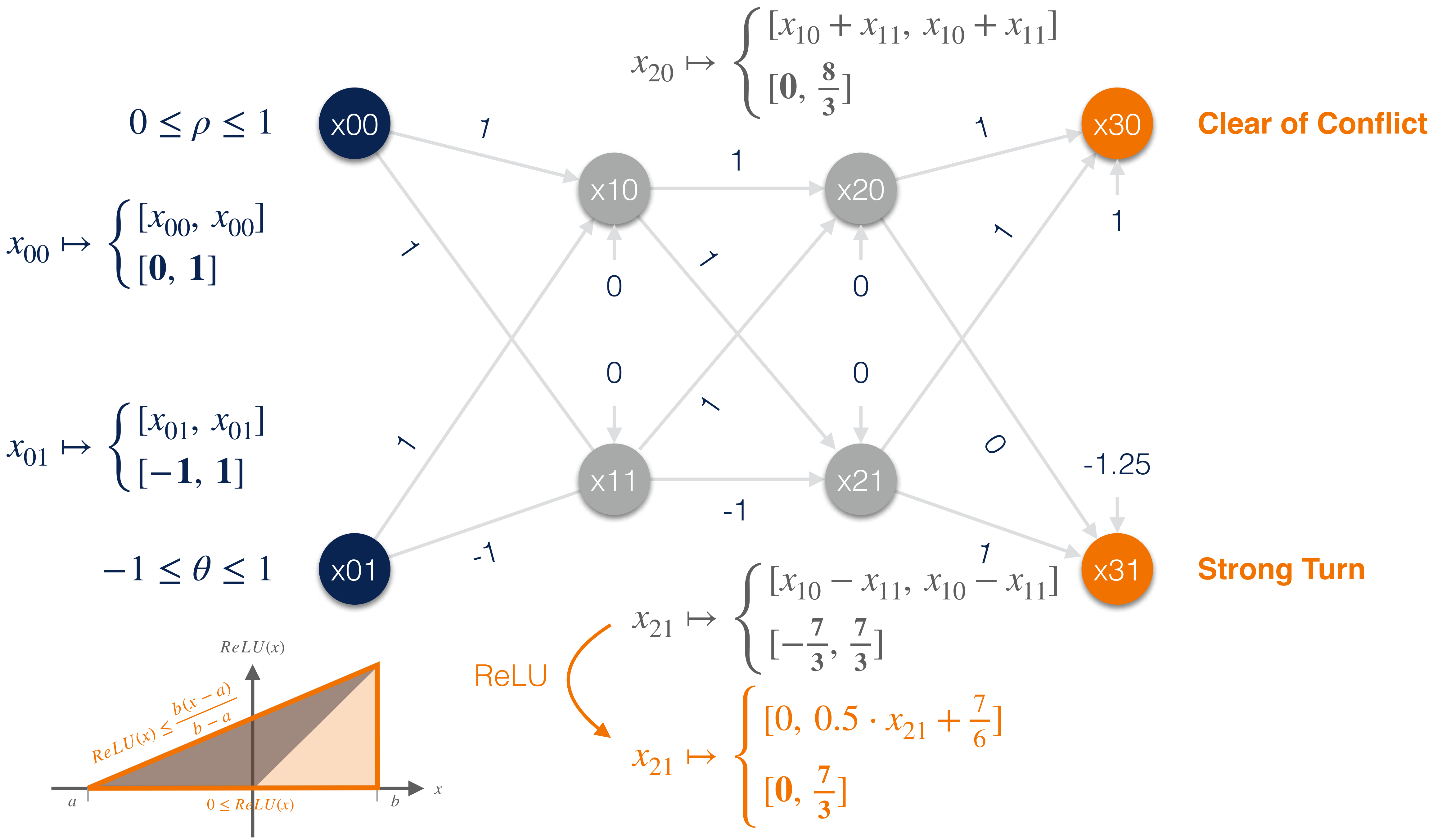


DeepPoly Domain [Singh19]

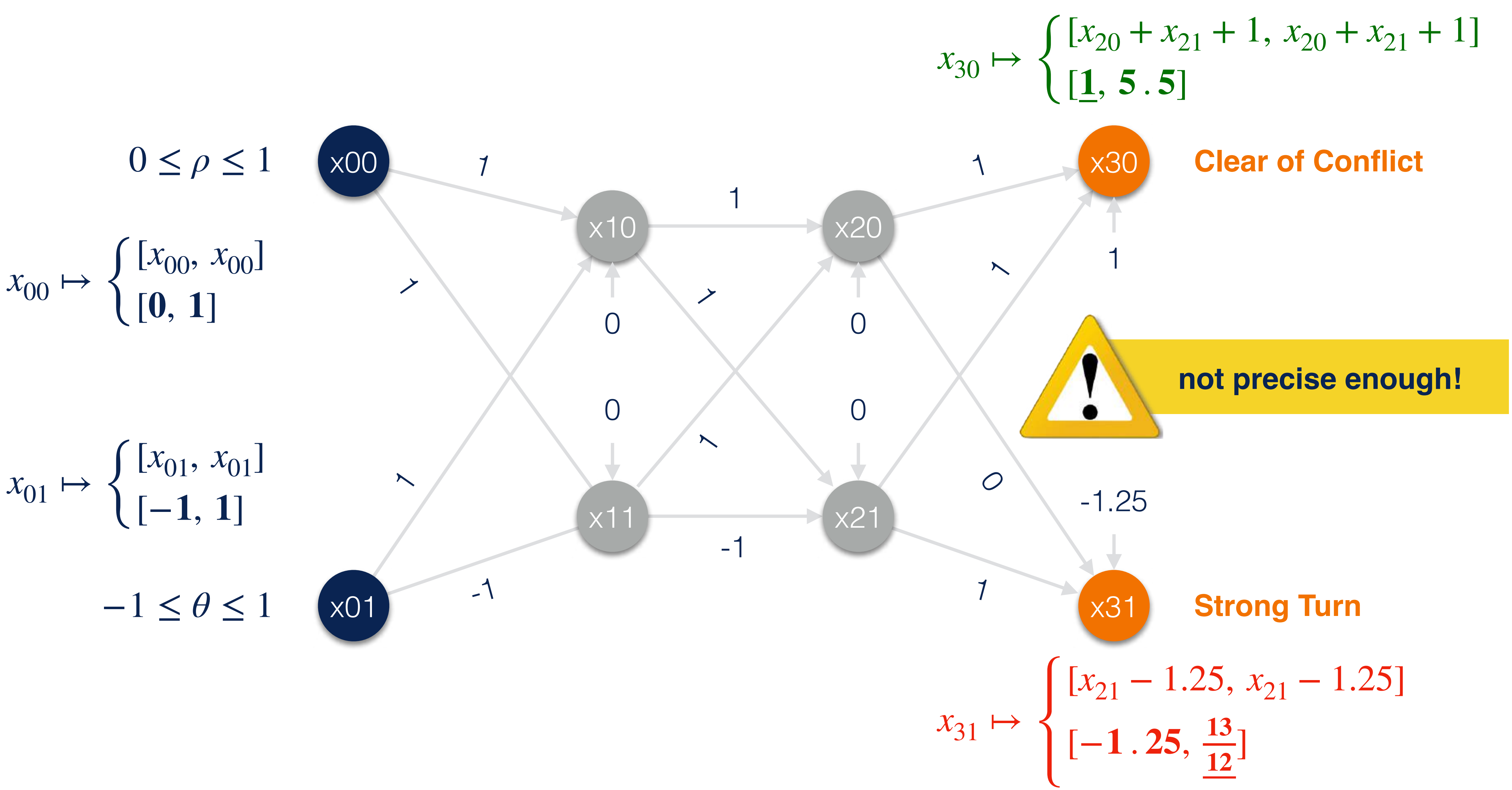




DeepPoly Domain [Singh19]

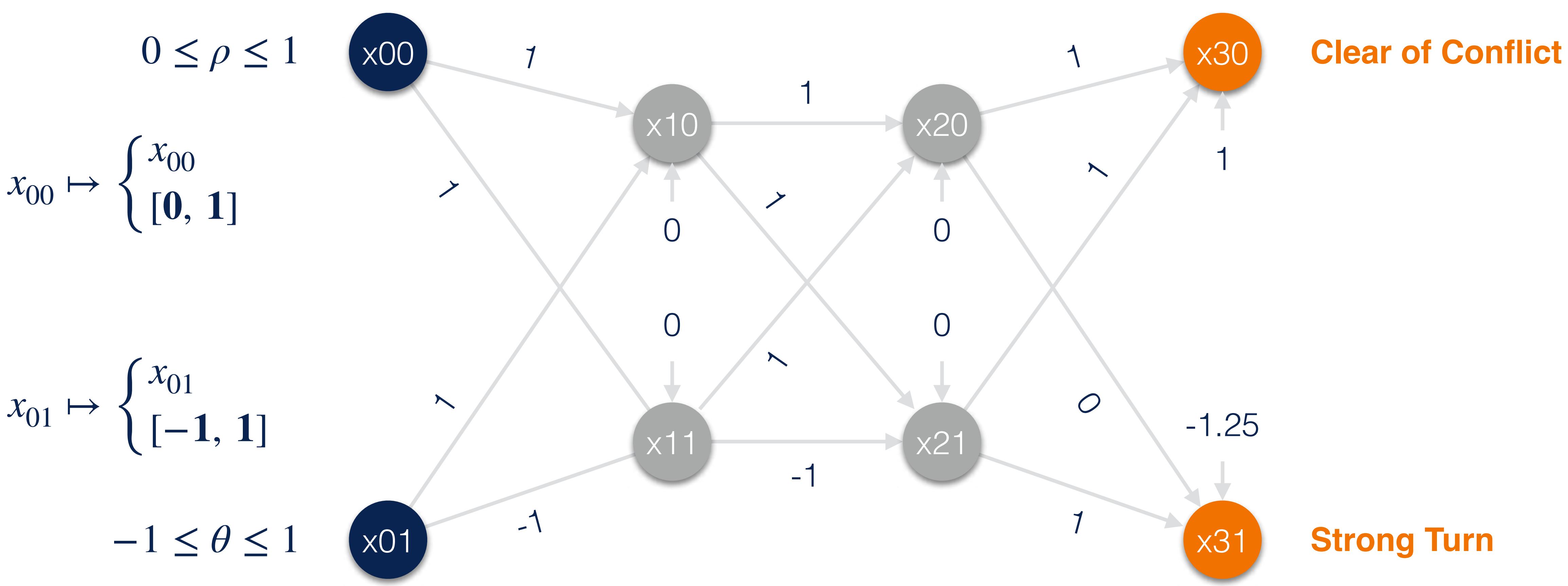


DeepPoly Domain [Singh19]



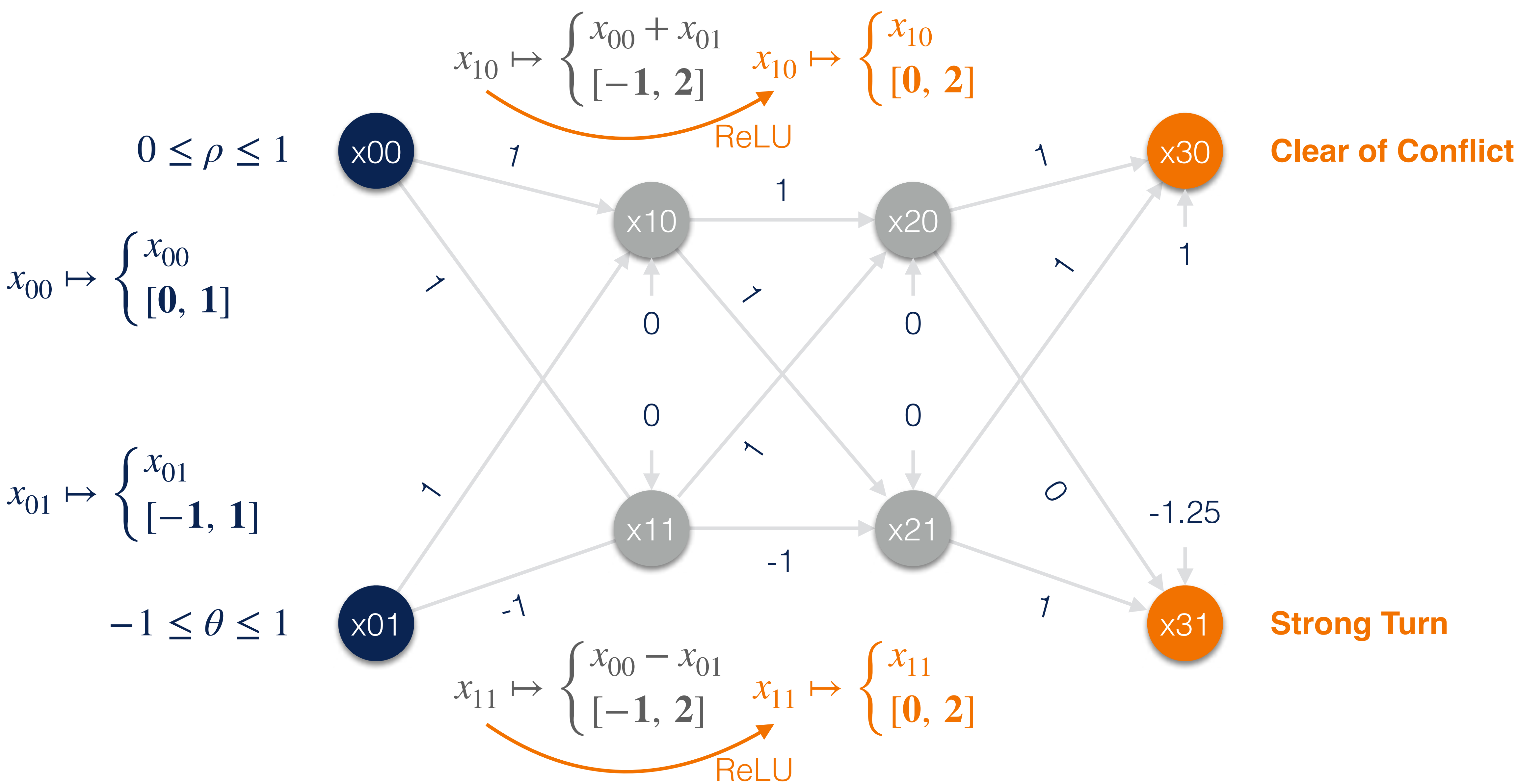
Interval Domain

with Symbolic Constant Propagation [Li19]



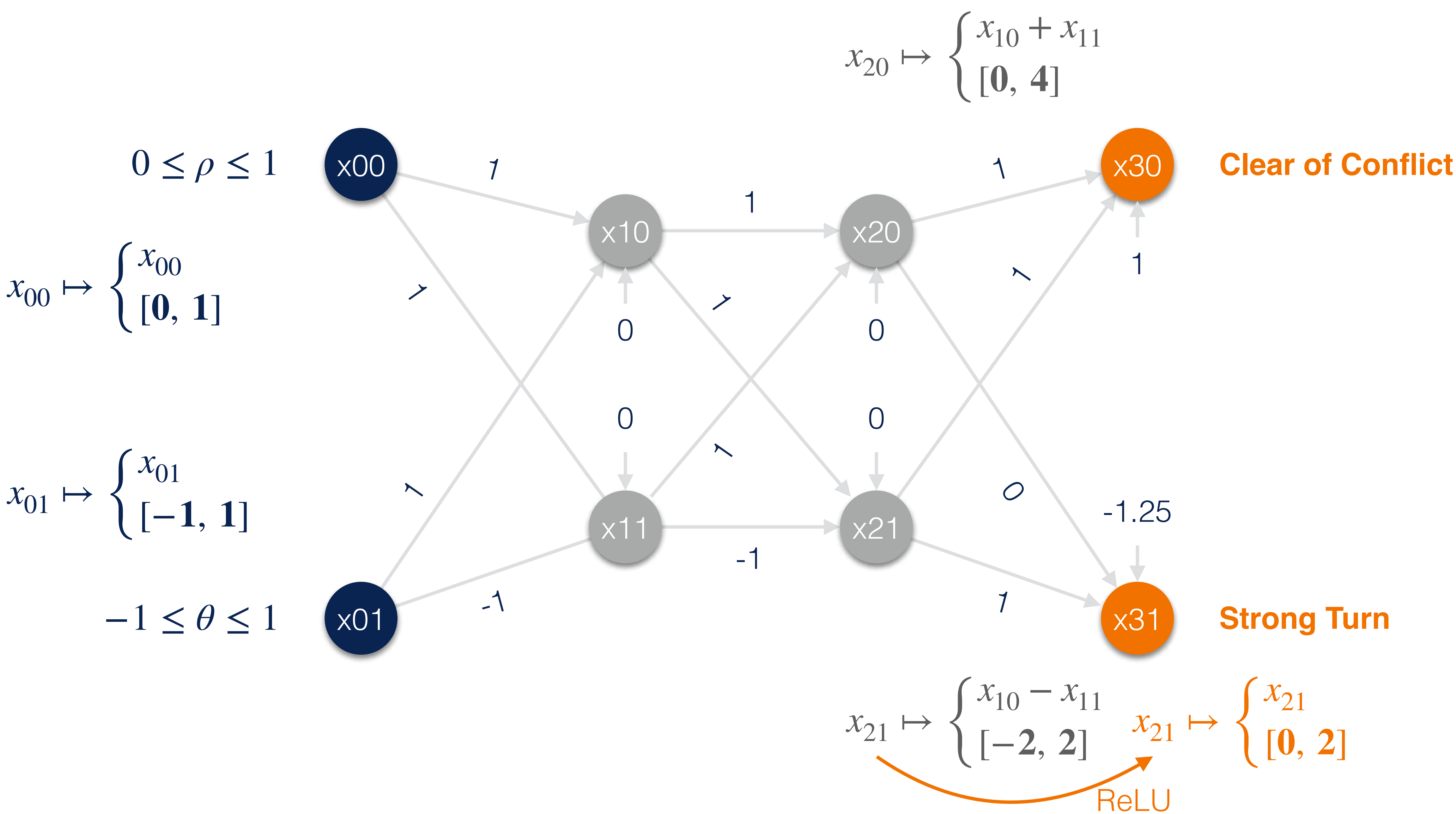
Interval Domain

with Symbolic Constant Propagation [Li19]



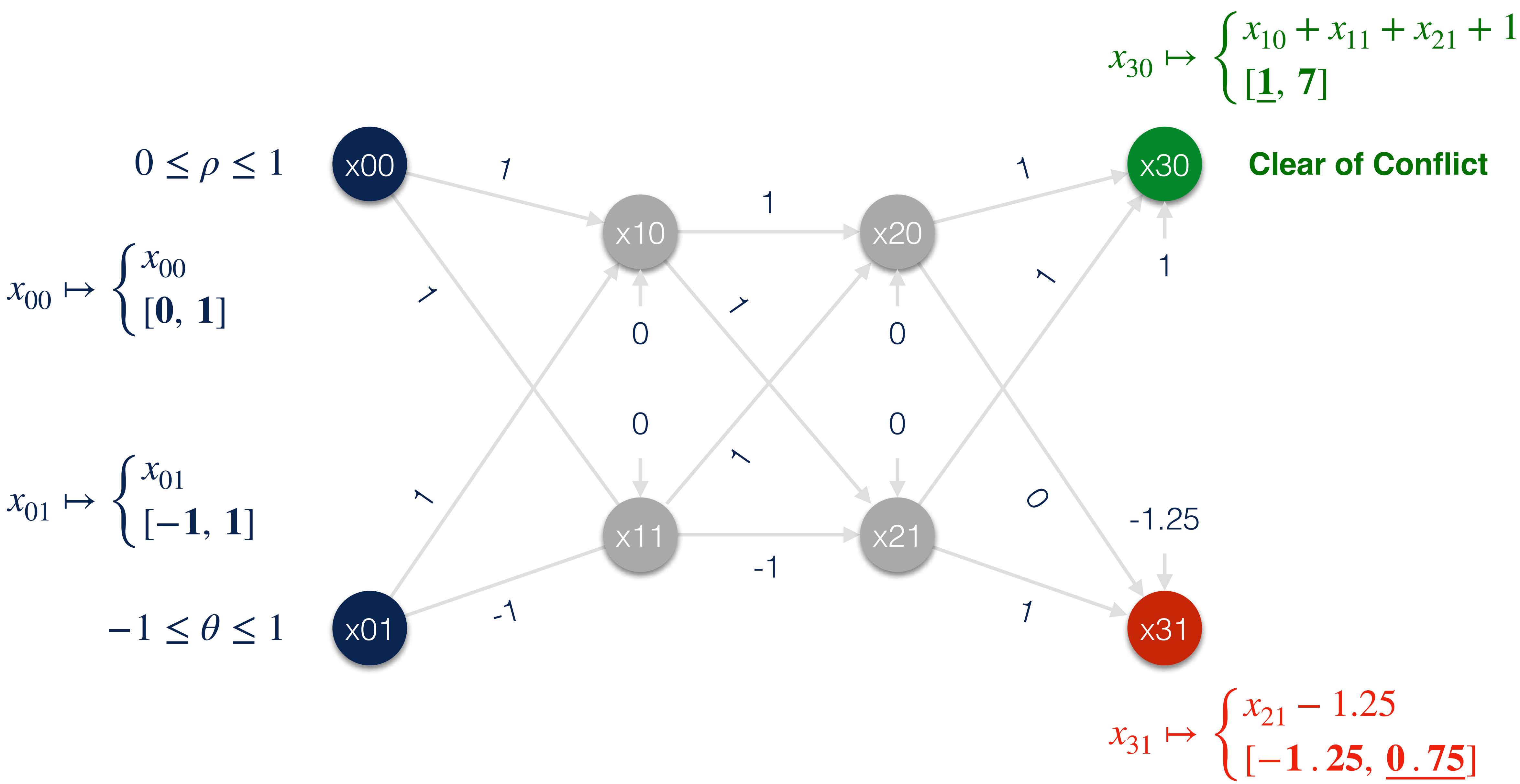
Interval Domain

with Symbolic Constant Propagation [Li19]



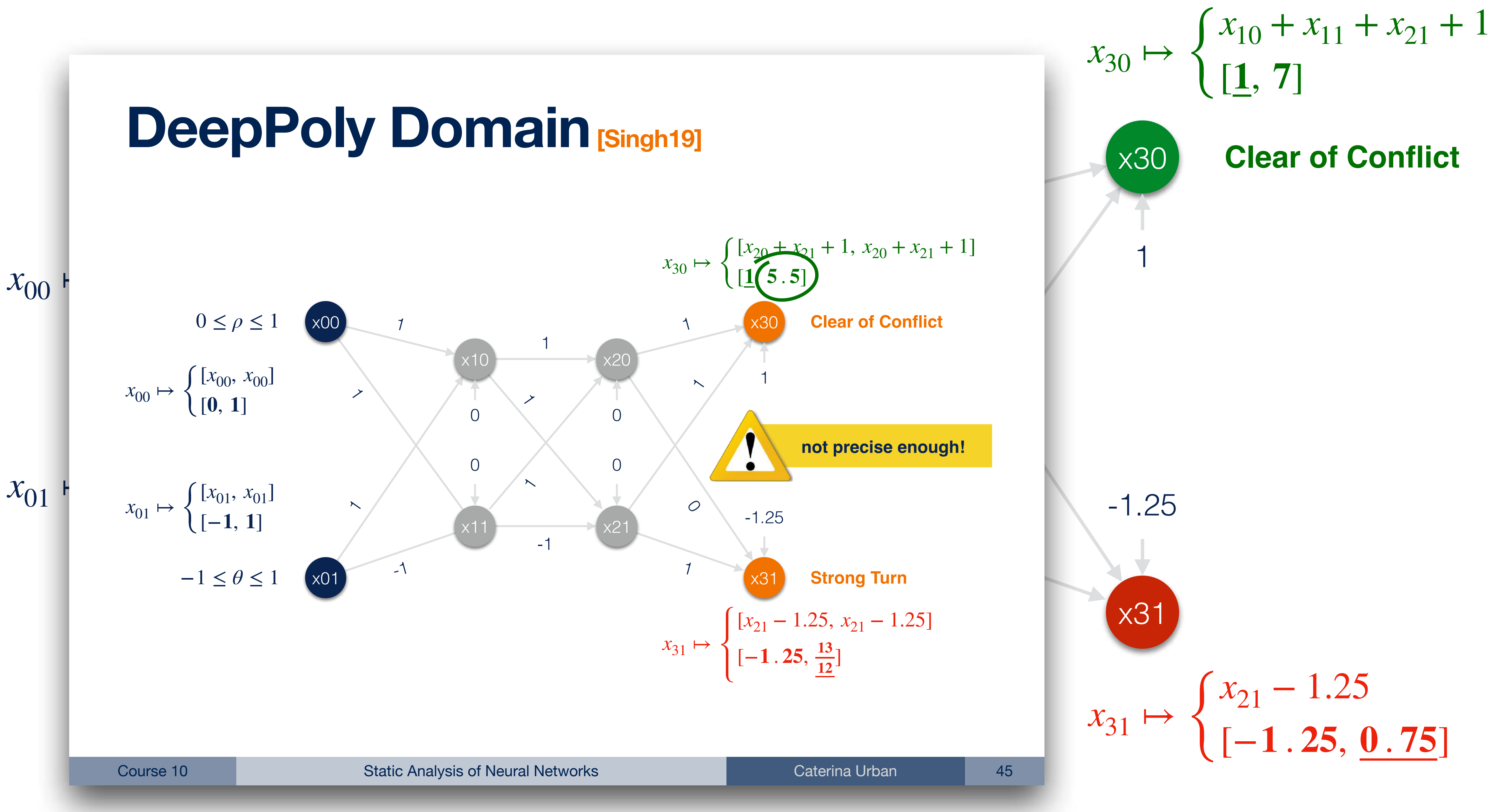
Interval Domain

with Symbolic Constant Propagation [Li19]



