



Umělá intelligence

PSY 481

Alan Turing



- Turingovo jméno pravděpodobně vybaví především ve dvou ustálených spojeních: Turingův stroj a Turingův test.
- pokus o matematické zachycení intuitivního pojmu vypočitatelnosti či ještě obecněji vyřešitelnosti
- Turing byl přesvědčen, že lidský mozek nemůže být ve své podstatě nic jiného než jakýsi (nesmírně komplikovaný) druh počítače.

Teorie informace



V roce 1948 publikoval Claude Shannon společně s matematikem Warrenem Weaverem článek „A mathematical theory of communication“.

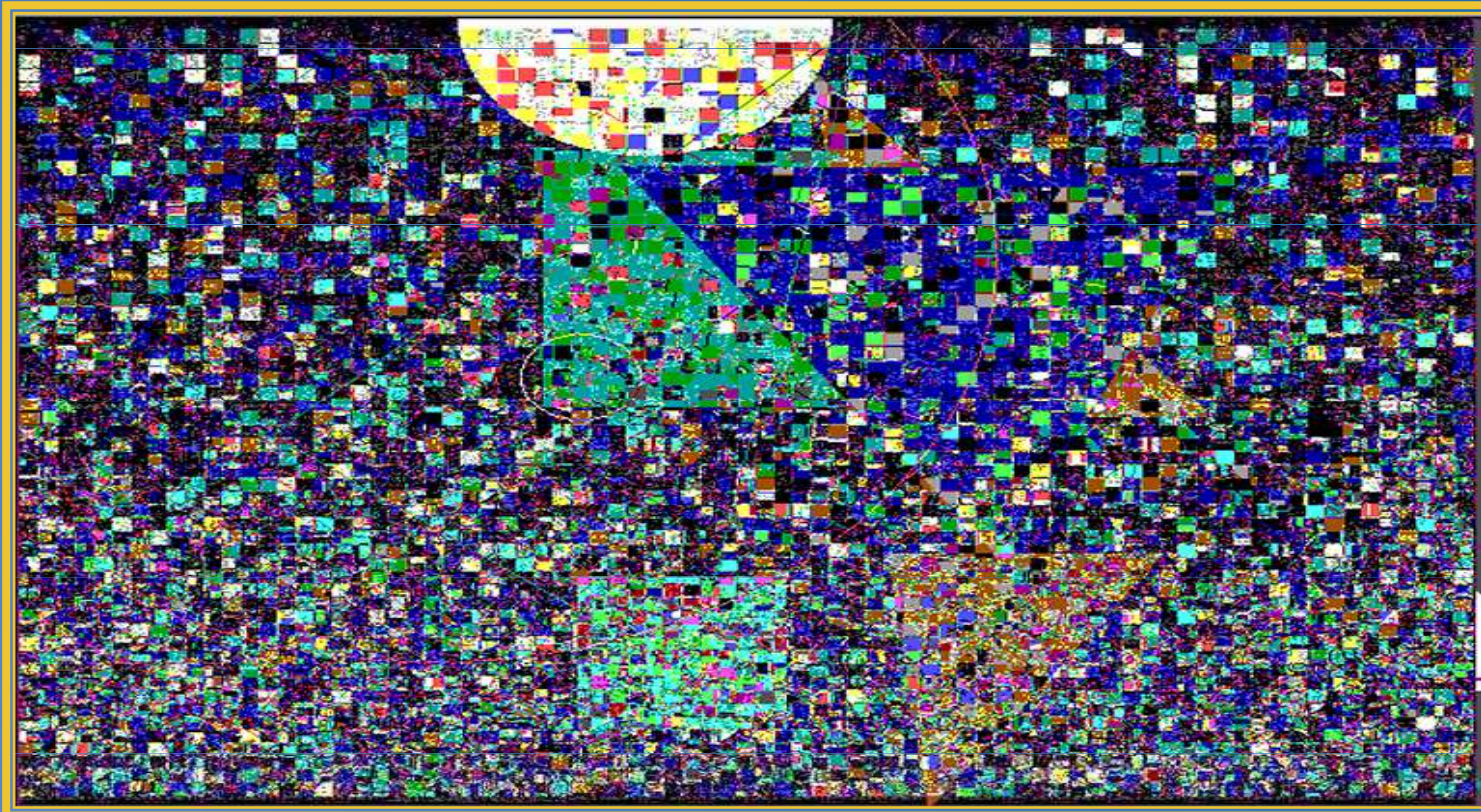
Norbert Wiener se k pojmu informace vyjádřil poněkud metaforicky: „Mechanický mozek neprodukuje myšlení „jako játra žluč“, jak si mysleli první materialisté, stejně jako jej neprodukuje ve formě energie jako svaly aktivitu. Informace je informace, ne hmota nebo energie.“

V samotné práci pak (stejně jako Shannon) hovoří o informaci jako o opaku entropie.

Teorie informace



Nehmotná informace je pevně vázána na fyzikální svět hmoty a energie a že každý přenos či záznam informace vyžaduje disipaci jisté energie, a tedy vzrůst termodynamické entropie.



