Characterizing NC^k

from words to trees and back to words

Guillaume Bonfante, Reinhard Kahle, Jean-Yves Marion
and Isabel Oitavem

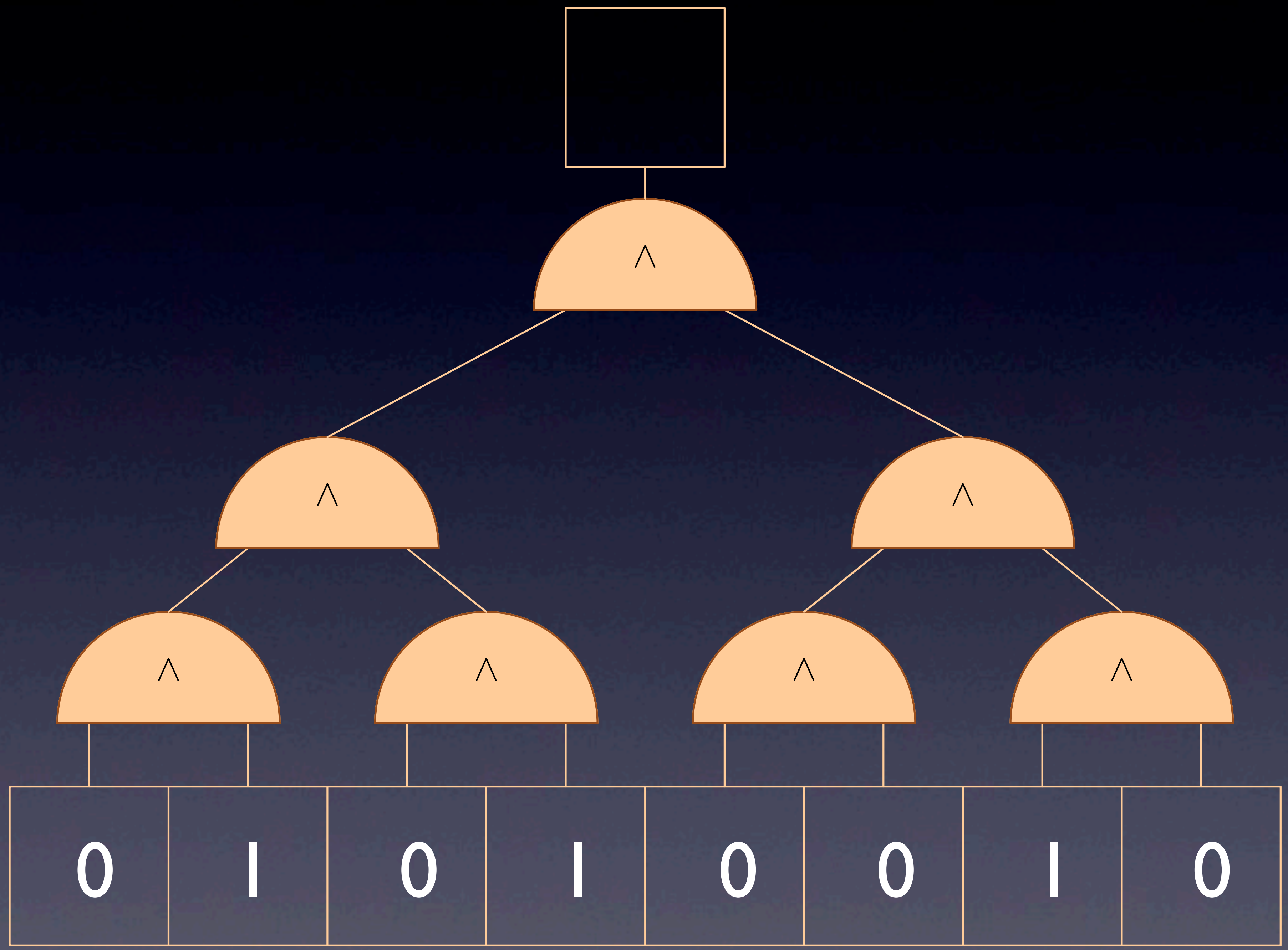
Implicit Computational Complexity and applications:
Resource control, security, real-number computation

Shonan Village - 2013

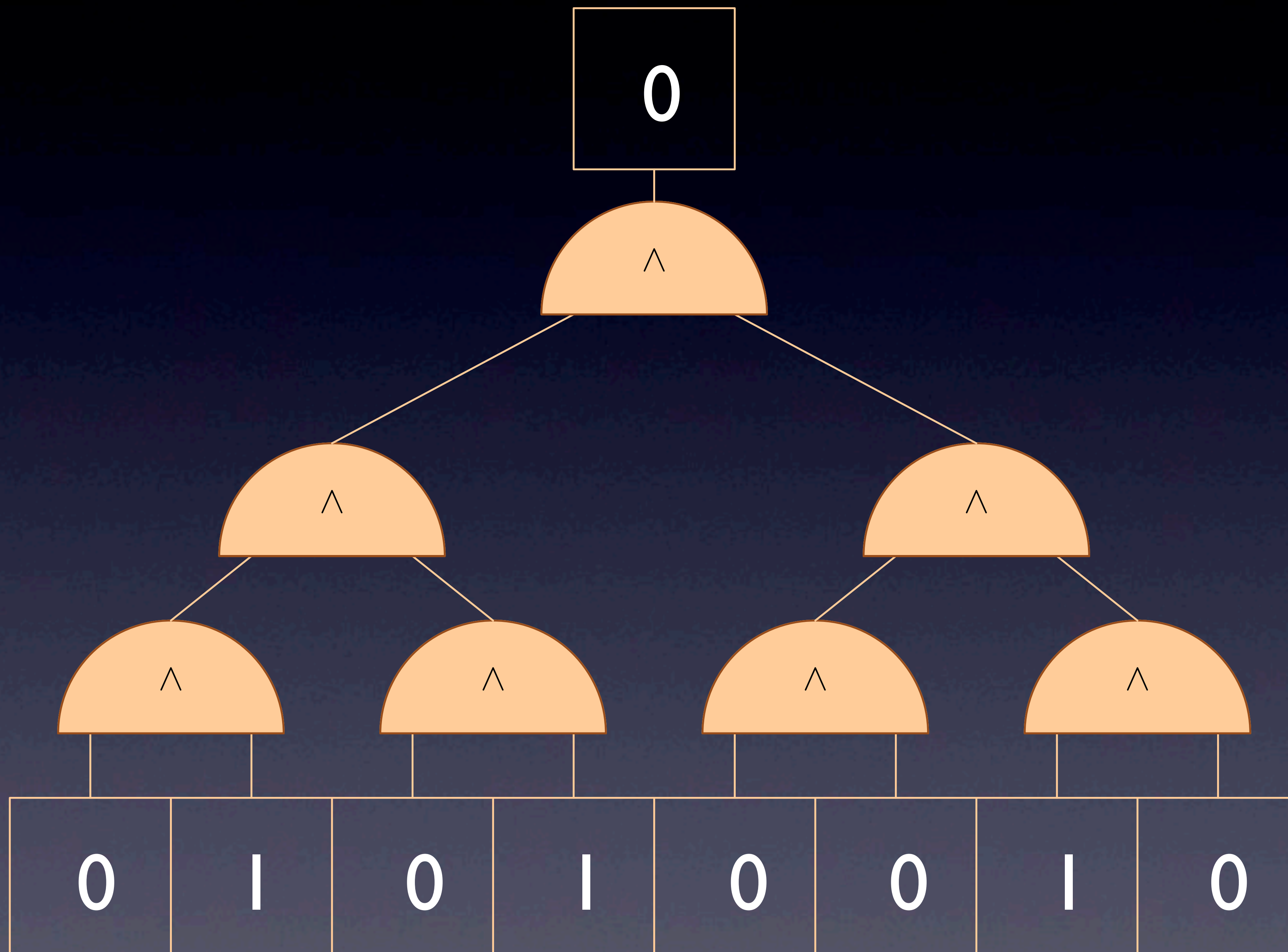
Is there a 0?

0 1 0 1 0 0 1 0

Is there a 0?



Is there a 0?

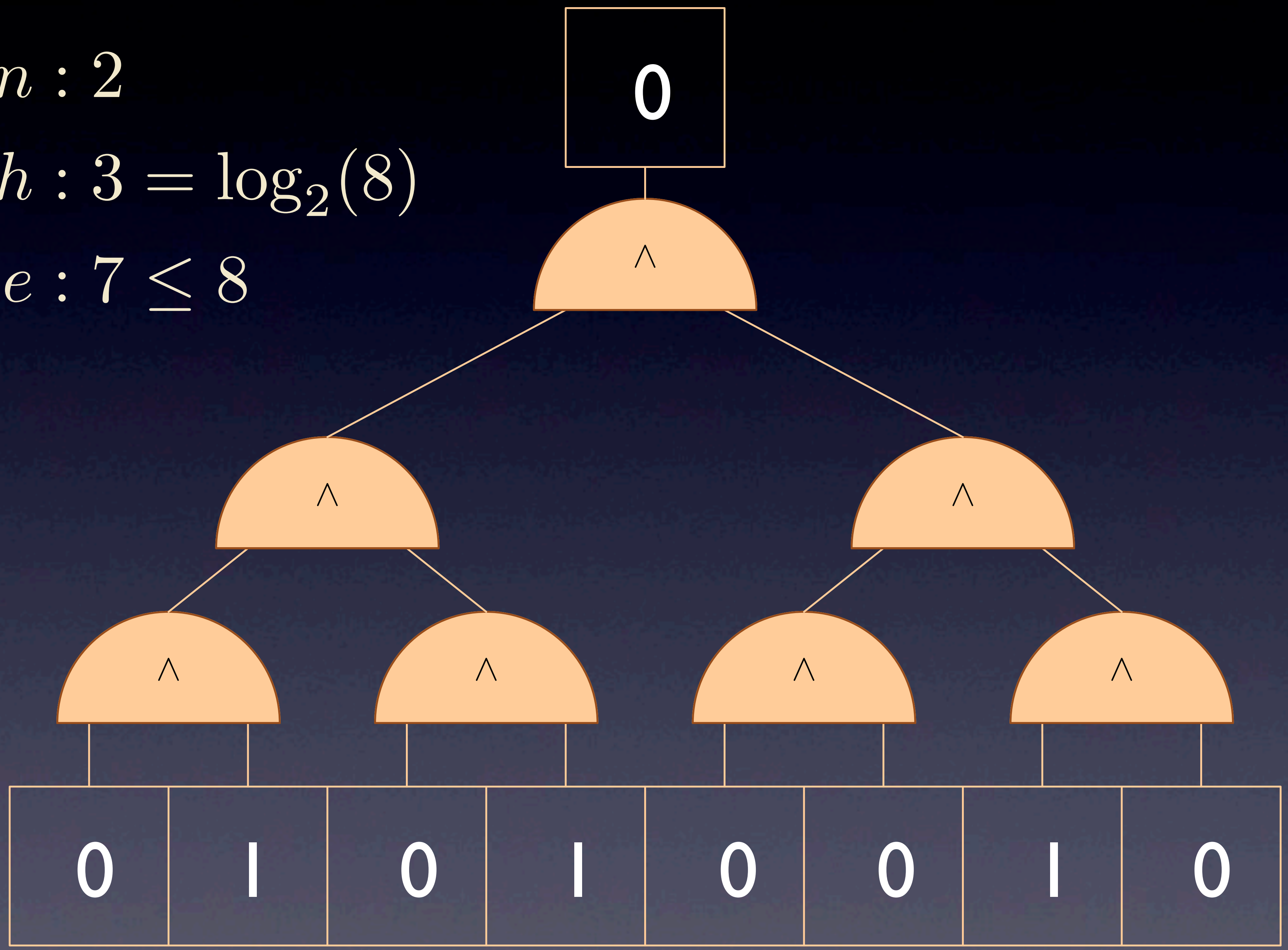


Is there a 0?