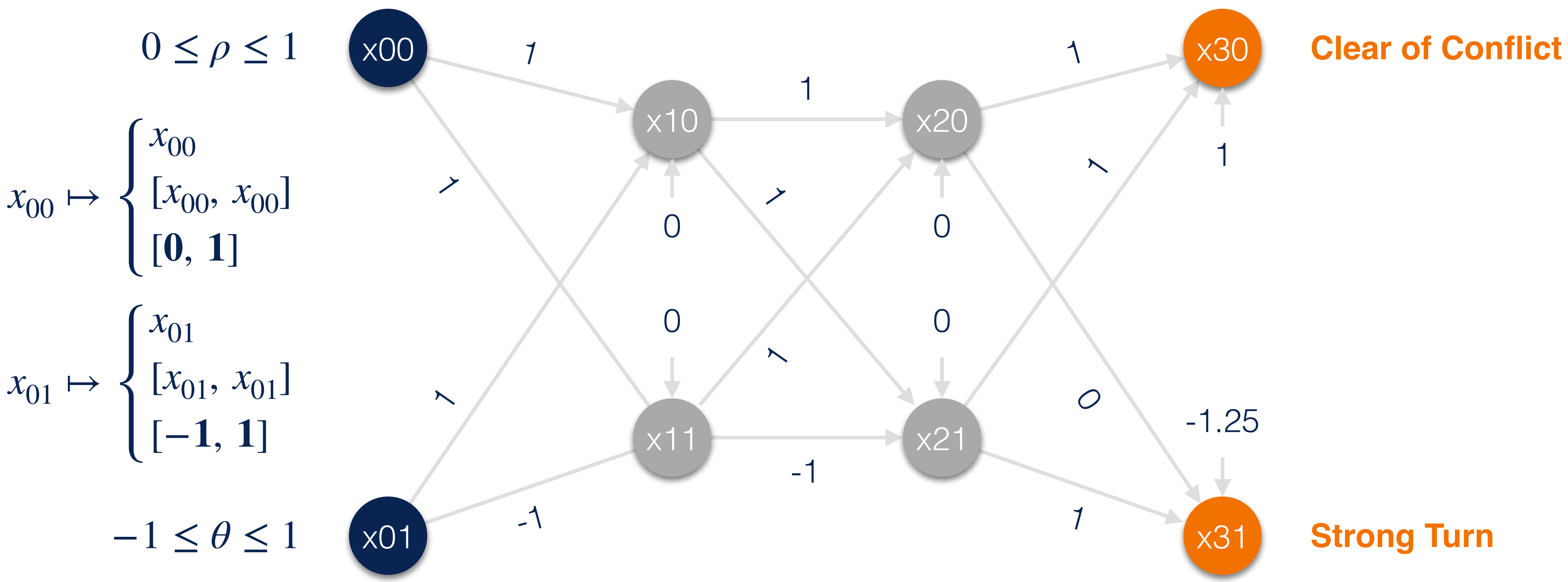
Interval Domain

with Symbolic Constant Propagation [Li19]

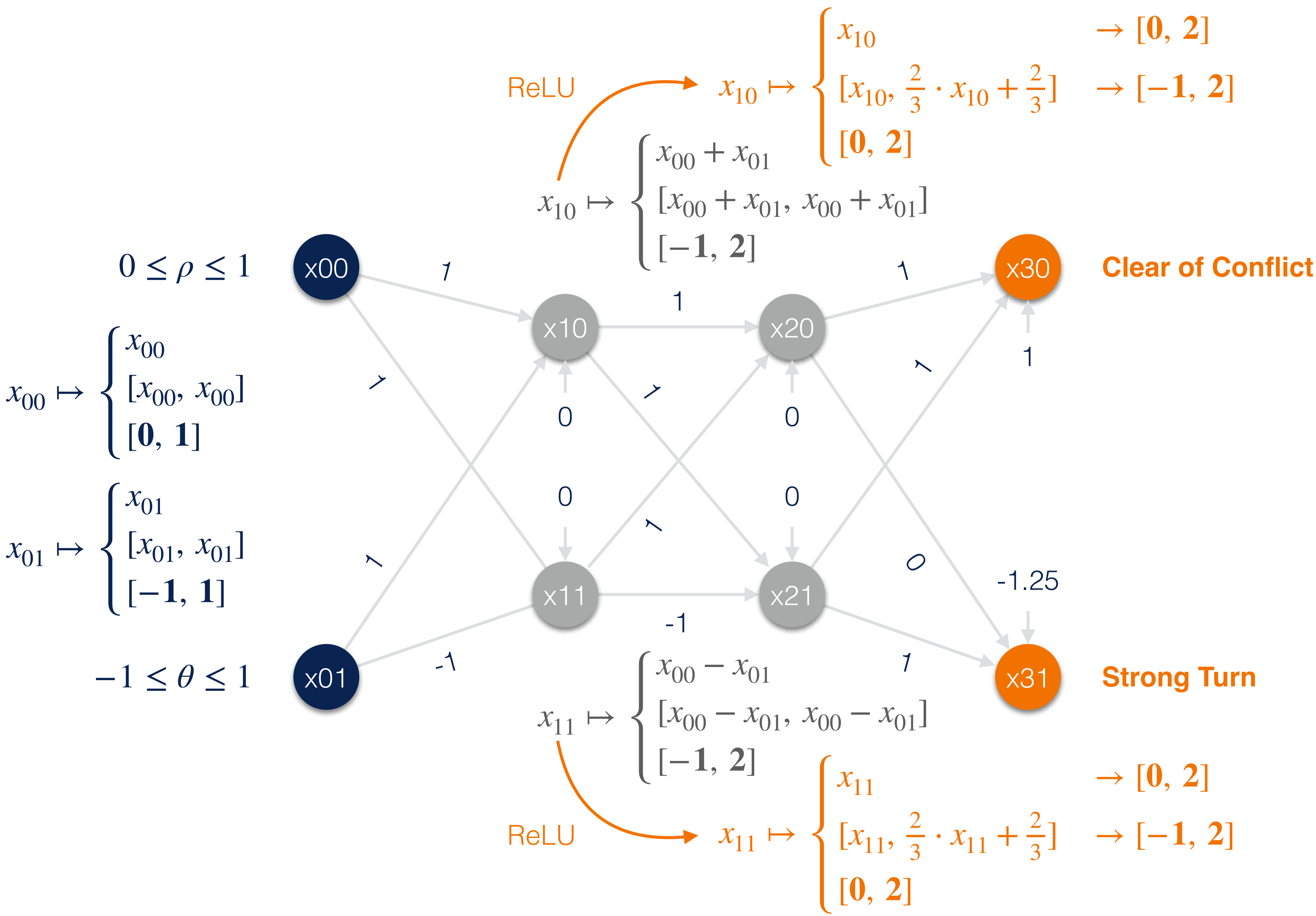


Product Domain

DeepPoly with Symbolic Constant Propagation



Product Domain



Product Domain

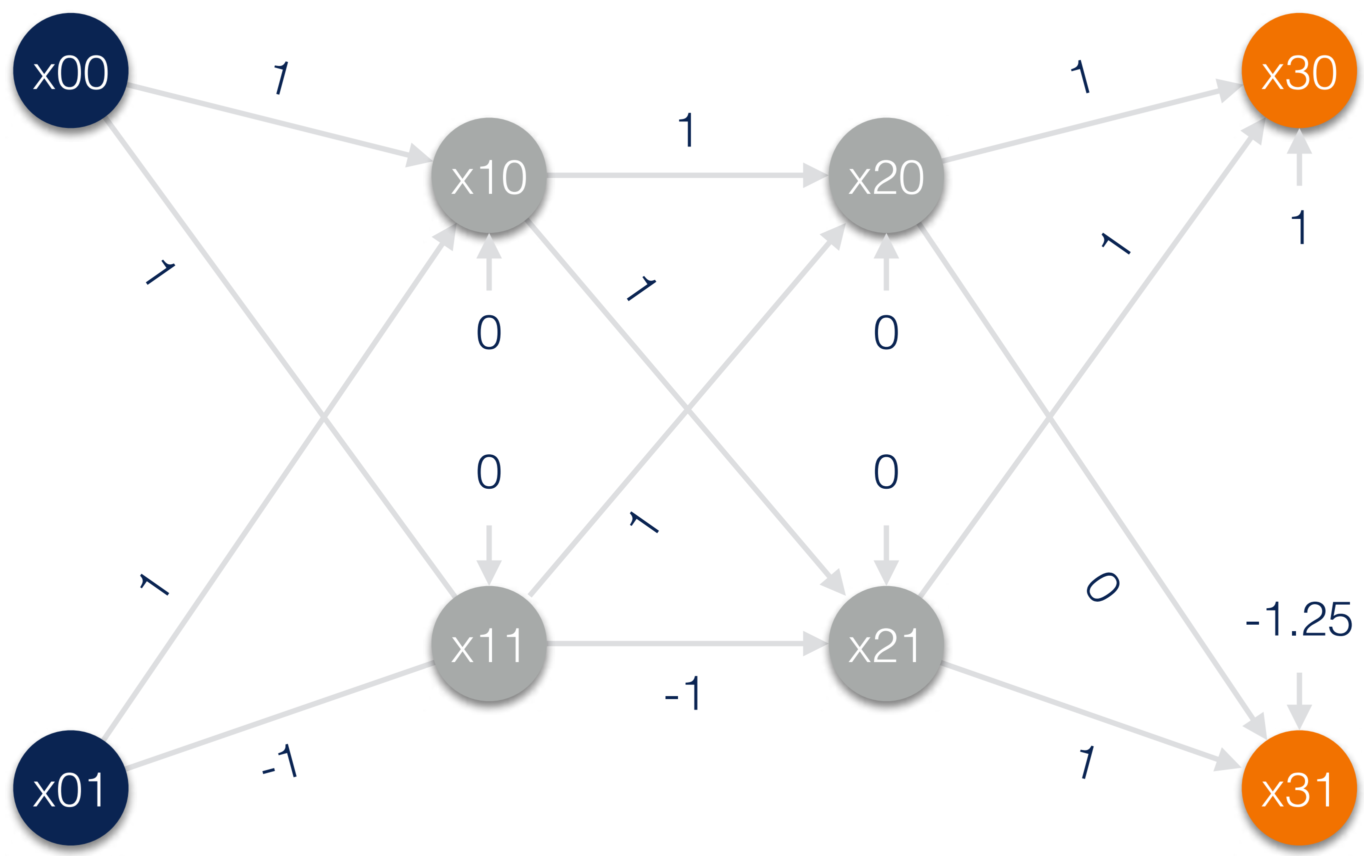
$$x_{20} \mapsto \begin{cases} x_{10}+x_{11} & \to [\mathbf{0},\ \mathbf{4}] \\ [x_{10}+x_{11},\ x_{10}+x_{11}] & \to [\mathbf{0},\ \frac{\mathbf{8}}{\mathbf{3}}] \\ [\mathbf{0},\ \frac{\mathbf{8}}{\mathbf{3}}] & \end{cases}$$

$0 \leq \rho \leq 1$

$$x_{00} \mapsto \begin{cases} x_{00} \\ [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} x_{01} \\ [x_{01},\ x_{01}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$-1 \leq \theta \leq 1$



Clear of Conflict

Strong Turn

$$x_{21} \mapsto \begin{cases} x_{10}-x_{11} & \to [\mathbf{-2},\ \mathbf{2}] \\ [x_{10}-x_{11},\ x_{10}-x_{11}] & \to [-\frac{\mathbf{7}}{\mathbf{3}},\ \frac{\mathbf{7}}{\mathbf{3}}] \\ [\mathbf{-2},\ \mathbf{2}] & \end{cases}$$

ReLU

$$x_{21} \mapsto \begin{cases} x_{21} & \to [\mathbf{0},\ \mathbf{2}] \\ [0,\ 0.5 \cdot x_{21} + 0.5] & \to [\mathbf{0},\ \frac{\mathbf{5}}{\mathbf{3}}] \\ [\mathbf{0},\ \frac{\mathbf{5}}{\mathbf{3}}] & \end{cases}$$

Product Domain

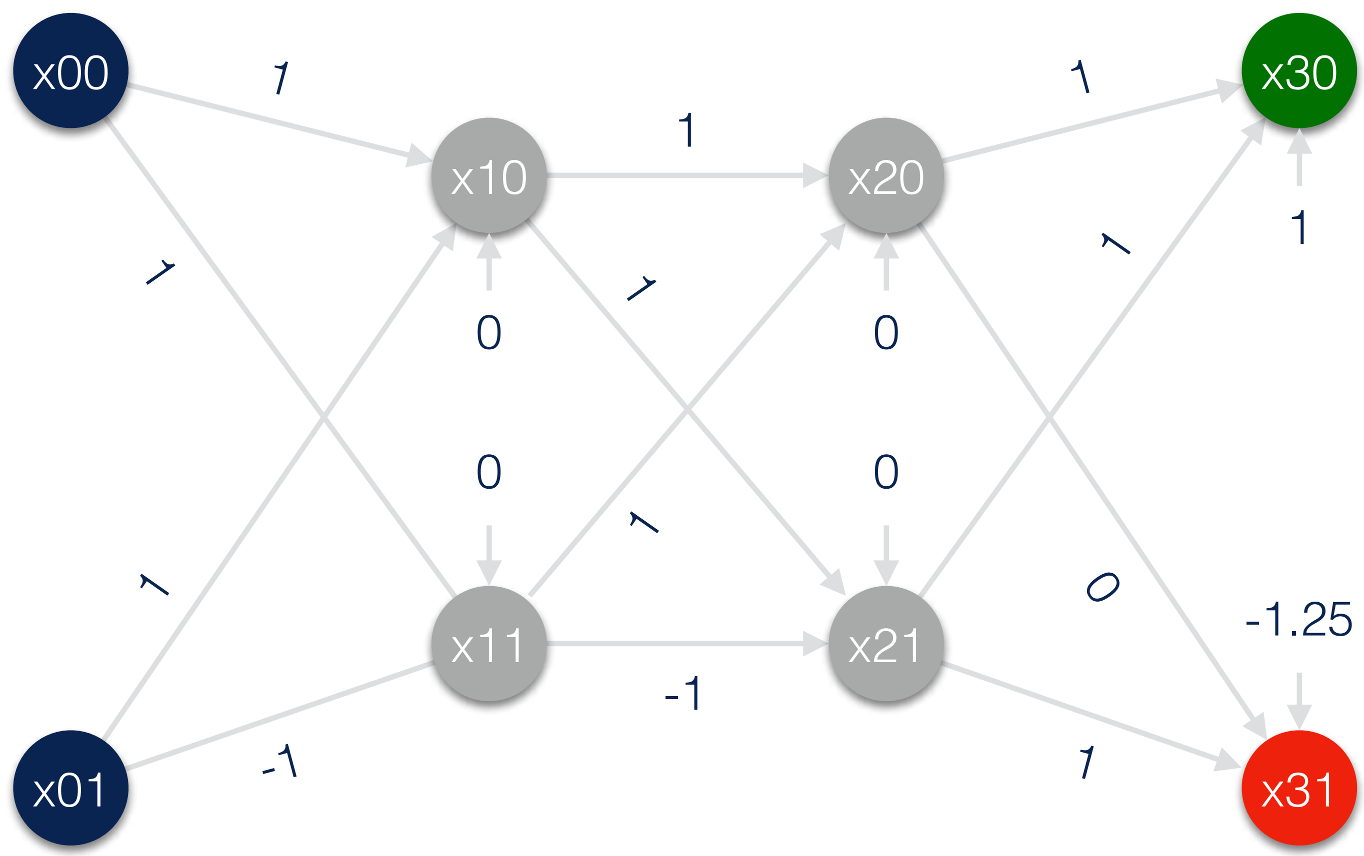
$$x_{30} \mapsto \begin{cases} x_{10} + x_{11} + x_{21} + 1 & \rightarrow [\mathbf{1},\ \mathbf{\tfrac{20}{3}}] \\ [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] & \rightarrow [\mathbf{1},\ \mathbf{4.5}] \\ [\underline{\mathbf{1}},\ \mathbf{4.5}] & \end{cases}$$

$0 \leq \rho \leq 1$

$$x_{00} \mapsto \begin{cases} x_{00} \\ [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} x_{01} \\ [x_{01},\ x_{01}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$-1 \leq \theta \leq 1$



Clear of Conflict

$$x_{31} \mapsto \begin{cases} x_{21} - 1.25 & \rightarrow [\mathbf{-1.25},\ \mathbf{\tfrac{5}{12}}] \\ [x_{21} - 1.25,\ x_{21} - 1.25] & \rightarrow [\mathbf{-1.25},\ \mathbf{\tfrac{5}{12}}] \\ [\mathbf{-1.25},\ \underline{\mathbf{\tfrac{5}{12}}}] & \end{cases}$$

Reading Assignment

Star-Based Reachability Analysis of Deep Neural Networks

Hoang-Dung Tran[1], Diago Manzanas Lopez[1], Patrick Musau[1], Xiaodong Yang[1], Luan Viet Nguyen[2], Weiming Xiang[1], and Taylor T. Johnson[1]

[1] Institute for Software Integrated Systems, Vanderbilt University, TN, USA
[2] Department of Computer and Information Science, University of Pennsylvania, PA, USA

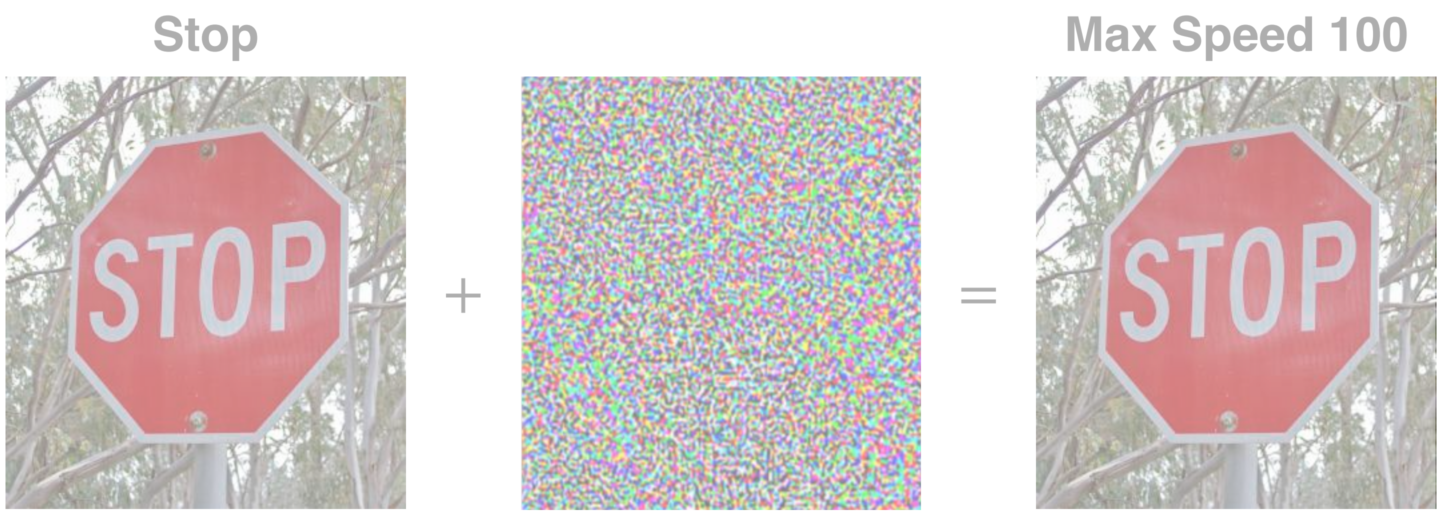
Abstract. This paper proposes novel reachability algorithms for both exact (sound and complete) and over-approximation (sound) analysis for deep neural networks (DNNs). The approach uses star sets as a symbolic representation of sets of states, which are known in short as stars and provide an effective representation of high-dimensional polytopes. Our star-based reachability algorithms can be applied to several problems in analyzing the robustness of machine learning methods, such as safety and robustness verification of DNNs. Our star-based reachability algorithms are implemented in a software prototype called the neural network verification (NNV) tool that is publicly available for evaluation and comparison. Our experiments show that when verifying ACAS Xu neural networks on a multi-core platform, our exact reachability algorithm is on average about 19 times faster than Reluplex, a satisfiability modulo theory (SMT)-based approach. Furthermore, our approach can visualize the precise behavior of DNNs because the reachable states are computed in the method. Notably, in the case that a DNN violates a safety property, the exact reachability algorithm can construct a complete set of counterexamples. Our star-based over-approximate reachability algorithm is on average 118 times faster than Reluplex on the verification of properties for ACAS Xu networks, even without exploiting the parallelism that comes naturally in our method. Additionally, our over-approximate reachability is much less conservative than DeepZ and DeepPoly, recent approaches utilizing zonotopes and other abstract domains that fail to verify many properties of ACAS Xu networks due to their conservativeness. Moreover, our star-based over-approximate reachability algorithm obtains better robustness bounds in comparison with DeepZ and DeepPoly when verifying the robustness of image classification DNNs.

1 Introduction

Deep neural networks (DNNs) have become one of the most powerful techniques to deal with challenging and complex problems such as image processing [15]

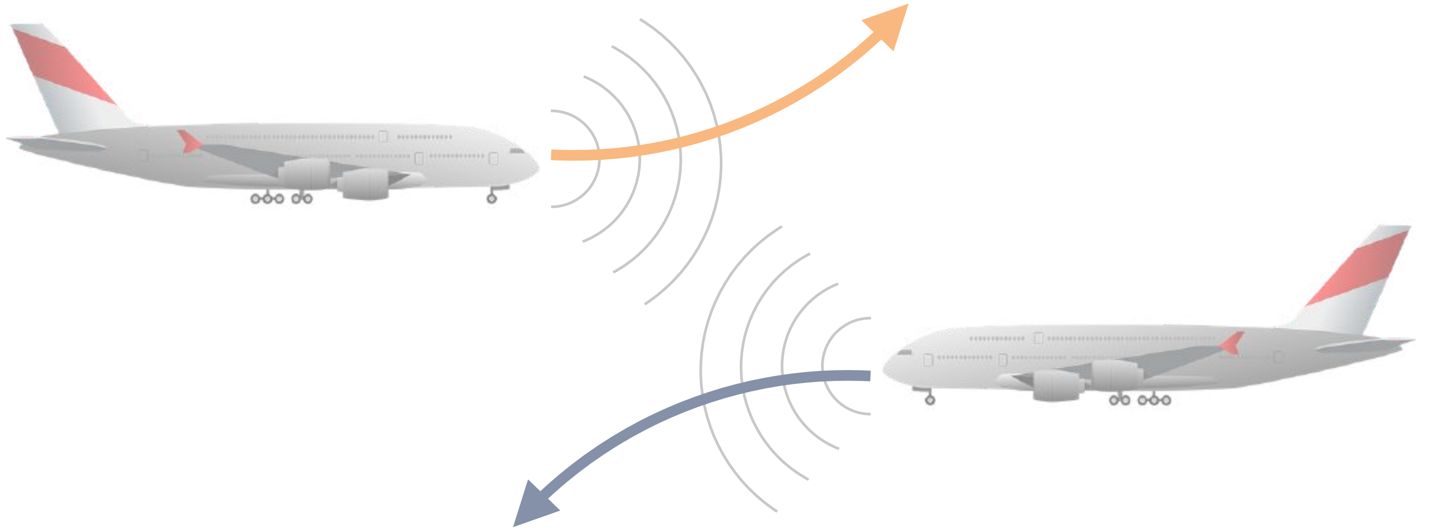
Stability

Goal G3 in [Kurd03]

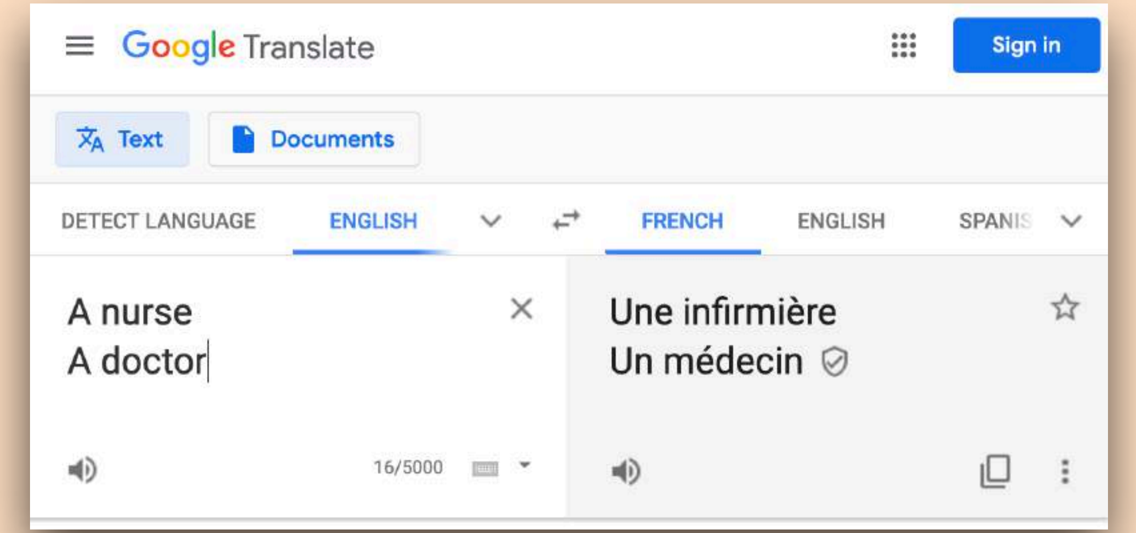


Safety

Goal G4 in [Kurd03]



Fairness



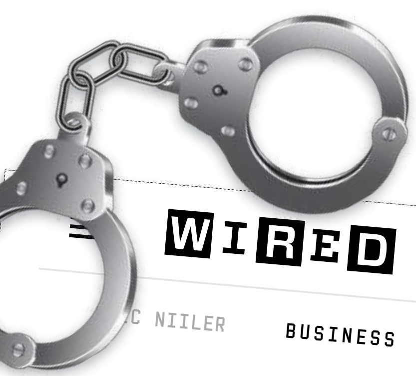
ML Impacts Our Society

WIRED

In 2019, predictive algorithms will start to make banking fair for all

AUTOMATED BACKGROUND CHECKS ARE DECIDING WHO'S FIT FOR A HOME

By Colin Lecher | @colinlecher | Feb 1, 2019, 8:00am EST



WIRED BUSINESS MORE SIGN IN SUBSCRIBE

C NIILER BUSINESS 03.25.2019 07:00 AM

Can AI Be a Fair Judge in Court? Estonia Thinks So

Estonia plans to use an artificial intelligence program to decide some small-claims cases, part of a push to make government services smarter.

