

JAK POČÍTAČE MYSLÍ?

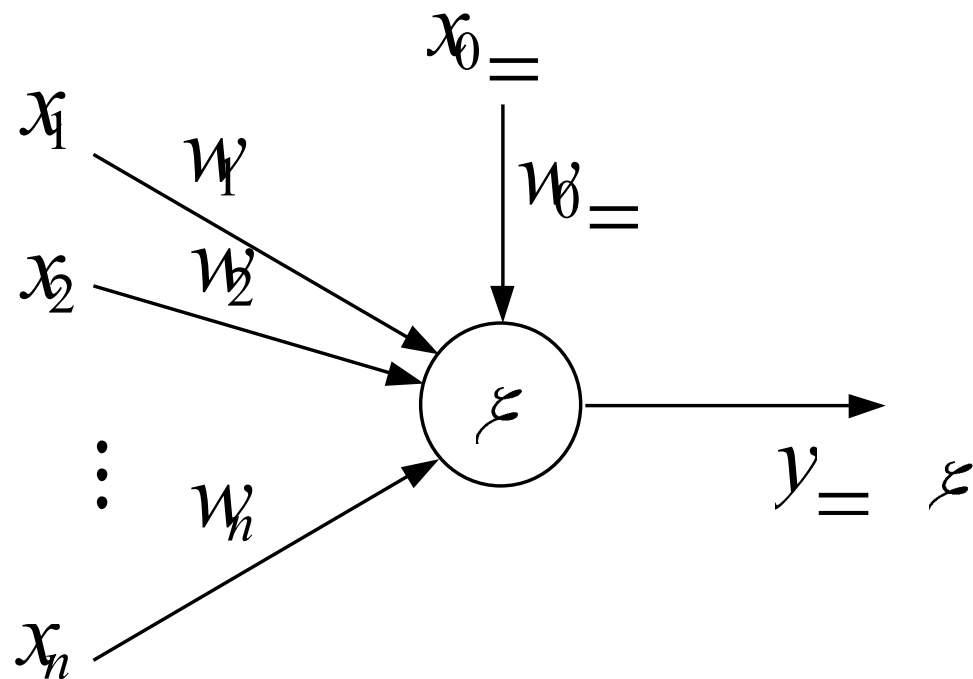
Michal Černý

Symbolický funkcionalismus

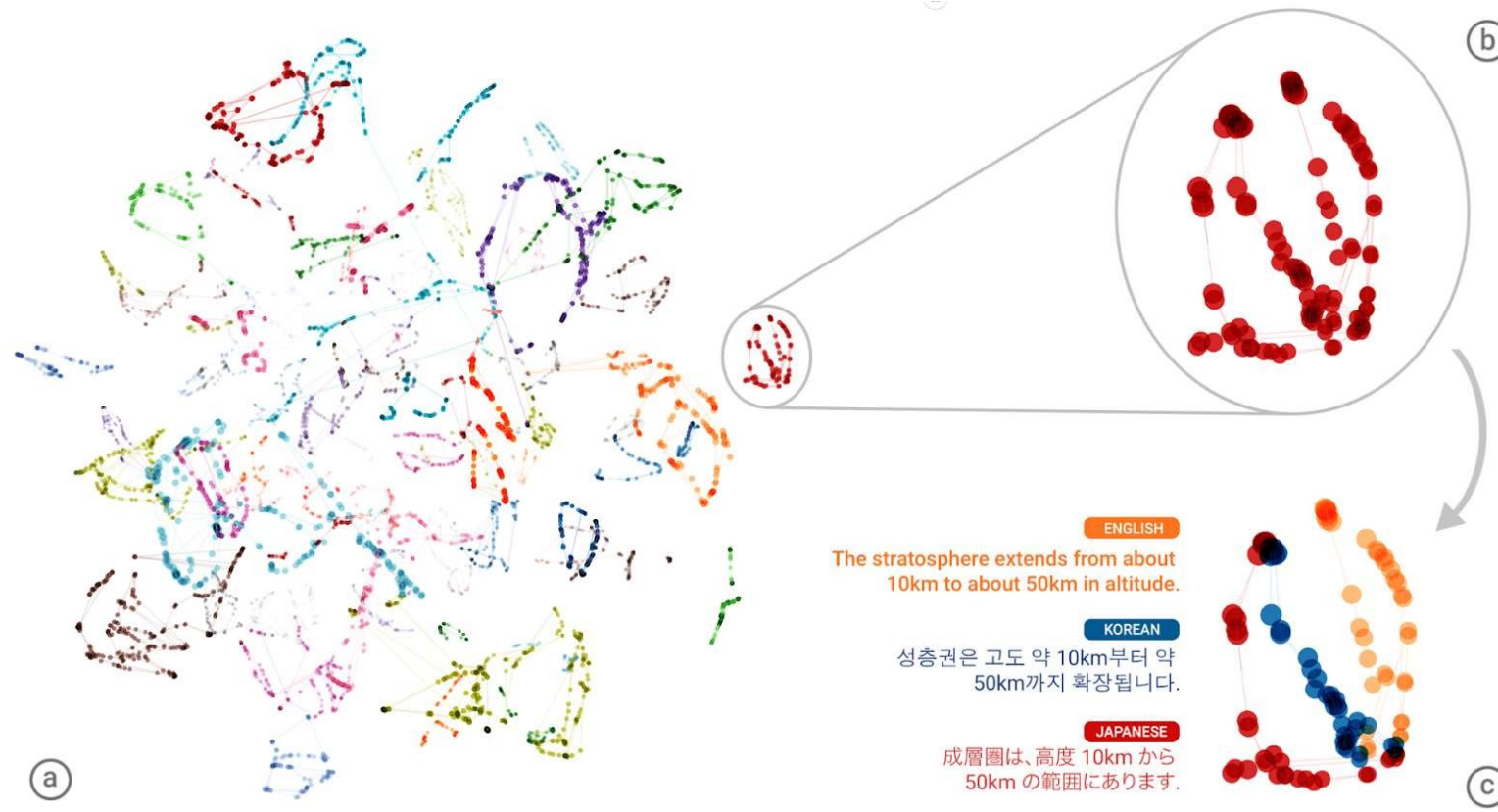
- *„Intelligentní chování daného systému je dosaženo interakcí mezi jednotlivými komponenty, které disponují odlišnou funkcionalitou, což je dosaženo tím, že v rámci systému hrají odlišnou roli.“*
- Existuje konečný automat (stroj), který posloupností kroků, která je jednoznačná dojde ke správnému výsledku (Turingův stroj) (podobnost s Carnotovým cyklem majícím ideální účinnost tehdy, když pracuje nekonečně pomalu)
- Tento přístup je klasickou formou AI

Neuronové sítě

- [Google vytvořil robotickou ruku, která se sama naučí, co má dělat](#)