$fan\!-\!in : 2$

$depth : 3 = \log_2(8)$

$size : 7 \leq 8$

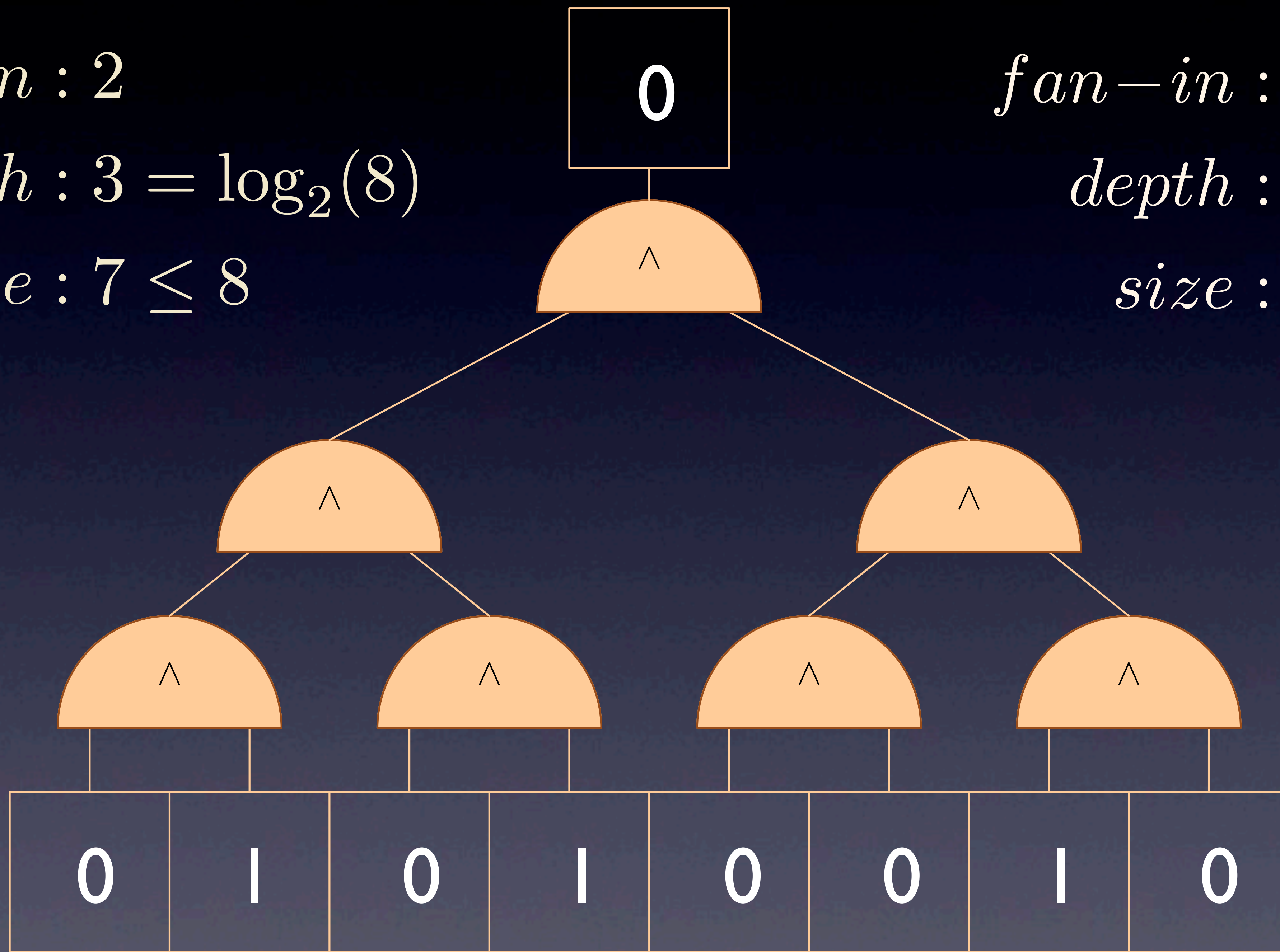


Is there a 0?

$fan{-}in : 2$

$depth : 3 = \log_2(8)$

$size : 7 \leq 8$



$fan{-}in : 2$

$depth : \log_2(n)$

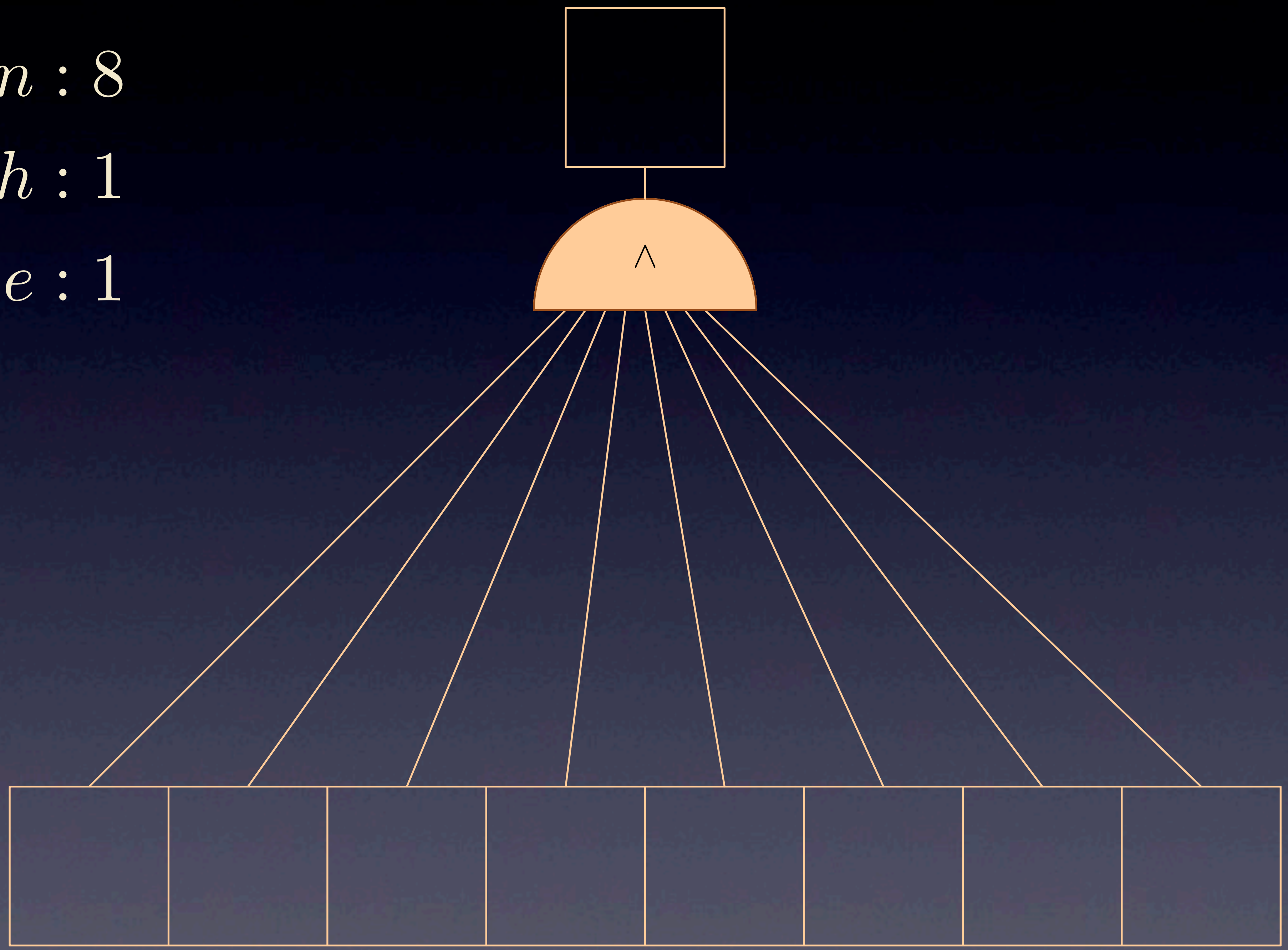
$size : n - 1$

Is there a 0?

$fan{-}in : 8$

$depth : 1$

$size : 1$

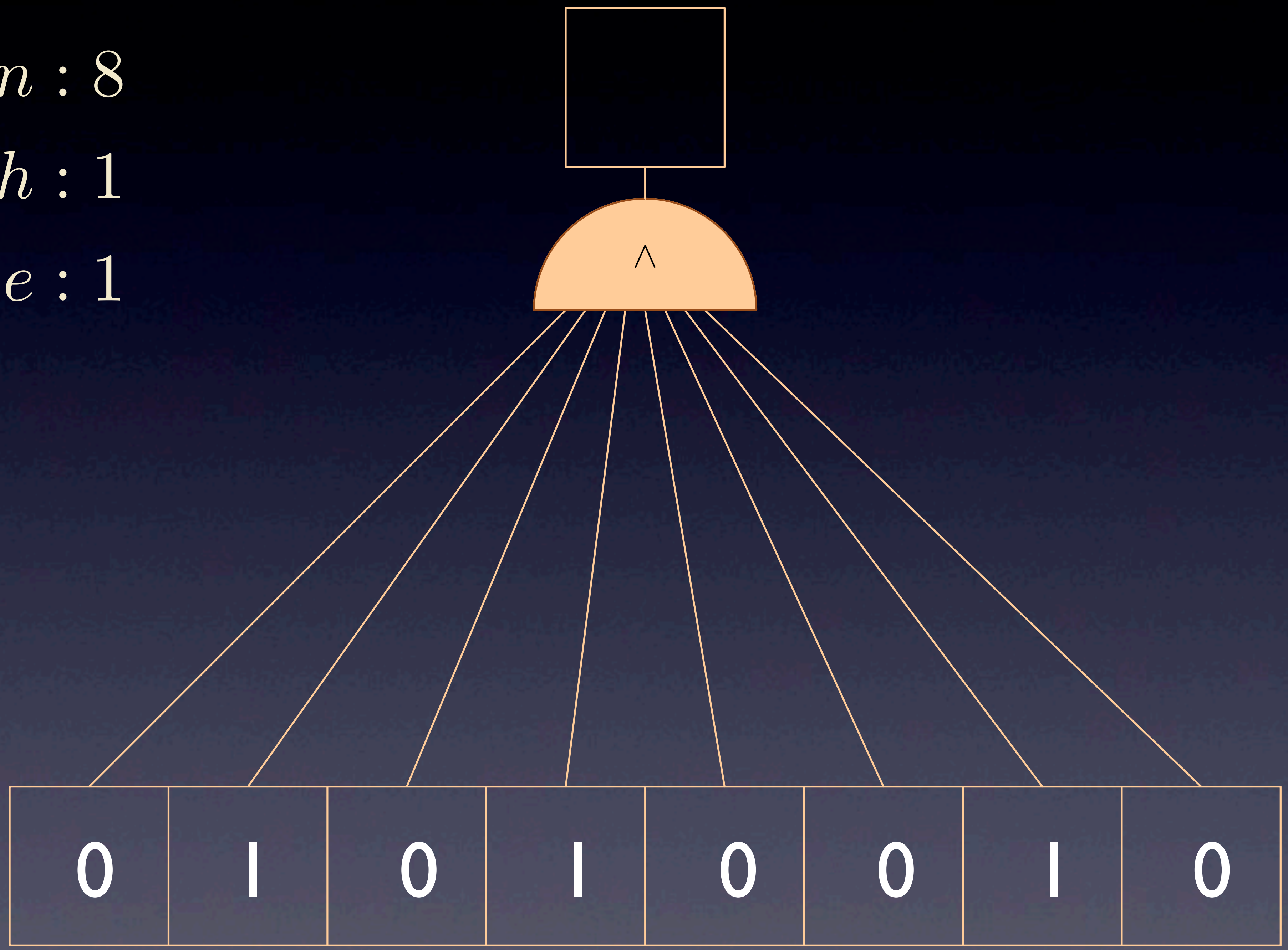


Is there a 0?