Filozofické pojetí informace

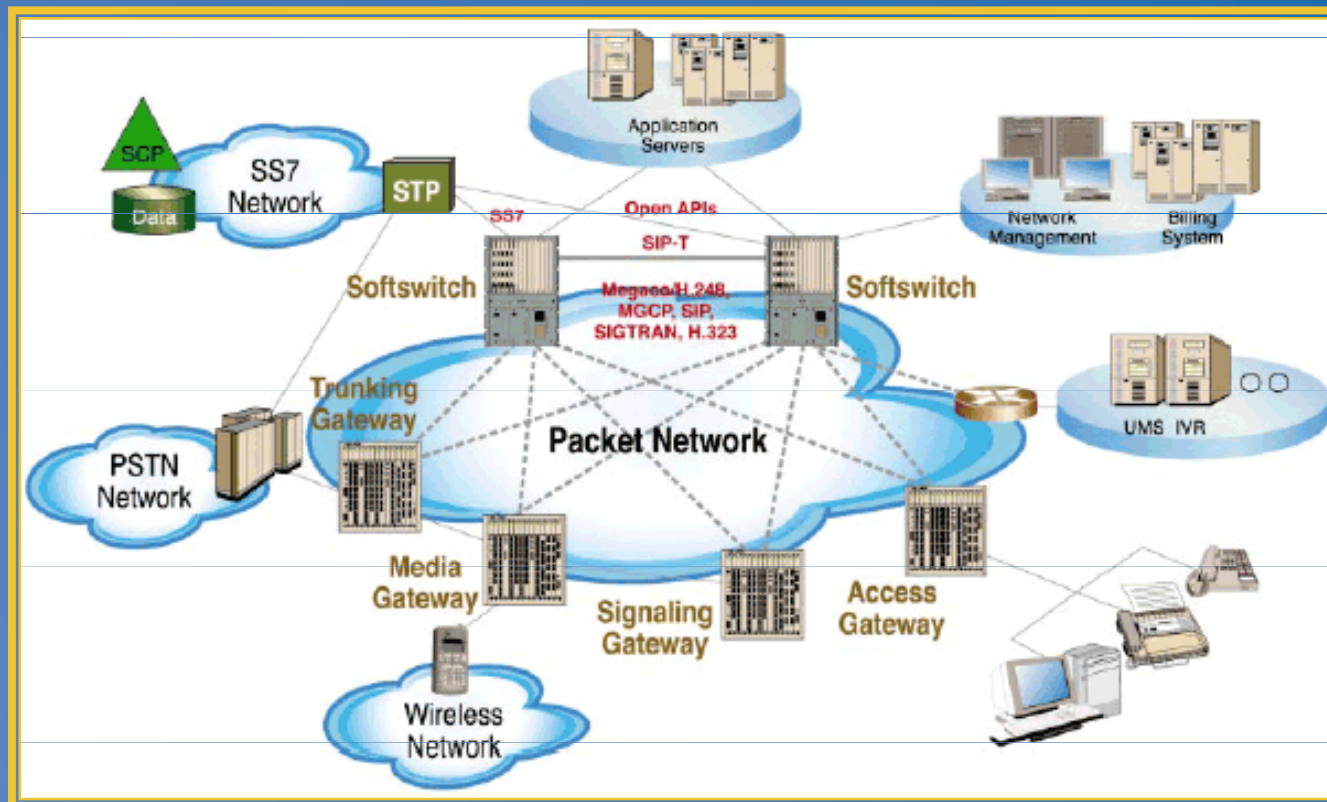


- Vlastnost hmotné reality být uspořádaná a její schopnost uspořádávat (forma existence hmoty vedle prostoru, času a pohybu).
- Význam přiřazený obrazům, údajům a z nich utvořeným lidským celkům. Informace představuje míru uspořádanosti systémů na rozdíl od entropie, tj. míry neuspořádanosti.

Komunikační pojetí informace



- Objektivní obsah komunikace mezi souvisejícími hmotnými objekty, projevující se změnou stavu těchto objektů.



Kybernetické pojetí informace



- Název pro obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním. Proces přijímání a využívání informace je procesem našeho přizpůsobování k nahodilostem vnějšího prostředí a aktivního života v tomto prostředí.
- Proces, kdy určitý systém předává jinému systému pomocí signálů zprávu, která nějakým způsobem mění stav přijímacího systému.

Matematický přístup k informaci



- Energetická veličina, jejíž hodnota je úměrná zmenšení entropie systému.
- Poznatek, který omezuje nebo odstraňuje nejistotu týkající se výskytu určitého jevu z dané množiny možných jevů.
- Obsah zprávy, který je definován jako záporný dvojkový logaritmus její pravděpodobnosti.

Obecná teorie informace



V nejobecnějším slova smyslu se informací chápe údaj o reálném prostředí, o jeho stavu a procesech v něm probíhajících. Informace snižuje nebo odstraňuje neurčitost systému (např. příjemce informace); množství informace je dáno rozdílem mezi stavem neurčitosti systému (entropie), kterou měl systém před přijetím informace a stavem neurčitosti, která se přijetím informace odstranila. V tomto smyslu může být informace považována jak za vlastnost organizované hmoty vyjadřující její hloubkovou strukturu (varietu), tak za produkt poznání fixovaný ve znakové podobě v informačních nosičích. V informační vědě se informací rozumí především sdělení, komunikovatelný poznatek, který má význam pro příjemce nebo údaj usnadňující volbu mezi alternativními rozhodovacími možnostmi. Významné pro informační vědu je také pojetí informace jako psychofyziologického jevu a procesu, tedy jako součásti lidského vědomí. V exaktní vědě se např. za informaci považuje sdělení, které vyhovuje přísným kritériím logiky či příslušné vědy. V oblasti výpočetní techniky se za informaci považuje kvantitativní vyjádření obsahu zprávy. Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami (0, 1) a vyjadřuje se jednotkou nazvanou bit.

