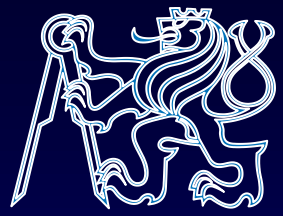
- ©Jan Schmidt 2012
  Katedra číslicového návrhu
  Fakulta informačních technologií
  České vysoké učení technické v Praze
- Zimní semestr 2012/13



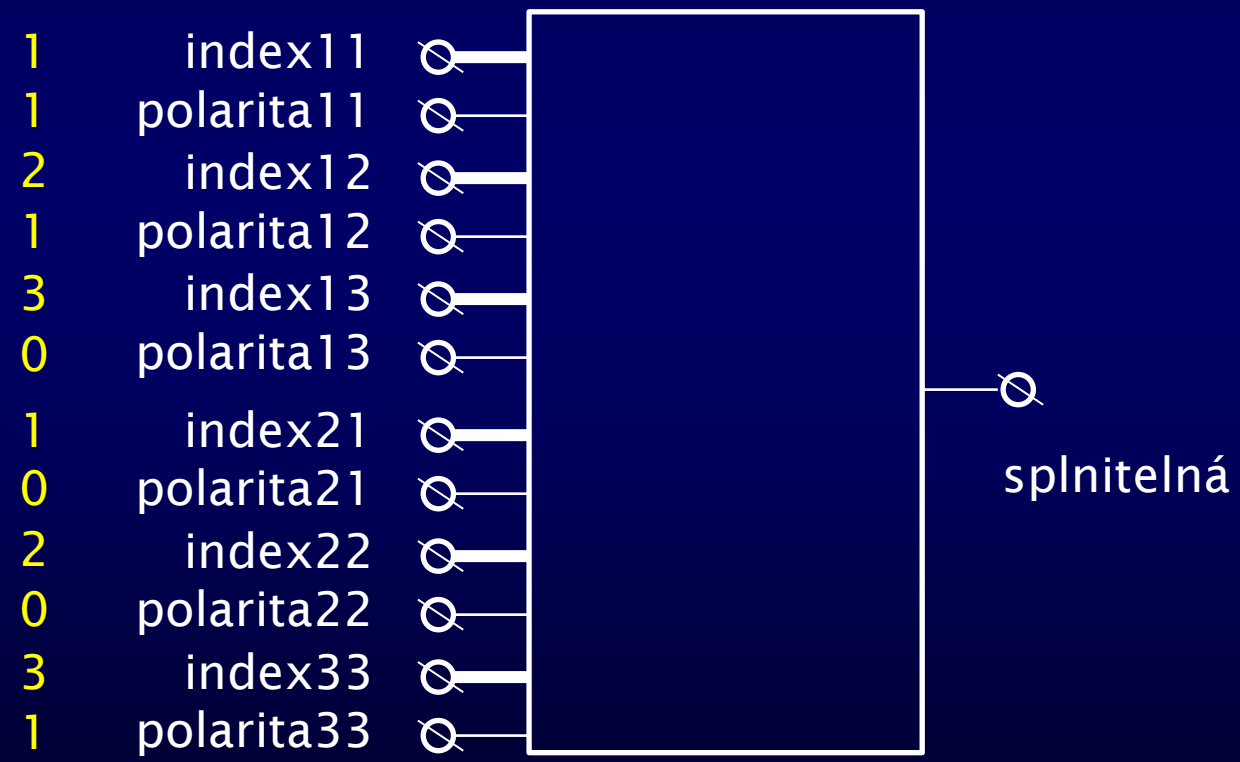
MI-PAA

5. Komunikační a obvodová složitost

- Obvodová složitost, třídy NC^0, AC^0, TC^0, NC^1
- Komunikační složitost
- Kolmogororova složitost, algoritmická náhodnost