$fan{-}in : 8$

$depth : 1$

$size : 1$

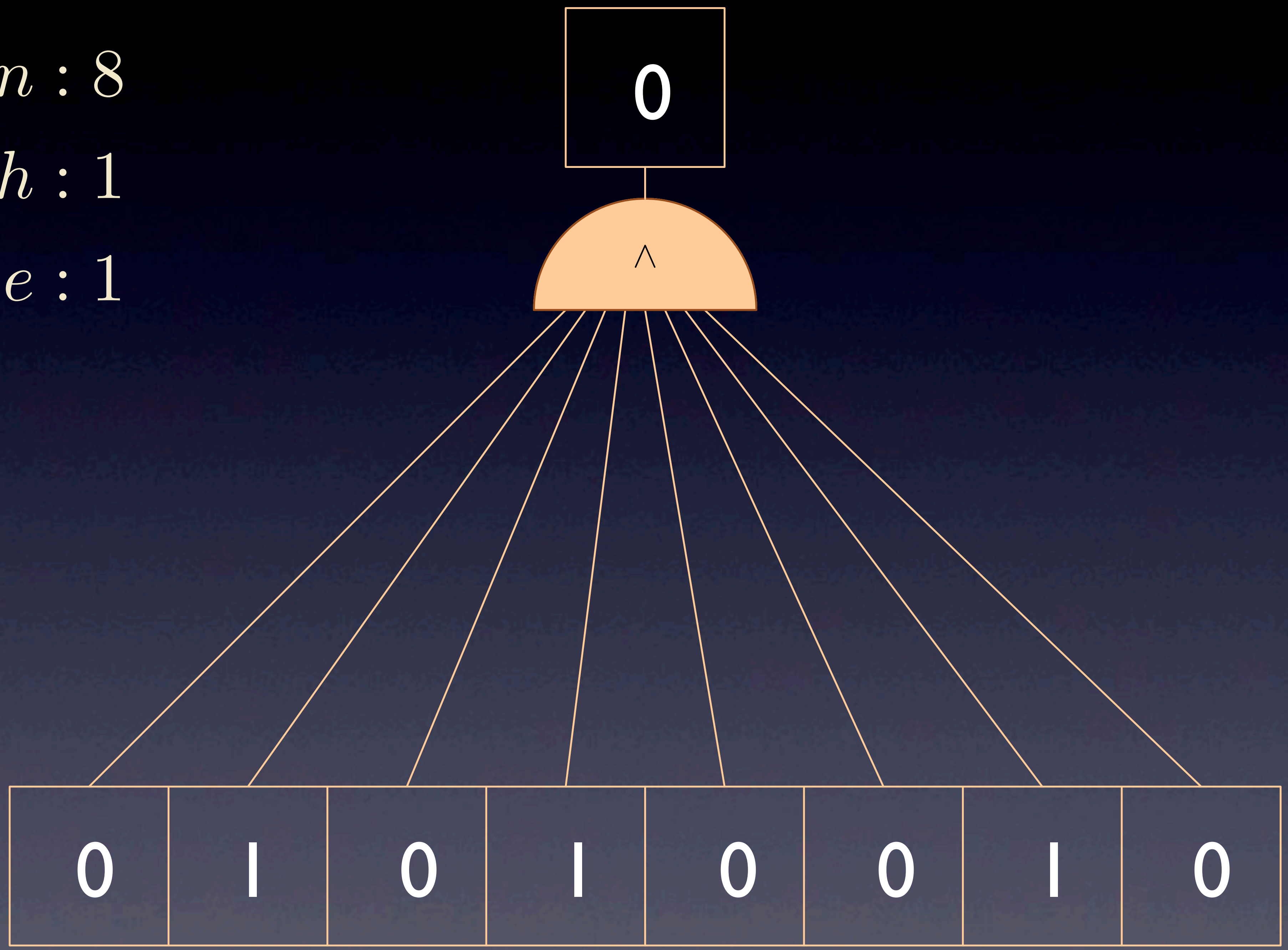


Is there a 0?

$fan\text{-}in : 8$

$depth : 1$

$size : 1$



Circuits NC^k

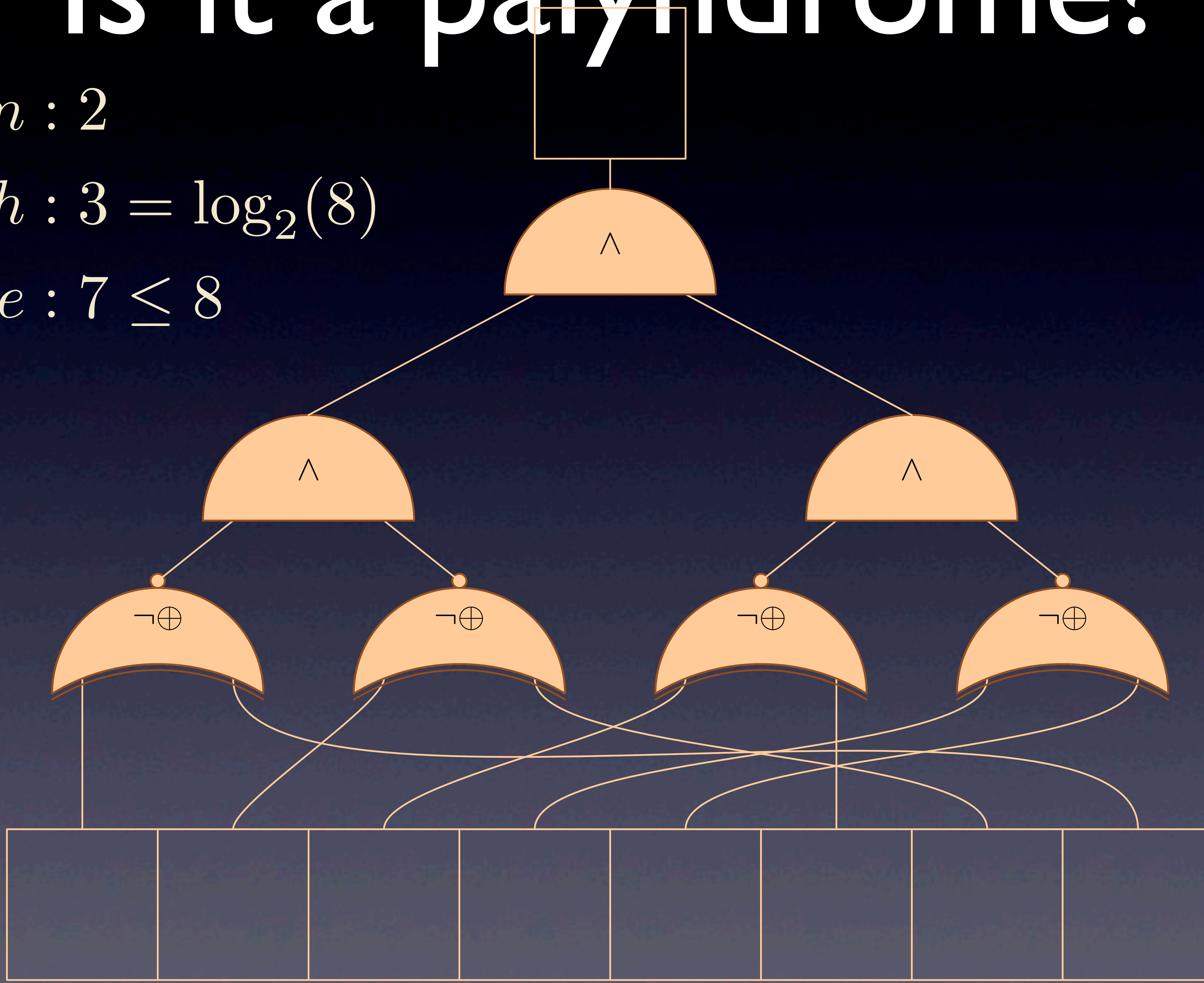
- For $k \geq 1$, NC^k is the class of uniform boolean circuits such that:
  - constant fan-in,
  - polynomial size (w.r.t. the size of inputs)
  - depth is bounded by $O(\log^k(n))$
  - LOGSPACE-Uniform

Is it a palyndrome?

$fan{-}in : 2$

$depth : 3 = \log_2(8)$

$size : 7 \leq 8$

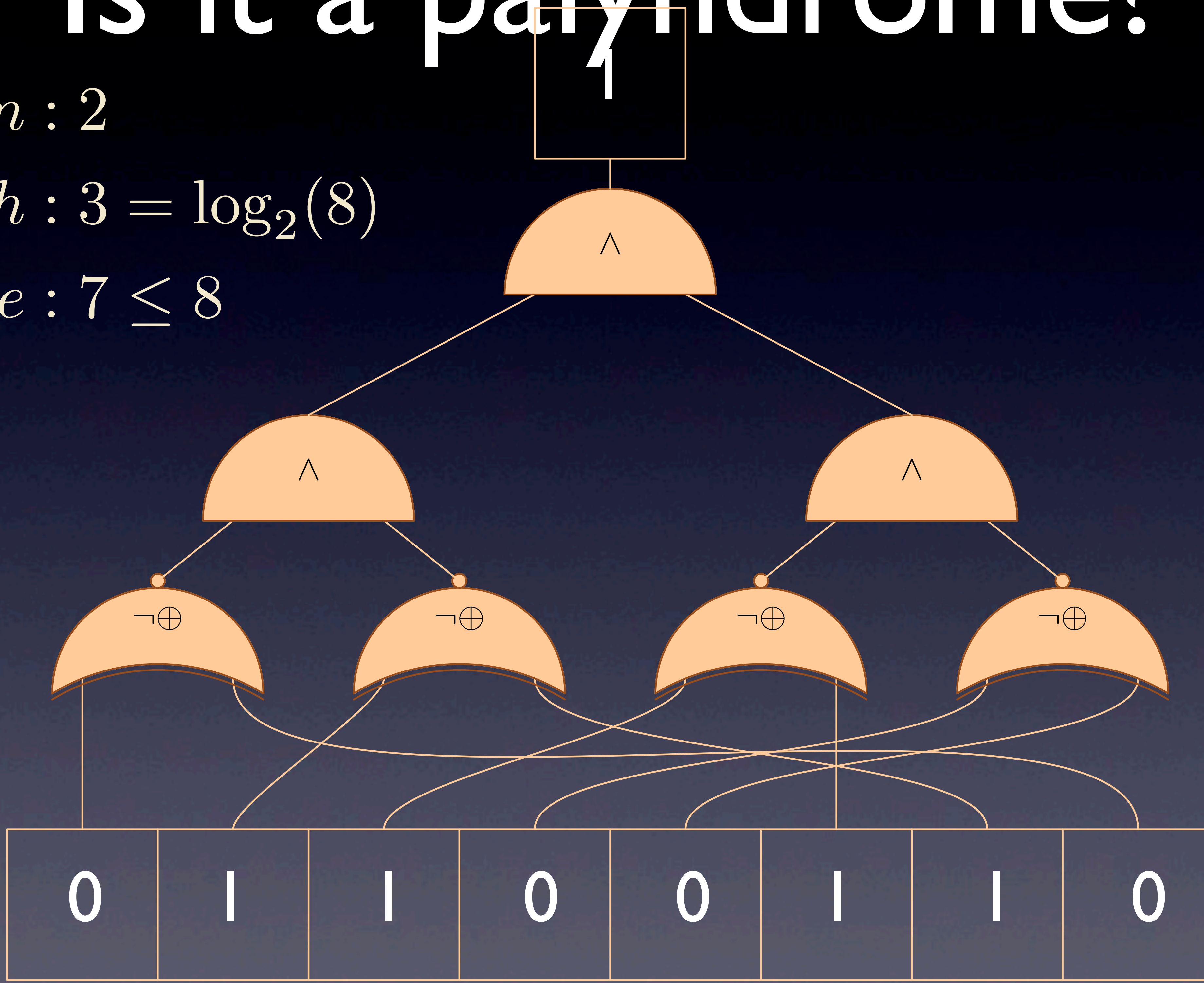


Is it a palyndrome?