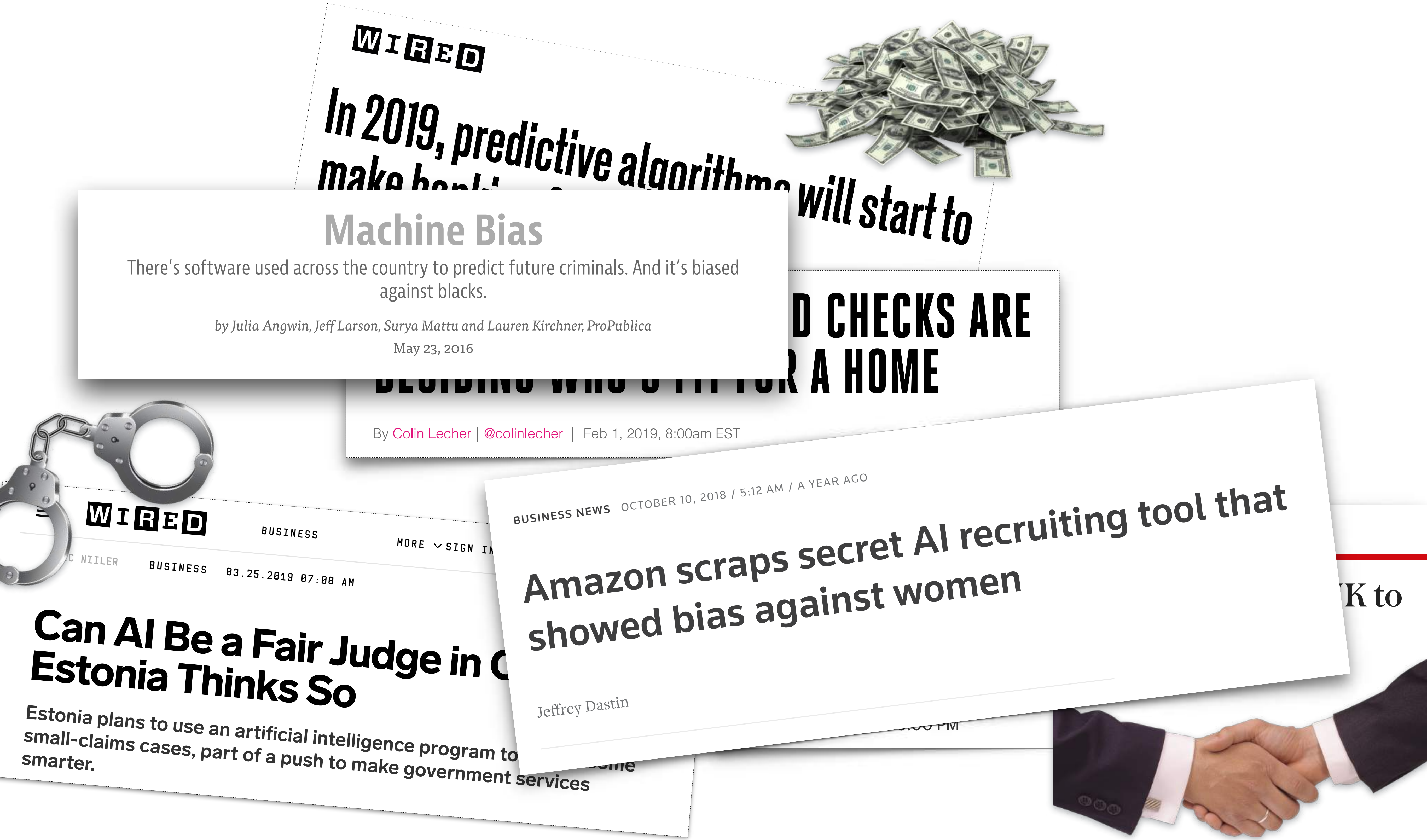
The Telegraph

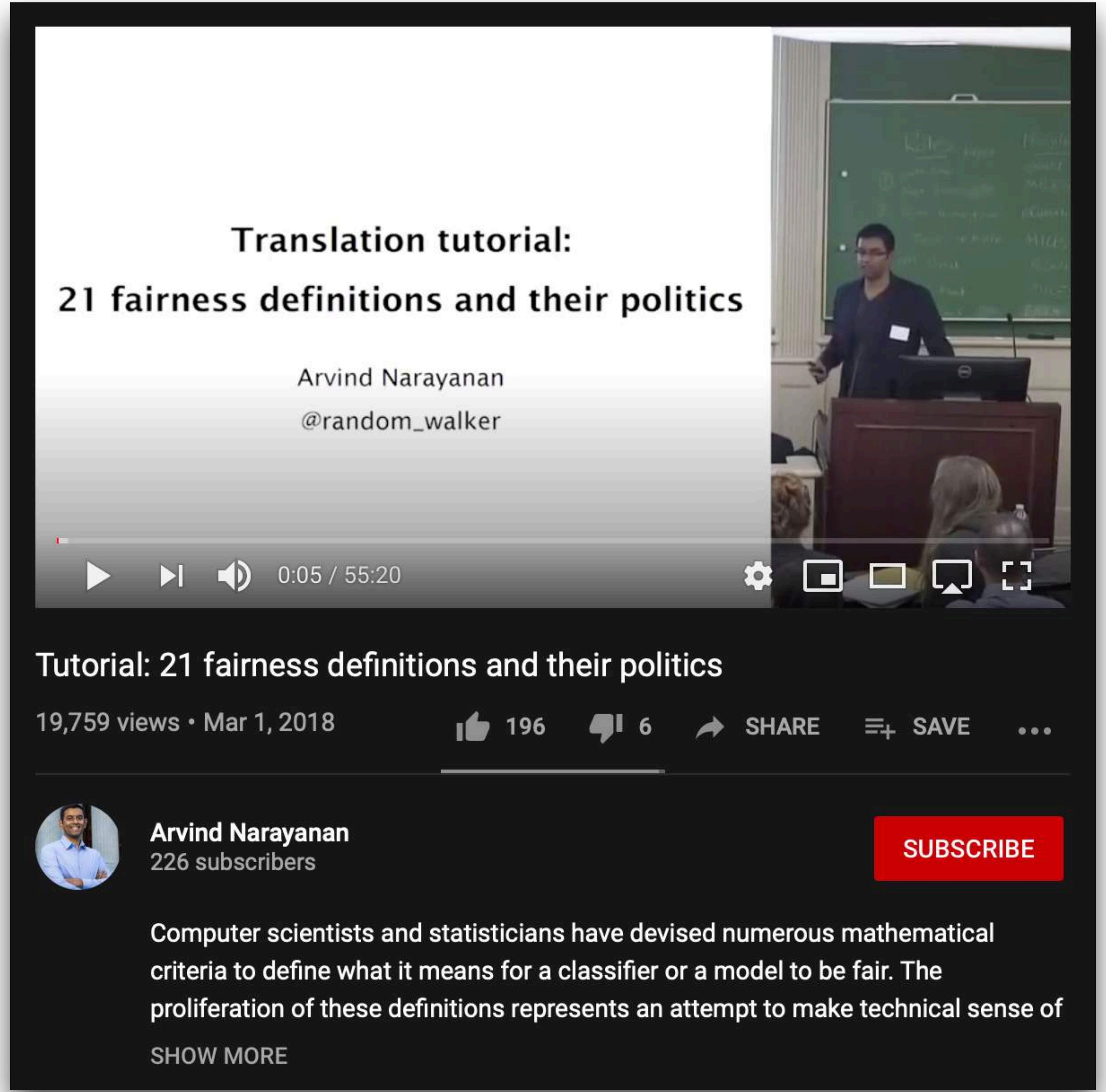
AI used for first time in job interviews in UK to find best applicants

By Charles Hymas

27 SEPTEMBER 2019 • 10:00 PM

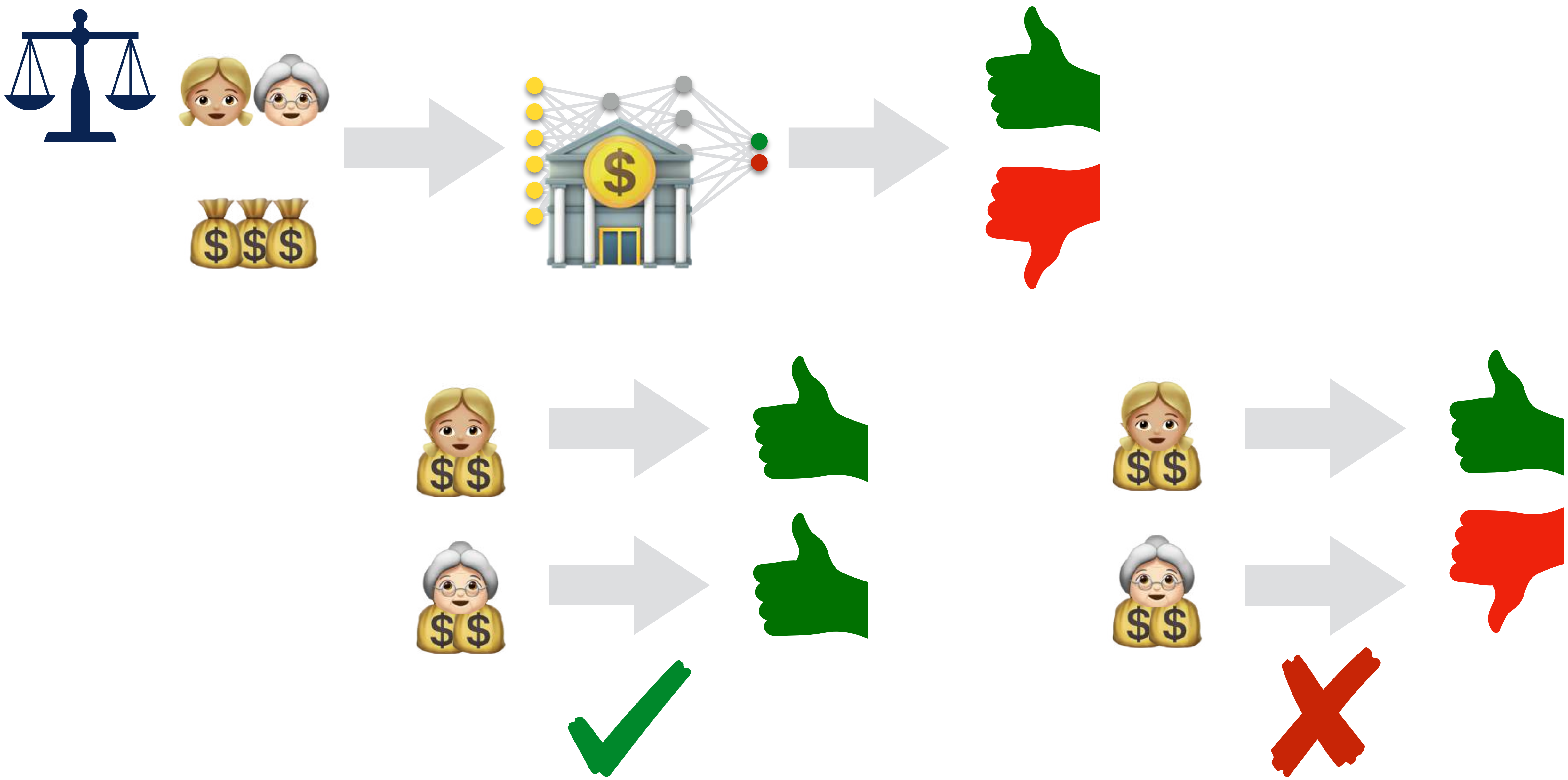
ML Impacts Our Society




Translation tutorial:
21 fairness definitions and their politics
Arvind Narayanan
@random_walker
0:05 / 55:20
Tutorial: 21 fairness definitions and their politics
19,759 views • Mar 1, 2018
196
6
SHARE
SAVE
Arvind Narayanan
226 subscribers
SUBSCRIBE
Computer scientists and statisticians have devised numerous mathematical criteria to define what it means for a classifier or a model to be fair. The proliferation of these definitions represents an attempt to make technical sense of
SHOW MORE

Dependency Fairness [Galhotra17]

The **classification** is **independent of** the values of the **sensitive inputs**



Dependency Fairness

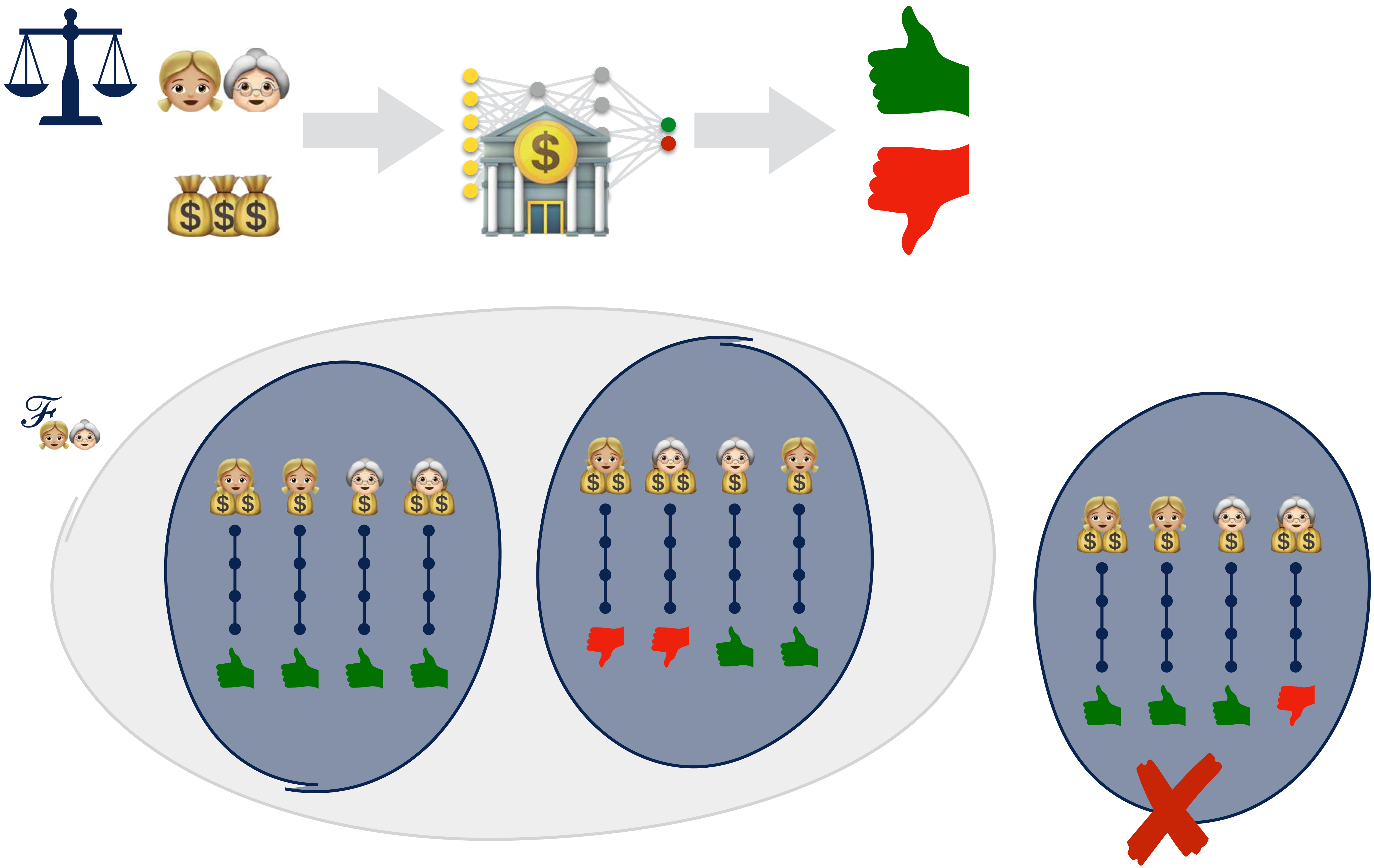
$$\mathcal{F}_i \stackrel{\text{def}}{=} \{[\![M]\!] \in \mathcal{P}(\Sigma^*) \mid \text{UNUSED}_i([\![M]\!])\}$$

$\mathcal{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\begin{aligned}\text{UNUSED}_i([\![M]\!]) \stackrel{\text{def}}{=} \; & \forall t \in [\![M]\!], v \in \mathcal{R} : \; t_0(x_{0,i}) \neq v \Rightarrow \exists t' \in [\![M]\!] : \\ & (\forall 0 \leq j \leq |L_0| : j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j})) \\ & \wedge t'_0(x_{0,i}) = v \\ & \wedge \max_j t_\omega(x_{N,j}) = \max_j t'_\omega(x_{N,j})\end{aligned}$$

Intuitively: **any possible classification outcome** is possible **from any value of the sensitive input** node $x_{0,i}$

Dependency Fairness



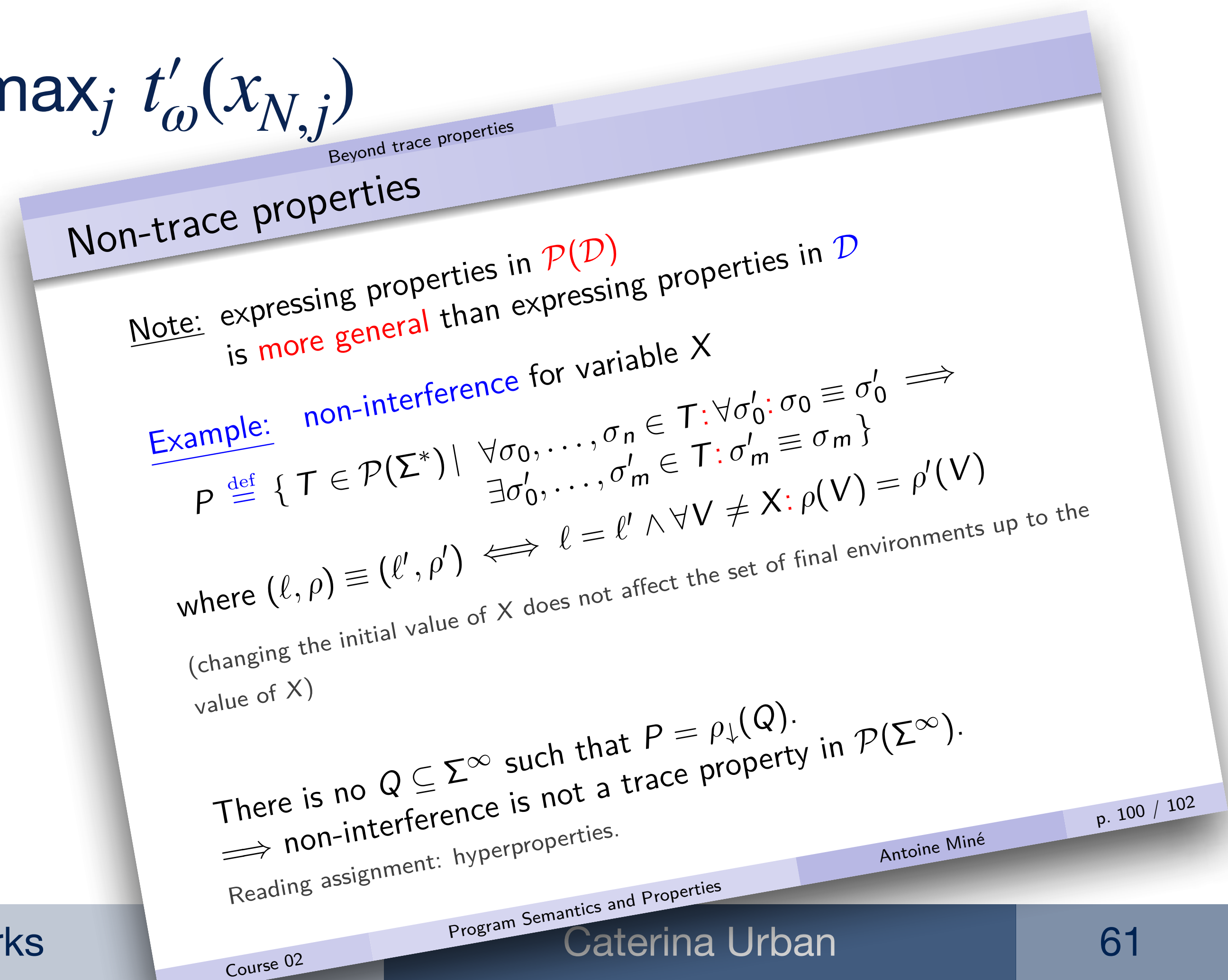
Dependency **Fairness**

$$\mathcal{F}_i \stackrel{\text{def}}{=} \{[\![M]\!] \in \mathcal{P}(\Sigma^*) \mid \text{UNUSED}_i([\![M]\!])\}$$

$\mathcal{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\begin{aligned}\text{UNUSED}_i([\![M]\!]) \stackrel{\text{def}}{=} \forall t \in [\![M]\!], v \in \mathcal{R}: t_0(x_{0,i}) \neq v \Rightarrow \exists t' \in [\![M]\!]:& \\ (\forall 0 \leq j \leq |L_0|: j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j}))& \\ \wedge\, t'_0(x_{0,i}) = v& \\ \wedge \max_j t_\omega(x_{N,j}) = \max_j t'_\omega(x_{N,j})&\end{aligned}$$

Intuitively: **any possible classification outcome** is possible **from any value of the sensitive input** node $x_{0,i}$


Beyond trace properties

Non-trace properties

Note: expressing properties in $\mathcal{P}(\mathcal{D})$ is more general than expressing properties in $\mathcal{D}$

Example: non-interference for variable X

$$P \stackrel{\text{def}}{=} \{ T \in \mathcal{P}(\Sigma^*) \mid \forall \sigma_0, \ldots, \sigma_n \in T: \forall \sigma'_0: \sigma_0 \equiv \sigma'_0 \Longrightarrow \exists \sigma'_0, \ldots, \sigma'_m \in T: \sigma'_m \equiv \sigma_m \}$$

where $(\ell, \rho) \equiv (\ell', \rho') \iff \ell = \ell' \wedge \forall V \neq X: \rho(V) = \rho'(V)$

(changing the initial value of X does not affect the set of final environments up to the value of X)

There is no $Q \subseteq \Sigma^\infty$ such that $P = \rho_\downarrow(Q)$.
$\Longrightarrow$ non-interference is not a trace property in $\mathcal{P}(\Sigma^\infty)$.

Reading assignment: hyperproperties.