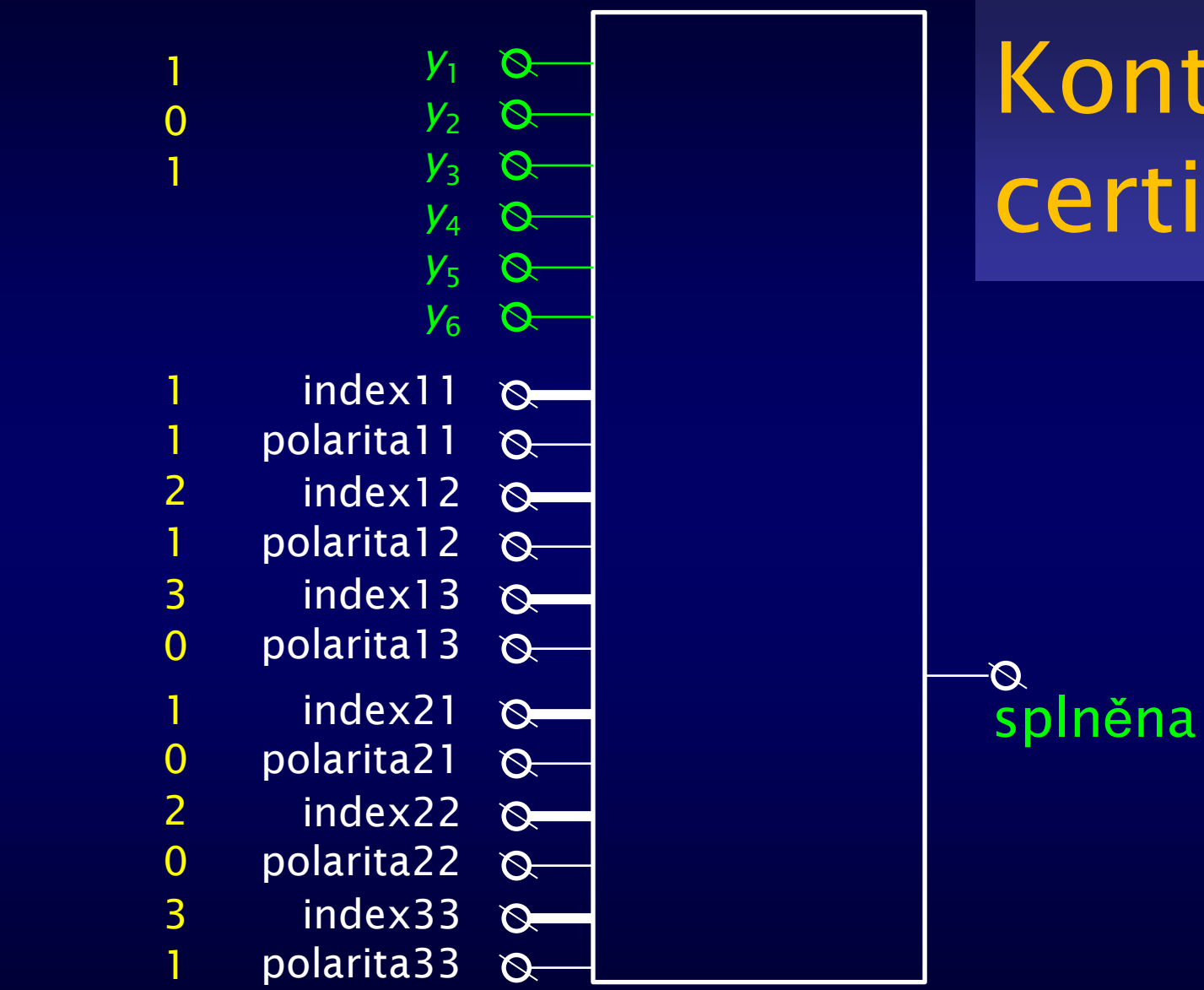
Obvod pro 3-SAT, 2 klausule

$(x_1'+x_2'+x_3)(x_1+x_2+x_3')$

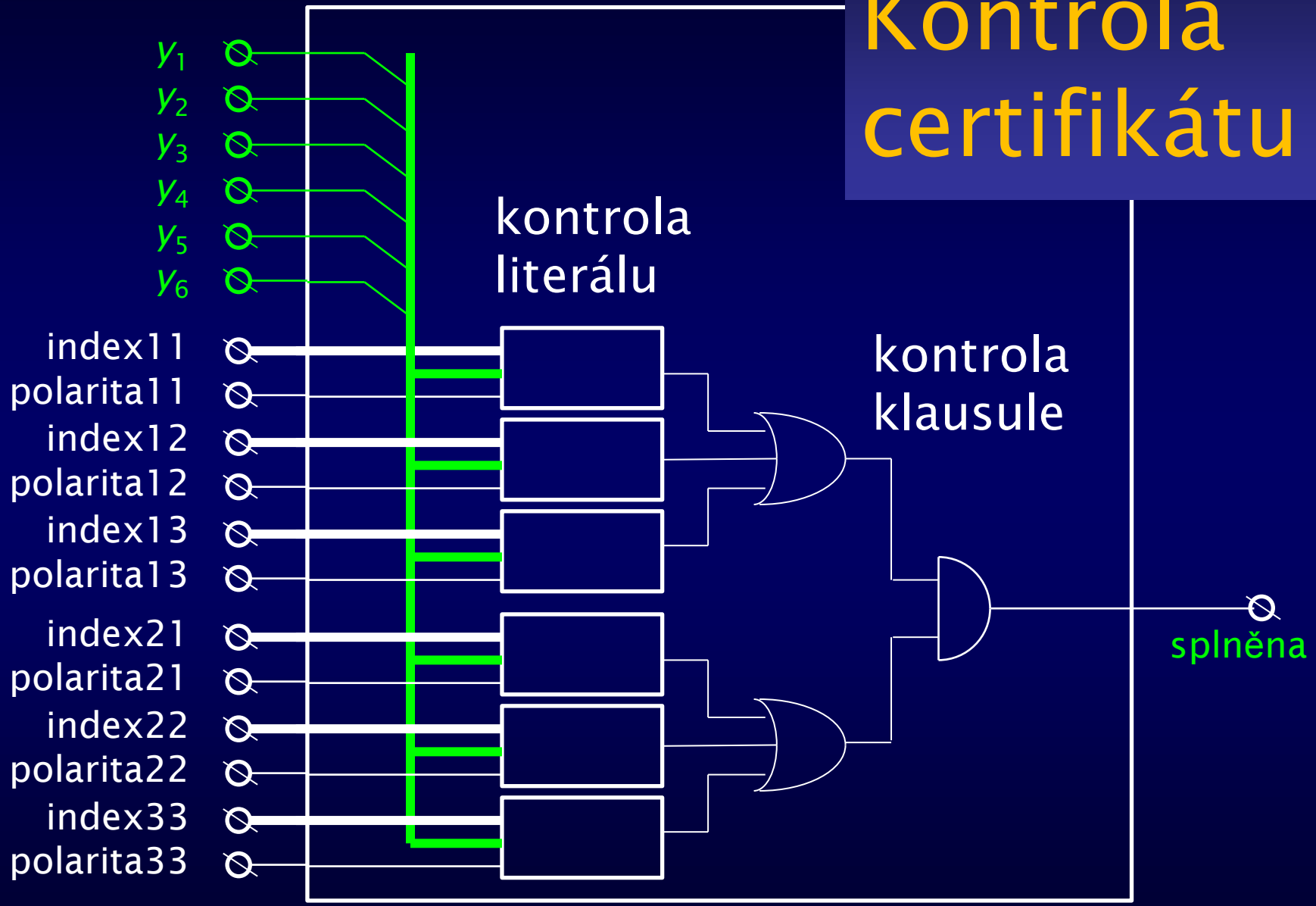


$(x_1'+x_2'+x_3)(x_1+x_2+x_3')$

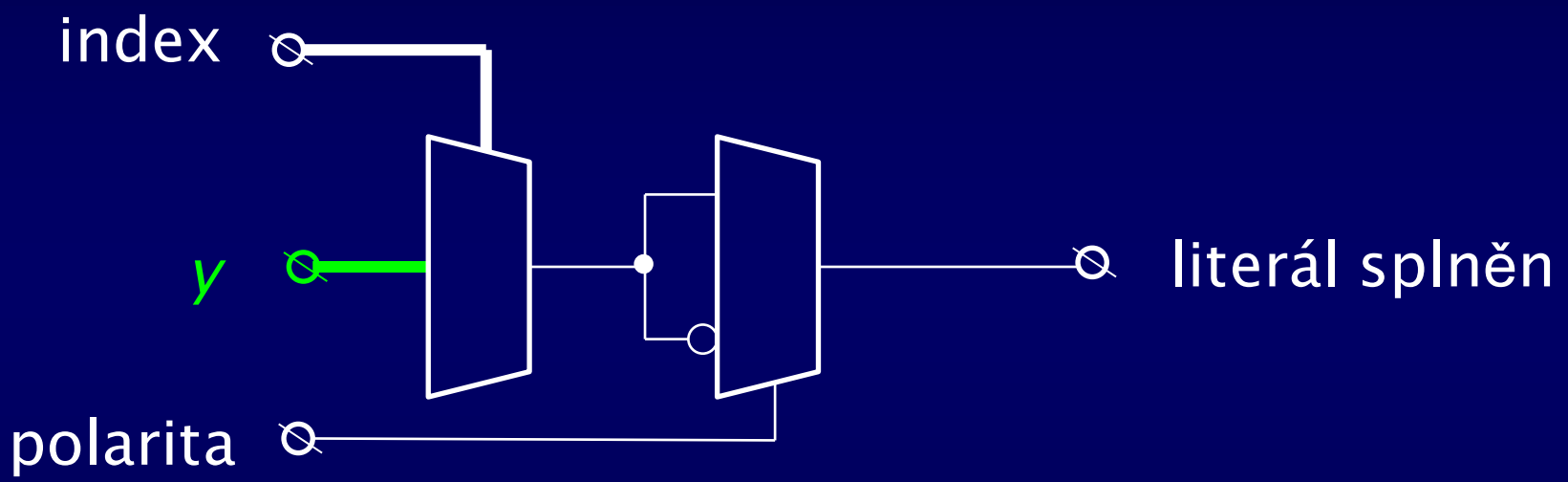
Kontrola certifikátu



Kontrola certifikátu



Kontrola literálu

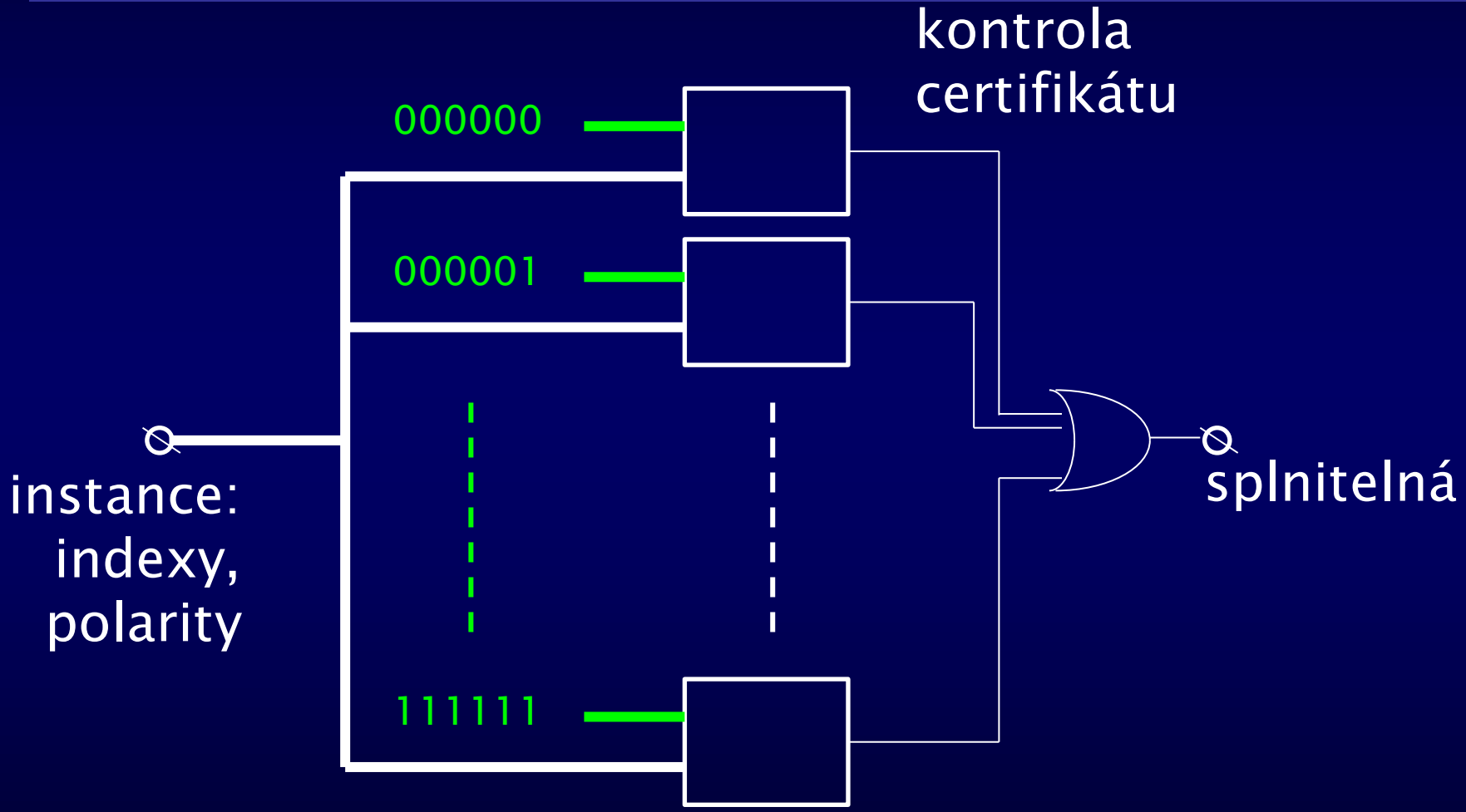


Vlastnosti kontrolního obvodu

- Formule 3-SAT
  - nechť má n klauzulí $\Rightarrow$ $3n$ literálů
  - pak má $O(n)$ proměnných
- Obvod
  - má $3n$ multiplexorů o $O(n)$ vstupech
  - má $3n$ multiplexorů o 2 vstupech
  - má n členů OR o 3 vstupech
  - má 1 člen AND o n vstupech
  - má 4 logické úrovně – konstantní (?) hloubka
  - algoritmicky sestrojitelný v polynomiálním čase

polynomiální velikost

Obvod na řešení 3-SAT, 2 klauzule



Vlastnosti

- Exponenciální velikost
- Konstantní (?) hloubka
- Algoritmicky sestrojitelný –
  ale v exponenciálním čase
- Polynomiální velikost, konstrukce
  v polynomiálním čase $\Rightarrow$ P = NP
- Polynomiální velikost $\Rightarrow$ $\Pi_2 = \Sigma_2$
  (kolaps polynomiální hierarchie,
  Karp–Liptonova věta)