$fan{-}in : 2$

$depth : 3 = \log_2(8)$

$size : 7 \leq 8$

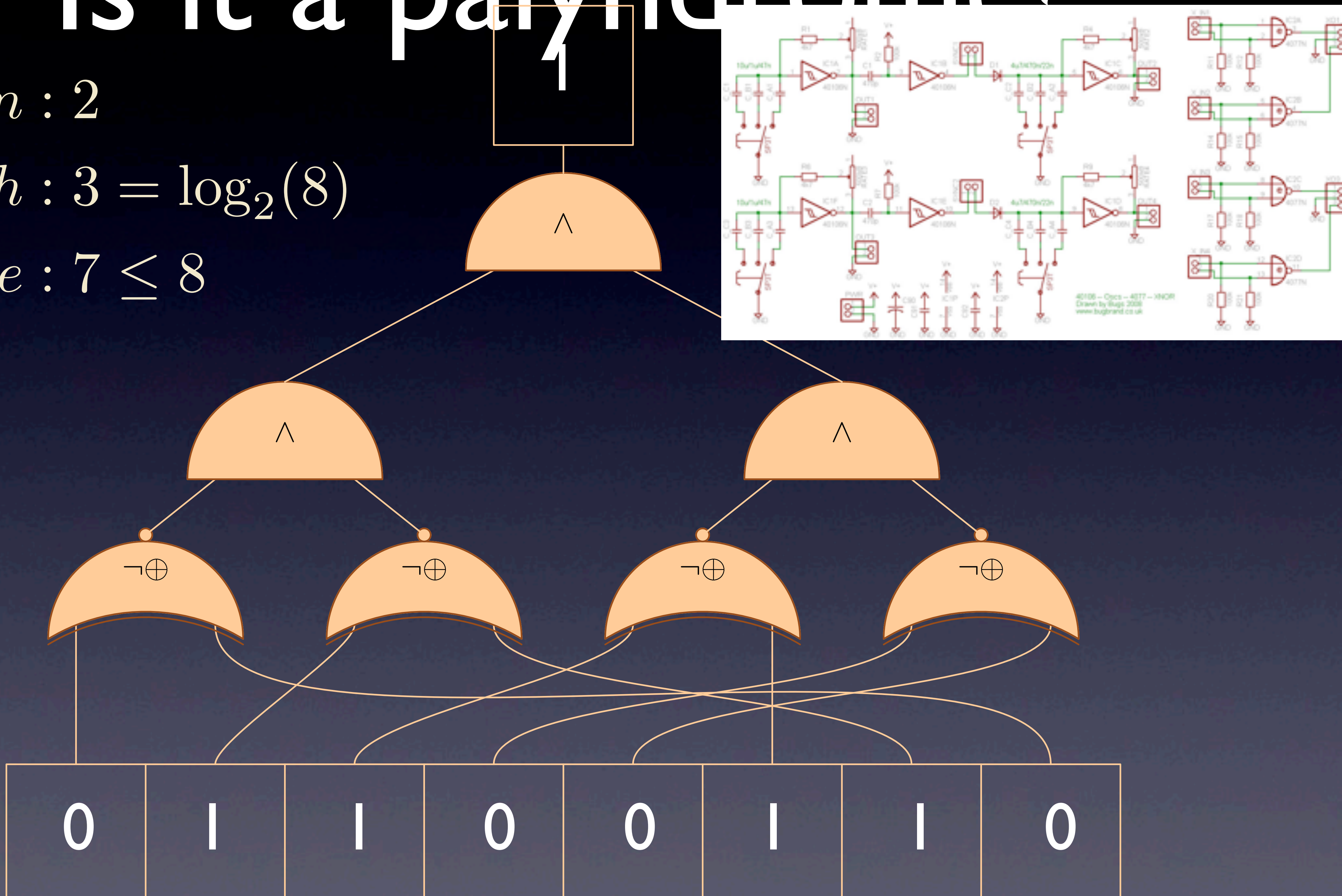


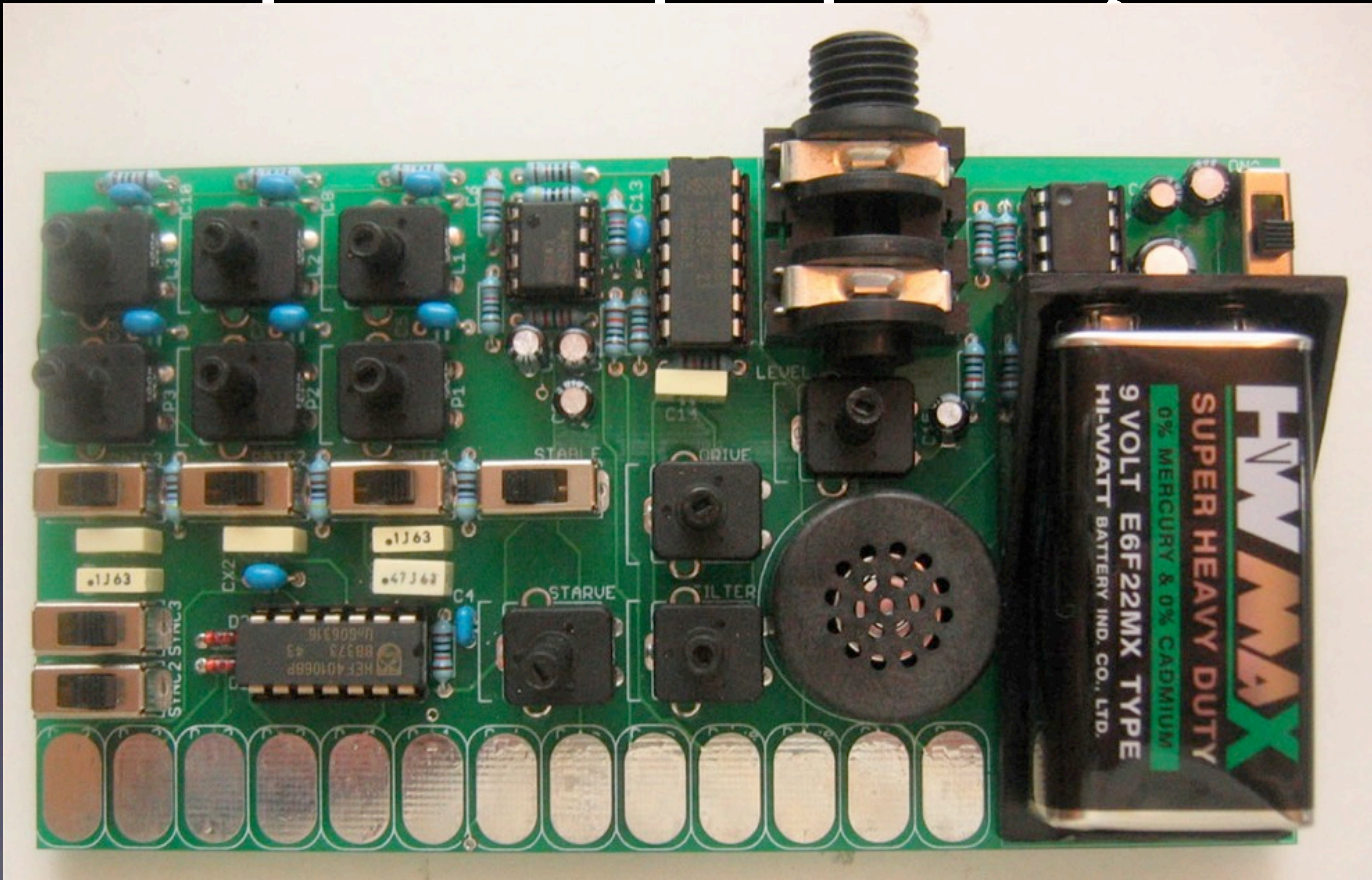
Is it a palyndrome?

$fan\!-\!in : 2$

$depth : 3 = \log_2(8)$

$size : 7 \leq 8$




LEVEL
DRIVE
STARVE
FILTER
SYNC2
SYNC3
C4
C13
HW MAX
SUPER HEAVY DUTY
0% MERCURY & 0% CADMIUM
9 VOLT E6F22MX TYPE
HI-WATT BATTERY IND. CO., LTD.

Implicit computational complexity

Implicit computational complexity

- A characterization of the classes NC^k

Implicit computational complexity

- A characterization of the classes NC^k
  - machine independent

Implicit computational complexity

- A characterization of the classes NC^k
  - machine independent
  - without a priori bounds (cf. Clote, Cobham)

Implicit computational complexity

- A characterization of the classes NC^k
  - machine independent
  - without a priori bounds (cf. Clote, Cobham)
  - Infinite structures (cf Cook, Buss)

Implicit computational complexity

- A characterization of the classes NC^k
  - machine independent
  - without a priori bounds (cf. Clote, Cobham)
  - Infinite structures (cf Cook, Buss)
  - Logical approaches (cf. Mogbil)