- [AlphaGo definitivně porazil člověka, a jelikož nemá dalšího soupeře, Google program ukončí](#)
- [Neuronová síť Googlu sledovala BBC tak dlouho, dokud se nenaučila odezírat z úst](#)
- [Google Brain: Vědci z Kalifornie proti sobě postavili tři umělé inteligence. Začaly spolupracovat](#)
- [Neuronová síť Googlu si vytvořila vlastní jazyk, kterým si pomáhá v překladech](#)
- [Chatbot jako psychoterapeut. Počítačový program pomáhá uprchlíkům](#)
- [Google uvolnil kód neuronové sítě. Snít může i vaše PC](#)
- [TEGA robot pro vzdělávání](#)
- [Čínská cesta k umělým učitelům](#)



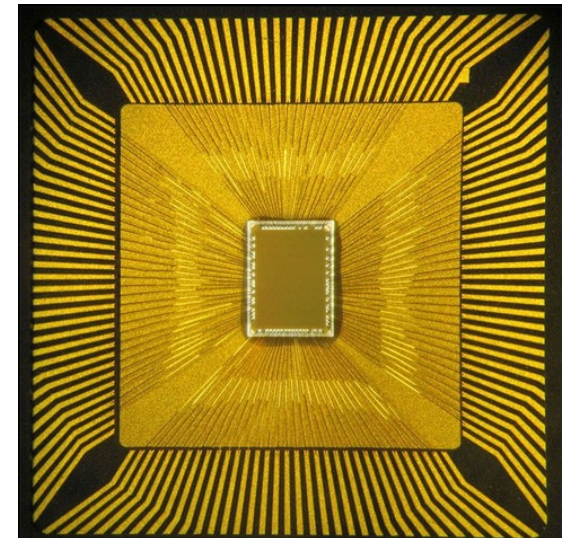
Co je jazyk? Geometrie!