Booleův obvod

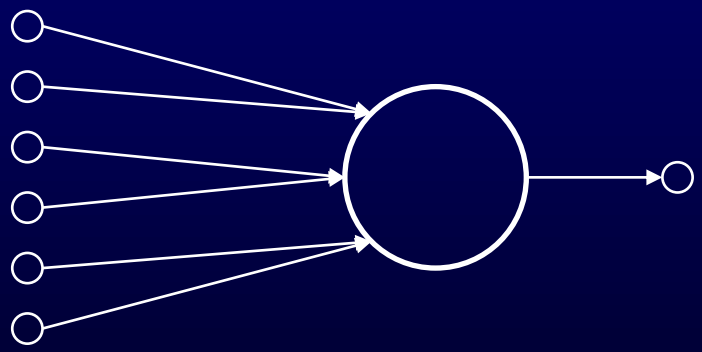
- Booleova funkce $f : \{0,1\}^n \rightarrow \{0,1\}^m$
- Orientovaný acyklický graf
  - Každý uzel se vstupním stupněm 0 je ohodnocen vstupní proměnnou x_i, $i=1..n$
  - Každý uzel s výstupním stupněm 0 je ohodnocen výstupní proměnnou y_i, $i=1..m$
  - Každý ostatní (vnitřní) uzel je ohodnocen hradlem ω_j ze zvolené báze Ω komutativních Booleových funkcí
- Počet vnitřních uzlů je složitost obvodu
- Délka nejdelší cesty grafem je hloubka obvodu
- Graf je stromem $\rightarrow$ Booleova formule

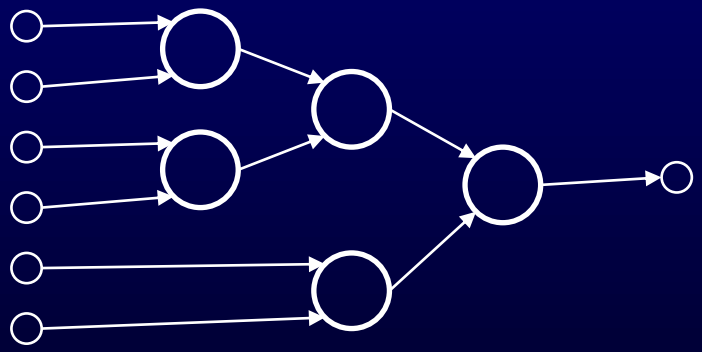
Báze

- Pro nekomutativní funkce bychom museli očíslovat vstupní hrany každého uzlu
- Báze je úplná, jestliže je možno s ní realizovat každou Booleovu funkci
- Některé úplné báze:
  - $\{\wedge,\neg\}$, $\{\vee,\neg\}$ AND-NOT, OR-NOT
  - $\{\neg\wedge\}$, $\{\neg\vee\}$ NAND, NOR
  - $\{\oplus,\neg\}$ GF(2)

Vstupní a výstupní větvení

- Vstupní stupeň vnitřního uzlu nazveme vstupní větvení (počet vstupů, fan-in)
- Výstupní stupeň vnitřního uzlu nazveme výstupní větvení (fan-out)
- Booleův obvod jako výpočetní model je robustní vůči omezení výstupního větvení
- Simulace vstupního větvení: logaritmická hloubka





Uniformní modely

- Turingův stroj řeší instance problému *libovolné velikosti* – uniformní výpočetní model
- Pro každou velikost instance potřebujeme jiný obvod:
  - povolíme zásobu předkonstruovaných obvodů → neuniformní model, výhoda proti Turingovu stroji
  - požadujeme, aby pro každou velikost instance bylo možno obvod efektivně vygenerovat Turingovým strojem → uniformní model

		Počet jedniček násobkem *m*		Počet jedniček alespoň *p*
P/poly polynomiální složitost	**$\{\wedge, \vee, \neg\}$**	**$\{\wedge, \vee, \neg,$ MOD *m*$\}$**	**$\{\wedge, \vee, \neg,$ MOD *m* $\forall m\}$**	**prahová funkce**
konstantní hloubka, vstupní větvení 2	NC^0			
konstantní hloubka	AC^0	$AC^0(m)$	ACC^0	TC^0
logaritmická hloubka, vstupní větvení 2	NC^1			

$$ACC^0 \subseteq TC^0 \subseteq NC^1$$

AC^0

- AC: Alternating Circuits – negace jen na vstupech, dále střídající se AND, OR $\rightarrow$ normální formy (DNF, CNF)
- AC^0 umí:
  - celočíselné sčítání
  - všechno, co lze popsat predikátovou logikou 1. řádu
- AC^0 neumí:
  - celočíselné násobení $\Rightarrow$ těžší než sčítání
  - paritu $\Rightarrow$ parita nemůže mít DNF, CNF polynomiální velikosti

Jeden z praktických důsledků

- Všechny Booleovy funkce nejvýše 2 vstupů:
  - Konstanty 0, 1
  - Opakovač a invertor
  - Funkce, které dostaneme z AND inverzí na vstupech nebo výstupu
  - Funkce, které dostaneme z XOR inverzí na vstupech nebo výstupu
- NPN třídy ekvivalence
- A když návrhový systém „neumí“ všechny třídy stejně? $\rightarrow$ výsledek větší, než má být

TC^0

- Prahová funkce: dává 1 od určitého počtu jedniček na vstupu výše – např. majorita
- TC^0 umí:
  - celočíselné násobení
  - modelovat neuronové sítě atd.