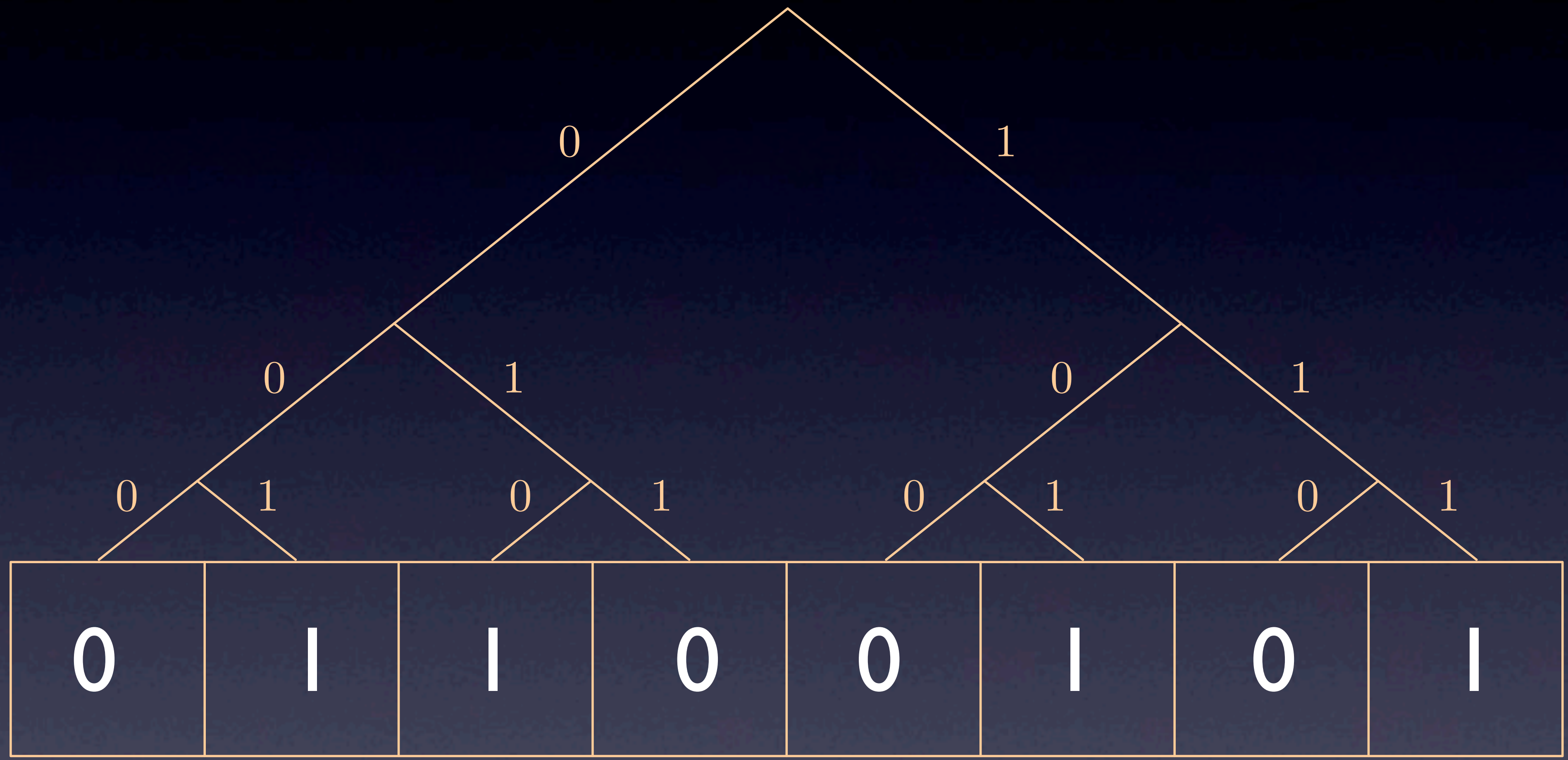
Implicit computational complexity

- A characterization of the classes NC^k
  - machine independent
  - without a priori bounds (cf. Clote, Cobham)
  - Infinite structures (cf Cook, Buss)
  - Logical approaches (cf. Mogbil)
  - Recursion Theory (Leivant, Bloch, Oitavem)

Computing on trees

0	1	1	0	0	1	0	1

Computing on trees

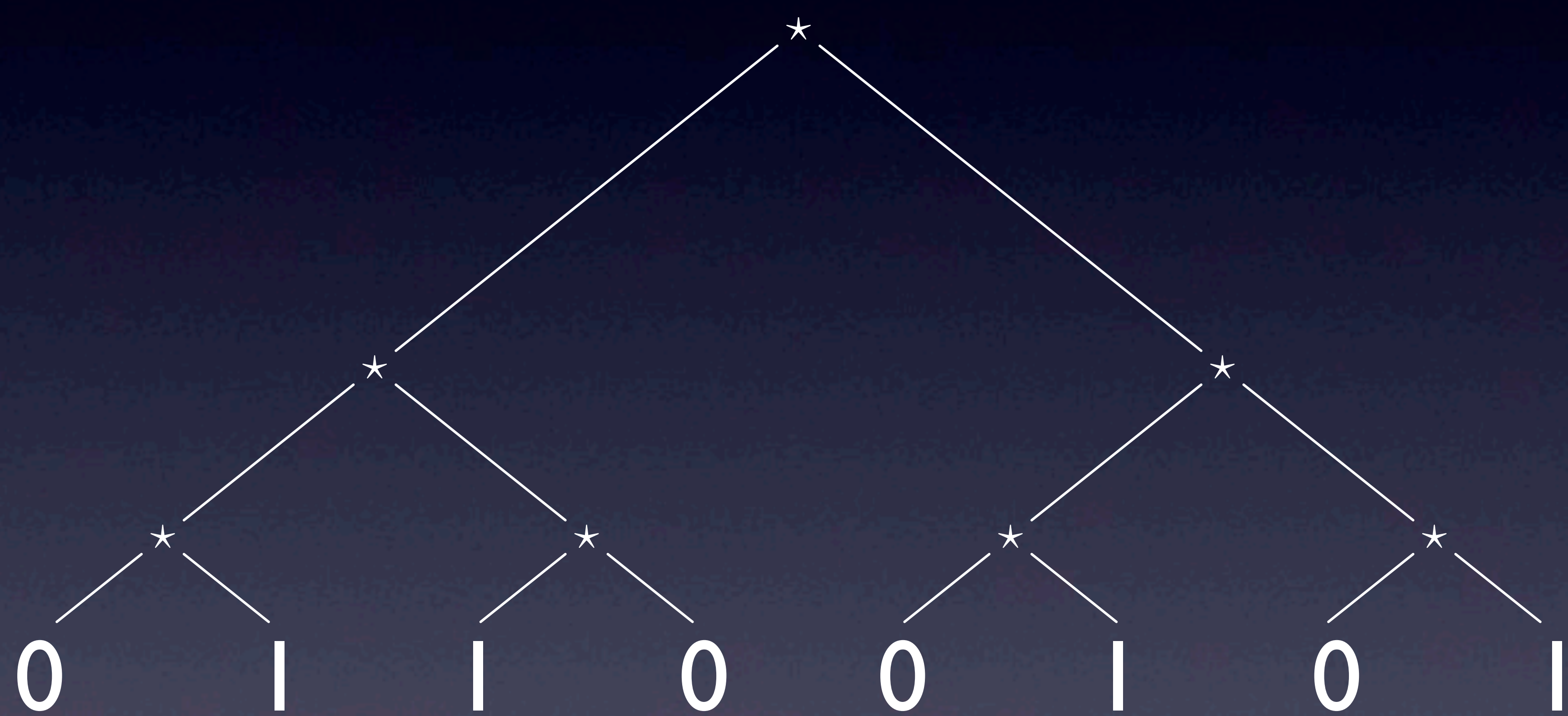


Computing on trees

0	1	1	0	0	1	0	1

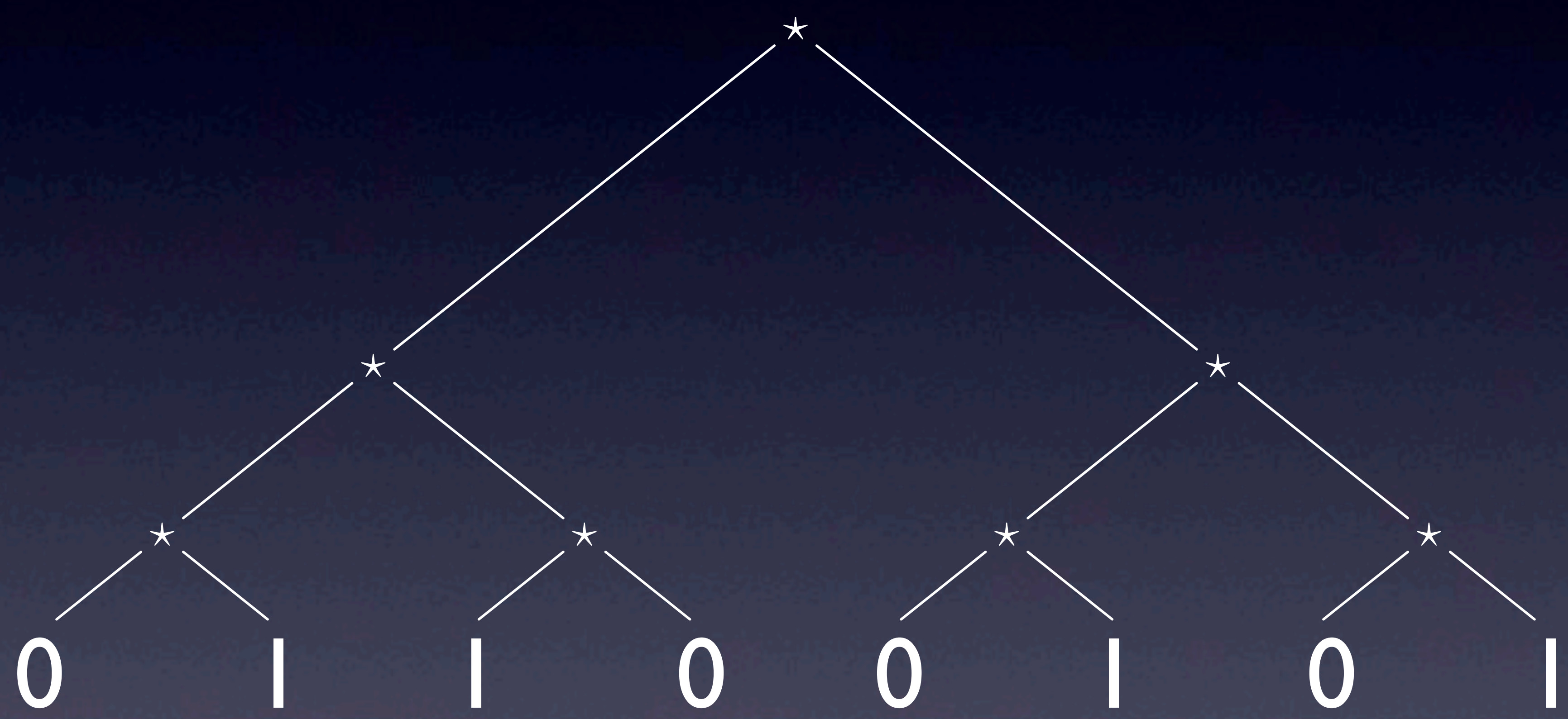
Computing on trees

0	1	1	0	0	1	0	1



Computing on trees

0	1	1	0	0	1	0	1



$$((0 \star 1) \star (1 \star 0)) \star ((0 \star 1) \star (0 \star 1))$$

Basic functions

$$\mathsf{d}_0(c) = \mathsf{d}_1(c) = c, \qquad c \in \{0,1\}$$

$$\mathsf{d}_0(t_0 \star t_1) = t_0,$$

$$\mathsf{d}_1(t_0 \star t_1) = t_1,$$

$$\mathsf{cond}(c, x_0, x_1, x_\star) = x_c, \qquad c \in \{0,1\}$$

$$\mathsf{cond}(t_0 \star t_1, x_0, x_1, x_\star) = x_\star$$

A characterization of NC by Leivant

Ramified Schematic Recurrence

$$f(c, \vec{u}; \vec{x}) = g_c(\vec{u}; \vec{x})$$

$$f(t_0 \star t_1, \vec{u}; \vec{x}) = g_\star(\vec{u}; f(t_0, \vec{u}; h_1(\vec{x})), \dots, f(t_0, \vec{u}; h_d(\vec{x})), f(t_1, \vec{u}; h'_1(\vec{x})), \dots, f(t_1, \vec{u}; h'_{d'}(\vec{x})))$$

Theorem (Leivant):

RSR characterize NC-computable functions

Mutual In-Place Recursion (MIP)

Definition

$(f_i)_{i \in I}$ is defined by MIP if for all i:

$$f_i(t_0 \star t_1, \vec{u}) = f_j(t_0, \sigma_{i,0}(t_0 \star t_1, \vec{u})) \star f_k(t_1, \sigma_{i,1}(t_0 \star t_1, \vec{u}))$$

$$f_i(c, \vec{u}) = g_{i,c}(\vec{u})$$

$g_{i,c}(\vec{u})$ range in $\{0, 1\}$

$\sigma_{i,b}$ is a destructor , $b = 0, 1$

Mutual In-Place Recursion (MIP)