Konekcionismus

- Výpočty získáme spojením jednoduchých objektů s výpočetní silou do sítě
- Představa sítě jako mozku – neurony a synaptický spojení
- Pracuje se s tzv. neuronovou sítí – každý uzel má určitou (většinou všechny stejnou) množinu operací, které umí a dohromady tvoří umělou inteligenci
- Příklad SyNAPSE – čip od IBM, který se umí sám učit (např. natáčet pátku v pin-pongu) – 265 neuronů a 65536 nebo 262144 synapsí

- . Inside IBM's cognitive chip. *Nature*. 2011-8-18, s. -. DOI: 10.1038/news.2011.486. Dostupné z: <http://www.nature.com/doi/finder/10.1038/news.2011.486>

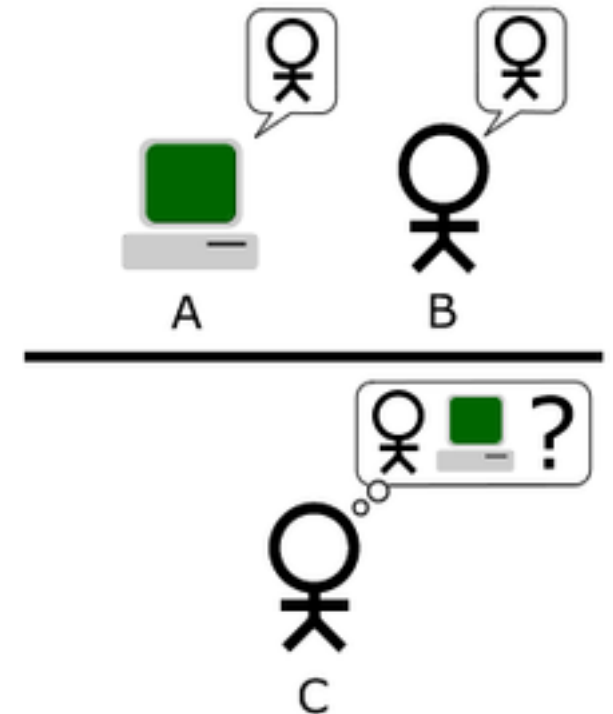


Robotický funkcionalismus

- Jako inteligentní chování je zde chápáno jako rozumná interakce mezi třemi entitami: *system, prostředí, úloha*
- Vychází tedy z myšlenek behaviorismu
- Intelligence je chápána jako instrumentální dovednost řešit nějakou úlohu
- Příklad: inteligentní umělé domácí roboti, zdravotnické systémy, výrobní linky a stroje....

Turingův test

- Umíme rozeznat člověka od počítače v běžné řeči?
- Historicky známé přístupy:
 - ELIZA Josepha Weizenbaum
 - Chatterboot (v česku například Pokec)
 - Botnet na Facebooku z Vancouveru
- <http://nlp-addiction.com/eliza/> a <http://alice.pandorabots.com/>
- Cena 100 000 dolarů pro první nerozpoznatelný počítač nebyla udělena.



Problém čínského pokoje

- *Searl, jenž nerozumí ani slovo čínsky se usadí v uzavřené místnosti plné knih, a návodů jak reagovat na jakoukoliv otázku v čínštině. Dejme tomu že v libovolném okamžiku, když dostane Searl vzkaz napsaný čínsky, dokáže pomocí knih a návodů zareagovat v čínštině. Není problém si představit konverzaci s Číňanem stojícím před pokojem a strkajícím si papírky na relativně velmi omezené téma. Toto téma lze samozřejmě nekonečně zobecňovat, až dojdeme k původnímu požadavku.*

TECHNICKÁ ŘEŠENÍ

Turingův stroj

- Na začátku výpočtu je Turingův stroj v počáteční konfiguraci a na pásce je zapsané vstupní slovo. Dále pracuje v jednotlivých krocích:
- pokud je aktuální stav zároveň stavem koncovým, výpočet končí
- čtecí hlava přečte jeden vstupní symbol z buňky, na které se právě nachází
- pokud je v přechodové funkci pro aktuální stav a pro přečtený symbol definovaný přechod, provede se (v případě více možných přechodů u nedeterministických strojů se vybere jeden náhodně):
 - změní se stav
 - na aktuální pozici hlavy se zapíše příslušný symbol
 - hlava se příslušným způsobem posune (či neposune)