Komputace



- Kolem roku 1950 se začínají objevovat nové koncepce způsobu nazírání lidské bytosti.
- Člověk je (metaforicky) viděn jako stroj a nastává i určitý posun v terminologii používané k popisu kognitivních procesů.
- Lidé jsou přirovnáváni k výpočetnímu zařízení, které se rodí s určitým hardwarem a je programováno zkušenostmi, socializací a zpětnou vazbou svého vlastního chování.
- Cílem psychologie je zjistit způsob, jakým lidé zpracovávají informace.
- Behavioristický model S-R se ukazuje jako nedostačující a pozornost se přesouvá k interním procesům a stavům.

Komputační teorie



Komputace se dá vyjádřit třemi samostatnými pojmy:

1. Matematická funkce
2. Algoritmus
3. Systémová architektura

Podobné rozdělení nacházíme i u Davida Marra. Ten rozděluje komputační teorii do 3 vrstev:

1. komputační
2. algoritmická
3. hardware

Marrova teorie



Komputační vrstva (není přesně ztotožnitelná přímo se slovem komputace – výpočet (proces). Marr jí vidí spíše jako otázku, co systém vykonává, než jak to vykonává. Komputační vrstva poskytuje abstraktní formulaci zpracovávané úlohy, včetně možností a omezení, které vstupují do hry

Algoritmická vrstva bere tyto informace v úvahu a v konkrétní rovině se snaží o tvorbu správných posloupností operátorů.

Hardware - tím, že zastřešujícím principem je komputační teorie, umožňuje být algoritmické vrstvě částečně nezávislá na použitém hardwaru. Přesněji architektura hardwaru musí být tak univerzální, aby na něm bylo možno provést libovolnou výpočetní úlohu.

Church-Turingova teze

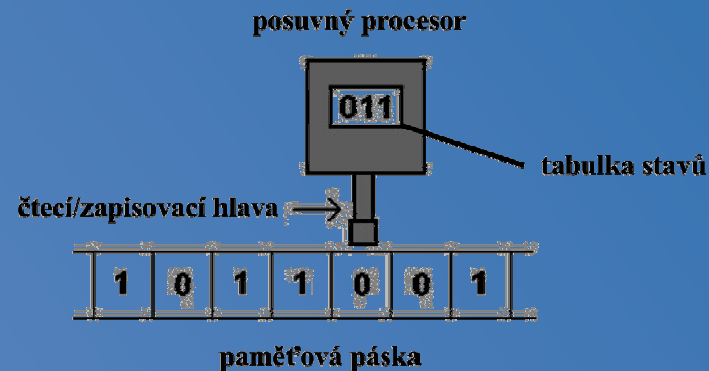


- Libovolný proces, který můžeme nazvat jako efektivní procedura může být realizován pomocí Turingova stroje. (Minsky, 1963, p.108)
- Všechny počítačové modely jsou stejné nebo méně výkonné než Turingův stroj (Luger, 1994).
- Složitost či efektivnost algoritmu je prokazatelná tím, jak ji lze provést Turingovým strojem (Crane, 2002).

Schéma Turingova stroje



Jedná se o zařízení, které obsahuje tabulku s konečným počtem fyzických nespecifikovaných stavů a posuvnou hlavu schopnou číst, zapisovat a mazat symboly (nejčastěji se používá 1 a 0, ale je možno použít libovolnou konečnou abecedu symbolů). Hlava se pohybuje v diskrétních krocích po libovolně dlouhé pásce (může být nekonečná), která je rozdělena na políčka obsahující vždy jeden symbol. Na začátku každého kroku ovlivňují činnost stroje dva vstupy. Jeden z pásky (symbol) a druhý z tabulky stavů, která dle daného symbolu přiřadí hlavě operaci, kterou má provést. Výsledná operace se skládá z instrukce, co provést s přečteným symbolem (nechat, smazat, přepsat) a určením směru posunu hlavy vlevo nebo vpravo.



Turingův stroj



•Current	•State 1			•State 2		
Char	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



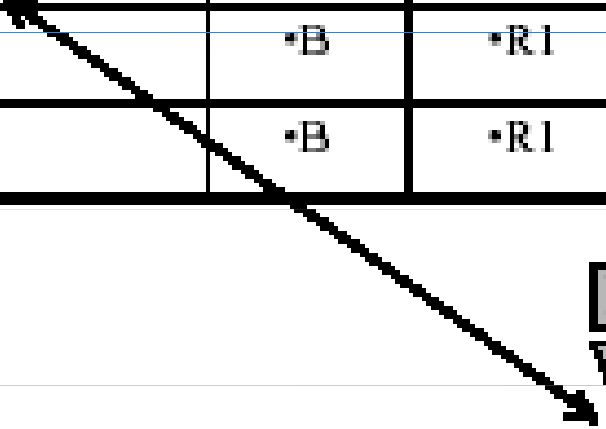
•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Action	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



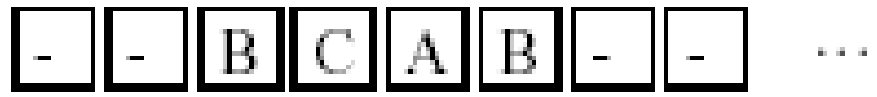
- - B C A B - - ...