Definition

$(f_i)_{i\in I}$ is defined by MIP if for all i:

$$f_i(t_0 \star t_1, \vec{u}) = f_j(t_0, \vec{\sigma}_{i,0}(t_0 \star t_1, \vec{u})) \star f_k(t_1, \vec{\sigma}_{i,1}(t_0 \star t_1, \vec{u}))$$

$$f_i(c, \vec{u}) = g_{i,c}(\vec{u})$$

$g_{i,c}(\vec{u})$ range in $\{0, 1\}$

$\sigma_{i,b}$ is a destructor , $b = 0, 1$

Computing the palyndrome

$$f_0(t_0 \star t_1) = f_1(t_0, t_1) \star f_1(t_1, t_0)$$
$$f_0(c) = 1$$
$$f_1(t_0 \star t_1, u) = f_1(t_0, \mathsf{d}_1(u)) \star f_1(t_1, \mathsf{d}_0(u))$$
$$f_1(c, c') = c == c'$$

with

$$\mathsf{d}_0(t_0 \star t_1) = t_0$$
$$\mathsf{d}_1(t_0 \star t_1) = t_1$$

Computing the palyndrome

$$f_0(((0 \star 1) \star (1 \star 0)) \star ((0 \star 1) \star (1 \star 0)))$$

Computing the palyndrome

$$f_0(((0 \star 1) \star (1 \star 0)) \star ((0 \star 1) \star (1 \star 0)))$$

$$= ((1 \star 1) \star (1 \star 1)) \star ((1 \star 1) \star (1 \star 1))$$

Computing the palyndrome

$$f_0(((0 \star 1) \star (1 \star 0)) \star ((0 \star 1) \star (1 \star 0)))$$

$$= ((1 \star 1) \star (1 \star 1)) \star ((1 \star 1) \star (1 \star 1))$$

$$f(t_0 \star t_1, u) = f(t_0, \mathsf{d}_0(u)) \star f(t_1, \mathsf{d}_1(u)),$$
$$f(c, u) = \wedge(c, \wedge(\mathsf{d}_0(u), \mathsf{d}_1(u)));$$

$$\mathsf{AND}(t_0 \star t_1) = f(t_0, t_0 \star t_1) \star f(t_1, t_0 \star t_1),$$
$$\mathsf{AND}(c) = c$$

Computing the palyndrome

Claim:

$$\text{AND}^{\log_2(n)}(t) = (((((b \star \cdots) \cdots)$$

with $b = 0$ iff t contains a 0

Computing the palyndrome

Claim:

$$\text{AND}^{\log_2(n)}(t) = (((((b \star \cdots) \cdots)$$

with $b = 0$ iff t contains a 0

Thus

$$\text{AND}^{\log_2(n)}(f_0(t)) = (((((b \star \cdots) \cdots)$$

with $b = 1$ iff t is a palyndrome

Time iteration

$$f(t'_1 \star t''_1, t_2, \ldots, t_k, s, \vec{u}) = h(f(t'_1, t_2, \ldots, t_k, s, \vec{u}), \vec{u})$$

$$f(c_1, t'_2 \star t''_2, t_3, \ldots, t_k, s, \vec{u}) = f(s, t'_2, t_3 \ldots, t_k, s, \vec{u})$$

$$\vdots$$

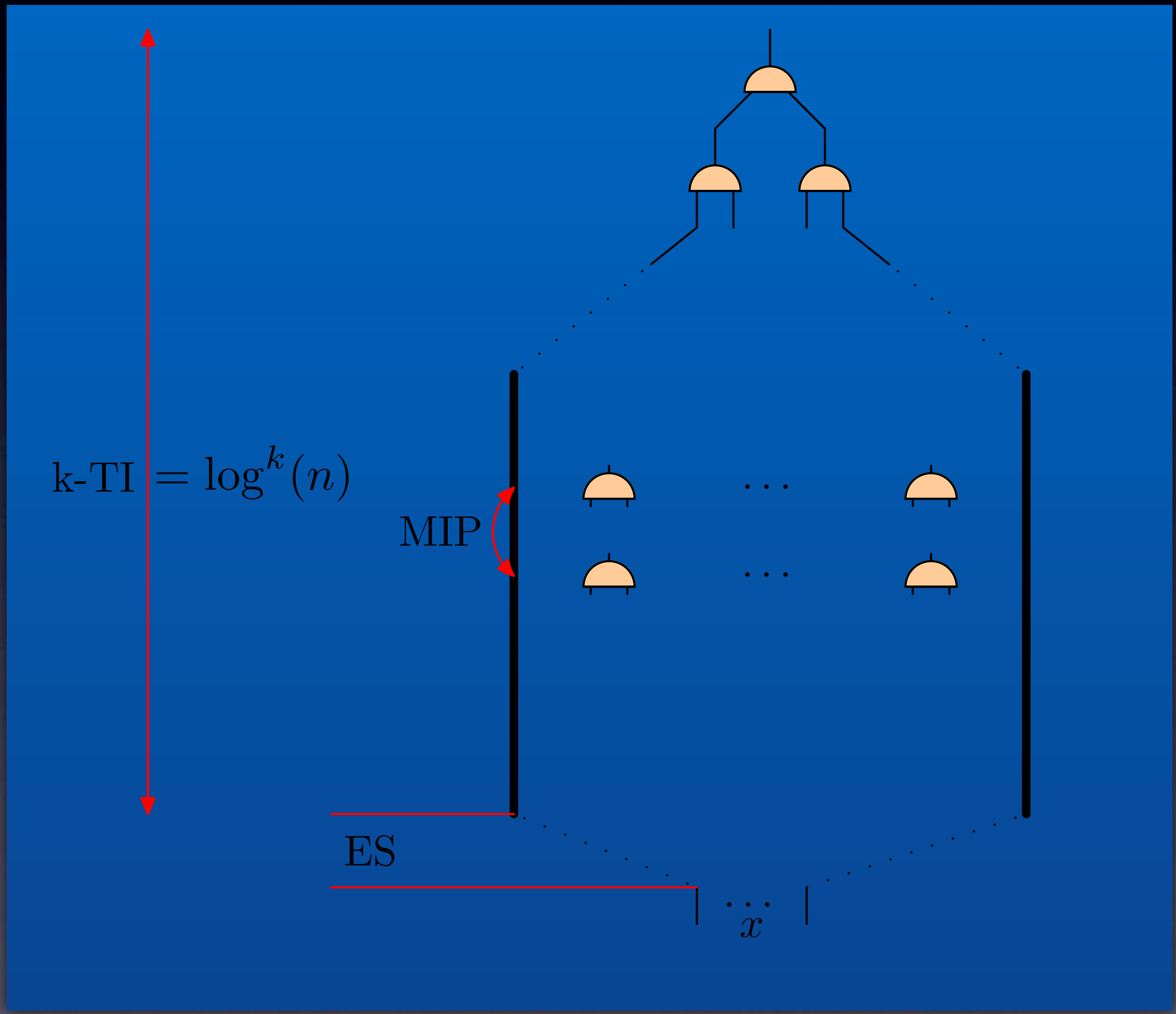
$$f(c_1, \ldots, c_{i-1}, t'_i \star t''_i, t_{i+1}, \ldots, t_k, s, \vec{u}) = f(c_1, \ldots, c_{i-2}, s, t'_i, t_{i+1}, \ldots, t_k, s, \vec{u})$$