Definice

Formálně je **Turingův stroj** definován jako šestice

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ kde:

- Q je konečná množina **stavů**
- Γ je konečná množina **páskových symbolů**
- $\Sigma \subseteq \Gamma, \Sigma \neq \emptyset$ je konečná množina **vstupních symbolů**
- $\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$ je **přechodová funkce**
- $q_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

Logické programování

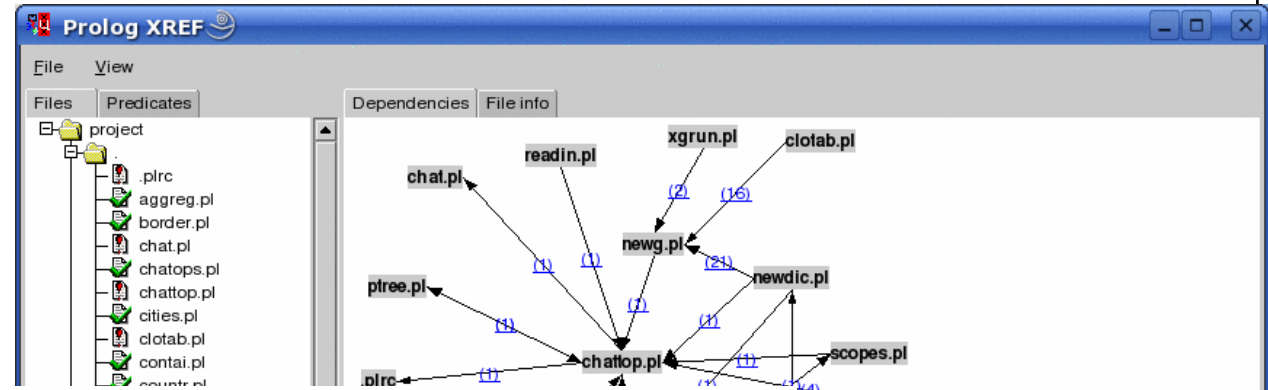
- Neprogramujeme postup řešení ale jen logická pravidla
- Program podle nich provádí jen logický důkaz
- Používá se Prolog nebo Gödel
 - Fakt: dívka(monika).
 - Otázka: ?- dívka(monika).
 - Odpověď: yes.
- Podporované možnosti: seznamy, pole, proměnné, řetězce, složitější struktury
- Základní myšlenka: musíme vytvořit databázi faktů a pravidel, ze kterých se pak vyvozuje nějaká informace

Prolog

```
c:\program files\win-prolog 4500\examples\salesman.pl
% initialise data, prepare graphics objects, and create the dialog

salesman :-
    tidy_salesman,
    init_salesman,
    Ds = [ws_caption,ws_maximizebox,ws_thickframe],
    Bs = [ws_child,ws_visible,ws_tabstop,bs_pushbutton],
    Ss = [ws_child,ws_visible,ss_left],
    Gs = [ws_child,ws_visible,ws_ex_clientedg
wcreate( dlg, `Travelling Salesman`,
wcreate( (dlg,3), button, `&Exhaustive`,
wcreate( (dlg,4), button, `&Heuristic`,
wcreate( (dlg,5), button, `&Stop`,
wcreate( (dlg,6), button, `&Close`,
wcreate( (dlg,8), static, ``,
wcreate( (dlg,9), grafix, ``,
set_buttons( 0, 0, 0, 1 ),
town_grafix,
window_handler( dlg, salesman_handler ),
call_dialog( dlg, _ ),
tidy_salesman.
```