•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•-	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2

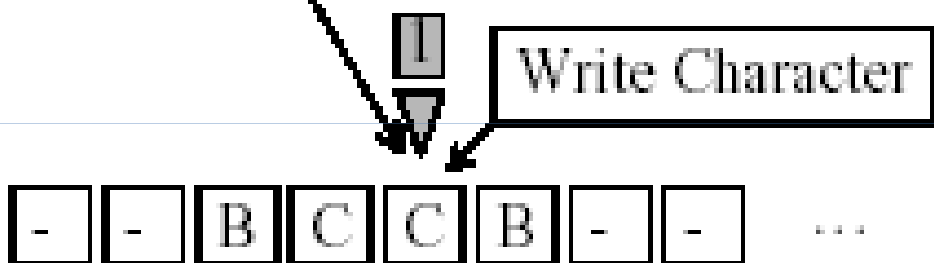


Read Character



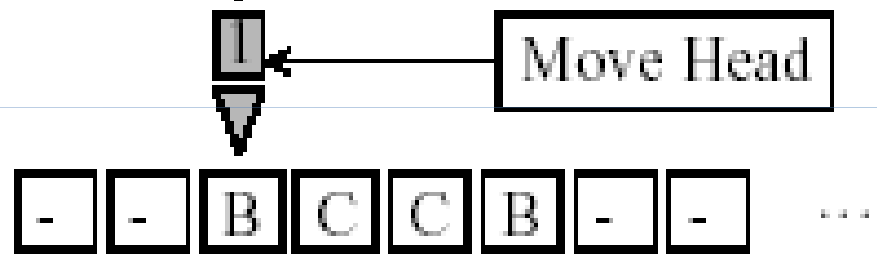


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



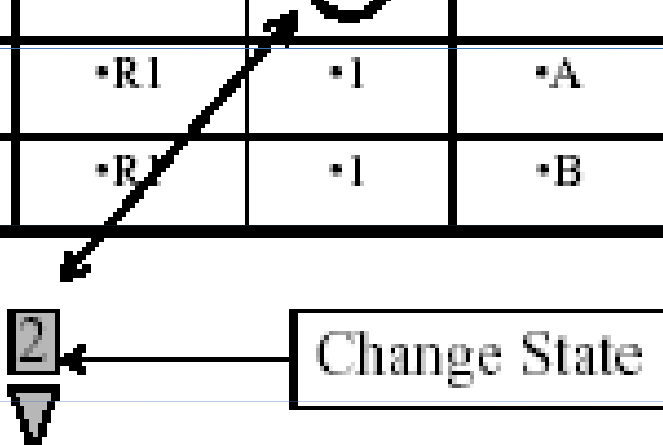


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Action	•State
•-	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2





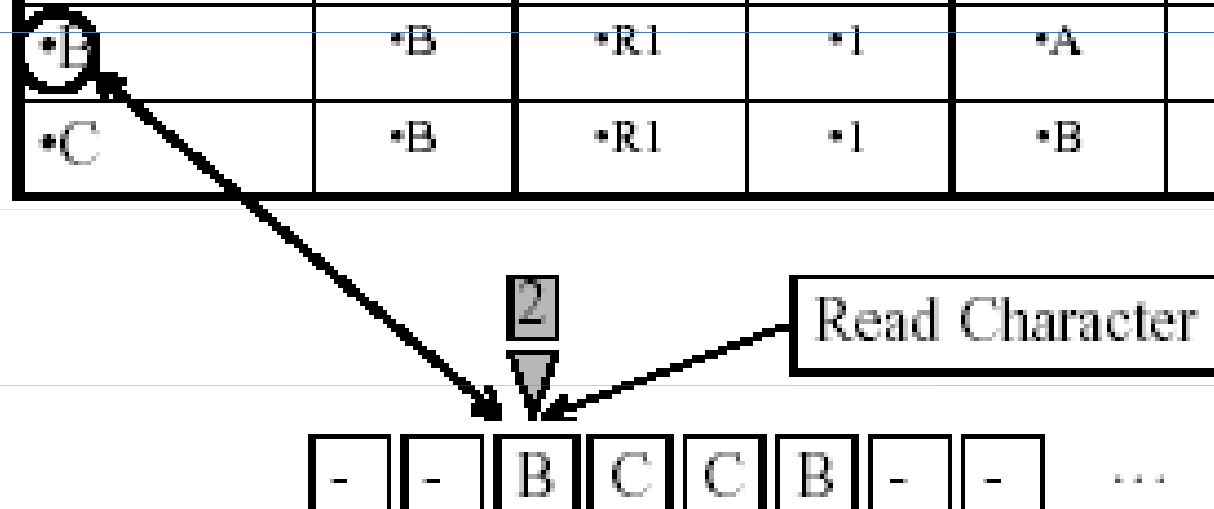
•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



- - B C C B - - ...

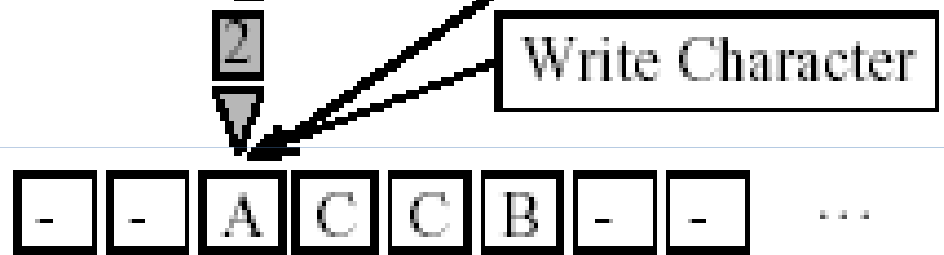


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



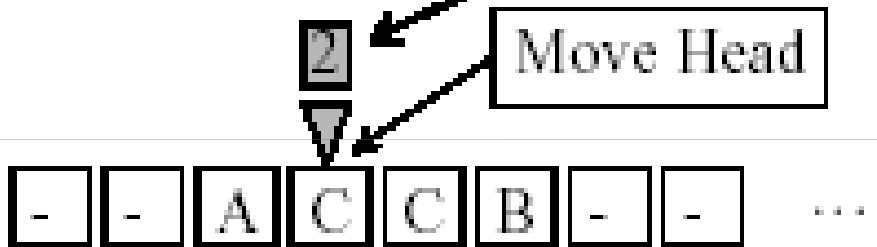


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Action	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