$$\vdots$$

$$f(c_1, \ldots, c_k, s, \vec{u}) = g(s, \vec{u})$$

MIP+TI^k = NC^k

Theorem (BKMO): MIP+TI^k = NC^k



Rational Bitwise Equations

$$f(w_0, \ldots, w_k) = w$$

$$|w_0| = |w|$$

$$w[p] = h(\phi_0(p), w_{e_1}[\phi_1(p)], \ldots, w_{e_m}[\phi_m(p)])$$

$\phi_0, \ldots, \phi_m$ some functional transducers on $\{0,1\}^*$

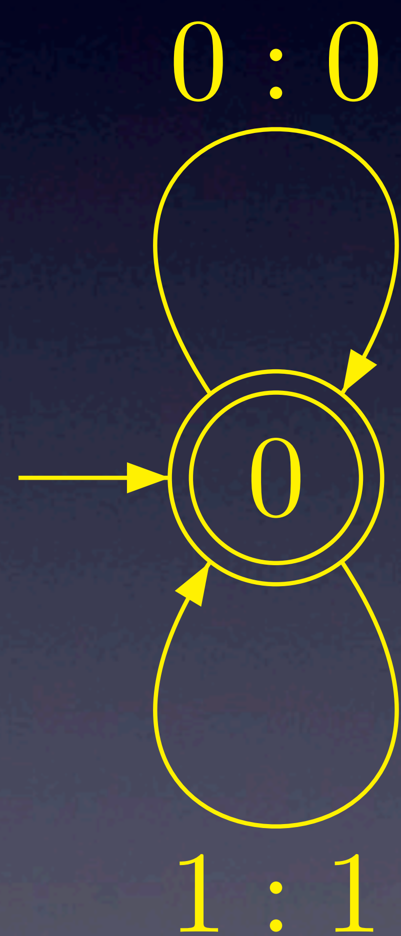
h is a finite mapping in $\{0,1\}$

On the palyndrome

$$f(w_0) = w$$

$$w[p] = \mathrm{XNOR}(w_0[1(p)], w_0[\bar{1}(p)])$$

$1:$



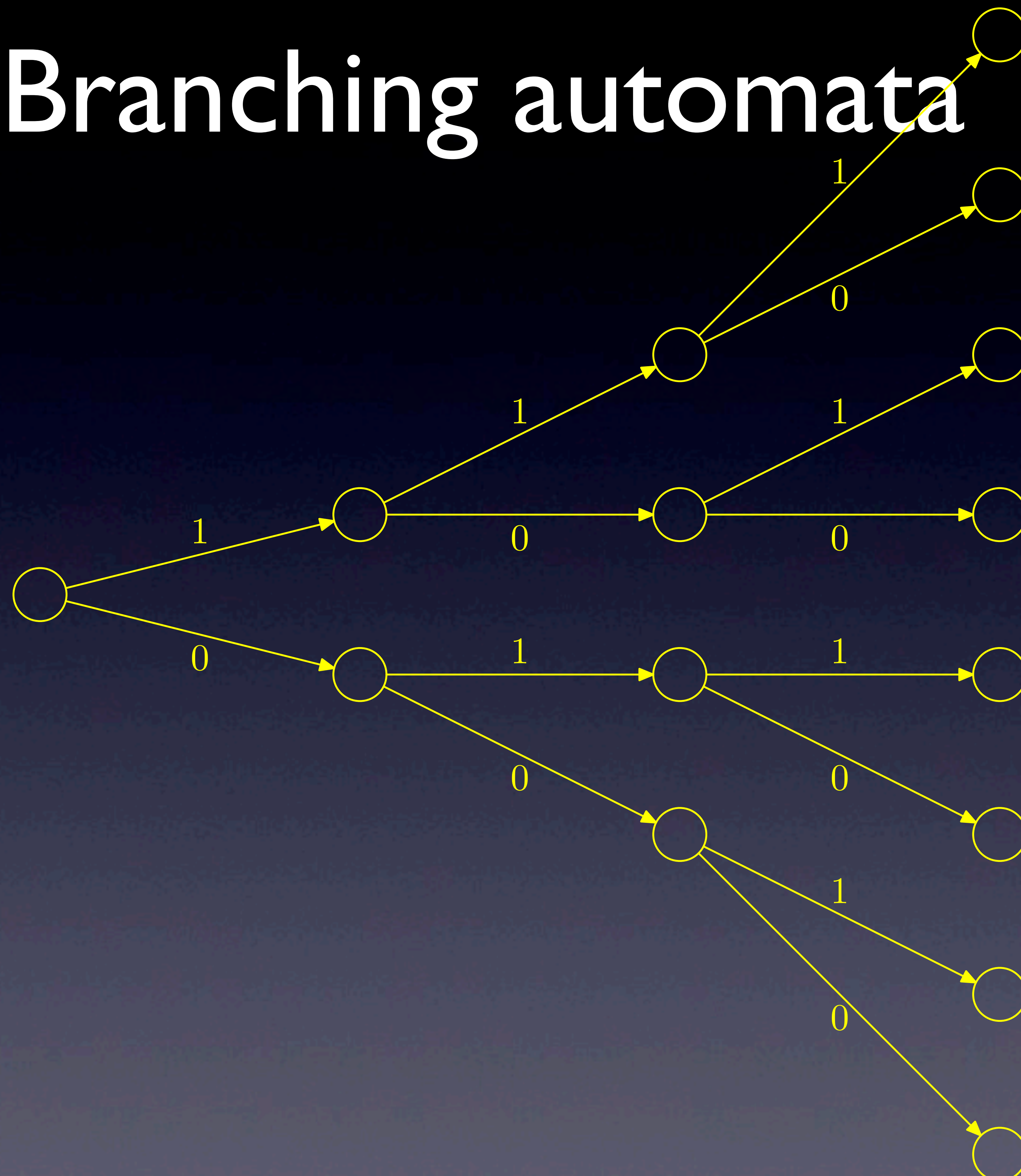
$\bar{1}:$

MIP = RBE

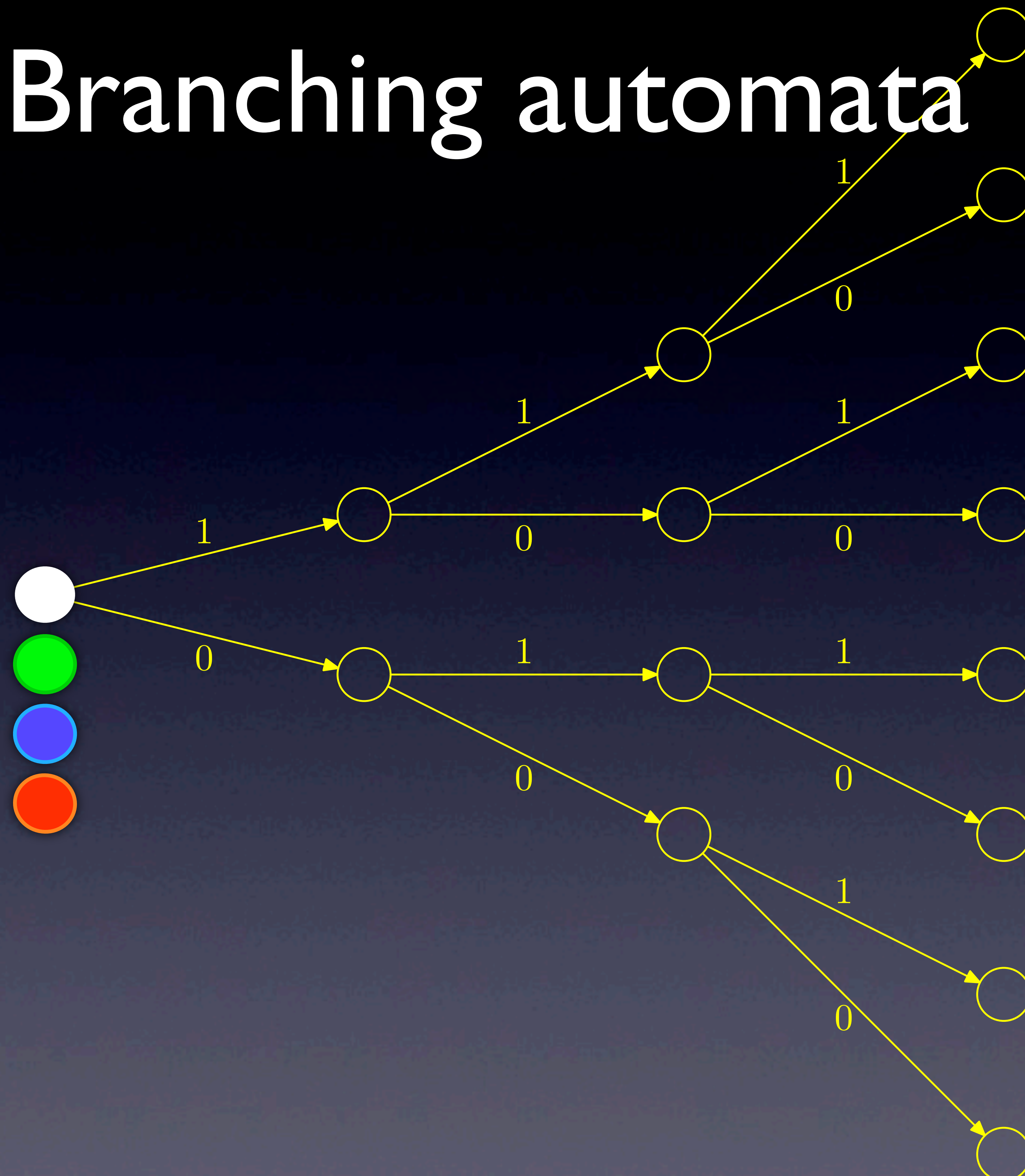
Theorem : MIP = RBE

A proof?