Prolog Source S C O R=481 C=29 L=26556 S=0



Java - LiterateZest/src/parser.pl - Eclipse SDK

```
parser.pl  semiotics.textile  Node.java  input.dot
```

```
S --> np, vp.
np --> det, n.
np --> n.
vp --> v, n.
vp --> v.
det --> [the].
det --> [a].
n --> [agent].
n --> [martinis].
v --> [likes].
v --> [drinks].

/* This grammar describes sentences like "the agent likes martinis":
digraph the_agent_likes_martinis

S; NP; V; VP; DET; N1[label="N"]; N2[label="N"]
the; agent; likes; martinis

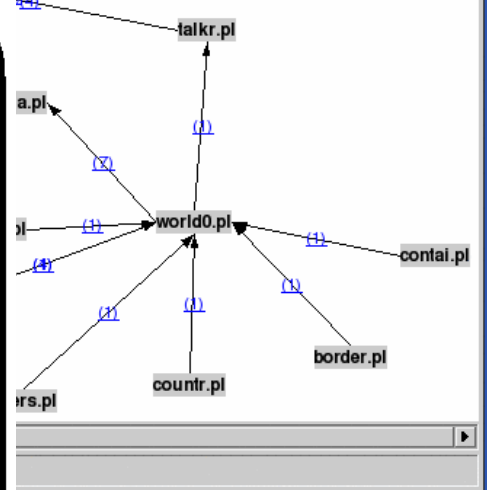
S -> NP; NP -> DET; NP -> N1
DET -> the[style=dashed]; N1 -> agent[style=dashed]

S -> VP; VP -> V; VP -> N2
V -> likes[style=dashed]; N2 -> martinis[style=dashed]
```

Zest Graph

The Zest Graph shows a parse tree for the sentence "the agent likes martinis". The root node is 'S', which branches into 'NP' and 'VP'. 'NP' branches into 'DET' (the) and 'N' (agent). 'VP' branches into 'V' (likes) and 'N' (martinis). The leaf nodes are 'the', 'agent', 'likes', and 'martinis'.

Writable Insert 34 : 2

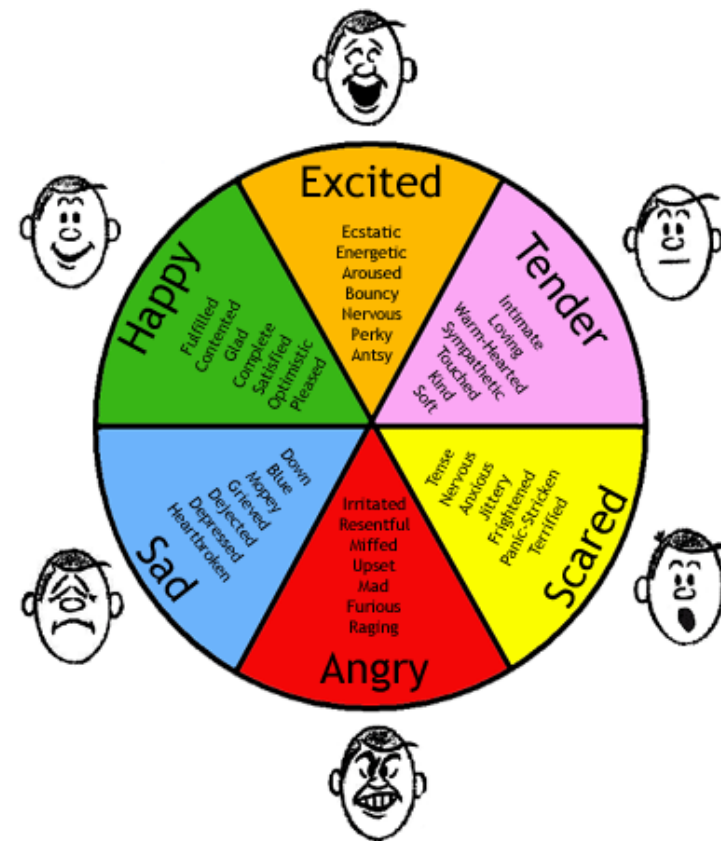


Další technické možnosti

- Genetické programování (Vytvoříme populaci entit a testujeme jejich chování. V druhém kroku vybereme ty nejlepší a snažíme se z nich vygenerovat novou nakříženou populaci. To opakujeme dokud nemáme dostatečně dobré řešení)
- Expertní systémy
- Dobývání znalostí (analýza obrazových a textových dat, získávání informací, které nejsou standardně dostupné přímo)
- Strojové učení (založené na statistických metodách, často se kombinuje s dalšími formami)