NC^1

- NC^1 umí:
  - vyhodnotit Booleovu formuli
  - rozpoznat regulární podmnožinu
- Existence problémů, které mají obvody v NC^1 ale nikoliv v TC^0, není jistá

Hloubka funkce

- Nechť
  - $f : \{0,1\}^n \to \{0,1\}$ nad bází $\{\wedge,\vee,\neg\}$ se vstupním větvením 2.
  - $B_0 \subset \{0,1\}^n$ jsou všechny vektory, pro které f nabývá hodnoty 0
  - $B_1 \subset \{0,1\}^n$ jsou všechny vektory, pro které f nabývá hodnoty 1
- Dva vzdálení, ale kooperující hráči A, B
  - A má vektor $a \in B_0$, B má vektor $b \in B_1$
  - mají najít pozici prvého bitu, ve kterém se jejich vektory liší
- Počet bitů, které si musí vyměnit, se rovná hloubce f.

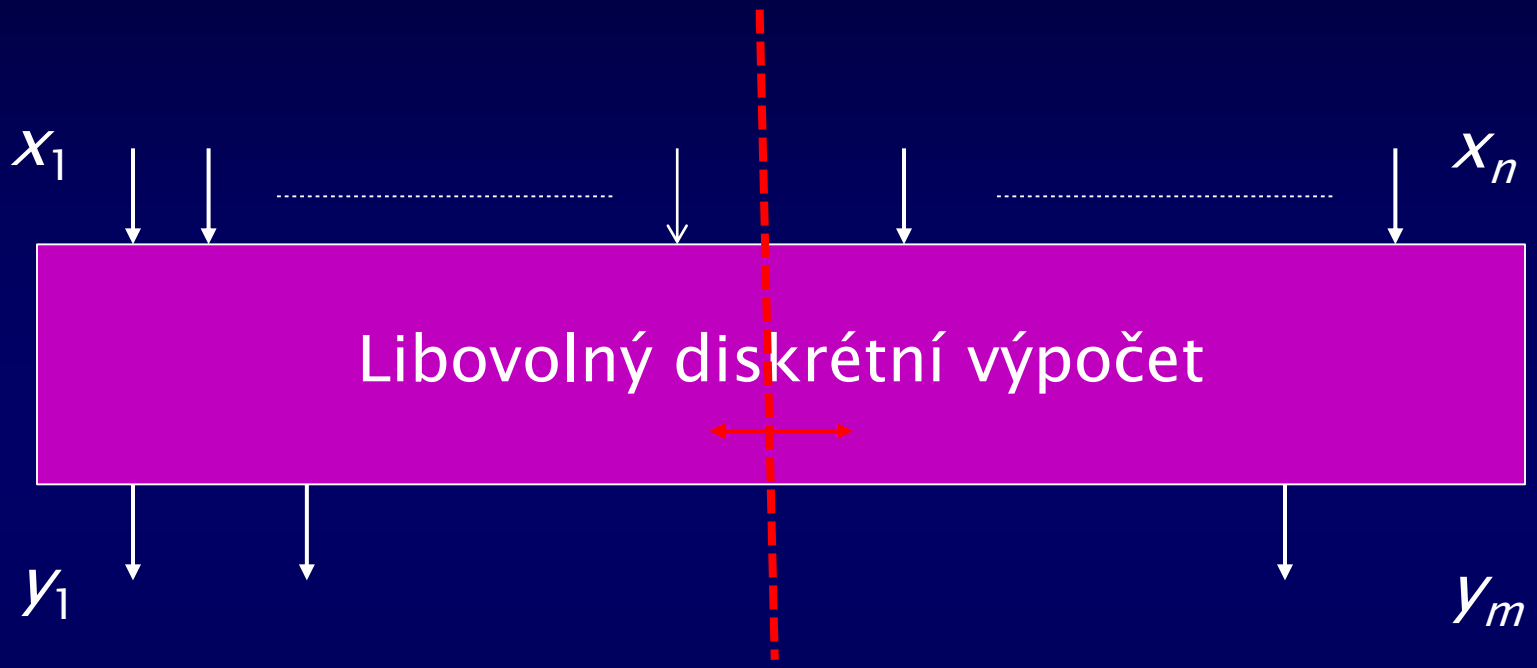
Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require superlogarithmic depth. *SIAM Journal of Computing, 3(2):255-265, May 1990.*

Důsledek

- Symetrické funkce: pravdivostní hodnota závisí jen na počtu vstupů v 1
- Symetrické funkce mají logaritmickou hloubku
- Důkaz: komunikační protokol