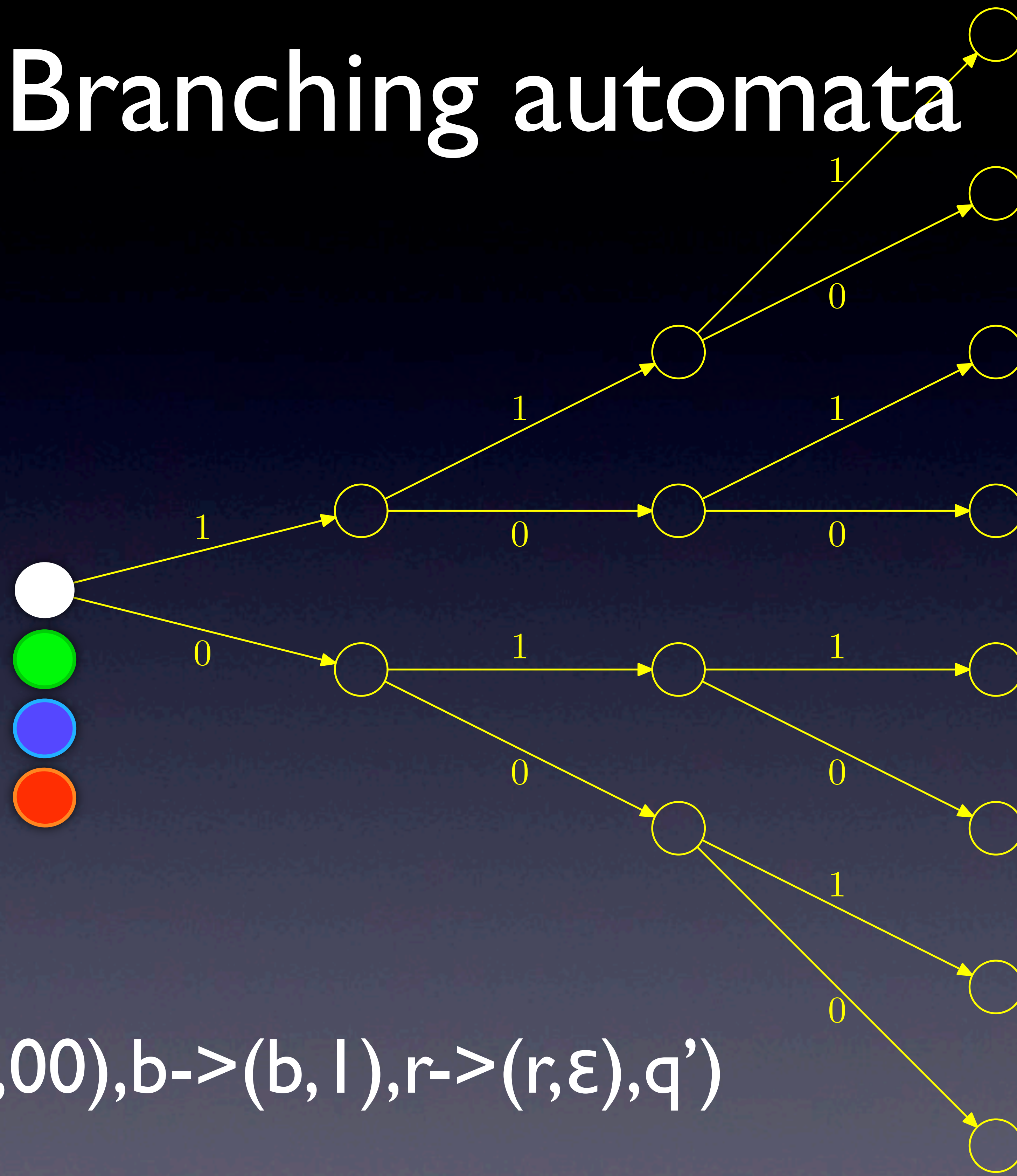
Branching automata



Branching automata

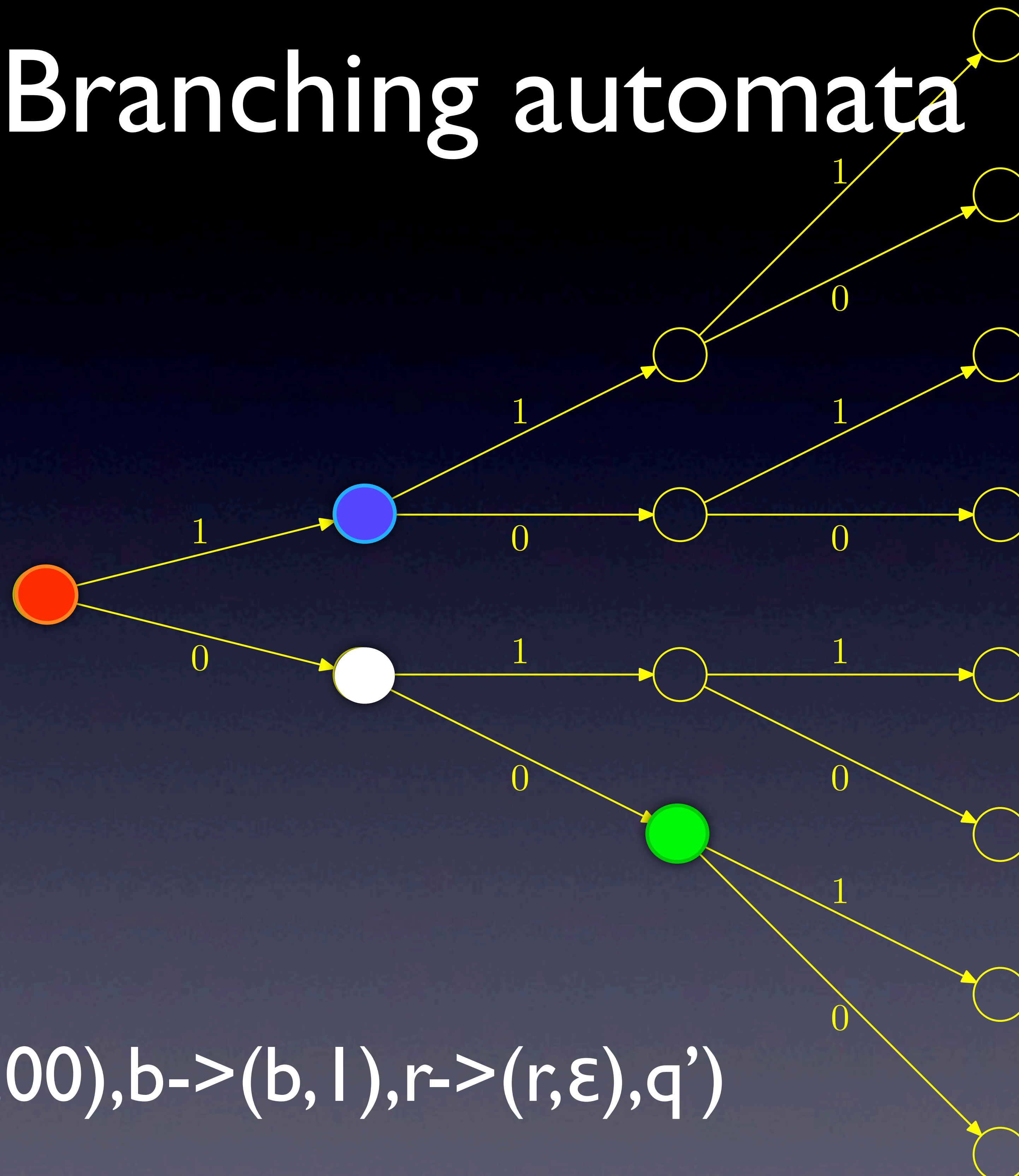


Branching automata



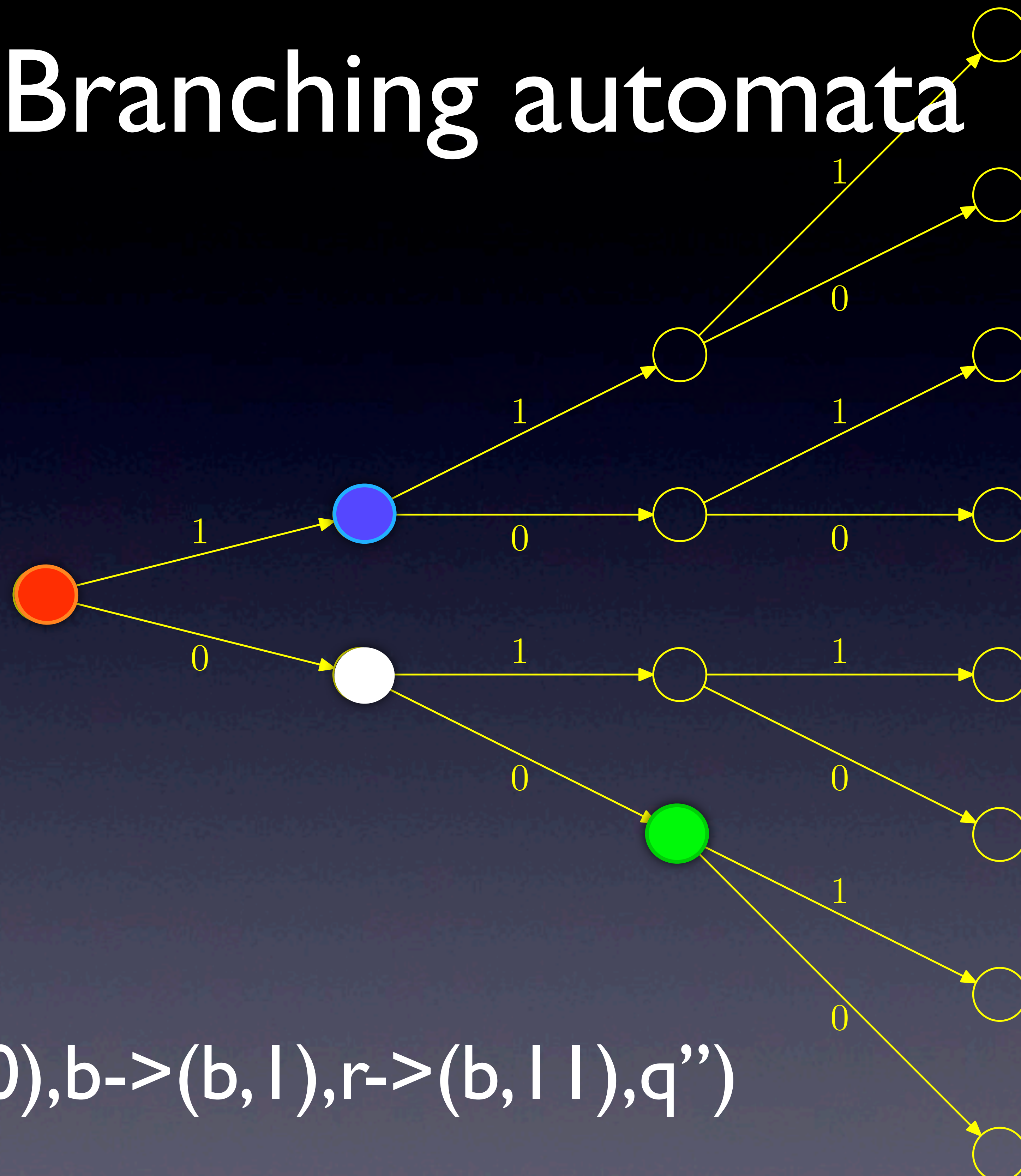
(q0,0,g->(r,00),b->(b,1),r->(r,ε),q')

Branching automata



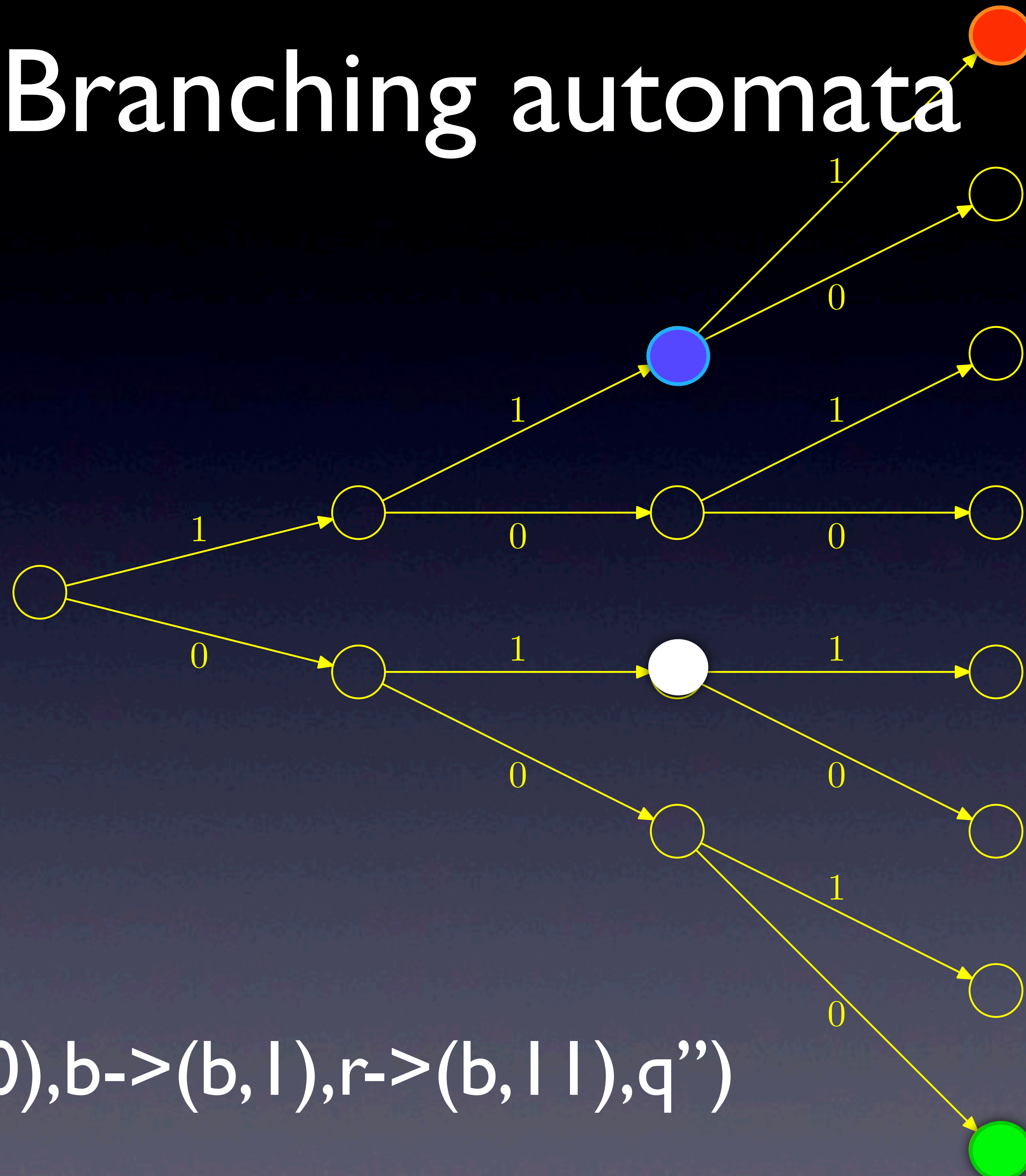
(q0,0,g->(r,00),b->(b,1),r->(r,ε),q')

Branching automata



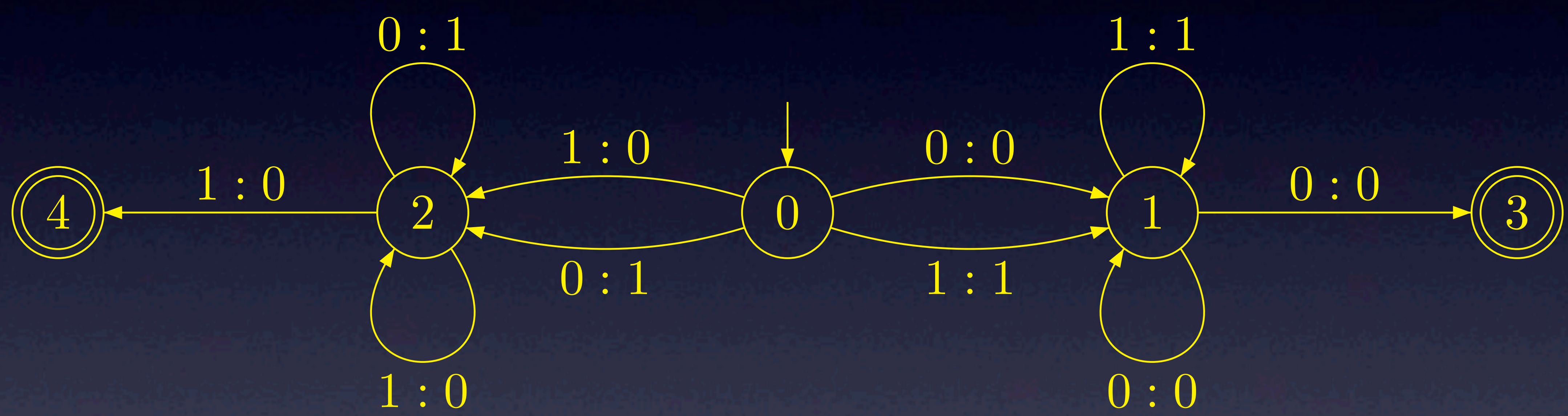
(q’,1,g->(g,0),b->(b,1),r->(b,11),q”)

Branching automata



(q’,1,g->(g,0),b->(b,1),r->(b,11),q”)

A proof?



Functional transducers

MIP == RBE iff FT == BA

Theorem (Elgot and Mezei, 1965):
Rational functions are the composition of a sequential and a co-sequential function

Exercice:
compute the finite transducer above with a BA

Conclusion

- A small step to NC^k
- A longer way to NC^0
- An even longer way to AC^k