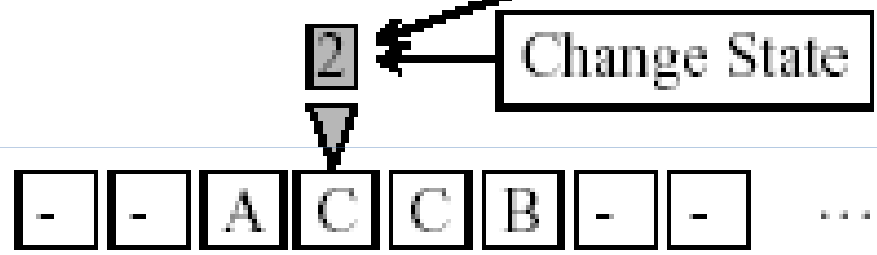


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



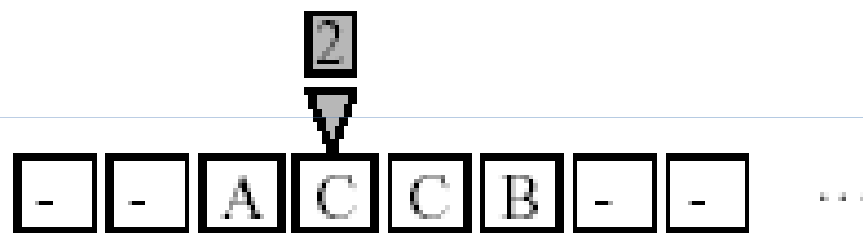


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



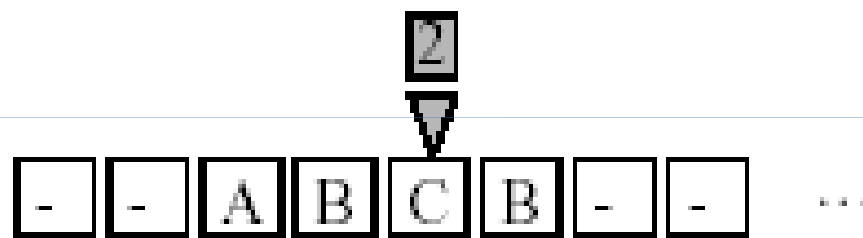


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



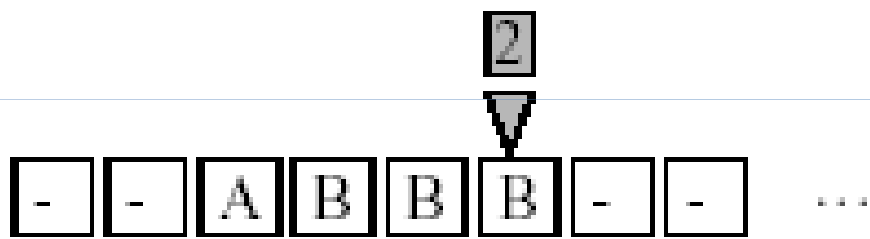


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Actio	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



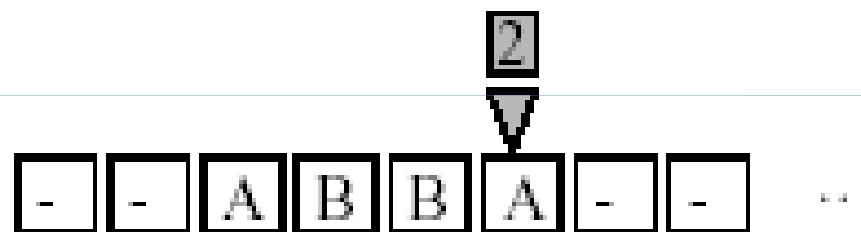


•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Action	•State
•-	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2





•Current Char	•State 1			•State 2		
	•Write	•Action	•State	•Write	•Action	•State
•_	•A	•L1	•2	•Halt	•Halt	•Halt
•A	•C	•L2	•2	•0	•R1	•2
•B	•B	•R1	•1	•A	•R1	•2
•C	•B	•R1	•1	•B	•R1	•2



Turingův stroj



- - A B B A - - ...

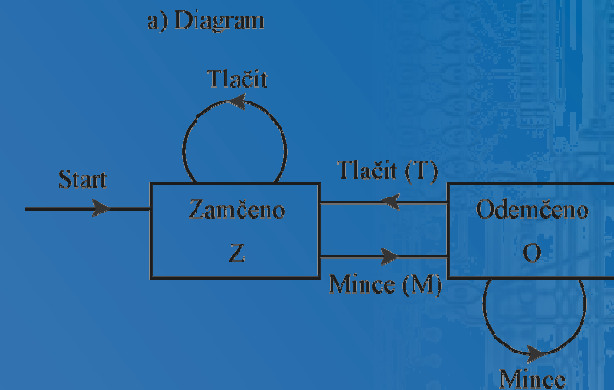
Finite state automata



Speciálním případem Turingova stroje jsou konečné automaty (finite state automata). Přesněji se jedná o automaty s konečným počtem stavů. Zastupují jednoduchou formu výpočetního zařízení, jejichž výchozím principem je právě Turingův stroj.