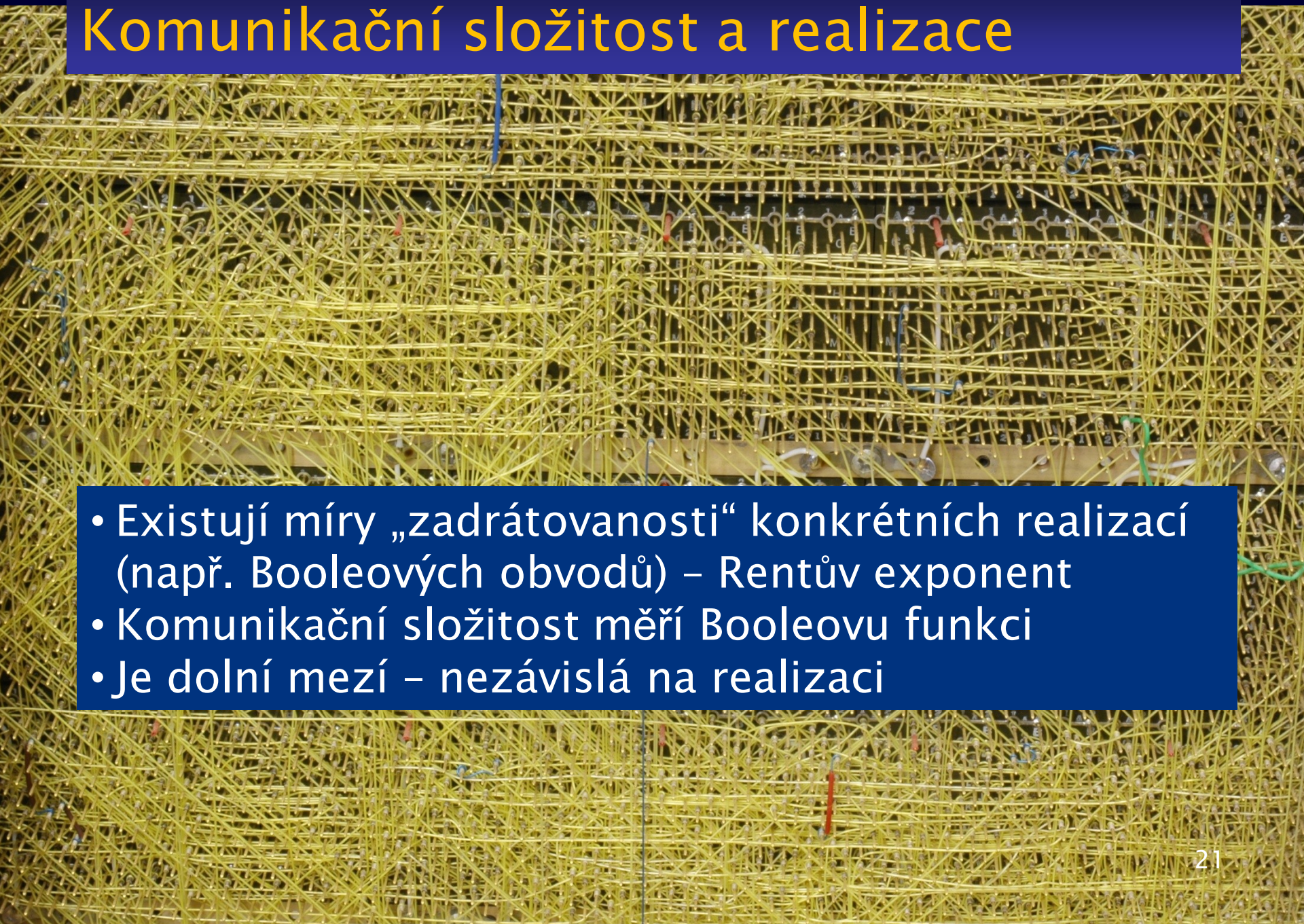
Brodal & Husfeldt: Symmetric Functions have Logarithmic Depth. Výzkumná zpráva BRICS RS961, Department of Computer Science University of Aarhus. 1996

Komunikační složitost



- Rozdělit vstupy na dvě stejné množiny (bez ohledu na výstupy) tak, aby počet bitů předávané informace pro provedení výpočtu byl minimální
- Počet bitů je (obousměrná) komunikační složitost

Komunikační složitost a realizace



- Existují míry „zadrátovanosti“ konkrétních realizací (např. Booleových obvodů) – Rentův exponent
- Komunikační složitost měří Booleovu funkci
- Je dolní mezí – nezávislá na realizaci

Složitost objektu

- V rozlišení 3200 × 2400 px je velikost 3.24 MB
- Program, který jej vygeneruje, je mnohem kratší
- Jak složitý je tento objekt?