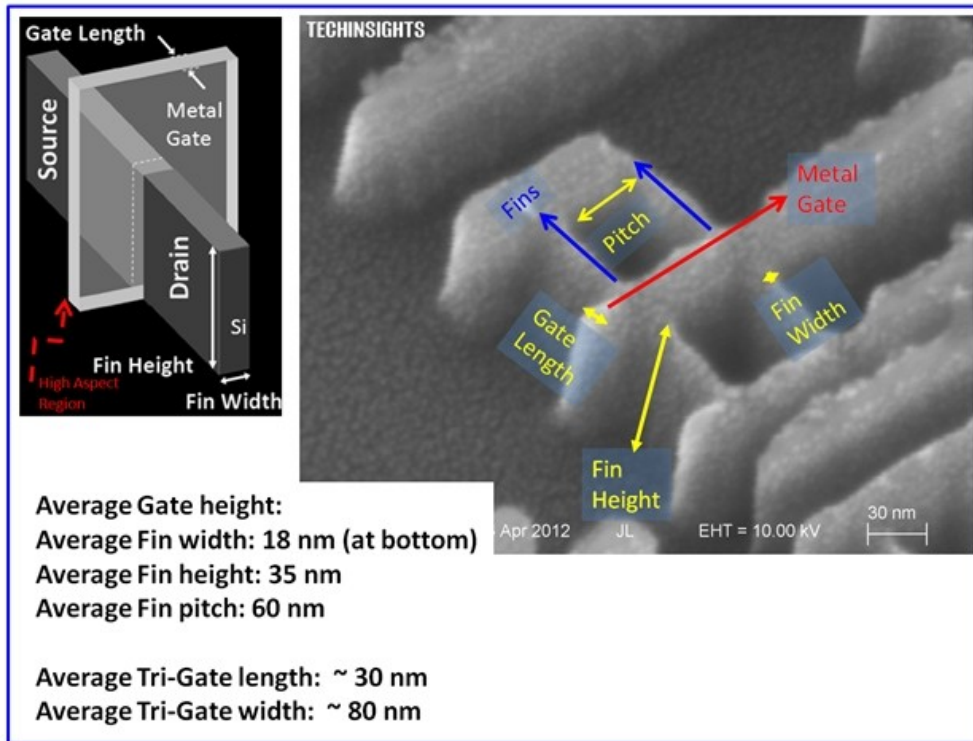
Počítačové zpracování emocí

- Člověk není jen racionální bytost, ale má také emoce, které jsou důležité pro pochopení obsahu (například ironie)
- Analýza emocí:
 - Z hlasu
 - Z fyziologických projevů (mrkání, tlak, teplota, galvanický odpor kůže,...)



KVANTOVÉ POČÍTÁNÍ

Kvantová mechanika v běžných procesorech



- Intel 2012: technologie 22 nm (procesor i5 Ivy Bridge)
- Technologická bariera: příliš krátká gate – překonána ploutvemi (fins)
- Ale elektrony se stále chovají jako nabitě kuličky

Proč kvantové počítače

- Efektivně lze řešit jen úlohy, které mají nejvýše kvadratickou složitost
- Kvantová mechanika umožňuje nový způsob práce s výpočty, takže lze změnit některé exponenciální problémy na lineární nebo kvadratické
- Typické výpočty:
 - Výpočet Fourierovy transformace v n -rozměrném prostoru
 - Black box problémy
 - Odhady Gaussovy sumy
 - Šachy
 - ...

qubit

- $|\mathbf{u}\rangle = A |\mathbf{1}\rangle + B |\mathbf{0}\rangle$,
kde $|\mathbf{u}\rangle$ je stav qubitu, A a B jsou kompletní čísla udávající pravděpodobnost stavu $|\mathbf{1}\rangle$ respektive $|\mathbf{0}\rangle$, která jsou normována na jedničku
- Během výpočtů může být $|\mathbf{u}\rangle$ jedna nebo nula, ale také cokoli mezi tím
- Až měření dává výsledek
- Algoritmus typicky není možné zkoumat „zevnitř“ ale jen analyzovat vstupy a výstupy

Realizace dvoustavového qubitu

- Spin elektronu
- Excitovaný vodíkový iont (dodáme energii právě nutnou k excitaci – pak je pravděpodobnost 1:1 že k ní dojde nebo ne)
- Polarizace fotonů

- Budoucnost? Více stavové quibity

- Současné zařízení: D-Wave One, který obsahuje 128qubitový procesor. Chlazení pomocí tekutého hélia. Drahé pomalé, špatně programovatelné,... [URL](#)

DĚKUJI ZA POZORNOST