Course 02 | Program Semantics and Properties | Antoine Miné | p. 100 / 102

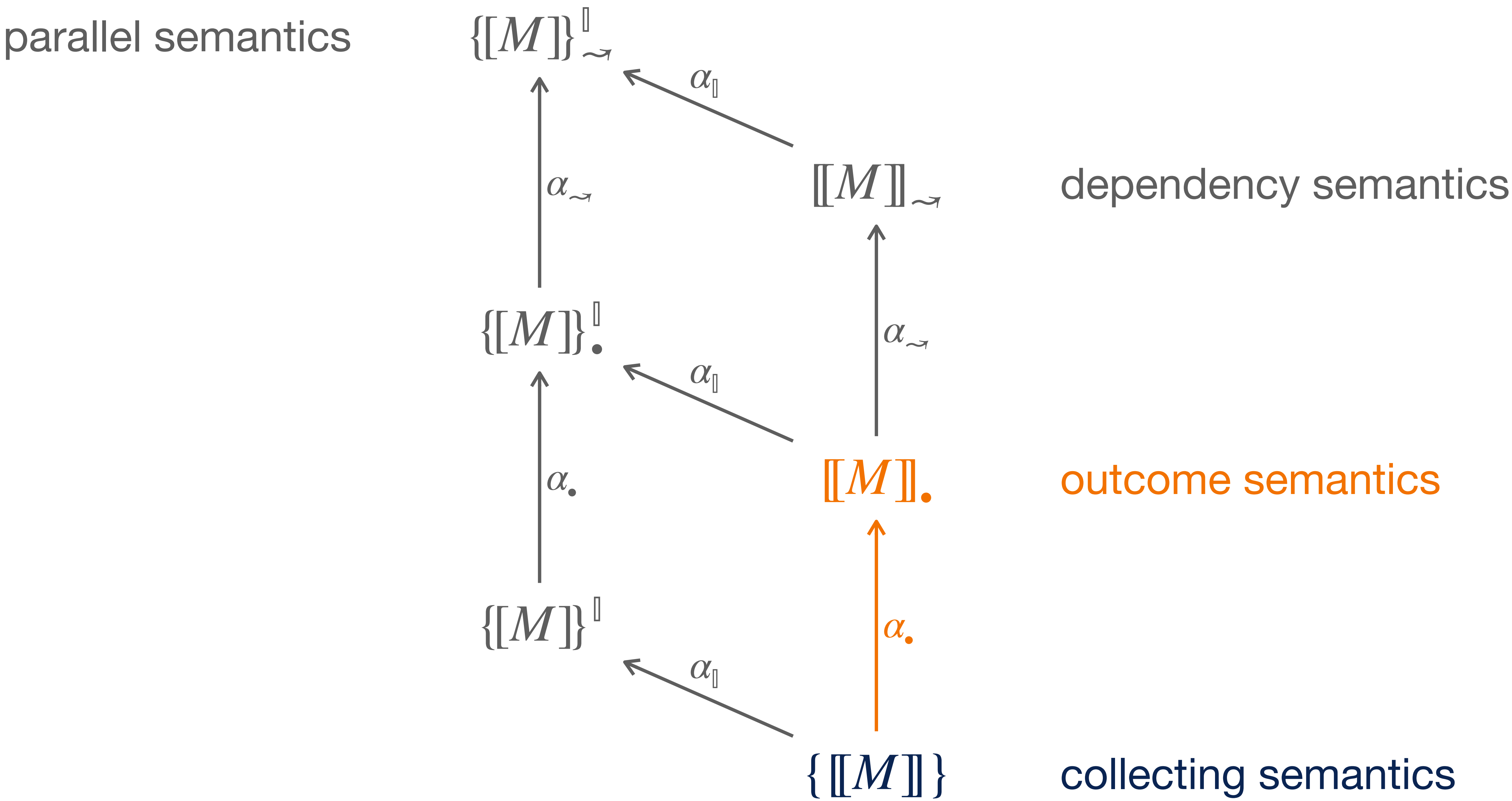
Theorem

$$M \models \mathcal{F}_i \Leftrightarrow \{[\![M]\!]\} \subseteq \mathcal{F}_i$$

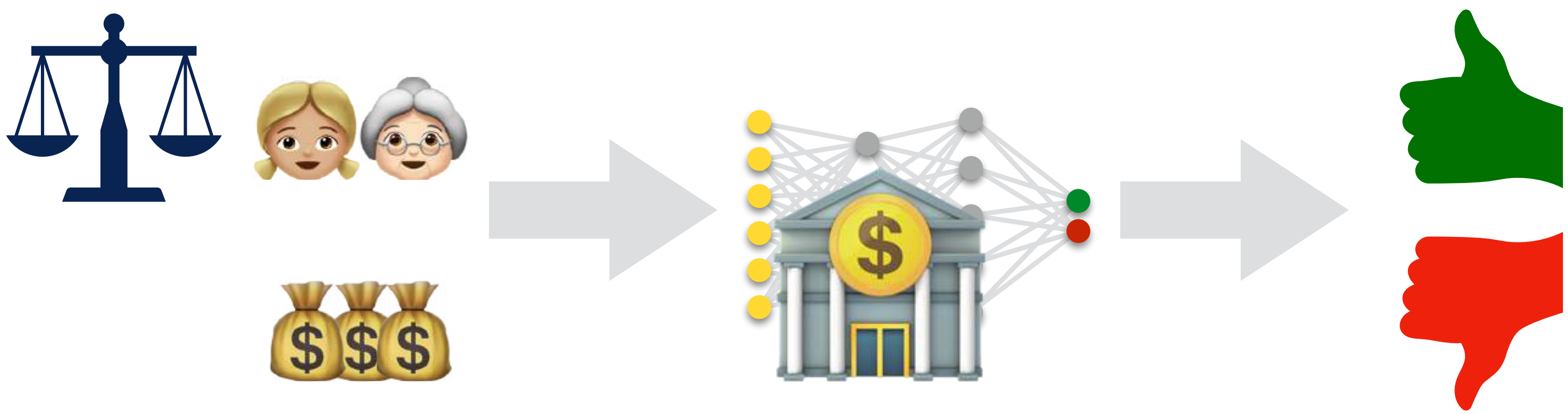
(Another) Hierarchy of Semantics

$$
\begin{tikzcd}
\text{parallel semantics} & \{[M]\}^{\parallel}_{\rightsquigarrow} & & \\
& & {[\![M]\!]_{\rightsquigarrow}} \arrow[ul, "\alpha_{\parallel}"'] & \text{dependency semantics} \\
& \{[M]\}^{\parallel}_{\bullet} \arrow[uu, "\alpha_{\rightsquigarrow}"] & & \\
& & {[\![M]\!]_{\bullet}} \arrow[ul, "\alpha_{\parallel}"'] \arrow[uu, "\alpha_{\rightsquigarrow}"] & \text{outcome semantics} \\
& \{[M]\}^{\parallel} \arrow[uu, "\alpha_{\bullet}"] & & \\
& & \{[\![M]\!]\} \arrow[ul, "\alpha_{\parallel}"'] \arrow[uu, "\alpha_{\bullet}"] & \text{collecting semantics}
\end{tikzcd}
$$

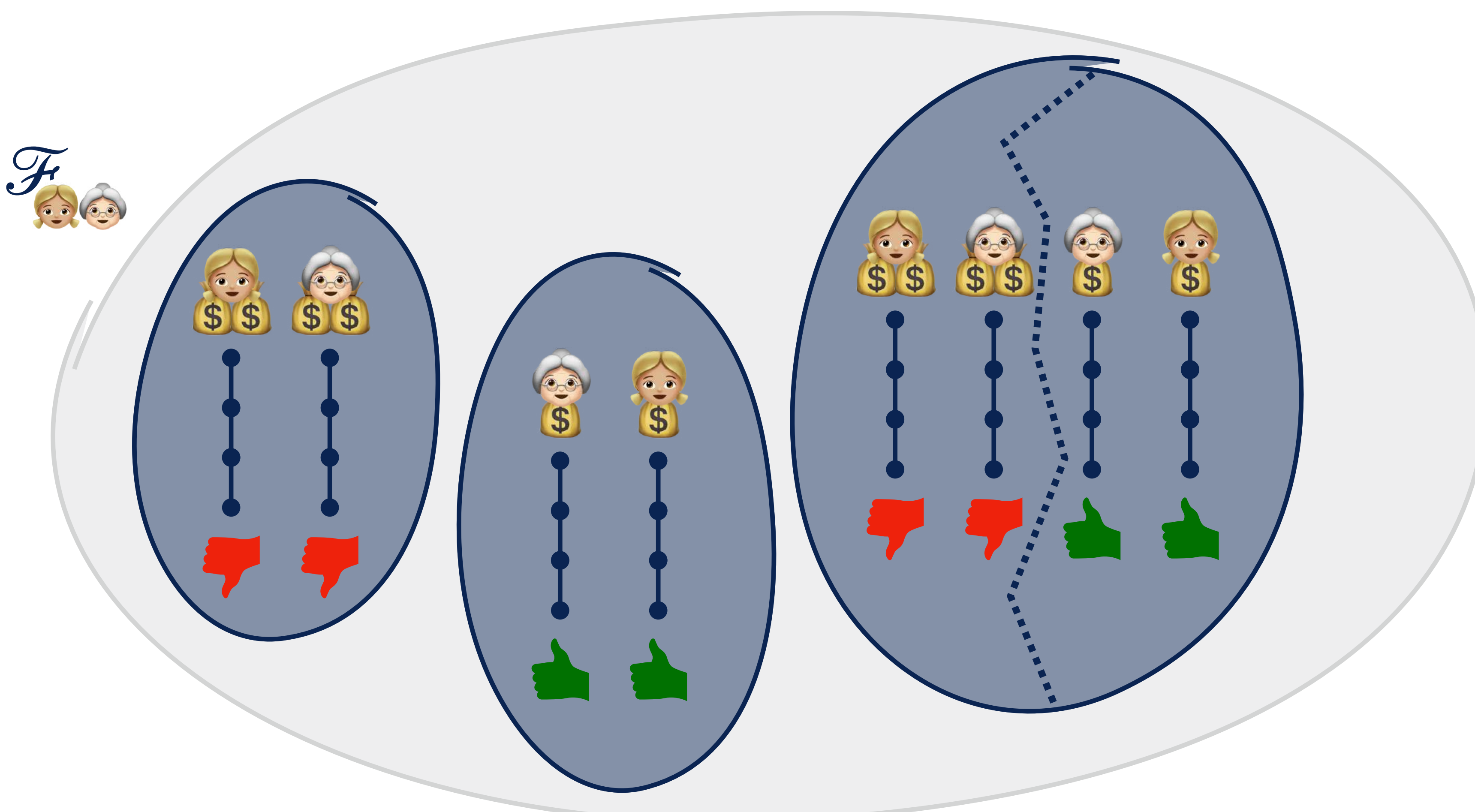
(Another) Hierarchy of Semantics



Outcome Semantics



💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the outcome classification** yields sets of traces that also satisfy dependency fairness



Outcome Semantics

$\mathbb{O} \stackrel{\text{def}}{=} \{\{\sigma \in \Sigma \mid \max_j \sigma(x_{N,j}) = i\} \mid 0 \leq i \leq |L_N|\}$ **outcomes**

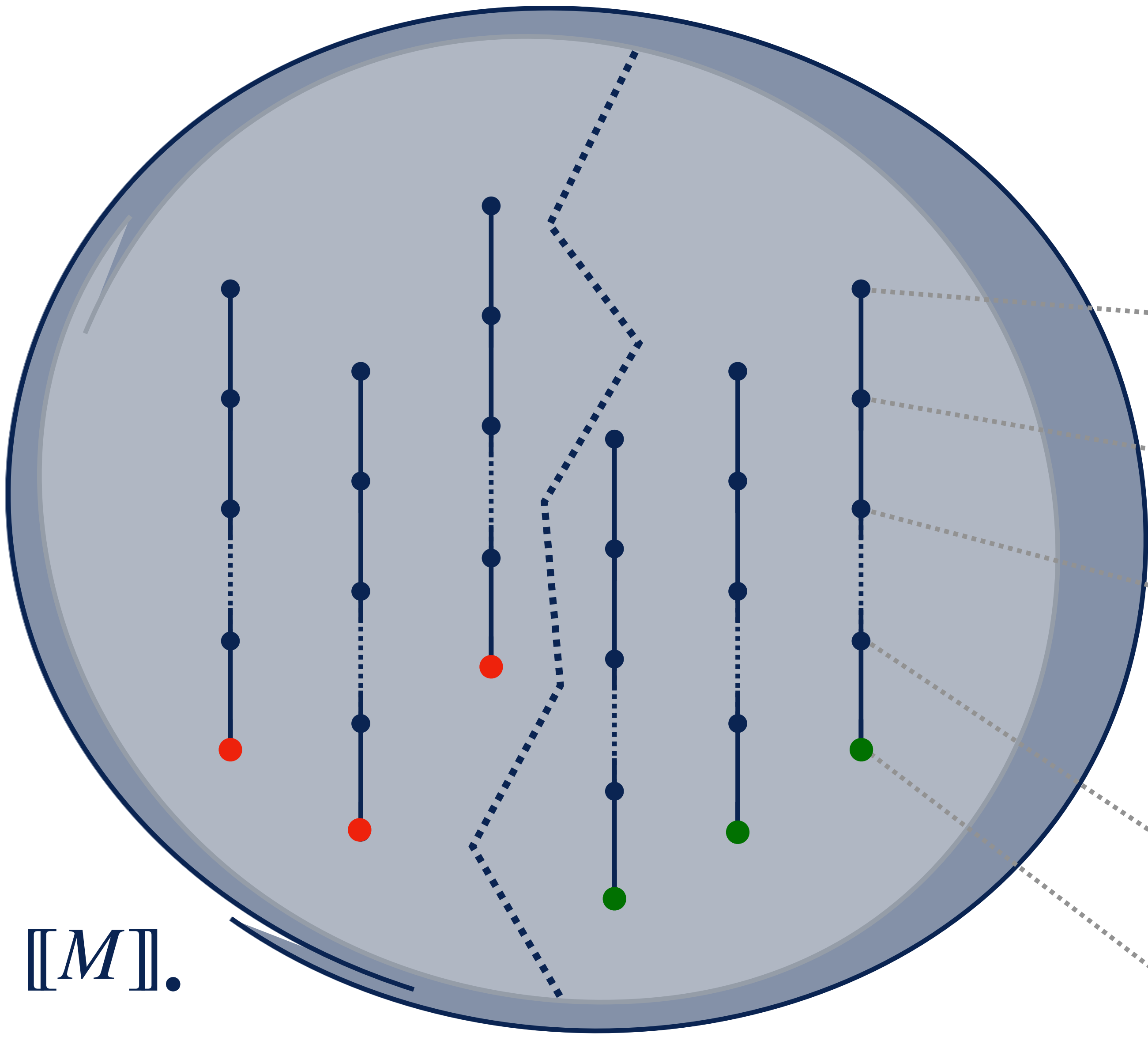
Lemma

$$M \models \mathcal{F}_i \Leftrightarrow \{\{t \in [\![M]\!] \mid t_\omega \in O\} \mid O \in \mathbb{O}\} \subseteq \mathcal{F}_i$$

$$\begin{tikzcd}
\langle \mathcal{P}(\mathcal{P}(\Sigma^*)), \subseteq \rangle \arrow[r, bend right, "\alpha_\bullet"'] & \langle \mathcal{P}(\mathcal{P}(\Sigma^*)), \subseteq_\bullet \rangle \arrow[l, bend right, "\gamma_\bullet"']
\end{tikzcd}$$

$\alpha_\bullet(S) \stackrel{\text{def}}{=} \{\{t \in T \mid t_\omega \in O\} \mid T \in S \wedge O \in \mathbb{O}\}$ **outcome abstraction**

Outcome Semantics



$[\![M]\!]$.

M

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

Outcome Semantics

$\mathbb{O} \overset{\text{def}}{=} \{\{\sigma \in \Sigma \mid \max_j \sigma(x_{N,j}) = i\} \mid 0 \leq i \leq |L_N|\}$ **outcomes**

Lemma

$$M \models \mathcal{F}_i \Leftrightarrow \{\{t \in [\![M]\!] \mid t_\omega \in O\} \mid O \in \mathbb{O}\} \subseteq \mathcal{F}_i$$

$$\begin{tikzcd}
\langle \mathcal{P}(\mathcal{P}(\Sigma^*)), \subseteq \rangle \arrow[r, bend right, "\alpha_\bullet"'] & \langle \mathcal{P}(\mathcal{P}(\Sigma^*)), \subseteq_\bullet \rangle \arrow[l, bend right, "\gamma_\bullet"']
\end{tikzcd}$$

$\alpha_\bullet(S) \overset{\text{def}}{=} \{\{t \in T \mid t_\omega \in O\} \mid T \in S \land O \in \mathbb{O}\}$ **outcome abstraction**

$$[\![M]\!]_\bullet \overset{\text{def}}{=} \alpha_\bullet(\{[\![M]\!]\}) = \{\{t \in [\![M]\!] \mid t_\omega \in O\} \mid O \in \mathbb{O}\}$$

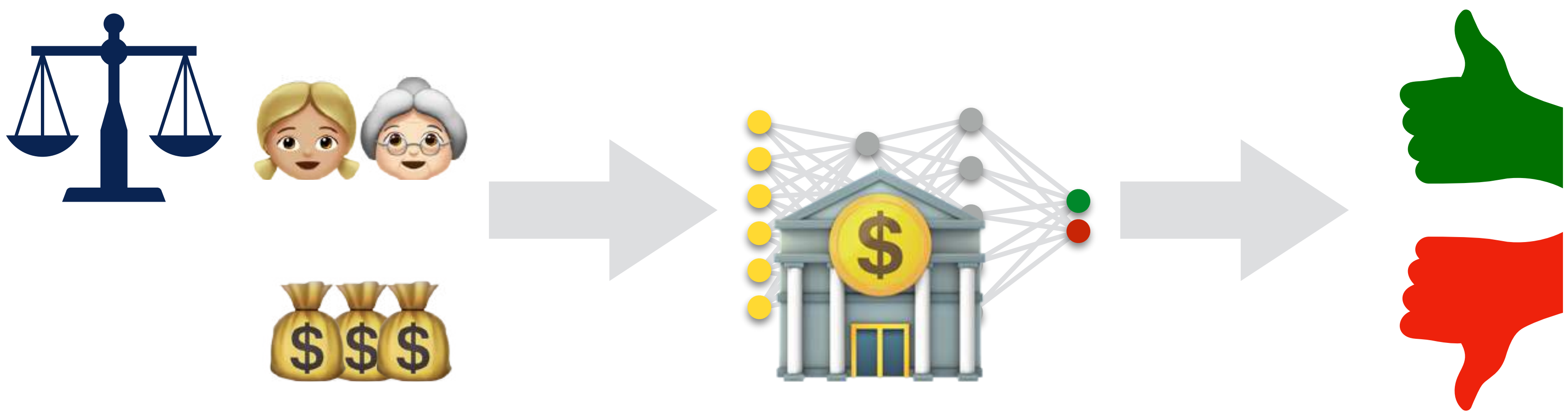
Theorem

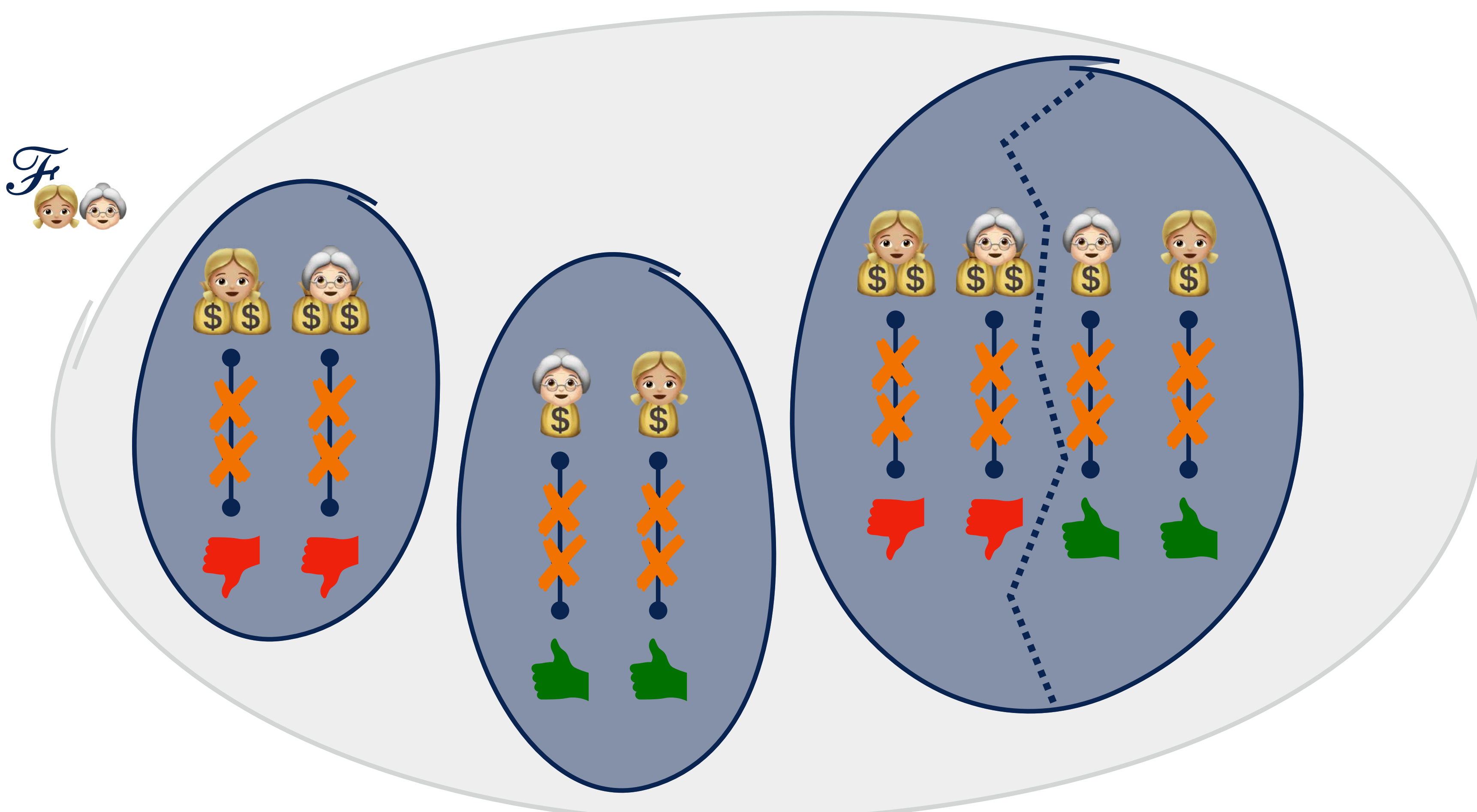
$$M \models \mathcal{F}_i \Leftrightarrow [\![M]\!]_\bullet \subseteq \alpha_\bullet(\mathcal{F}_i) \Leftrightarrow [\![M]\!]_\bullet \subseteq \mathcal{F}_i$$

(Another) Hierarchy of Semantics

$$
\begin{tikzcd}
\text{parallel semantics} & \{[M]\}^{\square}_{\rightsquigarrow} & & \\
& & \llbracket M \rrbracket_{\rightsquigarrow} \arrow[ul, "\alpha_{\square}"'] & \text{dependency semantics} \\
& \{[M]\}^{\square}_{\bullet} \arrow[uu, "\alpha_{\rightsquigarrow}"] & & \\
& & \llbracket M \rrbracket_{\bullet} \arrow[ul, "\alpha_{\square}"'] \arrow[uu, "\alpha_{\rightsquigarrow}"] & \text{outcome semantics} \\
& \{[M]\}^{\square} \arrow[uu, "\alpha_{\bullet}"] & & \\
& & \{\llbracket M \rrbracket\} \arrow[ul, "\alpha_{\square}"'] \arrow[uu, "\alpha_{\bullet}"] & \text{collecting semantics}
\end{tikzcd}
$$