Základní definice konečných automatů se dá shrnout do těchto bodů:

1. Počet stavů automatu je diskrétní a přesně rozlišitelný.
2. Počet stavů je konečný.
3. Vstupy a výstupy probíhají v libovolném z těchto stavů.
4. Neustále probíhá přechod mezi jednotlivými stavy
5. Systémy nemají žádnou externí paměť. Veškerá interní paměť jsou pouze stavy a jejich posloupnosti. To, jak stavy budou přecházet, je částečně dáno jejich obsahy, ale také informacemi které do tohoto systému vstupují



b) Tabulka stavů

Počáteční stav	Cílový stav	
	Vstup T	Vstup M
Z	Z	O
O	(Otočit) Z	Z

Architektura počítačů



- Britský matematik a vynálezce Charles Babbage se již ve 20. letech 19. století pokusil zkonstruovat mechanický výpočetní stroj, jehož činnost byla založena na programovatelných instrukcích.
- Babbage se pokusil o obnovení myšlenky, kterou se zabýval už Leibnitz.
- Leibnitz uvažoval o mechanickém uvažování jako o rozšířeném mechanickém počítání, ale nepodařilo se mu najít vhodný jazyk pro reprezentaci okolního světa.
- Babbagovi se to povedlo s tím, že použil Booleovskou algebru.
- Chtěl postavit takový přístroj, který by byl schopen počítat na aritmetické bázi.
- Přístroj měl navíc v sobě obsahovat prvky logické algebry, umožňující v mnohém napodobit lidské myšlení.
- Jednalo se Analytický stroj, teoretický koncept, který nebyl nikdy uskutečněn, ale který předznamenal způsob myšlení a pokusů o vytvoření myslícího stroje aktuální až do dnešních dní.

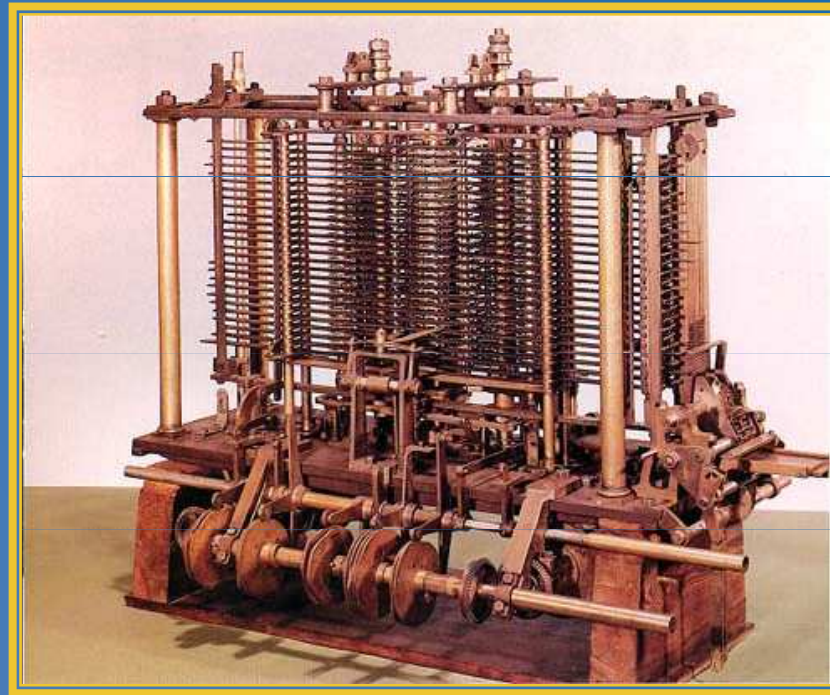
Architektura počítačů



- Předchůdcem Analytického stroje a jediným realizovaným projektem Charlese Babbage byl Derivační stroj, který v sobě obsahoval tabulky pro výpočet první derivace a byl založen na mechanickém principu.



Derivační stroj



Analytický stroj

Von neumannovská architektura

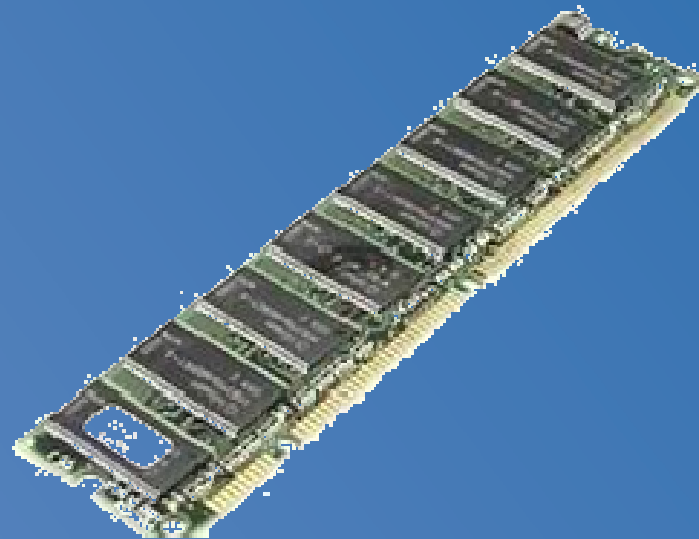


- Právě Babbageovy myšlenky měly zásadní vliv na tvorbu architektury, která je považována za standard v oblasti komputace.
- Již v jeho návrhu můžeme identifikovat základní prvky současných počítačů (centrální jednotka/procesor, paměť, vstupní a výstupní zařízení pro data).
- Norbert Wiener sepsal několik doporučení, určujících směr, kterým by se měla ubírat tvorba architektury budoucích strojů.
 1. *Používat číselnou formu reprezentace, ne mechanická kolečka (v extrémní podobě by se jednalo o rozdíl mezi digitálním a analogovým, zde spíše míněno z hlediska efektivity).*
 2. *Použít elektronky, pro jejich rychlost. Ne převodníky nebo relé.*
 3. *Používat dvojkový kód místo decimálního.*
 4. *Použít interní uložení programů, umístit odděleně od vstupních a výstupních dat.*
 5. *K internímu skladu by měl být rychlý přístup.*