Reguiieee at en.wikipedia

Kolmogorovova složitost

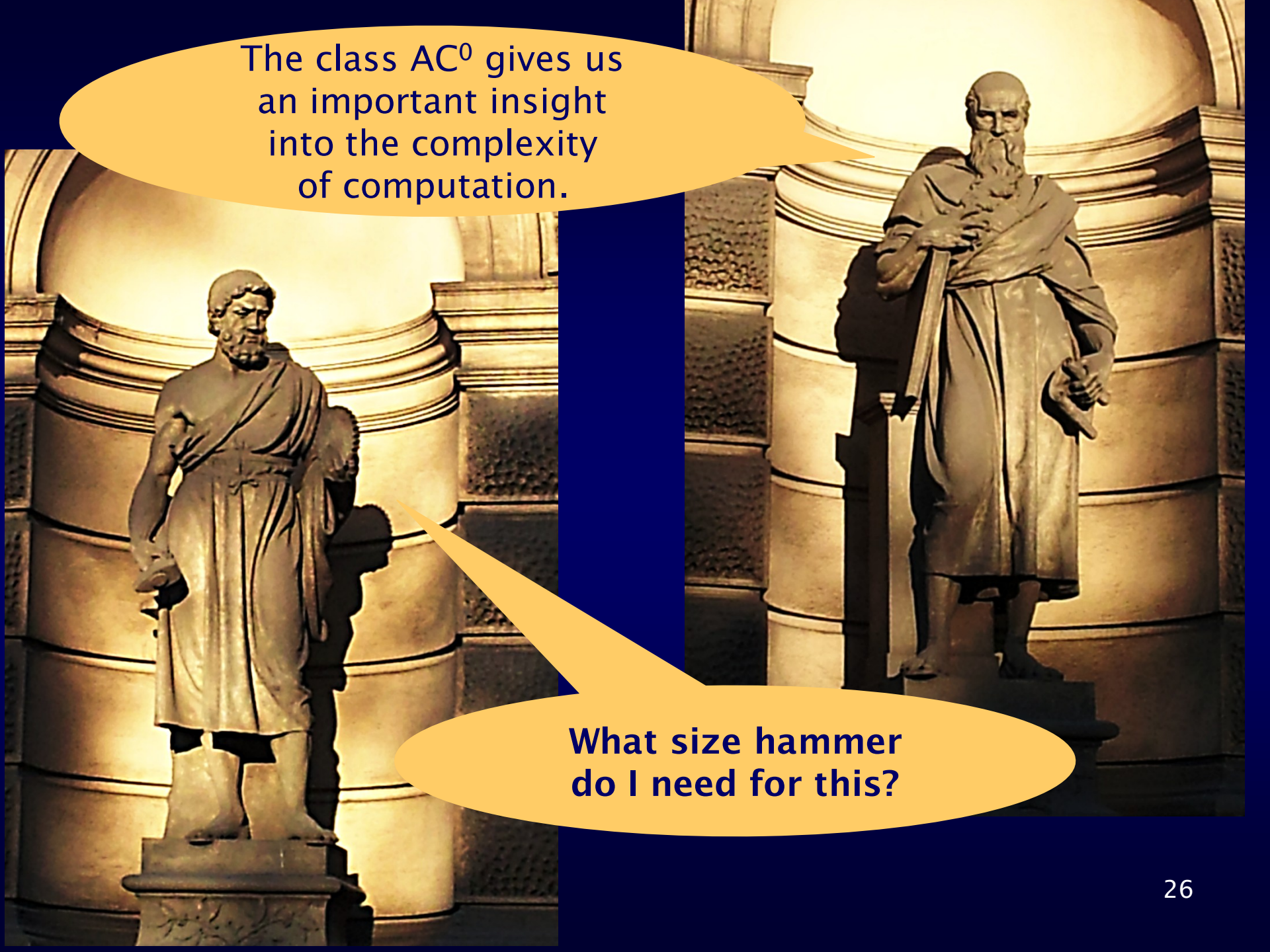
- Složitost popisu objektu vzhledem k jazyku (popisným prostředkům)
- Nechť
  - *w* je vstup Turingova stroje *M*, který vygeneruje objekt (řetěz) *s*
  - *m* je reprezentace *M* řetězem (program pro univerzální Turingův stroj)
- Pak zřetězení *w.m* je popisem *s*
- Délka nejkratšího takového zřetězení je Kolmogorovova složitost *s*

Vlastnosti

- Stanovení Kolmogorovovy složitosti je obecně nerozhodnutelný problém (je Turing-ekvivalentní s problémem zastavení Turingova stroje)
- Kolmogorovova složitost vzhledem ke dvěma různým strojům *M* se liší nejvýše o aditivní konstantu (velikost interpretu pro vzájemnou emulaci)
- Kolmogorovova složitost řetězu nemůže tedy být „o mnoho“ větší než jeho délka

Kolomogorovova (algoritmická) náhodnost

- Náhodný řetězec je ten, který je sám sobě nejkratším popisem (vzhledem k nějakému univerzálnímu Turingovu stroji)
- Pro každé *n* existuje alespoň jeden takový řetězec
- Mohou se lišit podle zvoleného univerzálního Turingova stroje


The class AC^0 gives us an important insight into the complexity of computation.
What size hammer do I need for this?