Dependency Semantics





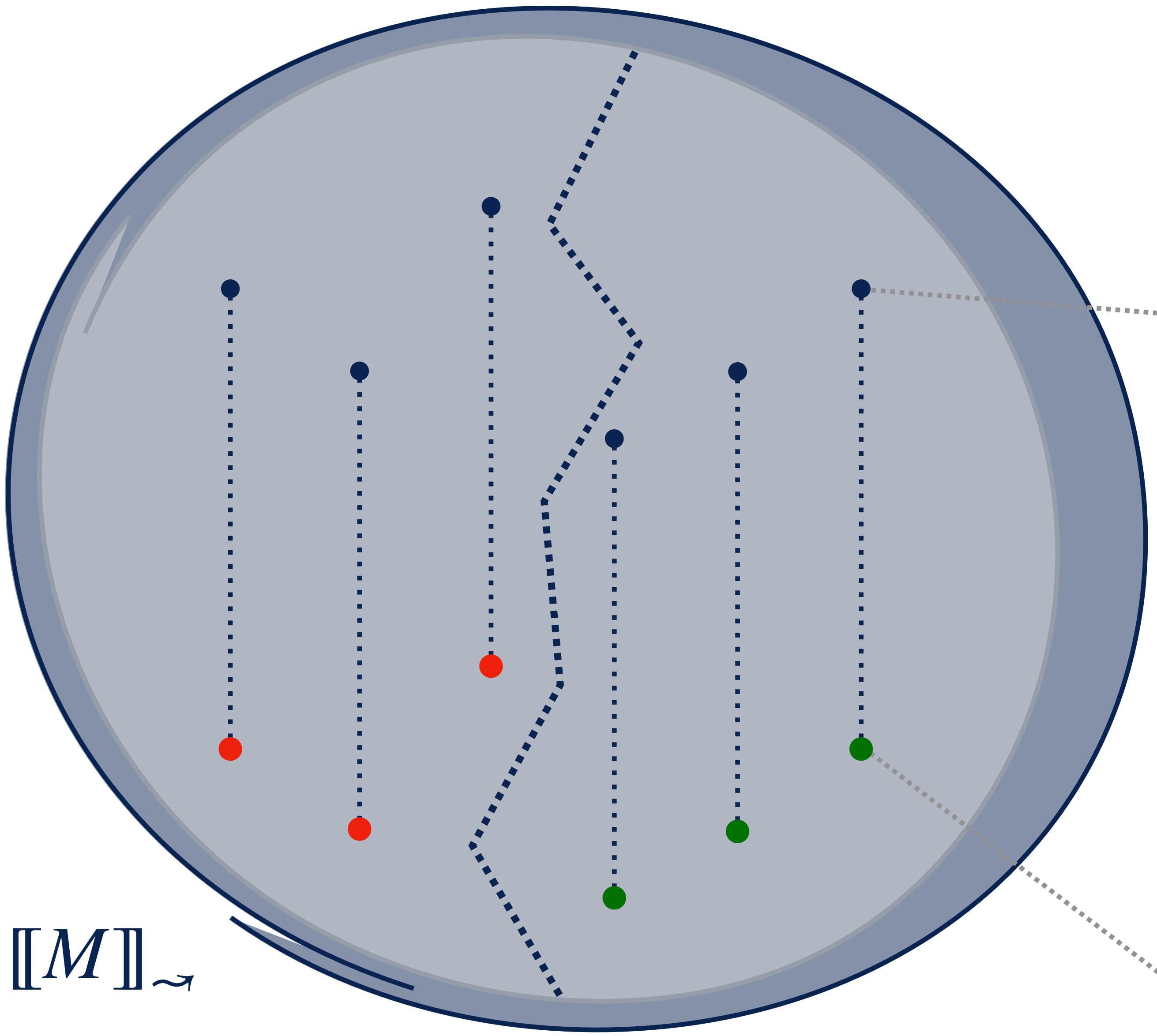
💡 to reason about dependency fairness **we do not need to consider all intermediate computations** between the initial and final states of a trace

Dependency Semantics

$$\begin{tikzcd} \langle \mathscr{P}(\mathscr{P}(\Sigma^*)), \subseteq_{\bullet} \rangle \arrow[r, bend right, "\alpha_{\rightsquigarrow}"'] & \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\rightsquigarrow} \rangle \arrow[l, bend right, "\gamma_{\rightsquigarrow}"'] \end{tikzcd}$$

$$\alpha_{\rightsquigarrow}(S) \stackrel{\text{def}}{=} \{\{\langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in T\} \mid T \in S\}$$ **dependency abstraction**

Dependency Semantics



$[\![M]\!]_{\rightsquigarrow}$

M

x00 = **input**()
x01 = **input**()

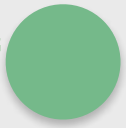
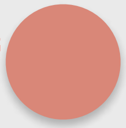
x10 = **-0.31** * x00 + **0.99** * x01 + (**-0.63**)
x11 = **-1.25** * x00 + (**-0.64**) * x01 + **1.88**

x10 = 0 **if** x10 < 0 **else** x10
x11 = 0 **if** x11 < 0 **else** x11

x20 = **0.40** * x10 + **1.21** * x11 + **0.00**
x21 = **0.64** * x10 + **0.69** * x11 + (**-0.39**)

x20 = 0 **if** x20 < 0 **else** x20
x21 = 0 **if** x21 < 0 **else** x21

x30 = **0.26** * x20 + **0.33** * x21 + **0.45**
X31 = **1.42** * x20 + **0.40** * x21 + (**-0.45**)

return ‘’ **if** x31 < 30 **else** ‘’

Dependency Semantics

$$\begin{tikzcd} \langle \mathscr{P}(\mathscr{P}(\Sigma^*)), \subseteq_{\bullet} \rangle \arrow[r, bend right, "\alpha_{\rightsquigarrow}"'] & \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\rightsquigarrow} \rangle \arrow[l, bend right, "\gamma_{\rightsquigarrow}"'] \end{tikzcd}$$

$$\alpha_{\rightsquigarrow}(S) \stackrel{\text{def}}{=} \{ \{ \langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in T \} \mid T \in S \}$$ **dependency abstraction**

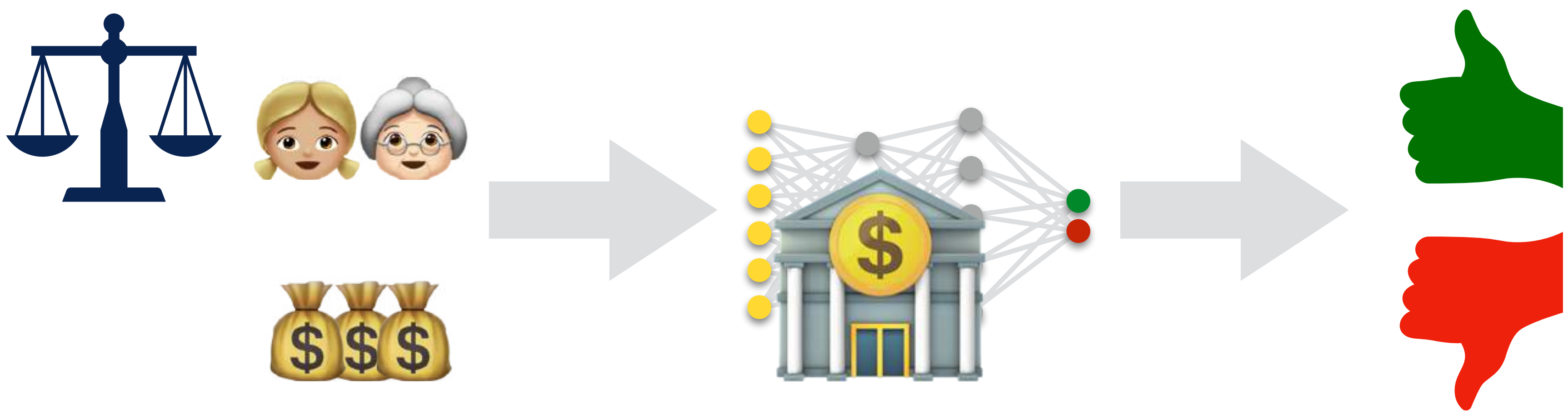
$$[\![M]\!]_{\rightsquigarrow} \stackrel{\text{def}}{=} \alpha_{\rightsquigarrow}([\![M]\!]_{\bullet}) = \{ \{ \langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in [\![M]\!] \wedge t_\omega \in O \} \mid O \in \mathbb{O} \}$$

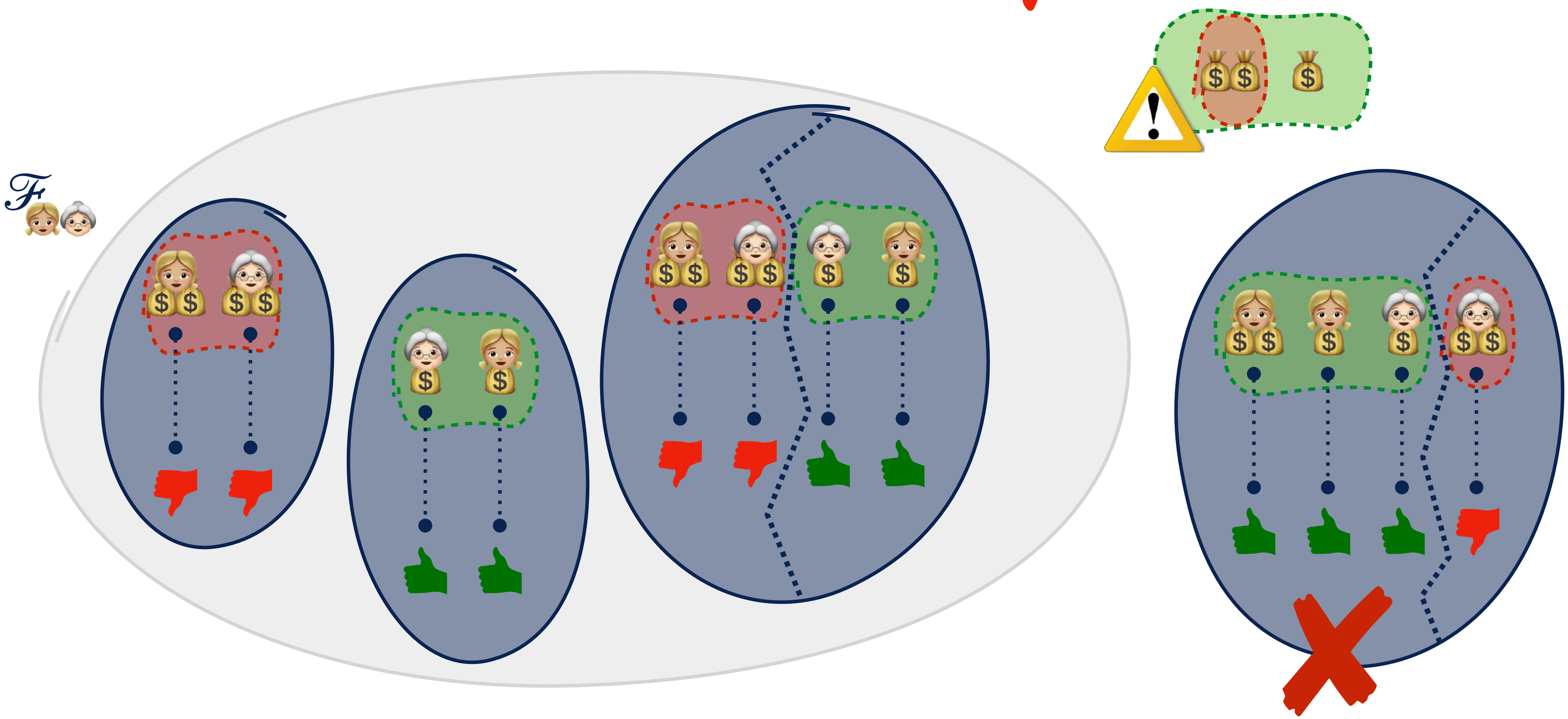
Theorem

$$M \models \mathscr{F}_i \Leftrightarrow [\![M]\!]_{\rightsquigarrow} \subseteq_{\rightsquigarrow} \alpha_{\rightsquigarrow}(\alpha_{\bullet}(\mathscr{F}_i)) \Leftrightarrow [\![M]\!]_{\rightsquigarrow} \subseteq_{\rightsquigarrow} \alpha_{\rightsquigarrow}(\mathscr{F}_i)$$

Dependency Semantics

💡 **partitioning with respect to the outcome** classification **induces a partition of the** space of **values** of the input nodes ***used* for classification**





Dependency Semantics

$$\begin{tikzcd} \langle \mathscr{P}(\mathscr{P}(\Sigma^*)), \subseteq_\bullet \rangle \arrow[r, bend right, "\alpha_\rightsquigarrow"'] & \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_\rightsquigarrow \rangle \arrow[l, bend right, "\gamma_\rightsquigarrow"'] \end{tikzcd}$$

$$\alpha_\rightsquigarrow(S) \overset{\text{def}}{=} \{\{\langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in T\} \mid T \in S\}$$

dependency abstraction

$$[\![M]\!]_\rightsquigarrow \overset{\text{def}}{=} \alpha_\rightsquigarrow([\![M]\!]_\bullet) = \{\{\langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in [\![M]\!] \land t_\omega \in O\} \mid O \in \mathbb{O}\}$$

Theorem

$$M \models \mathscr{F}_i \Leftrightarrow [\![M]\!]_\rightsquigarrow \subseteq_\rightsquigarrow \alpha_\rightsquigarrow(\alpha_\bullet(\mathscr{F}_i)) \Leftrightarrow [\![M]\!]_\rightsquigarrow \subseteq_\rightsquigarrow \alpha_\rightsquigarrow(\mathscr{F}_i)$$

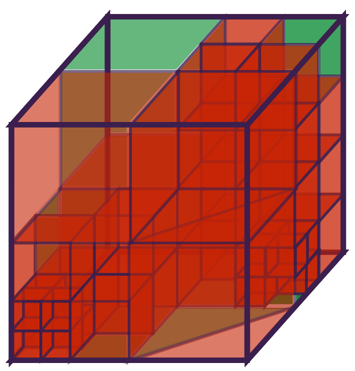
Lemma

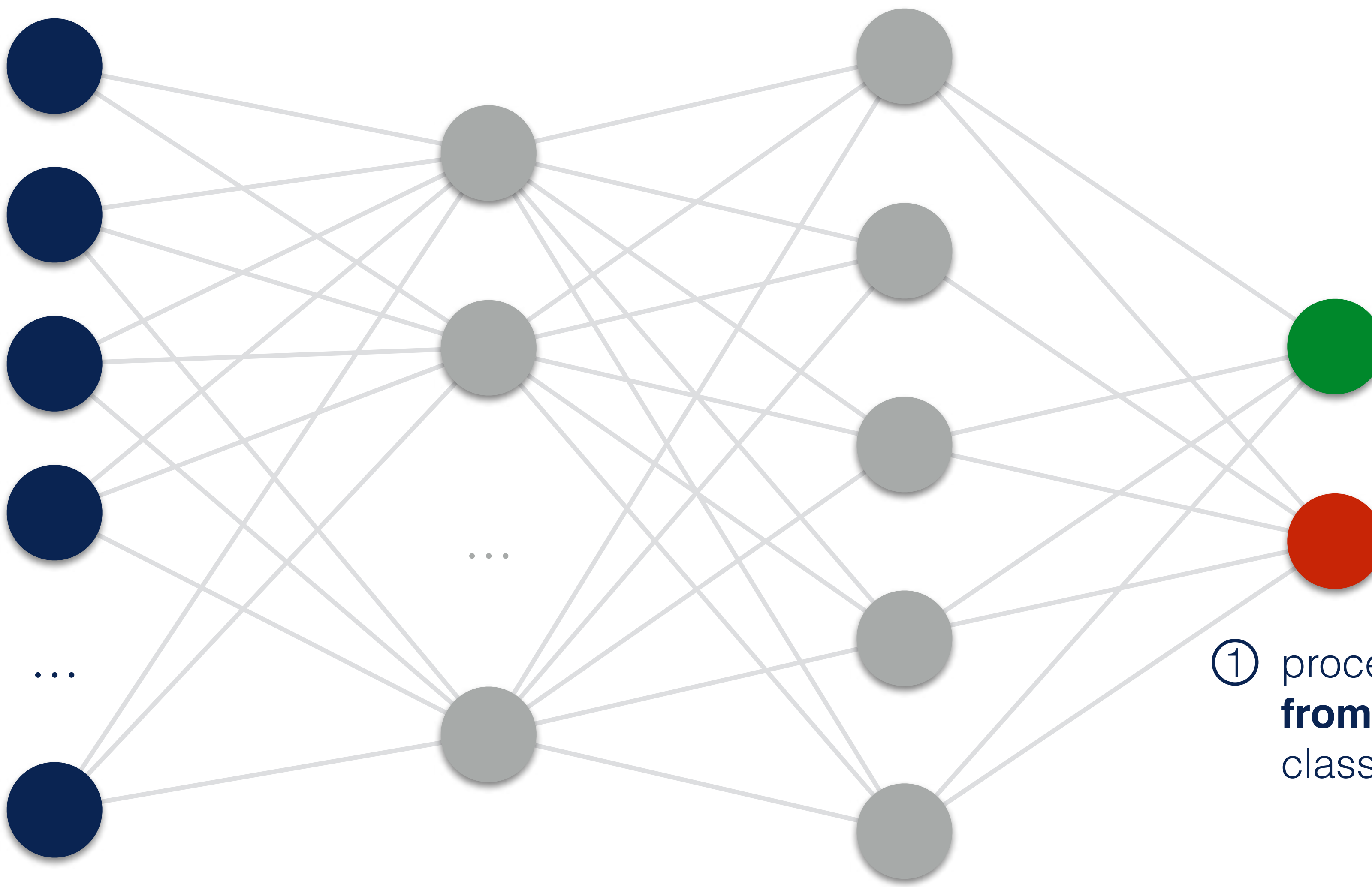
$$M \models \mathscr{F}_i \Leftrightarrow \forall A, B \in [\![M]\!]_\rightsquigarrow : (A_\omega \neq B_\omega \Rightarrow A_0|_{\neq i} \cap B_0|_{\neq i} = \emptyset)$$

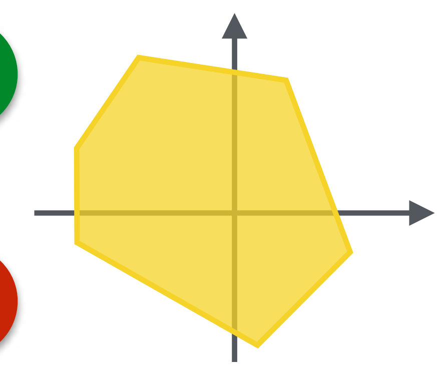
Naïve Abstraction

Naïve **Backward Analysis**

② **forget** the values of the **sensitive input** nodes



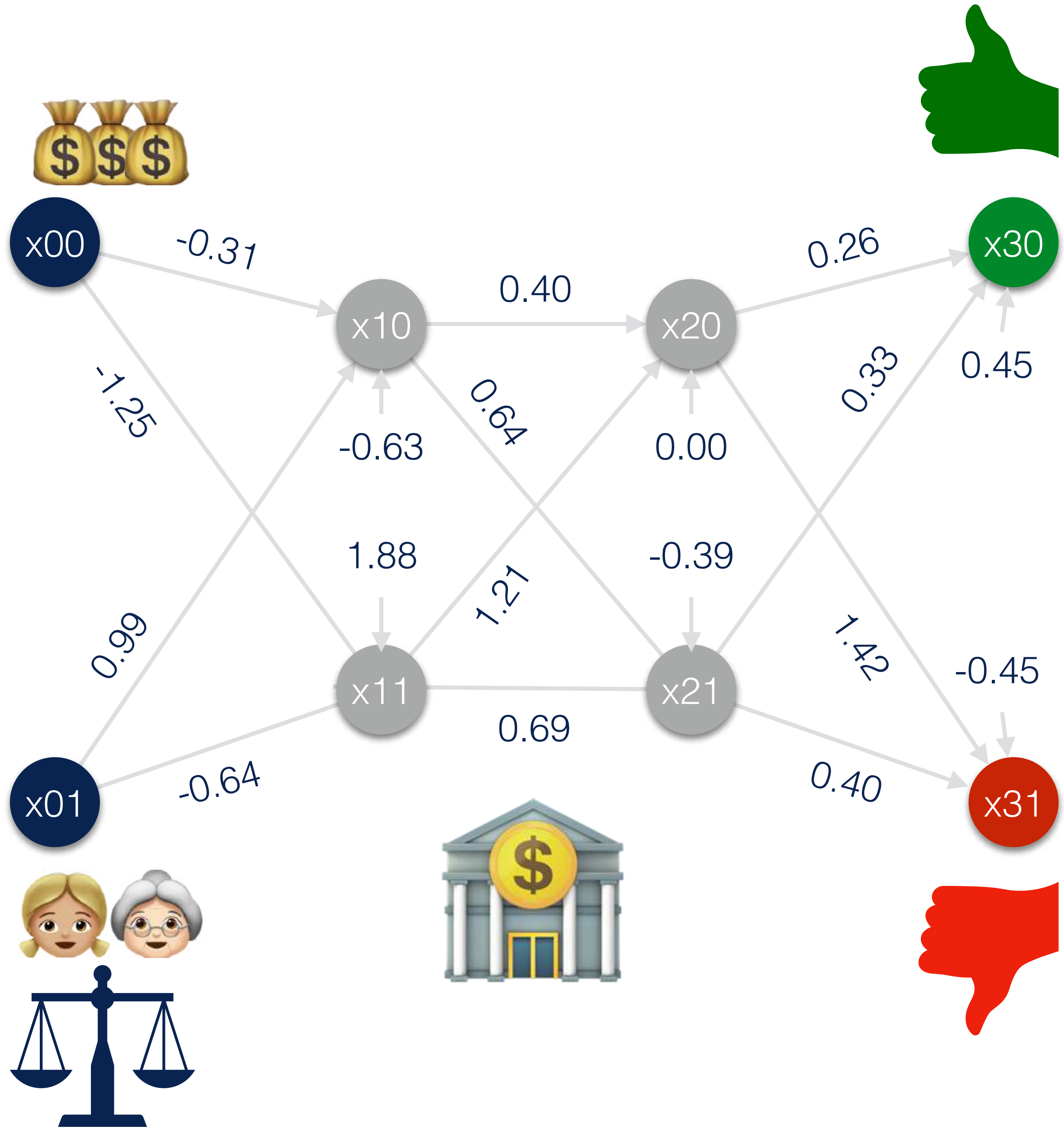




① proceed **backwards** **from all** possible classification **outcomes**

③ check for **intersection**:
empty → ✓ **fair**
otherwise → 🚨 **alarm**

Naïve Backward Analysis




```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21
1.16 * x20 + 0.07 * x21 ≤ 0.90    1.16 * x20 + 0.07 * x21 ≥ 0.90
x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)
x30 ≥ x31    x31 ≥ x30
return ‘ ’ if x31 < 30 else ‘ ’
```

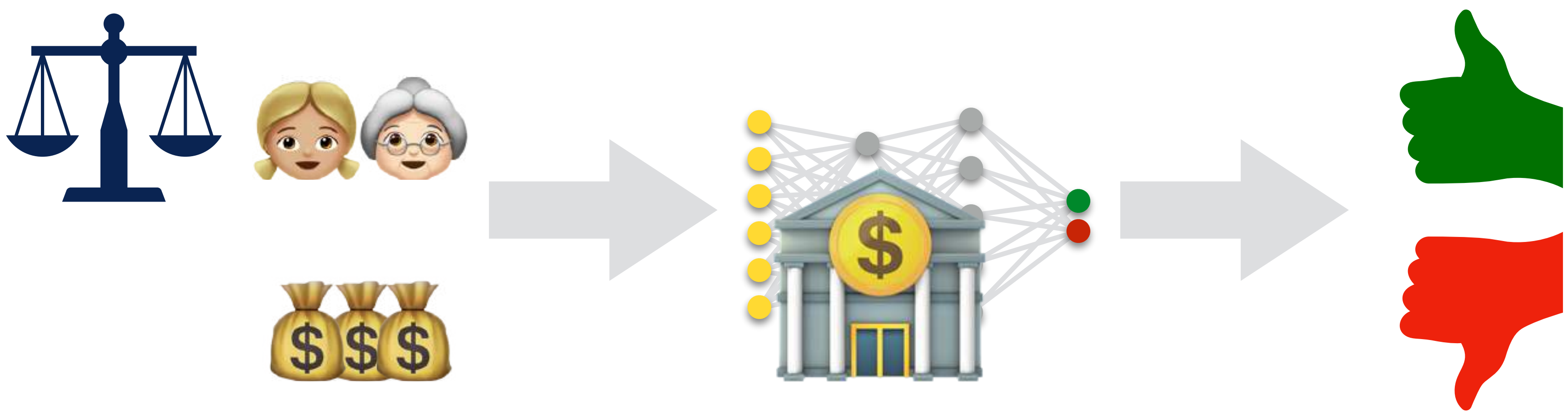
too many disjunctions!