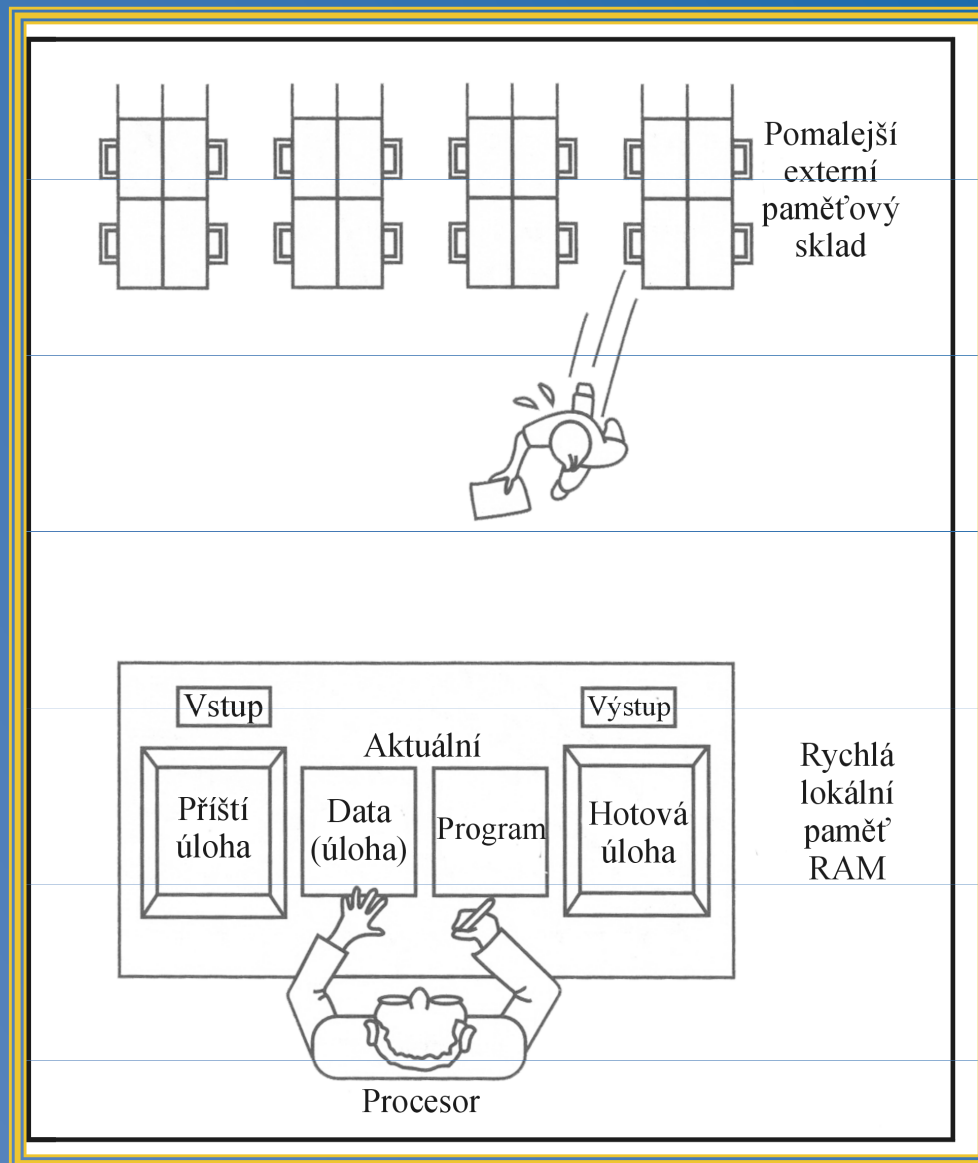
Von neumannovská architektura



Příspěvek samotného Johna von Neumanna do takto vylepšené architektury se zdá být zanedbatelný. V roce 1945 přišel s návrhem, že by výpočetní stroje měly obsahovat paměť rychle přístupnou pro procesor, ve které by byly uloženy aktuální program (soubor algoritmů) a také právě zpracovávaná data a jejich mezivýpočty. Narážel na pomalost tehdejších strojů, které ukládaly tyto informace vždy do externí paměti (z dnešního pohledu na pevný disk). Vznikla poslední část architektury počítače dnešních dní - operační paměť. Její uvedení do praxe neproběhlo okamžitě, ale muselo počkat na vynález paměti typu RAM.



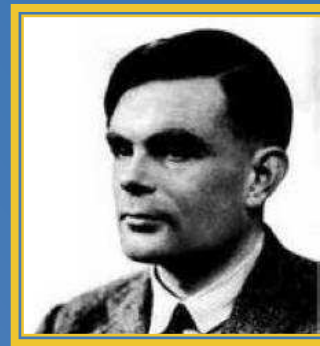
Von neumannovská architektura



Von Neumann versus Turing



Rozdíl mezi tímto typem architektury a Turingovým strojem je následující. Von neumannovská architektura je již navrhována se zřetelem k její praktické aplikaci (vycházela z Ch. Babbage) na rozdíl od Turingova stroje, jež byl vytvořen jako myšlenkový experiment, který si nekladal za cíl praktickou aplikaci. Turingovým cílem bylo vytvořit co nejjednodušší (s nejmenším počtem prvků) univerzální systém. Praktické provedení Turingova stroje by bylo zbytečné. Jeho jednoduchost spočívá v počtu použitých komponent a jejich funkcí, nikoliv však v rychlosti a možnostech v oblasti zpracování dat, což jsou vlastnosti v praxi považované za hlavní přednosti výpočetních systémů.