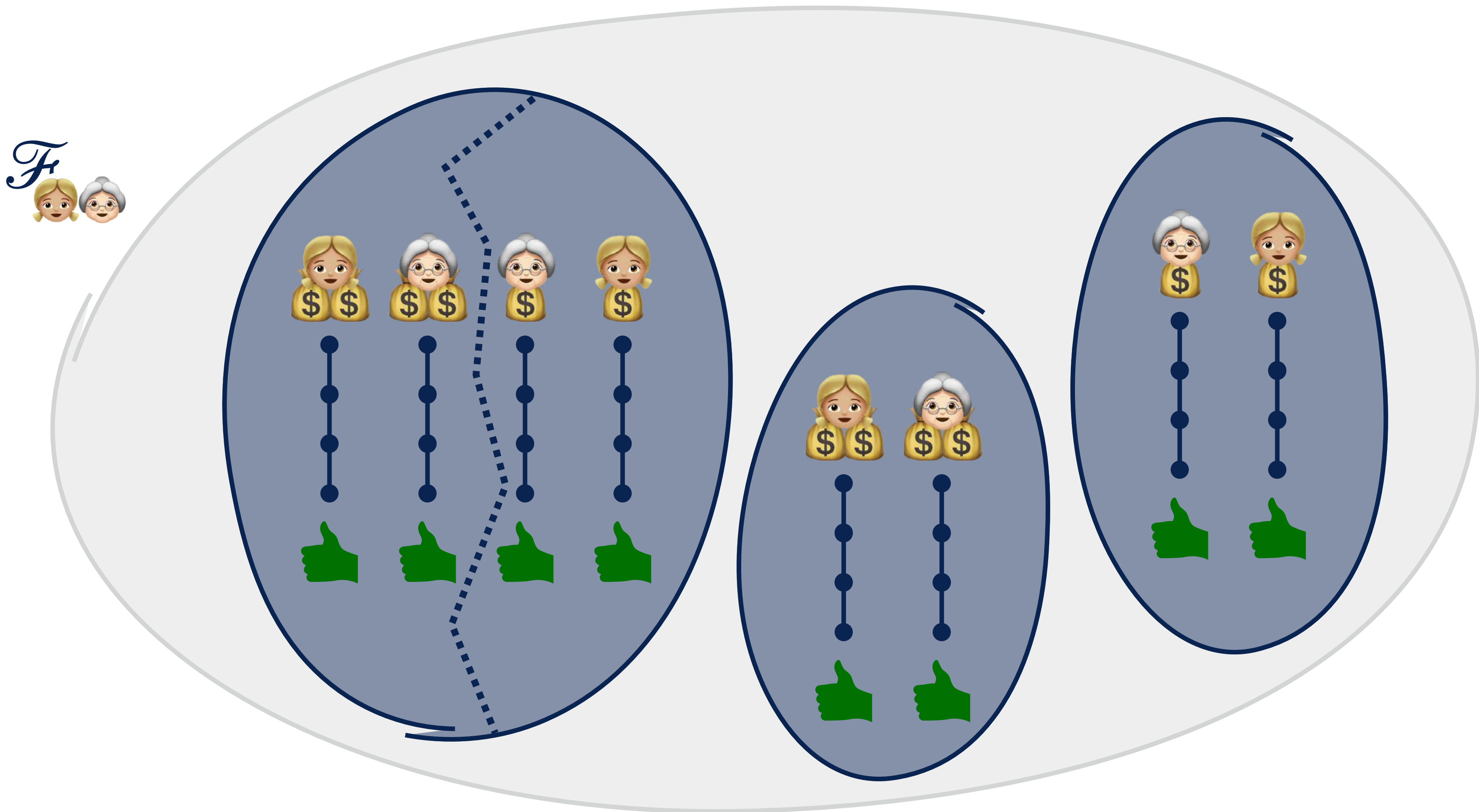
Back to the Semantics…

(Another) Hierarchy of Semantics

$$
\begin{tikzcd}
\text{parallel semantics} & \{[M]\}^{\parallel}_{\rightsquigarrow} & & \\
& & \llbracket M \rrbracket_{\rightsquigarrow} \arrow[ul, "\alpha_{\parallel}"'] & \text{dependency semantics} \\
& \{[M]\}^{\parallel}_{\bullet} \arrow[uu, "\alpha_{\rightsquigarrow}"] & & \\
& & \llbracket M \rrbracket_{\bullet} \arrow[uu, "\alpha_{\rightsquigarrow}"'] \arrow[ul, "\alpha_{\parallel}"'] & \text{outcome semantics} \\
& \{[M]\}^{\parallel} \arrow[uu, "\alpha_{\bullet}"] & & \\
& & \{\llbracket M \rrbracket\} \arrow[uu, "\alpha_{\bullet}"'] \arrow[ul, "\alpha_{\parallel}"'] & \text{collecting semantics}
\end{tikzcd}
$$

Parallel Semantics





💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the non-sensitive inputs** yields sets of traces that also satisfy dependency fairness

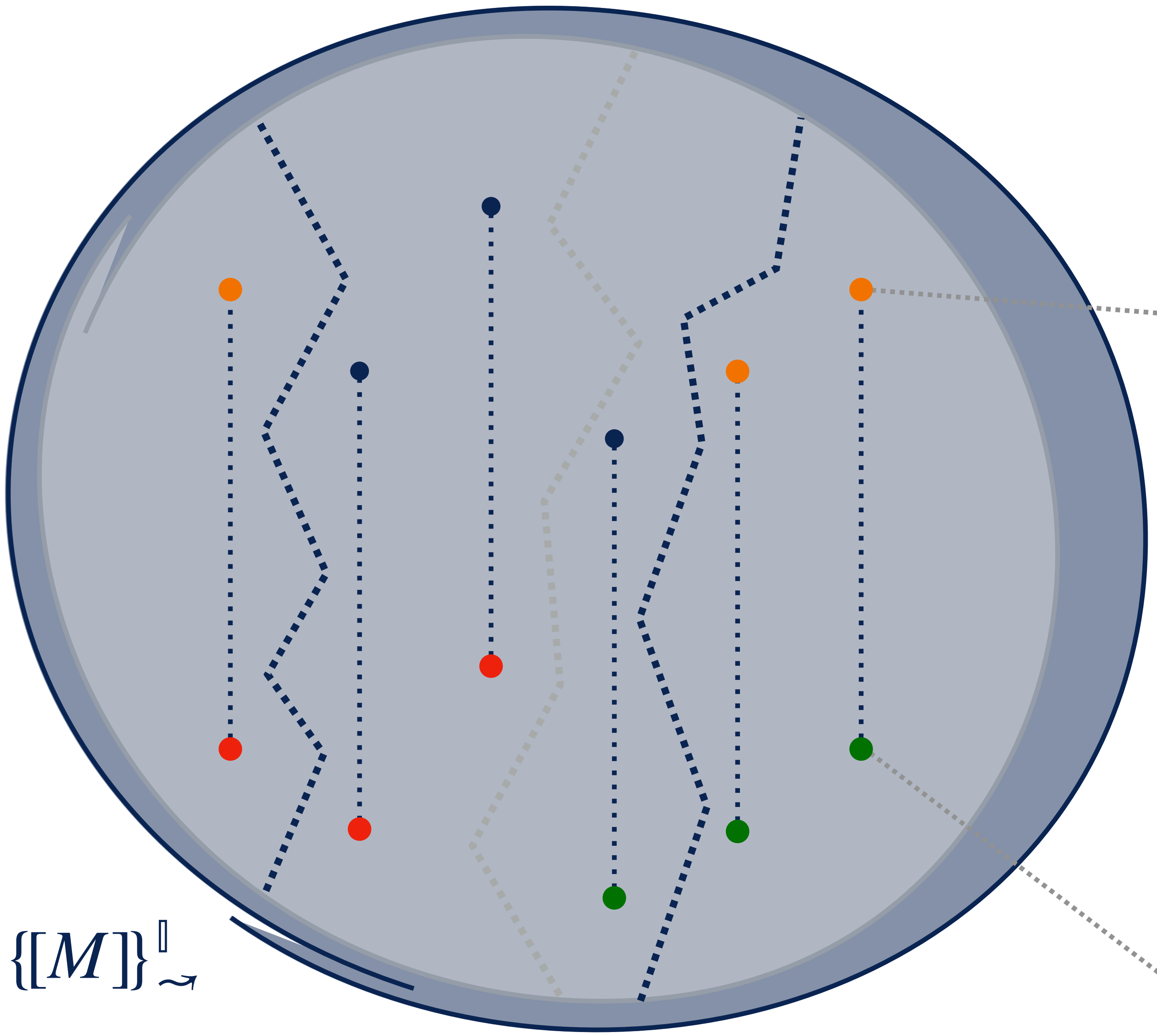
Parallel Semantics

$$\begin{tikzcd}[column sep=large] \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\rightsquigarrow} \rangle \arrow[r, bend right, "\alpha_{\parallel}"'] & \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\parallel} \rangle \arrow[l, bend right, "\gamma_{\parallel}"'] \end{tikzcd}$$

$$\alpha_{\parallel}(S) \stackrel{\text{def}}{=} \{\{\langle t_0, t_\omega \rangle \in R \mid t_0 \in I\} \mid R \in S \wedge I \in \parallel\}$$

parallel abstraction

Parallel Semantics



M

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

Parallel Semantics

$$\begin{tikzcd}
\langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\rightsquigarrow} \rangle \arrow[r, bend right, "\alpha_{\parallel}"'] & \langle \mathscr{P}(\mathscr{P}(\Sigma \times \Sigma)), \subseteq_{\parallel} \rangle \arrow[l, bend right, "\gamma_{\parallel}"']
\end{tikzcd}$$

$$\alpha_{\parallel}(S) \stackrel{\text{def}}{=} \{\{\langle t_0, t_\omega \rangle \in R \mid t_0 \in I\} \mid R \in S \wedge I \in \mathbb{I}\}$$ **parallel abstraction**

$$\{[M]\}^{\parallel}_{\rightsquigarrow} \stackrel{\text{def}}{=} \alpha_{\parallel}([\![M]\!]_{\rightsquigarrow})$$
$$= \{\{\langle t_0, t_\omega \rangle \in \Sigma \times \Sigma \mid t \in [\![M]\!] \wedge t_0 \in I \wedge t_\omega \in O\} \mid I \in \mathbb{I} \wedge O \in \mathbb{O}\}$$

Theorem

$$M \models \mathscr{F}_i \Leftrightarrow \{[M]\}^{\parallel}_{\rightsquigarrow} \subseteq_{\parallel} \alpha_{\parallel}(\alpha_{\rightsquigarrow}(\alpha_{\bullet}(\mathscr{F}_i))) \Leftrightarrow [\![M]\!]_{\rightsquigarrow} \subseteq_{\rightsquigarrow} \alpha_{\parallel}(\alpha_{\rightsquigarrow}(\mathscr{F}_i))$$

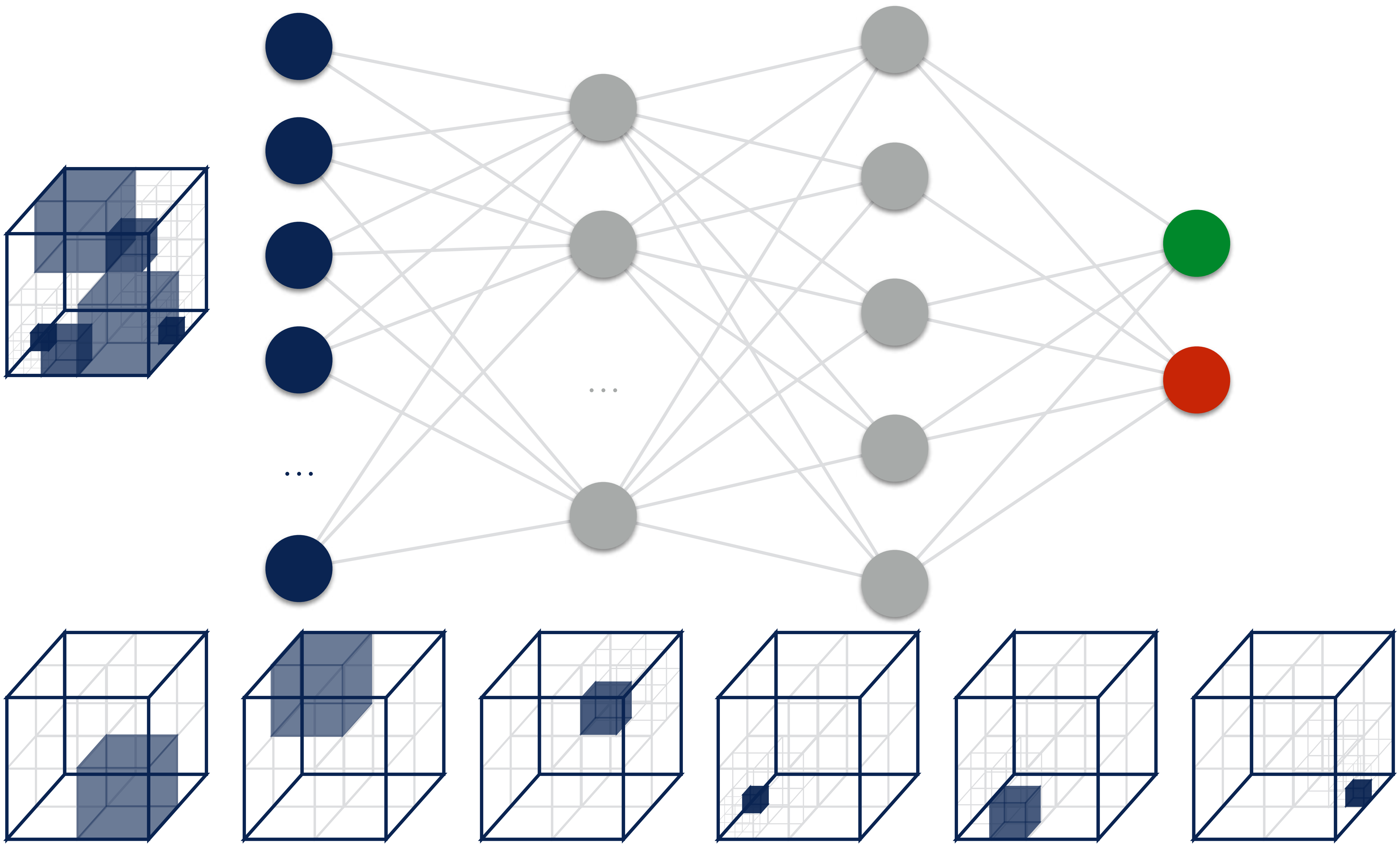
Lemma

$$M \models \mathscr{F}_i \Leftrightarrow \forall I \in \mathbb{I} \colon \forall A, B \in \{[M]\}^{\parallel}_{\rightsquigarrow} \colon (A^I_\omega \neq B^I_\omega \Rightarrow A^I_0|_{\neq i} \cap B^I_0|_{\neq i} = \emptyset)$$

Better Abstraction

Forward and Backward Analysis

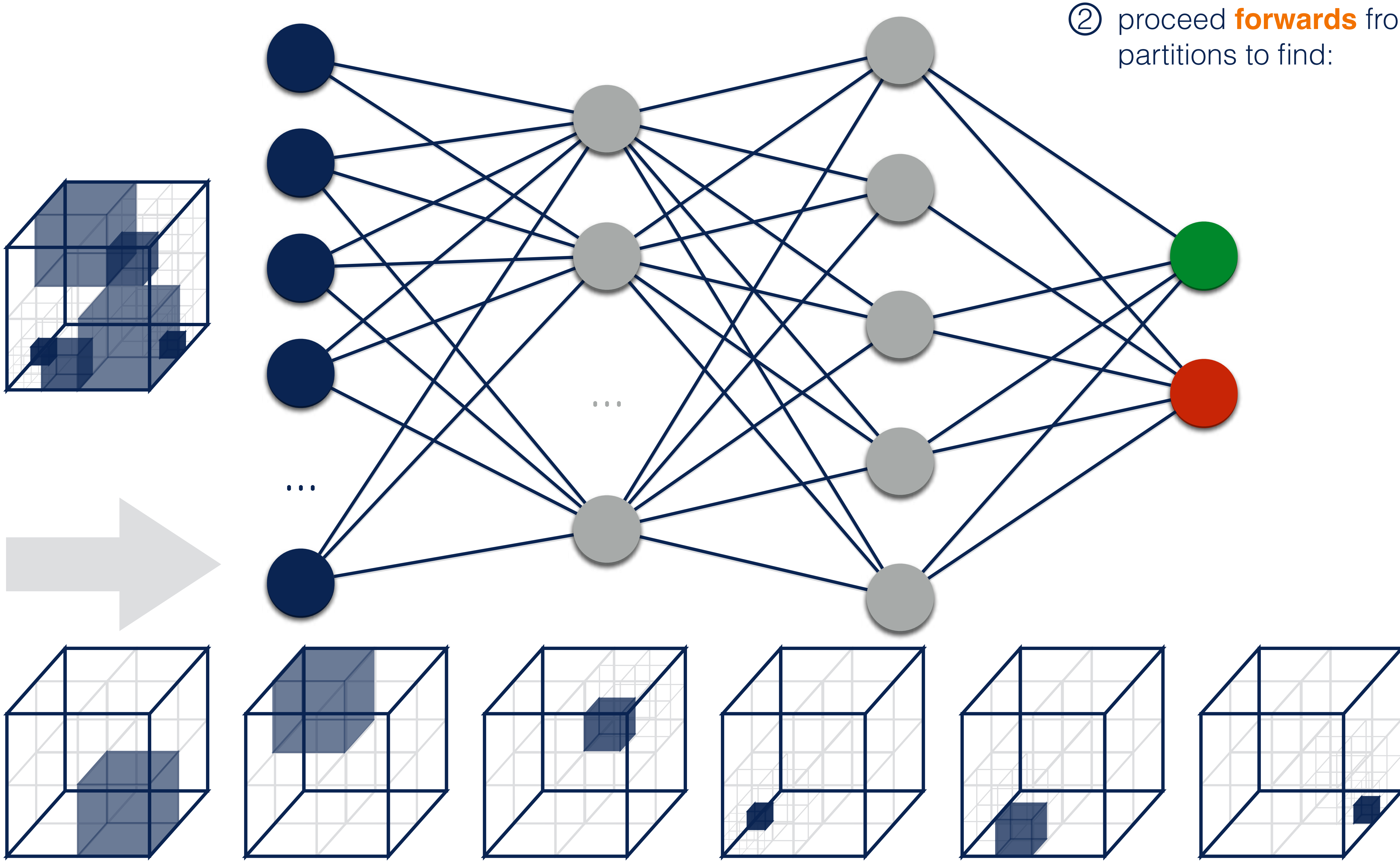
① **partition** the space of values of the **non-sensitive input** nodes



Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

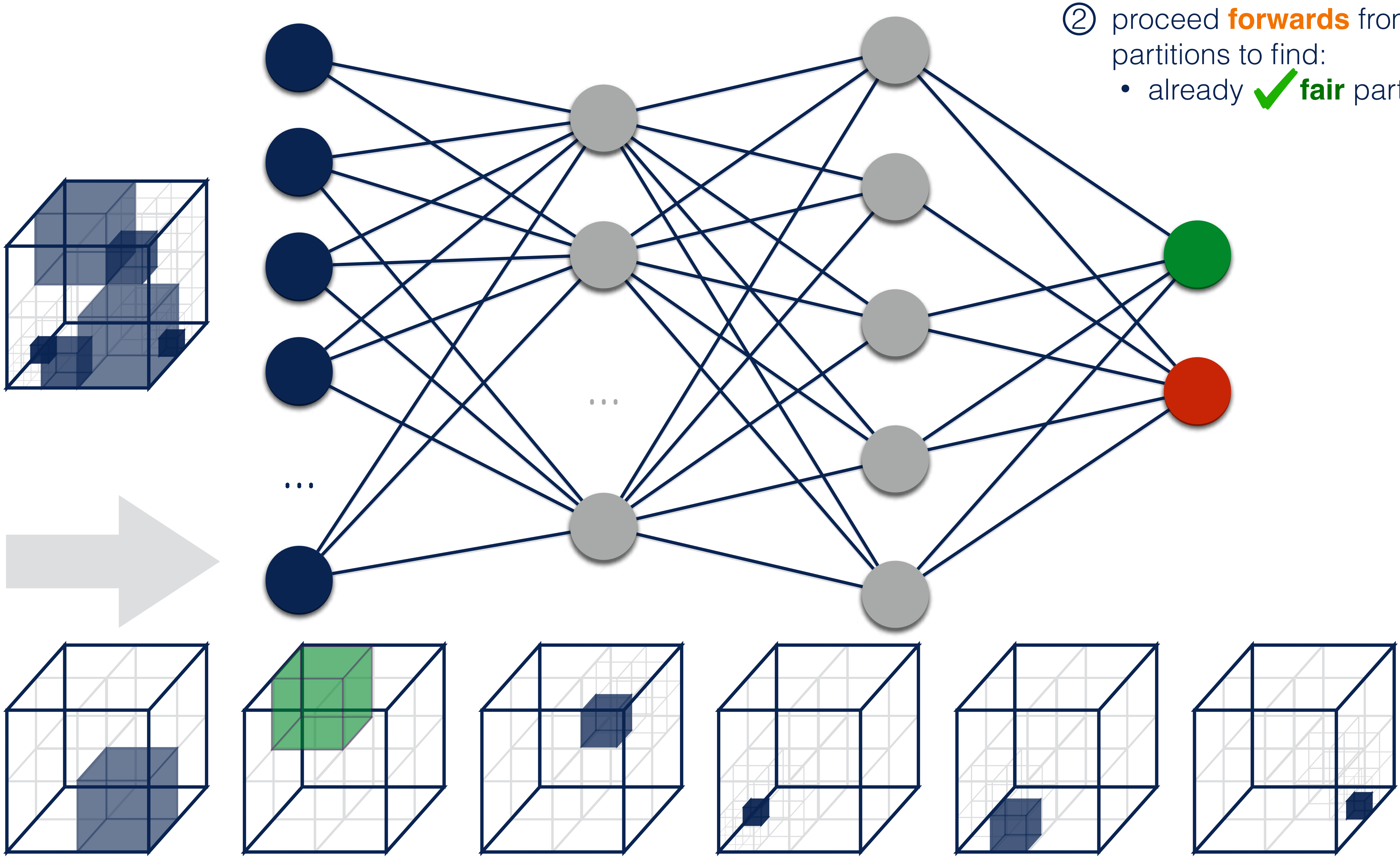
② proceed **forwards** from all partitions to find:



Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

② proceed **forwards** from all partitions to find:
- already ✓ **fair** partitions

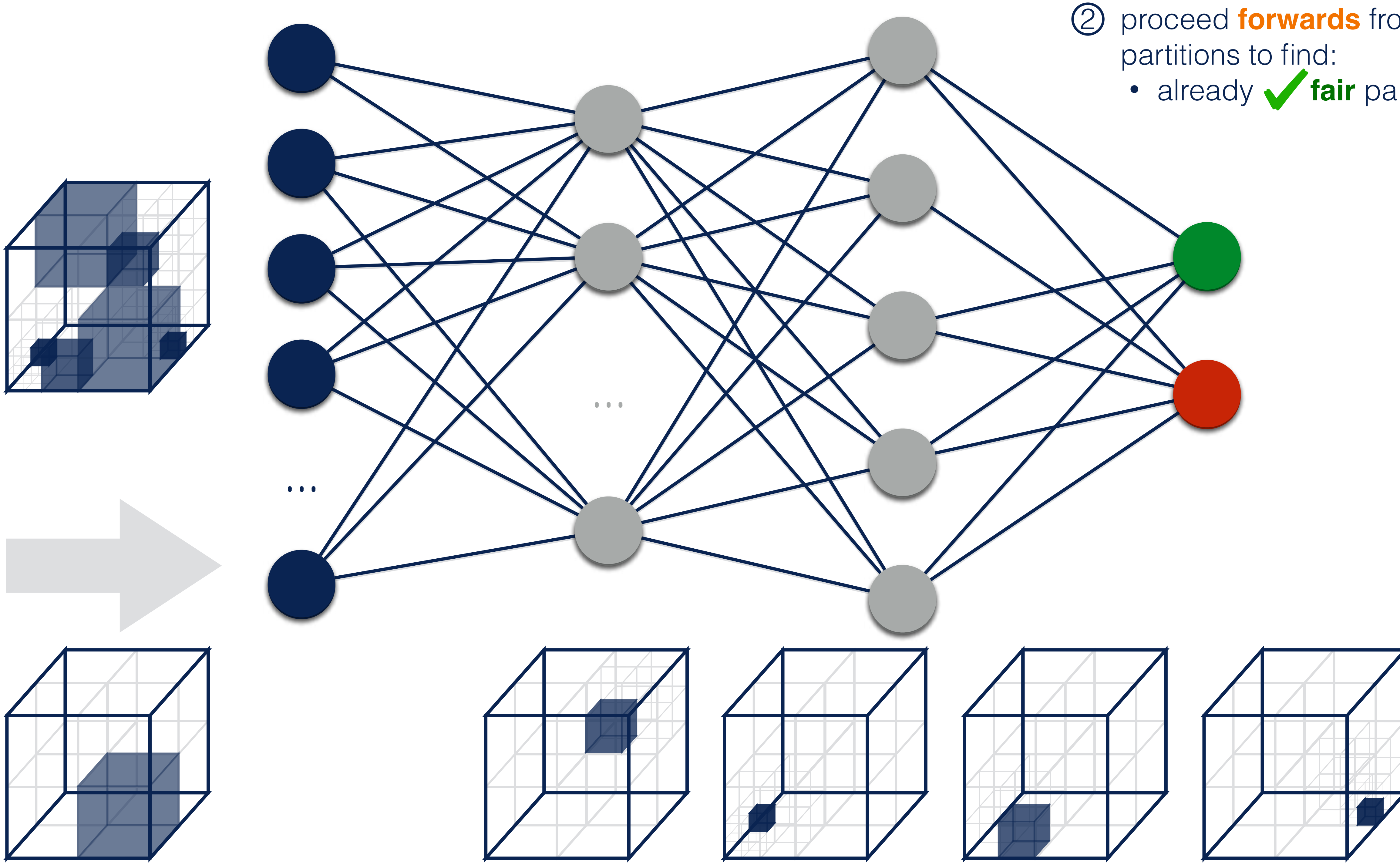


Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

② proceed **forwards** from all partitions to find:

- already ✓ **fair** partitions

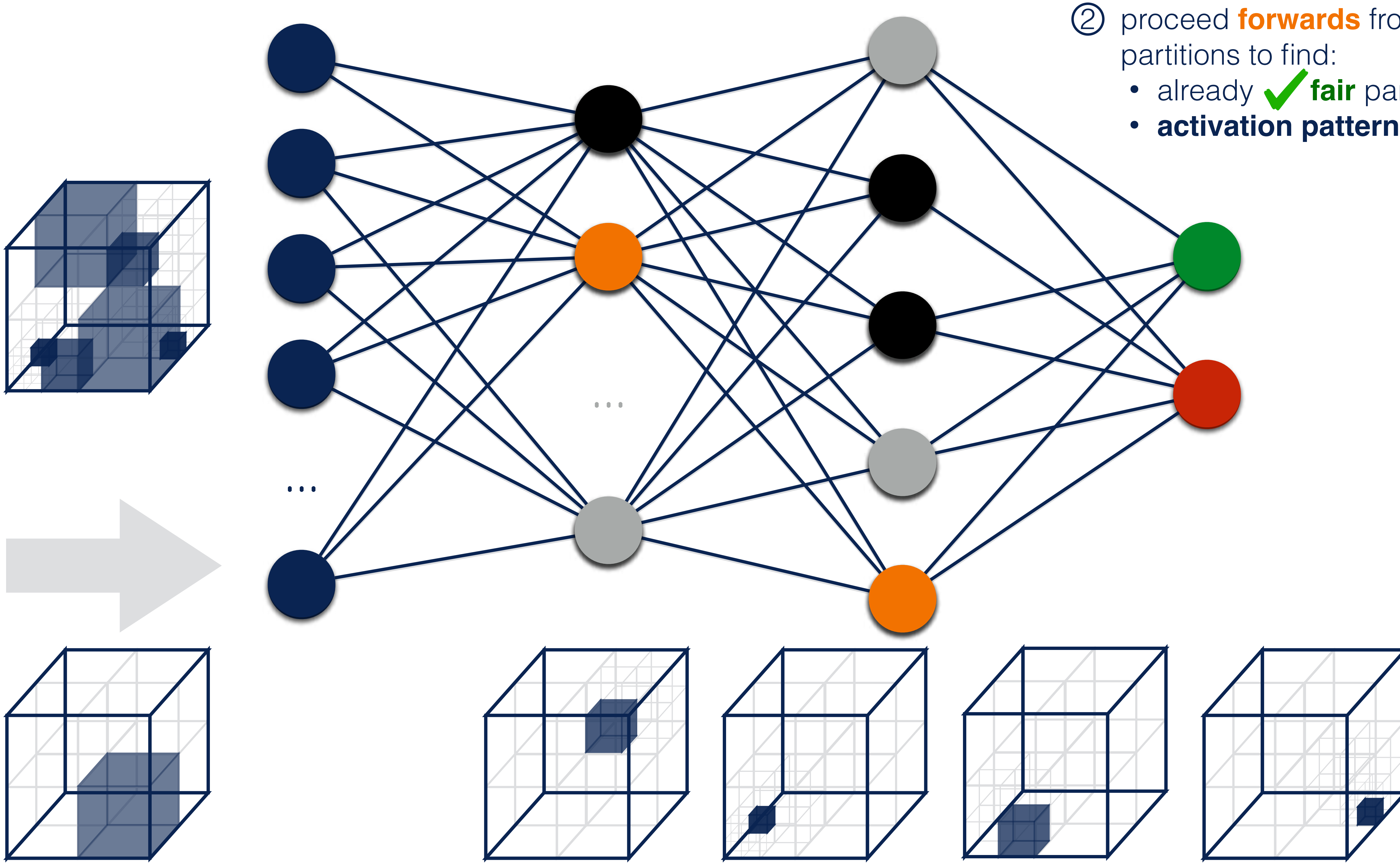


Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

② proceed **forwards** from all partitions to find:

- already ✓ **fair** partitions
- **activation patterns**

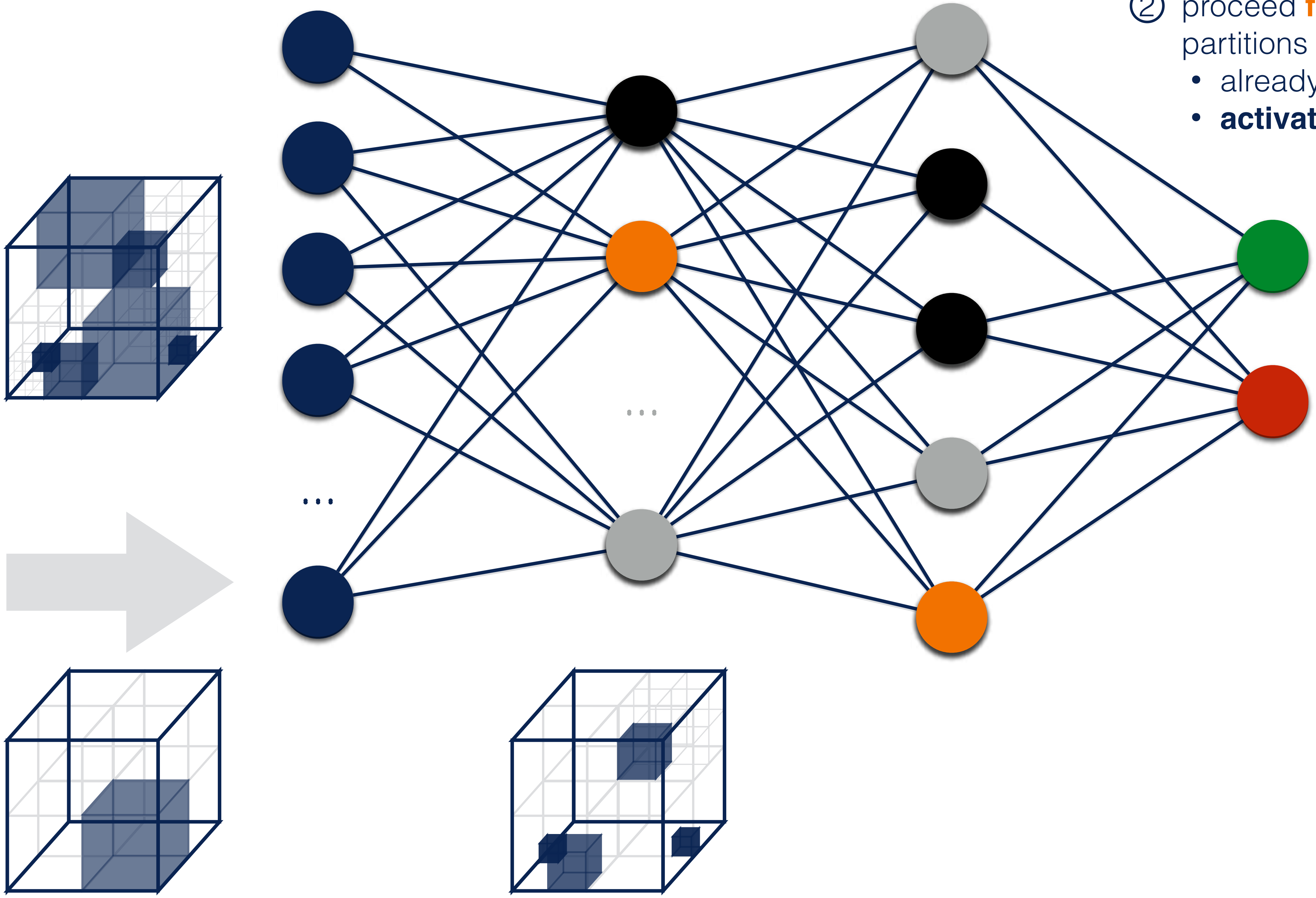


Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

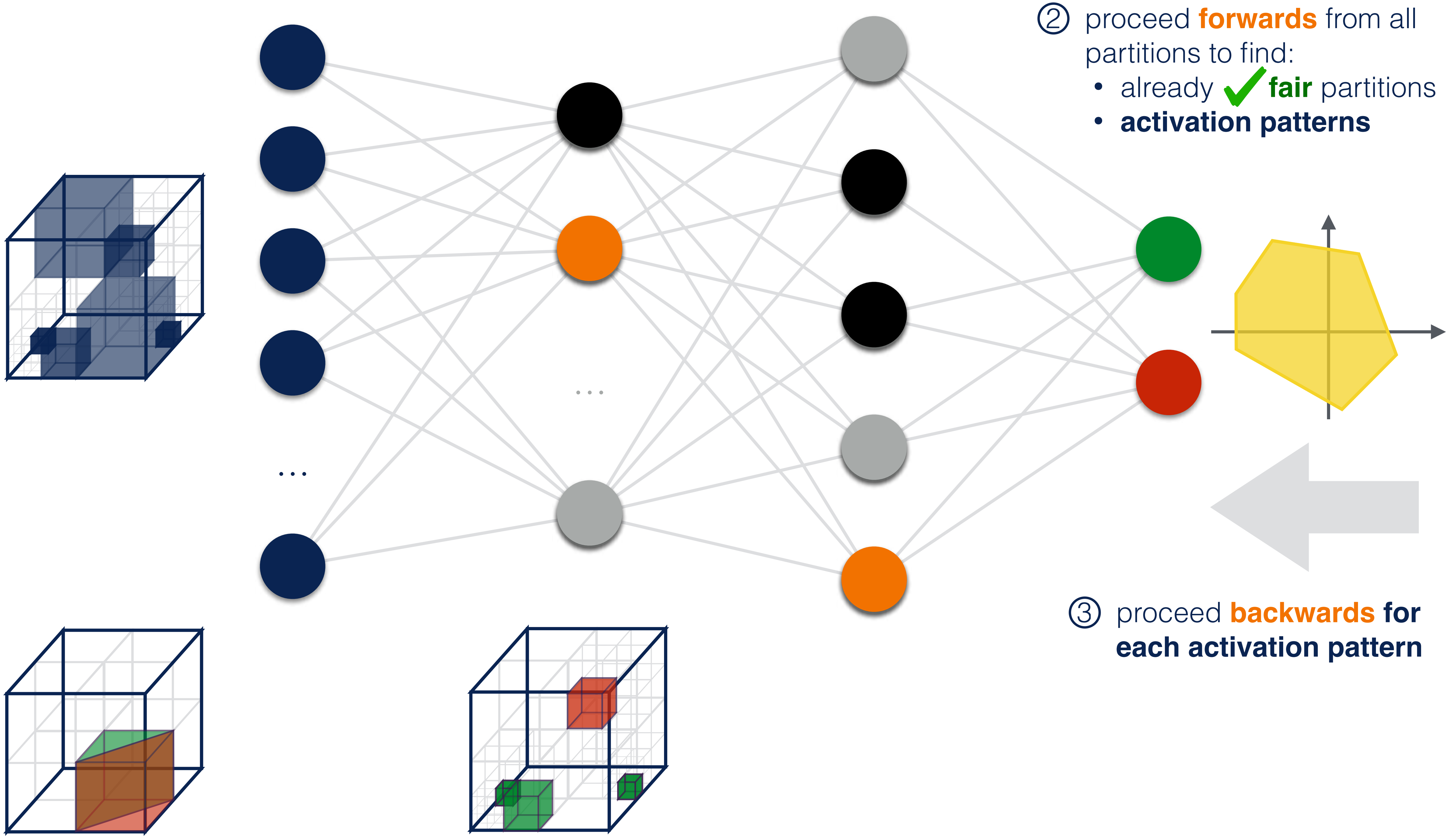
② proceed **forwards** from all partitions to find:

- already ✓ **fair** partitions
- **activation patterns**



Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

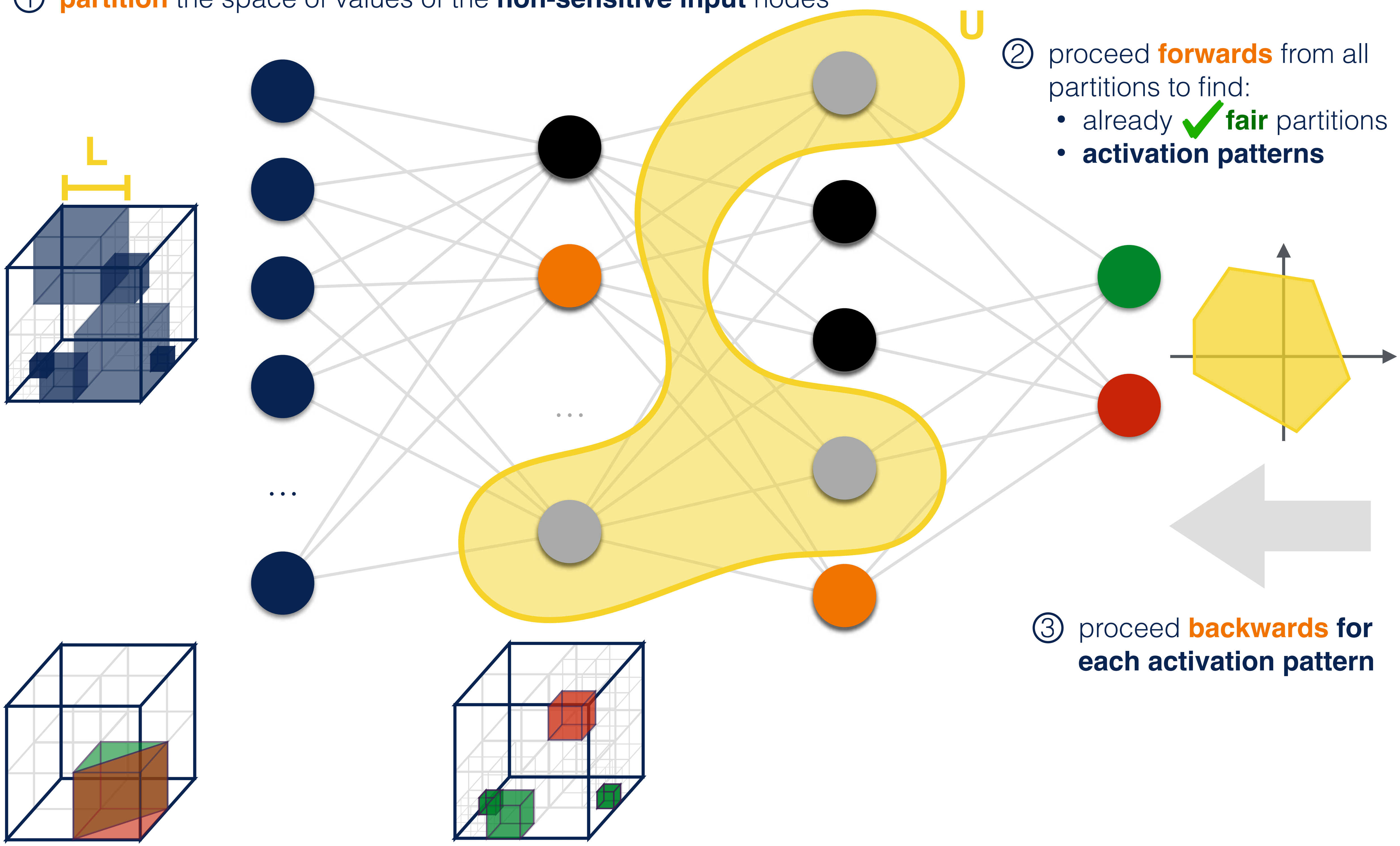


② proceed **forwards** from all partitions to find:
- already ✓ **fair** partitions
- **activation patterns**

③ proceed **backwards for each activation pattern**

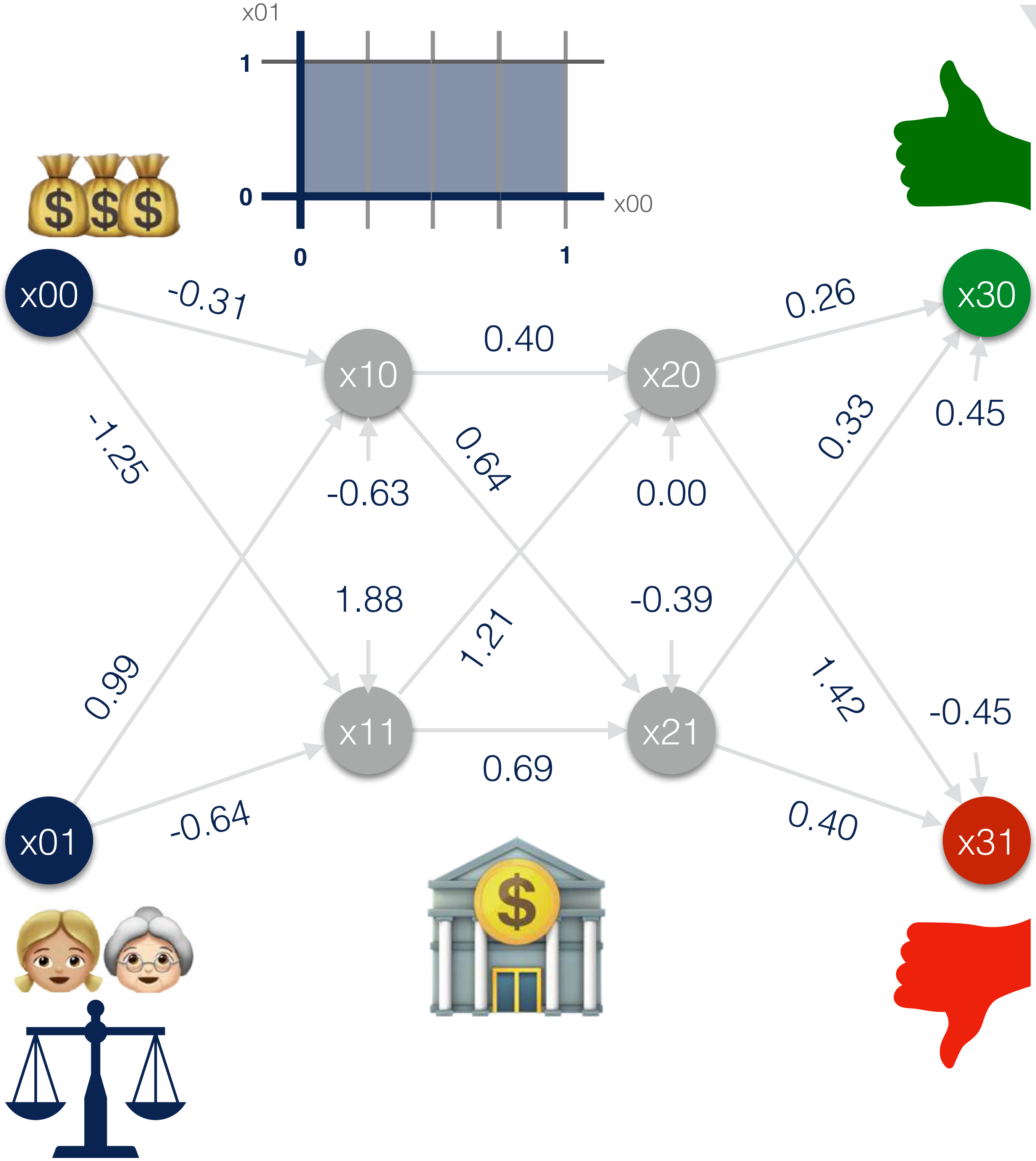
Forward and Backward Analysis

① **partition** the space of values of the **non-sensitive input** nodes



② proceed **forwards** from all partitions to find:
- already ✓ **fair** partitions
- **activation patterns**

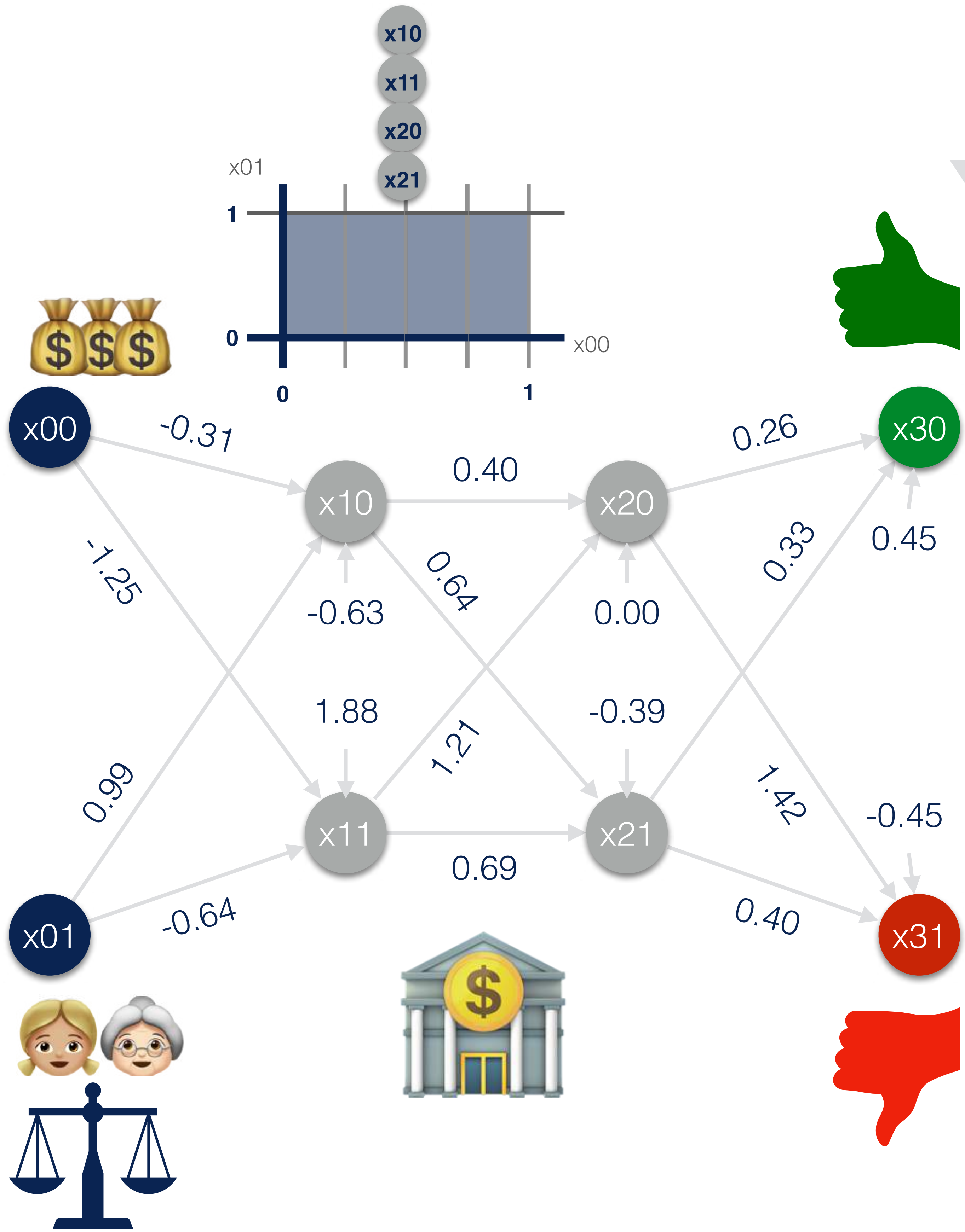
③ proceed **backwards for each activation pattern**