Reverzibilní komputace



V 60-tých létech začali uvažovat vývojáři počítačů o limitech výpočetních procesů.

Otázky se týkaly výpočetních možností materiálu:

Jaký maximální počet bitů zpracuje centimetr krychlový?

Jaké nejmenší množství tepla (energie) potřebujeme na zpracování jednoho bitu?

Reverzibita

Reverzibilní Turingův stroj umožňuje výpočet na pásku, která se může zpětně přepsat na původní.

Reverzibilní počítače jsou založeny na principu, že můžeme výpočet převést na původní zadání (neztrácíme cestou informaci na nereverzibilních bránách (hradlech) AND OR)

Landauerův princip



V roce 1961 uvažoval Landauer právě o množství energie spotřebované na zpracování jednoho bitu.

Došel k několika poznatkům a inovacím.

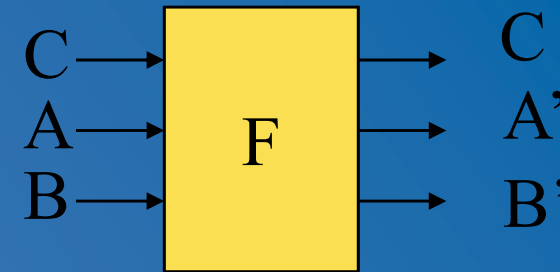
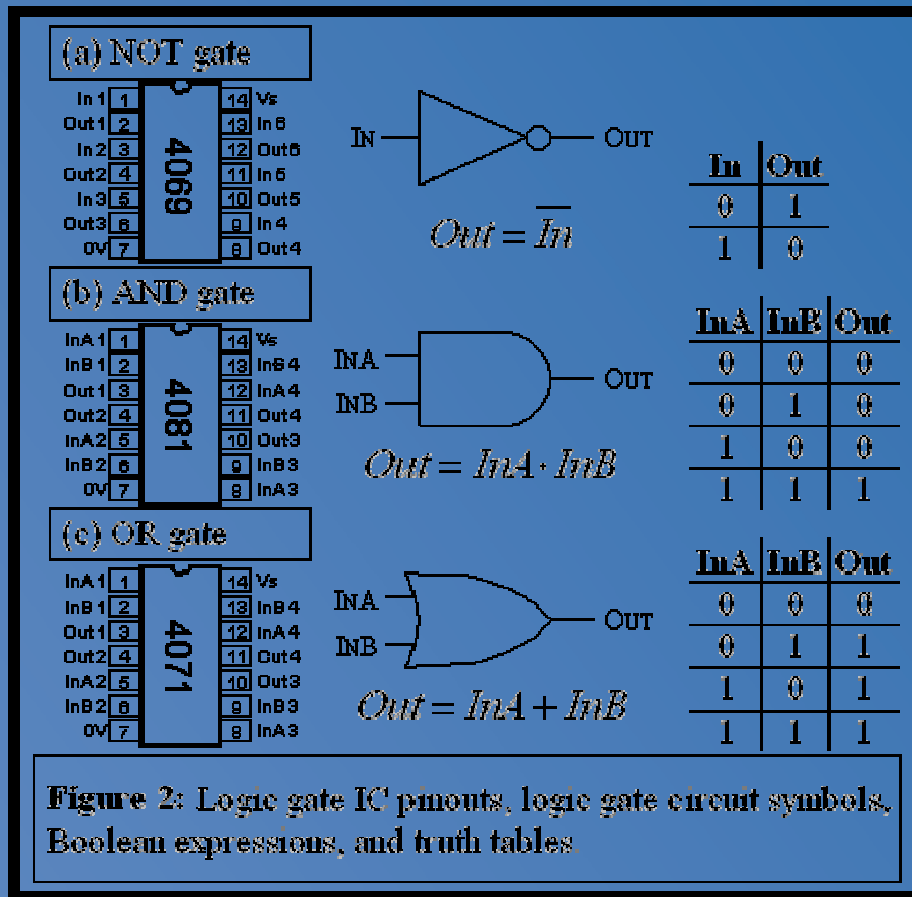
Existuje totiž vztah mezi logickou a fyzickou ireverzibilitou. Pokud totiž provádíme logickou nevratnou operaci ve fyzickém substrátu, který jí dokáže provést, je tato operace nevratná i z hlediska substrátu.

Příkladem logicky vratné operace je funkce NOT

Příkladem logicky nevratné operace je funkce AND

Landauerův přišel na to, že spotřeba energie stoupá díky tomu, že musí být během logické operace jeden bit vymazán. Což je podstata Landauerova principu.

Landauerův princip



C	A	B	C	A'	B'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Klasické logické brány

Fredkinova brána

Příště



V následující hodině se budeme zabývat

Historii umělé inteligence

Programy pro obecnou inteligenci

Úkol do příště:

Newell and Simon:

Human problem solving.

V informačním systému jej naleznete v sekci studijních materiálů.

Přečtěte si alespoň jeden záznam výpovědi zkoumané osoby o myšlenkových procesech, které používala během řešení kryptoaritmetické úlohy.

Konec



Děkuju za pozornost