L = 0.25

U = 2

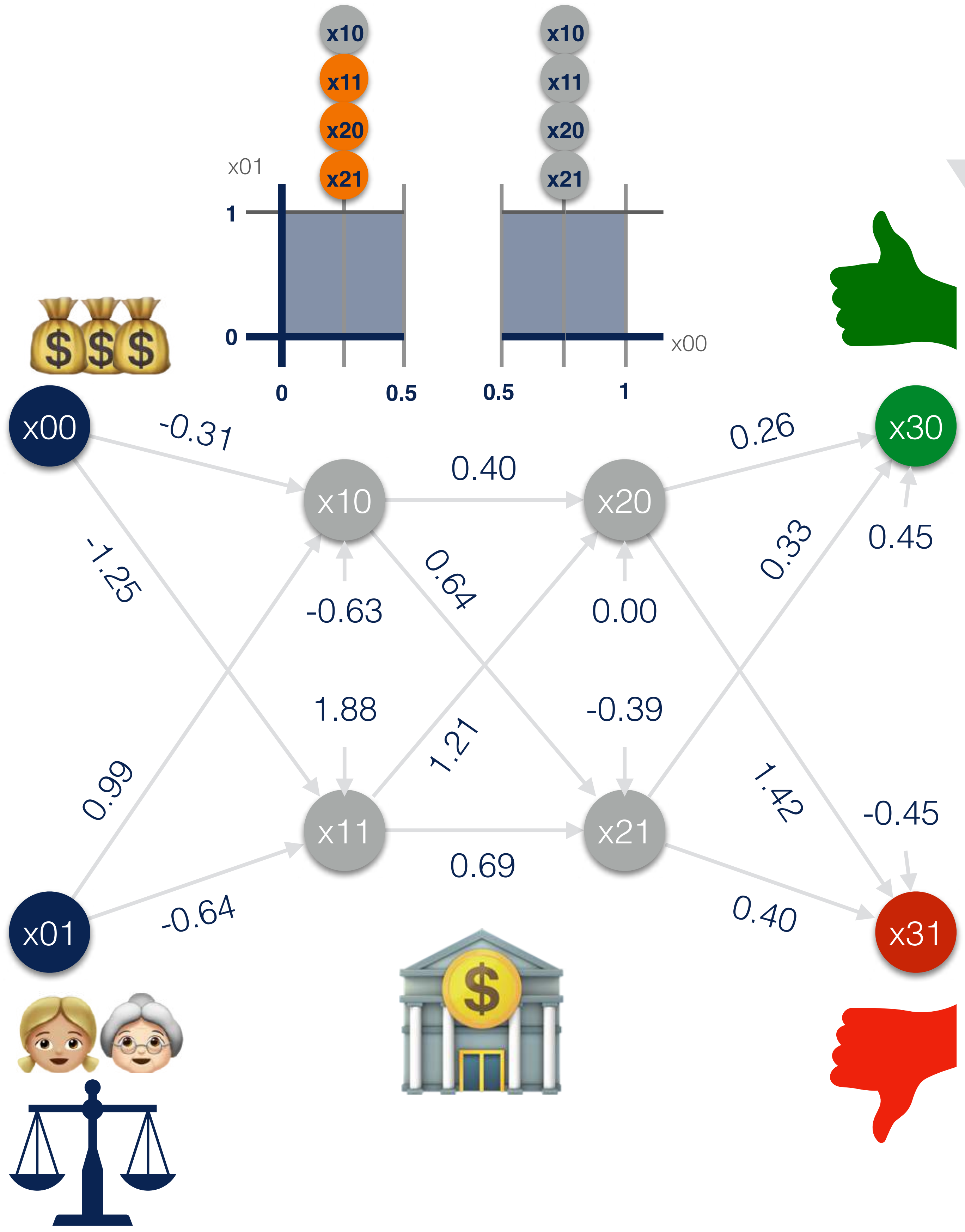
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```



L = 0.25

U = 2

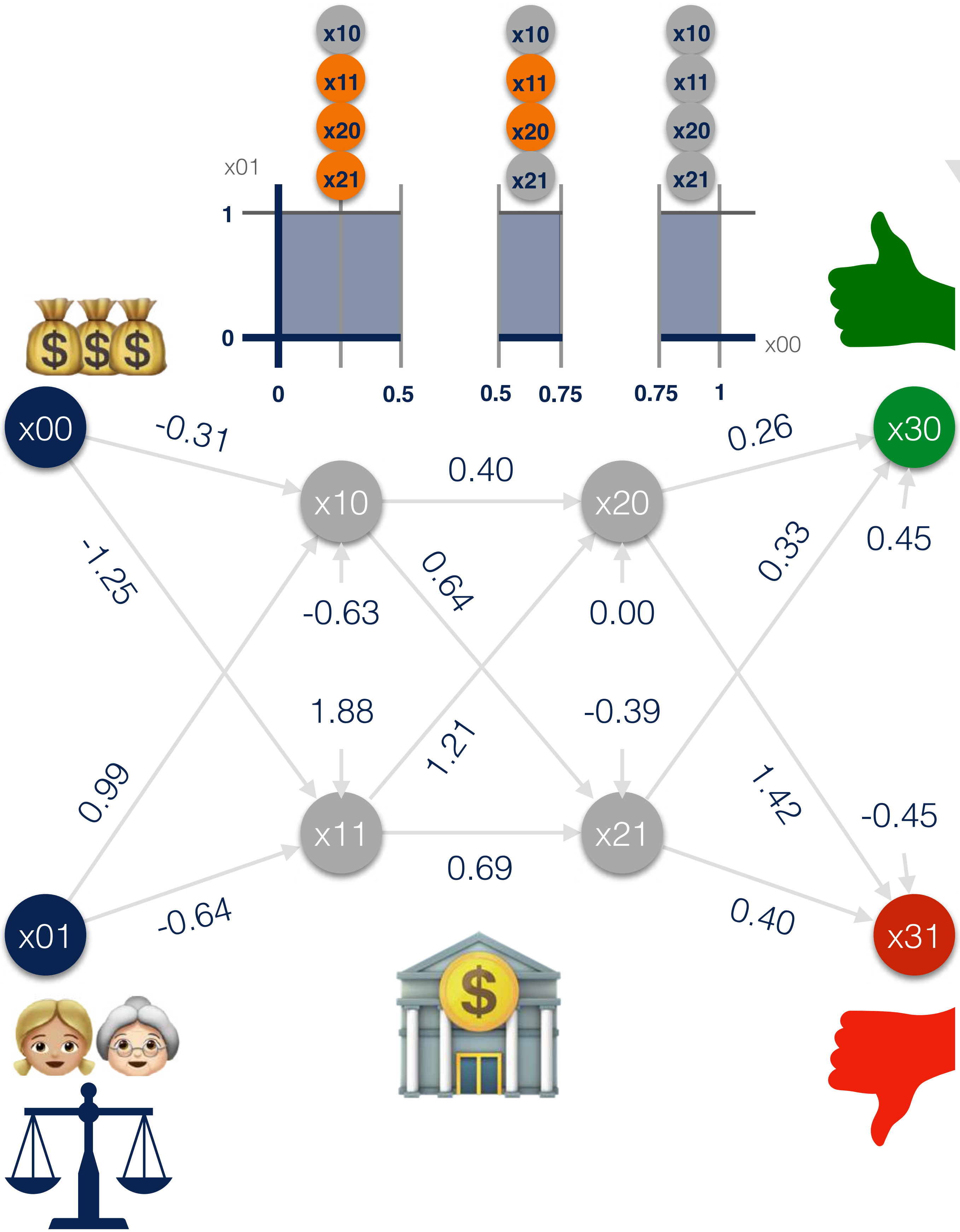
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```



L = 0.25

U = 2

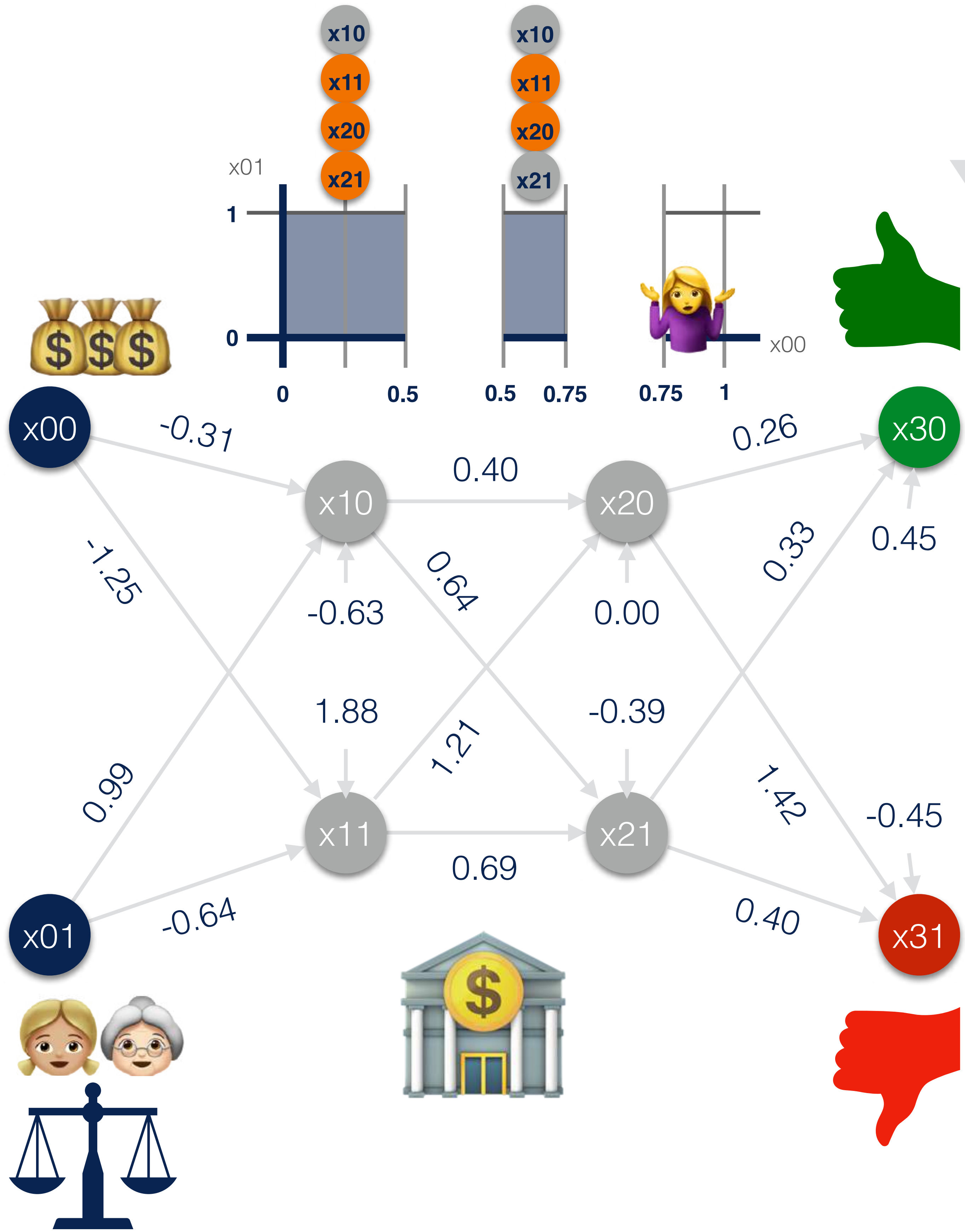
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```



L = 0.25

U = 2

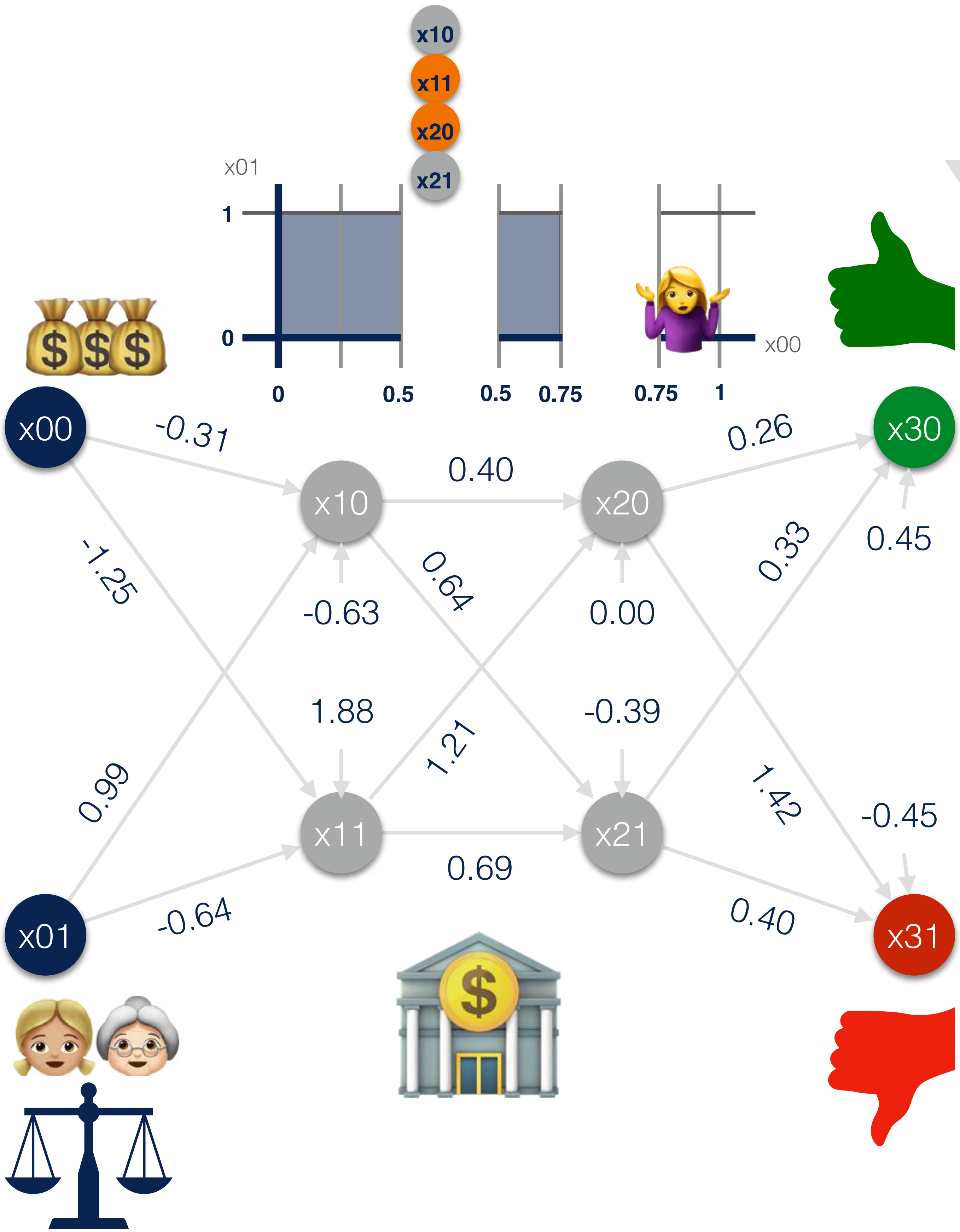
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```



L = 0.25

U = 2

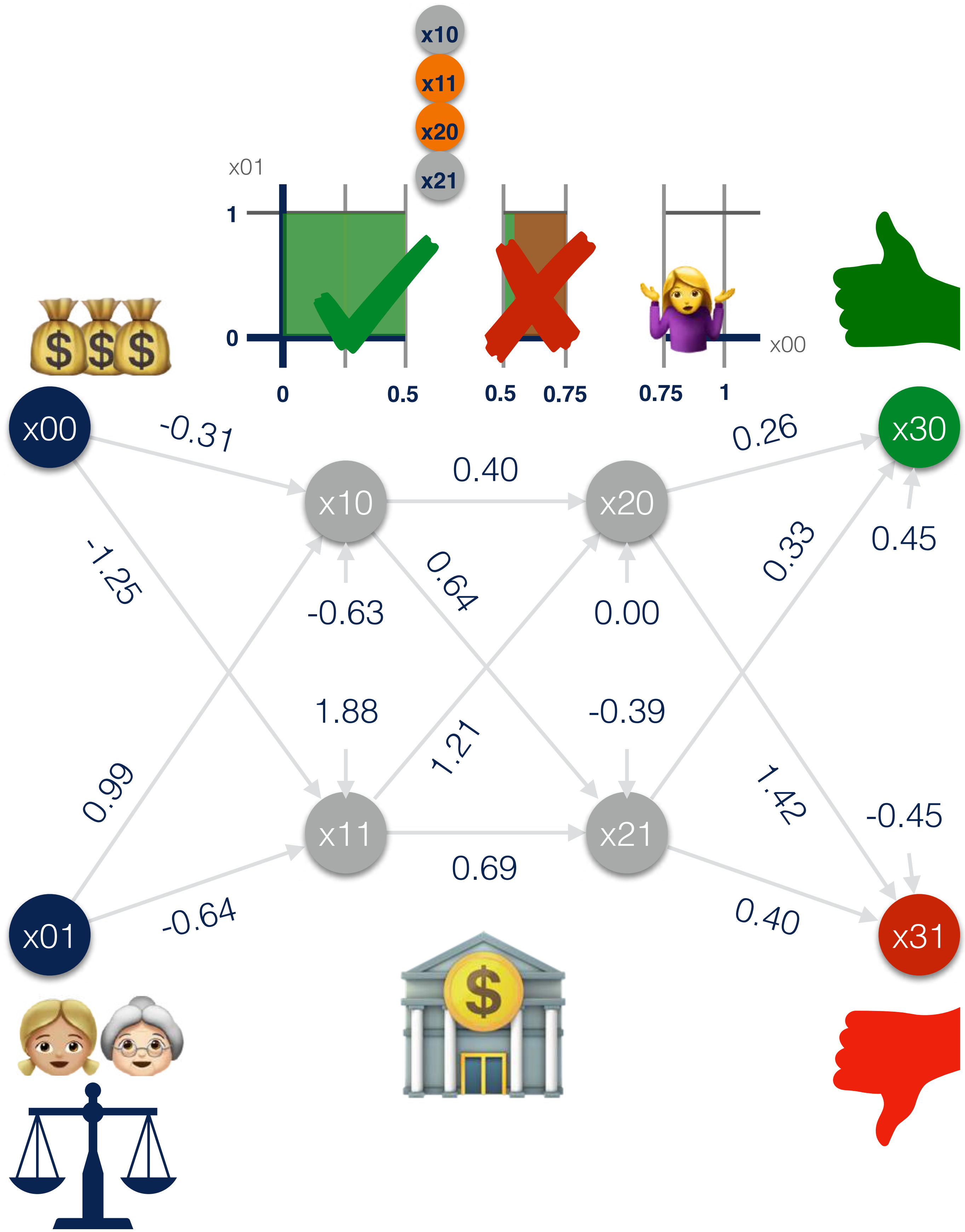
```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```



L = 0.25

U = 2

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```

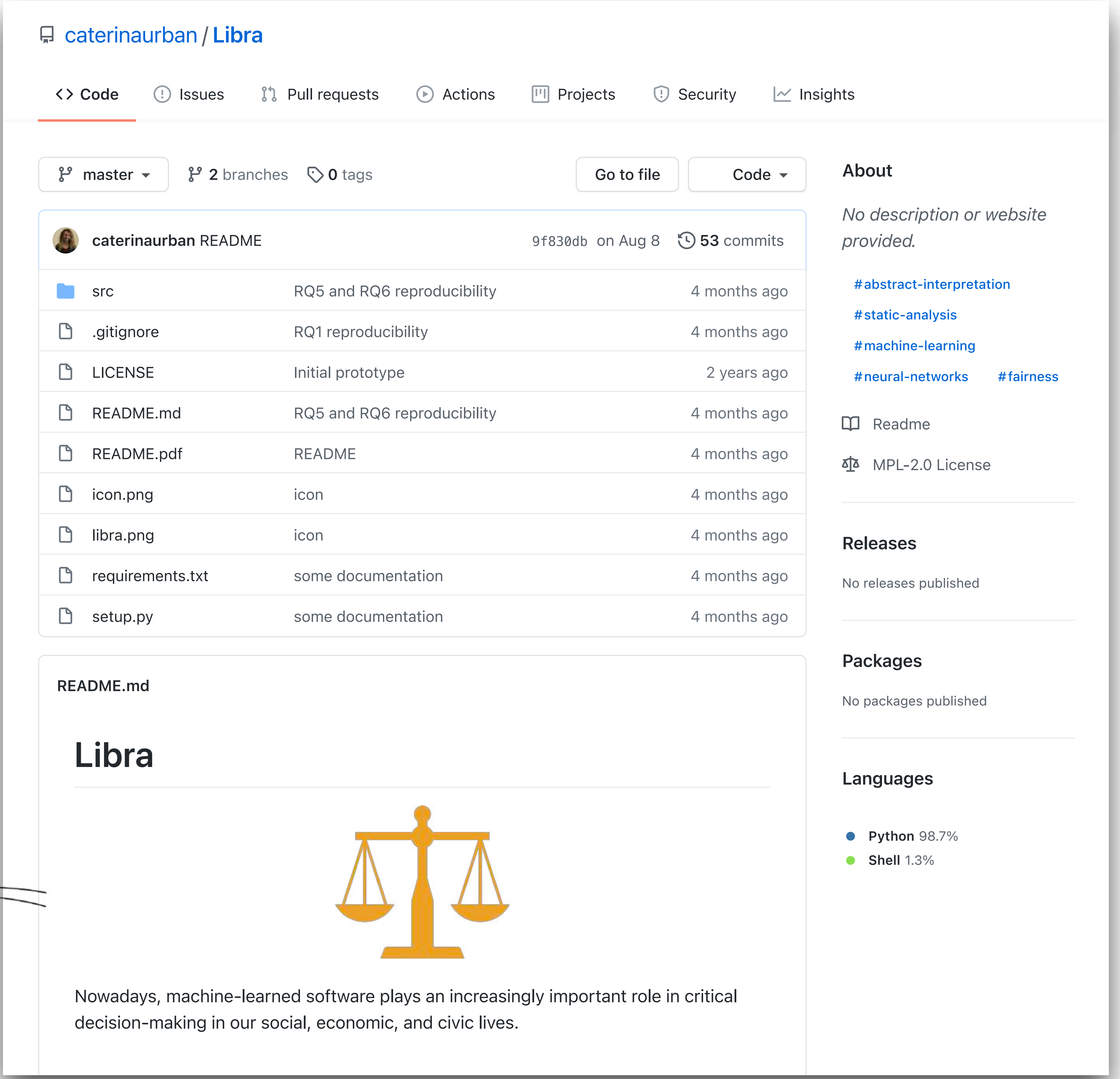


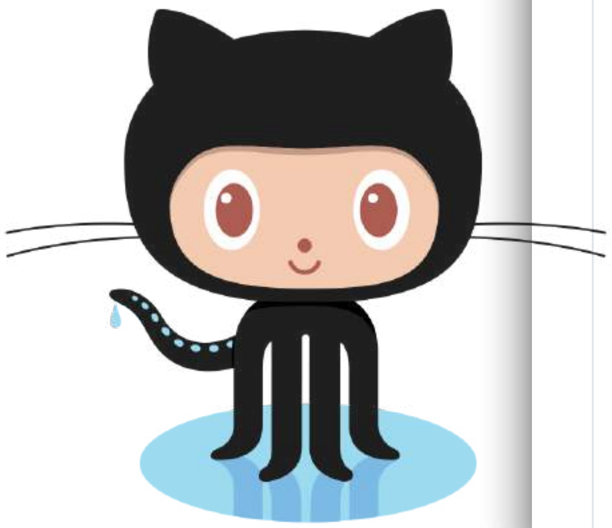
L = 0.25

U = 2

```
x00 = input()
x01 = input()
x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88
x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11
x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)
x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21
1.16 * x20 + 0.07 * x21 ≤ 0.90    1.16 * x20 + 0.07 * x21 ≥ 0.90
x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)
x30 ≥ x31    x31 ≥ x30
return ’ if x31 < 30 else ‘ ’
```

Libra


caterinaurban / Libra
Code
Issues
Pull requests
Actions
Projects
Security
Insights
master
2 branches
0 tags
Go to file
Code
caterinaurban README
9f830db on Aug 8
53 commits
src RQ5 and RQ6 reproducibility 4 months ago
.gitignore RQ1 reproducibility 4 months ago
LICENSE Initial prototype 2 years ago
README.md RQ5 and RQ6 reproducibility 4 months ago
README.pdf README 4 months ago
icon.png icon 4 months ago
libra.png icon 4 months ago
requirements.txt some documentation 4 months ago
setup.py some documentation 4 months ago
README.md
Libra
Nowadays, machine-learned software plays an increasingly important role in critical decision-making in our social, economic, and civic lives.
About
No description or website provided.
#abstract-interpretation
#static-analysis
#machine-learning
#neural-networks
#fairness
Readme
MPL-2.0 License
Releases
No releases published
Packages
No packages published
Languages
Python 98.7%
Shell 1.3%



Internship Opportunities

Inference of **Implicit Assumptions** on Input Data

Fairness of **Decision Tree Ensembles**

Static Analysis of **Neural Network Training**

Static Analysis of **Medical Data-Processing Software**

Bibliography

[Kurd03] **Zeshan Kurd, Tim Kelly**. Establishing Safety Criteria for Artificial Neural Networks. In KES, pages 63-169, 2003.

[Wang18] **Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, Suman Jana**. Formal Security Analysis of Neural Networks using Symbolic Intervals. In USENIX Security, pages 1599–1614, 2018.

[Li19] **Jianlin Li, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang**. Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification. In SAS, page 296–319, 2019.

[Singh19] **Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev**. An Abstract Domain for Certifying Neural Networks. In POPL, pages 41:1 - 41:30, 2019.

Bibliography

[Julian16] **Kyle D. Julian, Jessica Lopez, Jeffrey S. Brush, Michael P. Owen, Mykel J. Kochenderfer**. Policy Compression for Aircraft Collision Avoidance Systems. In DASC, pages 1–10, 2016.

[Katz17] **Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, Mykel J. Kochenderfer**. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In CAV, pages 97–117, 2017.

[Galhotra17] **Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou**. Fairness Testing: Testing Software for Discrimination. In FSE, pages 498–510, 2017.

[Urban20] **Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang**. Perfectly Parallel Fairness Certification of Neural Networks. In OOPSLA, pages 185:1–185:30, 2020.